Oracle® Communications Billing and Revenue Management

Collecting General Ledger Data



Release 16.26.0
G54448-01
October 2026

Oracle Communications Billing and Revenue Management Collecting General Ledger Data, Release 16.26.0

G54448-01

Copyright © 2023, 2026, Oracle and/or its affiliates.

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

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

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

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

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

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

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

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

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

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Contents

About This Content

1 About Collecting General Ledger Data

About Collecting G/L Data 1
About G/L IDs 2
- About Revenue G/L IDs 2
- About Accounts Receivable G/L IDs 2
- About Standard G/L IDs 3
- About the Contract G/L ID 4

2 G/L Reporting Case Studies

About G/L Case Studies 1
Case Study 1 2
Case Study 2 6
Case Study 3 12
Case Study 4 18
Case Study 5 22
Case Study 6 23

3 Setting Up Your General Ledger

Setting Up Your General Ledger 1
Creating a Chart of Accounts 2
Creating G/L IDs 3
Assigning G/L IDs to A/R Actions 4
Assigning G/L IDs to Prerated Events 5

4 Reporting Revenue for Groups of Customers

About Reporting Revenue for Groups of Customers 1
- Creating Nested Segments 1

Copyright © 2023, 2026, Oracle and/or its affiliates.

Changing the Default G/L Segment 2

5 General Ledger Configuration Options

Disabling G/L Collection in BRM 1
Disabling Creation of /journal Objects for Noncurrency Balance Impacts 2
Setting the Number of Paying Accounts per G/L Report 3
Generating G/L Reports Based on the Actual Billing Date 3
Segregating Unbilled Revenue by G/L Cycle within a Billing Cycle 5
Setting Up G/L Calendar Configuration 7
Creating Journal Entries for Zero-Balance Events 8

6 General Ledger Revenue Recognition with Configurable Days Per Month

About G/L Revenue Calculation Using Fixed Monthly Days 1
About G/L Revenue Based On Actual Days in the Month 1
About G/L Revenue Based On Fixed Days Per Month 2
How Fixed Day G/L Revenue Recognition Impacts Journal Entries 3
Example 1: Journal Splitting for a Quarterly Billing Cycle 4
Example 2: Journal Splitting for a Long Cycle 4
Example 3: Journal Splitting for a Short Cycle 5
About Rounding G/L Revenue 6
Enabling and Using Fixed-Day G/L Revenue Recognition 6
Configuring the G/L DOM 7
Configuring Fixed-Day G/L Revenue Recognition 7

7 Rounding General Ledger Data

About Rounding and G/L Reports 1
Rounding G/L Report Data After Billing 1
Rounding G/L Report Data Prior to Billing 2

8 Generating General Ledger Reports

About G/L Reports 1
About the Monthly Export Report 2
About the Preview G/L Report 3
About Individual G/L Reports 4
Overview of the G/L Report Process 4
Running G/L Jobs in Business Operations Center 4
Generating G/L Reports Manually 5
Creating Preview G/L Reports with pin_ledger_report 5

Copyright © 2023, 2026, Oracle and/or its affiliates.

Creating Monthly Export G/L Reports with pin_ledger_report 6
Creating Individual G/L Reports with pin_ledger_report 6
Requirement for Creating a Report for Billed Revenue 7

9 Configuring General Ledger Report XML Output Files

Configuring Preview and Monthly Report XML Output Files 1
About Preview and Monthly Export Report XML Output Files 2
Sample Configuration File for Preview and Monthly Export Reports 2
Configuring Individual Report XML Output Files 3
Setting Up Schedules for Individual G/L Reports 5
About Individual Report XML Output Files 6
Sample Configuration File for Individual G/L Reports 7
Validating and Troubleshooting Exported G/L Reports 9

10 Troubleshooting and Correcting G/L Exports

Retrieving Audit Data for Exported G/L Reports 1
Using Updated G/L Data After an Export 2
Correcting Incorrect Data or Replacing Lost XML Output Files 3
Restarting a Failed Export 3
Purging G/L Export Data 3

11 About Deliverable-Based Revenue Recognition

About Revenue Recognition 1
About Deliverable-Based Revenue Recognition 2
About the Standalone Selling Price 2
About the Revenue Earning Schedule 3
About Calculating the Revenue Allocation 3
About Deliverables 3
About Subscription Terms 4
About the Revenue from Usage Charges and Administrative Fees 4
Sample Revenue Recognition Scenarios 5
Scenario 1: One-Year Contract with a Linear Earning Schedule 5
Scenario 2: One-Year Contract for a Physical Good with Free Maintenance 5
Setting Up Deliverable-Based Revenue Recognition 6
Enabling Deliverable-Based Revenue Recognition 6

12 About Event-Based Revenue Recognition

About Event-Based Revenue Recognition 1

Copyright © 2023, 2026, Oracle and/or its affiliates.

Examples of Billed and Unbilled Revenue 1
Billed and Unbilled Usage Fees 2
Billed and Unbilled Purchase and Cancellation Fees 2
Billed Cycle Arrears Fees 2
Billed and Unbilled Cycle Forward Arrears Fees 3
Billed and Unbilled Nonrated Events 3
Billed and Unbilled Cycle Forward Fees 4
About Unbilled Cycle Forward Fees 4
About Earned and Unearned Revenue 5
About Previously Billed Earned Revenue 6
About Incremental Reports and Cumulative Revenue 6
About Reversing G/L Entries 7
About Adjustments and G/L Reporting 7
How BRM Calculates Earned and Unearned Revenue 8
Calculating Billed Earned Revenue 8
Calculating Previously Billed Earned Revenue 9
Calculating Unearned Revenue 10

13 General Ledger Utilities

load_pin_config_export_gl 1
load_pin_glchartaccts 1
load_pin_glid 2
pin_ledger_report 3

Copyright © 2023, 2026, Oracle and/or its affiliates.

About This Content

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.

Copyright © 2023, 2026, Oracle and/or its affiliates.

1

About Collecting General Ledger Data

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

Topics in this document:

- About Collecting G/L Data
- About G/L IDs

About Collecting G/L Data

The general ledger is a list of accounts in which your company records financial transactions for accounting purposes. It shows how much revenue is recorded for each type of charge, such as how much revenue is generated by purchase fees or usage fees. Each type of transaction is recorded in a separate general ledger (G/L) account.

In BRM, you group related G/L accounts that you want to report a financial transaction against in a G/L ID. Each G/L ID is associated with a debit account and a credit account. For example, the revenue earned for telephony services could be in a G/L ID that is associated with:

- Debit account: Telephony liability
- Credit account: Telephony revenue

The general process for collecting G/L data is:

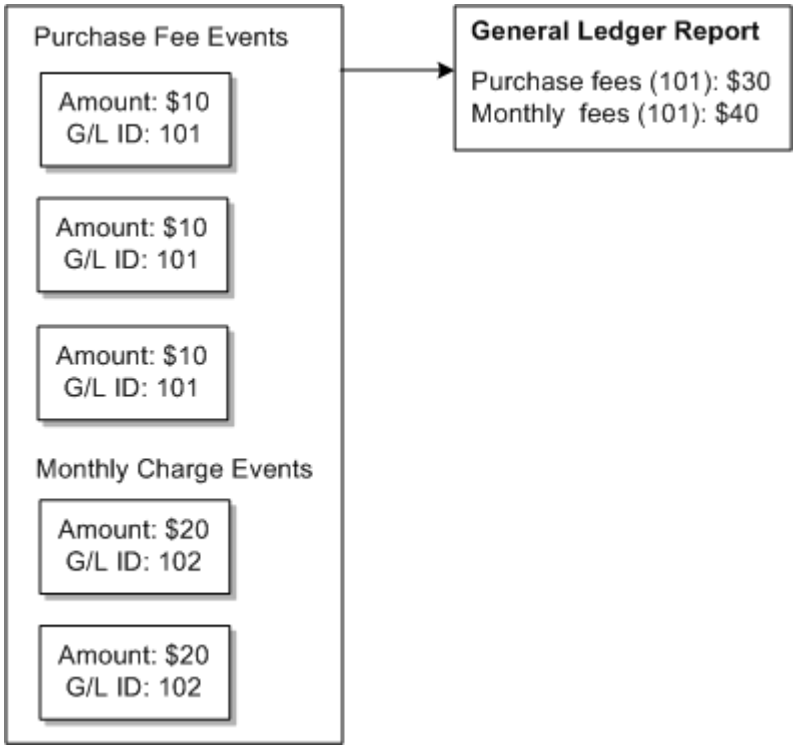
1. You set up a general ledger in BRM that matches the corporate general ledger in your external financial system. In BRM, this involves defining your general ledger accounts and then adding them to G/L IDs.

   When you create your general ledger in BRM, you set up your G/L accounts and G/L IDs for the revenue recognition scheme that your company supports: deliverable-based revenue recognition or event-based revenue recognition. See "About Deliverable-Based Revenue Recognition" and "About Event-Based Revenue Recognition".
2. You associate G/L IDs with everything that has a balance impact, such as charges, discounts, accounts receivable actions, and so on.
3. When an event with a balance impact occurs, BRM stores the revenue data along with its associated G/L ID information in a **/journal** object in the BRM database.
4. You generate a G/L report at the end of your company's reporting period.
5. BRM does the following to generate the G/L report:
   - Retrieves the revenue data from the **/journal** objects and uses the G/L IDs to compile the total amounts for each type of revenue.
   - Saves the report to a **/ledger_report** object in the BRM database.
   - Exports the report to an XML output file in your specified directory.
6. You retrieve the report XML output file and then post it to the corporate general ledger on your external financial system.

Copyright © 2023, 2026, Oracle and/or its affiliates.

Figure 1-1 shows how multiple purchase events and monthly fee events are compiled into a G/L report.

Figure 1-1 G/L IDs in Events and in a G/L Report



About G/L IDs

BRM contains four types of G/L IDs: revenue, accounts receivable (AR), standard, and contract. The ones you use depend on the revenue recognition scheme your company is using and the type of financial transaction you are tracking.

About Revenue G/L IDs

You use revenue G/L IDs in deliverable-based revenue recognition (see "About Deliverable-Based Revenue Recognition"). Revenue G/L IDs record financial transactions from charges that are associated with deliverables and that occur when a customer cancels their contract early.

Revenue G/L IDs report earned net revenue.

About Accounts Receivable G/L IDs

You use accounts receivable (A/R) G/L IDs in deliverable-based revenue recognition (see "About Deliverable-Based Revenue Recognition"). A/R G/L IDs record financial transactions from:

- Recurring, one-time, and charge selector charges
- Recurring, one-time, and simple discounts

Copyright © 2023, 2026, Oracle and/or its affiliates.

- Taxes
- Early termination fees, which can occur when customers cancel their contracts before the commitment period ends

A/R G/L IDs report billed net, billed tax, unbilled net, and unbilled tax revenue.

About Standard G/L IDs

You can use standard G/L IDs in both event-based revenue recognition and deliverable-based revenue recognition. See "About Event-Based Revenue Recognition" and "About Deliverable-Based Revenue Recognition".

In event-based revenue recognition, standard G/L IDs track all financial transactions in your system, including charges, discounts, tax codes, early termination fees, and A/R actions.

In deliverable-based revenue recognition, standard G/L IDs track financial transactions for:

- Service usage charges, such as the amount of data downloaded
- Charges and discounts that are not associated with deliverables
- A/R actions, such as payments, write-offs, and adjustments
- Fees for canceling services in no-contract subscriptions

Because standard G/L IDs can be used for multiple types of financial transactions, BRM allows you to map them to any debit account and credit account. When you create standard G/L IDs, ensure that the mappings are valid for your business needs.

Table 1-1 shows the recommended debit and credit accounts to use for each financial transaction type.

Table 1-1 Recommended Debit and Credit Accounts for Financial Transactions

Financial Transaction Type	Report Type	Amount Represents	Debit Account Type	Credit Account Type
Charge	Billed Unbilled	Net	Asset	Liability
Charge	Earned	Net	Liability	Revenue
Credit adjustment Write-off	Billed Unbilled	Net	Asset	Expense
Debit adjustment	Billed Unbilled	Net	Asset	Liability
Debit adjustment	Earned	Net	Liability	Revenue
Payment	Billed Unbilled	Net	Asset	Cash

By default, standard G/L IDs report billed, unbilled, and earned revenue with either a net or a tax attribute.

In addition to the default revenue types, standard G/L IDs can also report billed earned, billed unearned, previous billed earned, unbilled earned, and unbilled unearned revenue with a net, tax, gross, or discount attribute. Use these revenue types and attributes only if your company uses event-based revenue recognition. To be able to report these revenue types and attributes, create **Individual** G/L reports. See "About Individual G/L Reports".

Copyright © 2023, 2026, Oracle and/or its affiliates.

About the Contract G/L ID

You use the contract G/L ID in deliverable-based revenue recognition (see "About Deliverable-Based Revenue Recognition"). The contract G/L ID uses journal entries to track the amount paid and owed by your customers throughout their contract commitment term. BRM automatically creates the contract G/L ID for you.

Contract G/L IDs report earned net revenue.

Copyright © 2023, 2026, Oracle and/or its affiliates.

2

G/L Reporting Case Studies

Learn how revenue is reported for Oracle Communications Billing and Revenue Management (BRM) general ledger (G/L) reports through the illustrative examples in this chapter.

Topics in this document:

- About G/L Case Studies
- Case Study 1
- Case Study 2
- Case Study 3
- Case Study 4
- Case Study 5
- Case Study 6

For more information, see the following:

- About Collecting General Ledger Data
- Generating General Ledger Reports

About G/L Case Studies

The following case studies show the results of various plans on customer accounts. Each study contains a short description of the plan followed by actual outputs from running the **pin_ledger_report** utility. Figure 2-1 shows the **pin_glid** file used when generating the sample reports.

Copyright © 2023, 2026, Oracle and/or its affiliates.

Figure 2-1 Modified pin_glid File with G/L Entries

```
#====================================================================
# Default G/L ID
#====================================================================
glid(
     id          0
     descr       undefined
     gl_acct     unbilled                  net         undef A/R     undef off
     gl_acct     billed                    net         undef A/R     undef off
)
#====================================================================
# G/L IDs for purchase rate
#====================================================================
glid (
     id          101
     descr       Purchase Fees
     gl_acct billed_earned                 gross       10000      40000
     gl_acct billed_earned                 disc        40005      10000
     gl_acct unbilled_earned               gross       10000      40000
     gl_acct unbilled_earned               disc        40005      10000
)
#====================================================================
# G/L IDs for cycle_forward Internet Access
#====================================================================
glid (
     id          102
     descr       Monthly Fees -- Internet Access
     gl_acct billed_earned                 gross       10000      40001
     gl_acct billed_earned                 disc        40005      10000
     gl_acct prev_billed_earned            gross       10000      40001
     gl_acct prev_billed_earned            disc        40005      10000
     gl_acct billed_unearned               gross       10000      40001-001
     gl_acct billed_unearned               disc        40005      10000
     gl_acct unbilled_earned               gross       10000      40001
     gl_acct unbilled_earned               disc        40005      10000
     gl_acct unbilled_unearned             gross       10000      40001-001
     gl_acct unbilled_unearned             disc        40005      10000
)
#====================================================================
# G/L IDs for Email Service
#====================================================================
glid (
     id          104
     descr       Monthly Fees - Email Service
     gl_acct billed_earned                 gross       10000      40003
     gl_acct billed_earned                 disc        40005      10000
     gl_acct prev_billed_earned            gross       10000      40003
     gl_acct prev_billed_earned            disc        40005      10000
     gl_acct billed_unearned               gross       10000      40001-001
     gl_acct billed_unearned               disc        40005      10000
     gl_acct unbilled_earned               gross       10000      40003
     gl_acct unbilled_earned               disc        40005      10000
     gl_acct unbilled_unearned             gross       10000      40001-001
     gl_acct unbilled_unearned             disc        40005      10000
)

glid (
     id          109
     descr       Payments
     gl_acct billed_earned                 gross       10000      50000
)
```

```
10000 = AR Account
50000 = Cash Accoun
40005 = Discount Account
40000= Revenue Account
40001 = Revenue Account
40003 = Revenue Account
40003-001 = Deferred Rev
```

Case Study 1

Plan details:

- Customer created: July 6, 2000
- G/L posting: Monthly G/L posting
- Purchase: Plan 1 - Measured Web access with discounts from an example **.ipl** file.
  - Monthly email fee: $3.00 (G/L ID 104)
  - Monthly proratable IP fee: $9.95 (G/L ID 102)

Copyright © 2023, 2026, Oracle and/or its affiliates.

- Accrual-based accounting

Figure 2-2 shows the G/L output for July.

Figure 2-2 July G/L Output

Ledger Output for July 2000

General Ledger Data

Type : unbilled_earned
Posting Start : Sat Jul 1 00:00:00 2000
Posting End : Tue Aug 1 00:00:00 2000

ACCT #: 0.0.0.1-9267 DATE: 08/06/2000 BILL #: B1-3 ITEM #: B1-3,1 (Cycle forward)

104 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	2.52	0.00	40003	0.00	2.52
NET	-- N/A --	2.52	0.00	-- N/A --	0.00	2.52
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-9267 DATE: 08/06/2000 BILL #: B1-3 ITEM #: B1-3,3 (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	8.35	0.00	40001	0.00	8.35
NET	-- N/A --	8.35	0.00	-- N/A --	0.00	8.35
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

General Ledger Data

Type : unbilled_unearned
Posting Start : Sat Jul 1 00:00:00 2000
Posting End : Tue Aug 1 00:00:00 2000

ACCT #: 0.0.0.1-9267 DATE: 08/06/2000 BILL #: B1-3 ITEM #: B1-3,1 (Cycle forward)

104 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	0.48	0.00	40001-001	0.00	0.48
NET	-- N/A --	0.48	0.00	-- N/A --	0.00	0.48
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

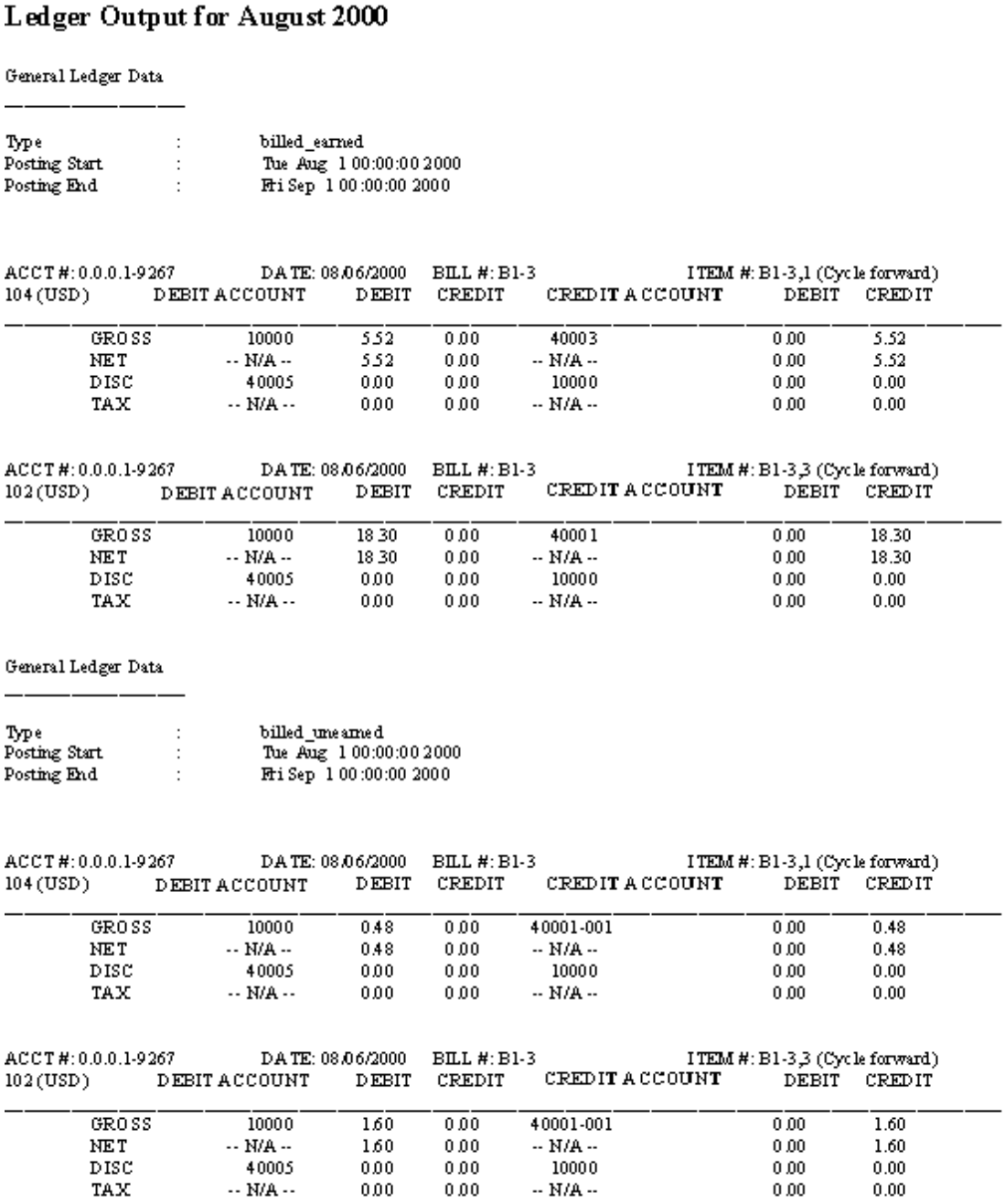
ACCT #: 0.0.0.1-9267 DATE: 08/06/2000 BILL #: B1-3 ITEM #: B1-3,3 (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	1.60	0.00	40001-001	0.00	1.60
NET	-- N/A --	1.60	0.00	-- N/A --	0.00	1.60
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

Figure 2-3 shows the G/L output for August.

Copyright © 2023, 2026, Oracle and/or its affiliates.

Figure 2-3 August G/L Output



Ledger Output for August 2000

General Ledger Data

Type : billed_earned
Posting Start : Tue Aug 1 00:00:00 2000
Posting End : Fri Sep 1 00:00:00 2000

ACCT #: 0.0.0.1-9267 DATE: 08/06/2000 BILL #: B1-3 ITEM #: B1-3,1 (Cycle forward)

104 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	5.52	0.00	40003	0.00	5.52
NET	-- N/A --	5.52	0.00	-- N/A --	0.00	5.52
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-9267 DATE: 08/06/2000 BILL #: B1-3 ITEM #: B1-3,3 (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	18.30	0.00	40001	0.00	18.30
NET	-- N/A --	18.30	0.00	-- N/A --	0.00	18.30
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

General Ledger Data

Type : billed_unearned
Posting Start : Tue Aug 1 00:00:00 2000
Posting End : Fri Sep 1 00:00:00 2000

ACCT #: 0.0.0.1-9267 DATE: 08/06/2000 BILL #: B1-3 ITEM #: B1-3,1 (Cycle forward)

104 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	0.48	0.00	40001-001	0.00	0.48
NET	-- N/A --	0.48	0.00	-- N/A --	0.00	0.48
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-9267 DATE: 08/06/2000 BILL #: B1-3 ITEM #: B1-3,3 (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	1.60	0.00	40001-001	0.00	1.60
NET	-- N/A --	1.60	0.00	-- N/A --	0.00	1.60
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

The unbilled revenue for July 1 through August 1 appears on the billed report of August 1 through September 1. To do reversals, you need to reverse the data in your company's accounting software.

Figure 2-4 is an example of the September G/L report:

Copyright © 2023, 2026, Oracle and/or its affiliates.

Figure 2-4 September G/L Report

Ledger Output for September 2000

General Ledger Data

Type : billed_earned
Posting Start : Fri Sep 1 00:00:00 2000
Posting End : Sun Oct 1 00:00:00 2000

ACCT #: 0.0.0.1-9267 DATE: 09/06/2000 BILL #: B1-6 ITEM #: B1-6,1 (Cycle forward)

104 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	2.50	0.00	40003	0.00	2.50
NET	-- N/A --	2.50	0.00	-- N/A --	0.00	2.50
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-9267 DATE: 09/06/2000 BILL #: B1-6 ITEM #: B1-6,3 (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	8.29	0.00	40001	0.00	8.29
NET	-- N/A --	8.29	0.00	-- N/A --	0.00	8.29
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

General Ledger Data

Type : billed_unearned
Posting Start : Fri Sep 1 00:00:00 2000
Posting End : Sun Oct 1 00:00:00 2000

ACCT #: 0.0.0.1-9267 DATE: 09/06/2000 BILL #: B1-6 ITEM #: B1-6,1 (Cycle forward)

104 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	0.50	0.00	40001-001	0.00	0.50
NET	-- N/A --	0.50	0.00	-- N/A --	0.00	0.50
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-9267 DATE: 09/06/2000 BILL #: B1-6 ITEM #: B1-6,3 (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	1.66	0.00	40001-001	0.00	1.66
NET	-- N/A --	1.66	0.00	-- N/A --	0.00	1.66
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

General Ledger Data

Type : prev_billed_earned
Posting Start : Fri Sep 1 00:00:00 2000
Posting End : Sun Oct 1 00:00:00 2000

ACCT #: 0.0.0.1-9267 DATE: 08/06/2000 BILL #: B1-3 ITEM #: B1-3,1 (Cycle forward)

104 (USD)	A/R ACCOUNT	DEBIT	CREDIT	OFFSET ACCOUNT	DEBIT	CREDIT
GROSS	10000	0.48	0.00	40003	0.00	0.48
NET	-- N/A --	0.48	0.00	-- N/A --	0.00	0.48
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-9267 DATE: 08/06/2000 BILL #: B1-3 ITEM #: B1-3,3 (Cycle forward)

102 (USD)	A/R ACCOUNT	DEBIT	CREDIT	OFFSET ACCOUNT	DEBIT	CREDIT
GROSS	10000	1.60	0.00	40001	0.00	1.60
NET	-- N/A --	1.60	0.00	-- N/A --	0.00	1.60
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

Copyright © 2023, 2026, Oracle and/or its affiliates.

From this report onwards, the monthly fees of $9.95 and $3.00 become earned revenue for all successive months. However, because the monthly fee is proratable, the amount depends on the number of days per month. Therefore, the amount earned in September is prorated for 30 days (25/30 * 9.95 = 8.29). The amount that was reported as unearned revenue in August becomes earned revenue in September.

Case Study 2

Plan details:

- Customer created: July 6, 2000
- G/L posting: Daily G/L posting
- Purchase: Plan 1 – Measured Web access with discounts from an example **.ipl** file.
  - Monthly email fee: $3.00 (G/L ID 104)
  - Monthly proratable IP fee: $9.95 (G/L ID 102)
- Accrual-based accounting

Figure 2-5 shows the G/L report for July 6.

Copyright © 2023, 2026, Oracle and/or its affiliates.

Figure 2-5 July 6 G/L Report

Ledger Output for July 6, 2000

General Ledger Data

Type : billed_earned
Posting Start : Thu Jul 6 00:00:00 2000
Posting End : Fri Jul 7 00:00:00 2000

ACCT #: 0.0.0.1-9267 DATE: 07/06/2000 BILL #: -- N/A -- ITEM #: P-1 (Payment)

109 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	0.00	12.95	50000	12.95	0.00
NET	-- N/A --	0.00	12.95	-- N/A --	12.95	0.00
DISC	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

General Ledger Data

Type : unbilled_earned
Posting Start : Thu Jul 6 00:00:00 2000
Posting End : Fri Jul 7 00:00:00 2000

ACCT #: 0.0.0.1-9267 DATE: -- N/A -- BILL #: -- N/A -- ITEM #: -- N/A -- (Cycle forward)

104 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	0.10	0.00	40003	0.00	0.10
NET	-- N/A --	0.10	0.00	-- N/A --	0.00	0.10
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-9267 DATE: -- N/A -- BILL #: -- N/A -- ITEM #: -- N/A -- (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	0.32	0.00	40001	0.00	0.32
NET	-- N/A --	0.32	0.00	-- N/A --	0.00	0.32
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

General Ledger Data

Type : unbilled_unearned
Posting Start : Thu Jul 6 00:00:00 2000
Posting End : Fri Jul 7 00:00:00 2000

ACCT #: 0.0.0.1-9267 DATE: -- N/A -- BILL #: -- N/A -- ITEM #: -- N/A -- (Cycle forward)

104 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	2.90	0.00	40001-001	0.00	2.90
NET	-- N/A --	2.90	0.00	-- N/A --	0.00	2.90
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-9267 DATE: -- N/A -- BILL #: -- N/A -- ITEM #: -- N/A -- (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	9.63	0.00	40001-001	0.00	9.63
NET	-- N/A --	9.63	0.00	-- N/A --	0.00	9.63
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

The full payment was received for the cycle forward fees because the customer account was created with the credit card payment method.

Copyright © 2023, 2026, Oracle and/or its affiliates.

The cycle forward fee remains unbilled until the first billing cycle on August 6. In the G/L reports for July 7 and July 15, BRM decreases the unbilled_unearned revenue and increases the unbilled_unearned revenue amounts.

Figure 2-6 shows the July 7 G/L report.

Figure 2-6 July 7 G/L Report

Ledger Output for July 7, 2000

General Ledger Data

Type : unbilled_earned
Posting Start : Fri Jul 7 00:00:00 2000
Posting End : Sat Jul 8 00:00:00 2000

ACCT #: 0.0.0.1-9267 DATE: -- N/A -- BILL #: -- N/A -- ITEM #: -- N/A -- (Cycle forward)

104 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	0.19	0.00	40003	0.00	0.19
NET	-- N/A --	0.19	0.00	-- N/A --	0.00	0.19
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-9267 DATE: -- N/A -- BILL #: -- N/A -- ITEM #: -- N/A -- (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	0.64	0.00	40001	0.00	0.64
NET	-- N/A --	0.64	0.00	-- N/A --	0.00	0.64
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

General Ledger Data

Type : unbilled_unearned
Posting Start : Fri Jul 7 00:00:00 2000
Posting End : Sat Jul 8 00:00:00 2000

ACCT #: 0.0.0.1-9267 DATE: -- N/A -- BILL #: -- N/A -- ITEM #: -- N/A -- (Cycle forward)

104 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	2.81	0.00	40001-001	0.00	2.81
NET	-- N/A --	2.81	0.00	-- N/A --	0.00	2.81
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-9267 DATE: -- N/A -- BILL #: -- N/A -- ITEM #: -- N/A -- (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	9.31	0.00	40001-001	0.00	9.31
NET	-- N/A --	9.31	0.00	-- N/A --	0.00	9.31
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

Figure 2-7 shows the July 15 G/L report.

Copyright © 2023, 2026, Oracle and/or its affiliates.

Figure 2-7 July 15 G/L Report

Ledger Output for July 15, 2000

General Ledger Data

Type : unbilled_earned
Posting Start : Sat Jul 15 00:00:00 2000
Posting End : Sun Jul 16 00:00:00 2000

ACCT #: 0.0.0.1-9267 DATE: -- N/A -- BILL #: -- N/A -- ITEM #: -- N/A -- (Cycle forward)

104 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	0.97	0.00	40003	0.00	0.97
NET	-- N/A --	0.97	0.00	-- N/A --	0.00	0.97
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-9267 DATE: -- N/A -- BILL #: -- N/A -- ITEM #: -- N/A -- (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	3.21	0.00	40001	0.00	3.21
NET	-- N/A --	3.21	0.00	-- N/A --	0.00	3.21
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

General Ledger Data

Type : unbilled_unearned
Posting Start : Sat Jul 15 00:00:00 2000
Posting End : Sun Jul 16 00:00:00 2000

ACCT #: 0.0.0.1-9267 DATE: -- N/A -- BILL #: -- N/A -- ITEM #: -- N/A -- (Cycle forward)

104 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	2.03	0.00	40001-001	0.00	2.03
NET	-- N/A --	2.03	0.00	-- N/A --	0.00	2.03
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-9267 DATE: -- N/A -- BILL #: -- N/A -- ITEM #: -- N/A -- (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	6.74	0.00	40001-001	0.00	6.74
NET	-- N/A --	6.74	0.00	-- N/A --	0.00	6.74
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

When billing is run on August 6, the amounts change from unbilled to billed as shown in Figure 2-8. The entire amount is now earned and reported in the billed_earned revenue type.

Copyright © 2023, 2026, Oracle and/or its affiliates.

Figure 2-8 August 6 G/L Report

Ledger Output for August 6, 2000

General Ledger Data

Type : billed_earned
Posting Start : Sun Aug 6 00:00:00 2000
Posting End : Mon Aug 7 00:00:00 2000

ACCT #: 0.0.0.1-9267 DATE: 08/06/2000 BILL #: B1-3 ITEM #: B1-3,1 (Cycle forward)

104 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	3.10	0.00	40003	0.00	3.10
NET	-- N/A --	3.10	0.00	-- N/A --	0.00	3.10
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-9267 DATE: 08/06/2000 BILL #: -- N/A -- ITEM #: P-3 (Payment)

109 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	0.00	12.95	50000	12.95	0.00
NET	-- N/A --	0.00	12.95	-- N/A --	12.95	0.00
DISC	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-9267 DATE: 08/06/2000 BILL #: B1-3 ITEM #: B1-3,3 (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	10.27	0.00	40001	0.00	10.27
NET	-- N/A --	10.27	0.00	-- N/A --	0.00	10.27
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

General Ledger Data

Type : billed_unearned
Posting Start : Sun Aug 6 00:00:00 2000
Posting End : Mon Aug 7 00:00:00 2000

ACCT #: 0.0.0.1-9267 DATE: 08/06/2000 BILL #: B1-3 ITEM #: B1-3,1 (Cycle forward)

104 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	2.90	0.00	40001-001	0.00	2.90
NET	-- N/A --	2.90	0.00	-- N/A --	0.00	2.90
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-9267 DATE: 08/06/2000 BILL #: B1-3 ITEM #: B1-3,3 (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	9.63	0.00	40001-001	0.00	9.63
NET	-- N/A --	9.63	0.00	-- N/A --	0.00	9.63
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

From the report run on August 7, the earned revenue is now posted under the prev_billed_earned revenue type as shown in Figure 2-9.

Copyright © 2023, 2026, Oracle and/or its affiliates.

Figure 2-9 August 7 G/L Report

Ledger Output for August 7, 2000

General Ledger Data

Type : billed_unearned
Posting Start : Mon Aug 7 00:00:00 2000
Posting End : Tue Aug 8 00:00:00 2000

ACCT #: 0.0.0.1-9267 DATE: 08/06/2000 BILL #: B1-3 ITEM #: B1-3,1 (Cycle forward)

104 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	2.81	0.00	40001-001	0.00	2.81
NET	-- N/A --	2.81	0.00	-- N/A --	0.00	2.81
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-9267 DATE: 08/06/2000 BILL #: B1-3 ITEM #: B1-3,3 (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	9.31	0.00	40001-001	0.00	9.31
NET	-- N/A --	9.31	0.00	-- N/A --	0.00	9.31
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

General Ledger Data

Type : prev_billed_earned
Posting Start : Mon Aug 7 00:00:00 2000
Posting End : Tue Aug 8 00:00:00 2000

ACCT #: 0.0.0.1-9267 DATE: 08/06/2000 BILL #: B1-3 ITEM #: B1-3,1 (Cycle forward)

104 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	0.10	0.00	40003	0.00	0.10
NET	-- N/A --	0.10	0.00	-- N/A --	0.00	0.10
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

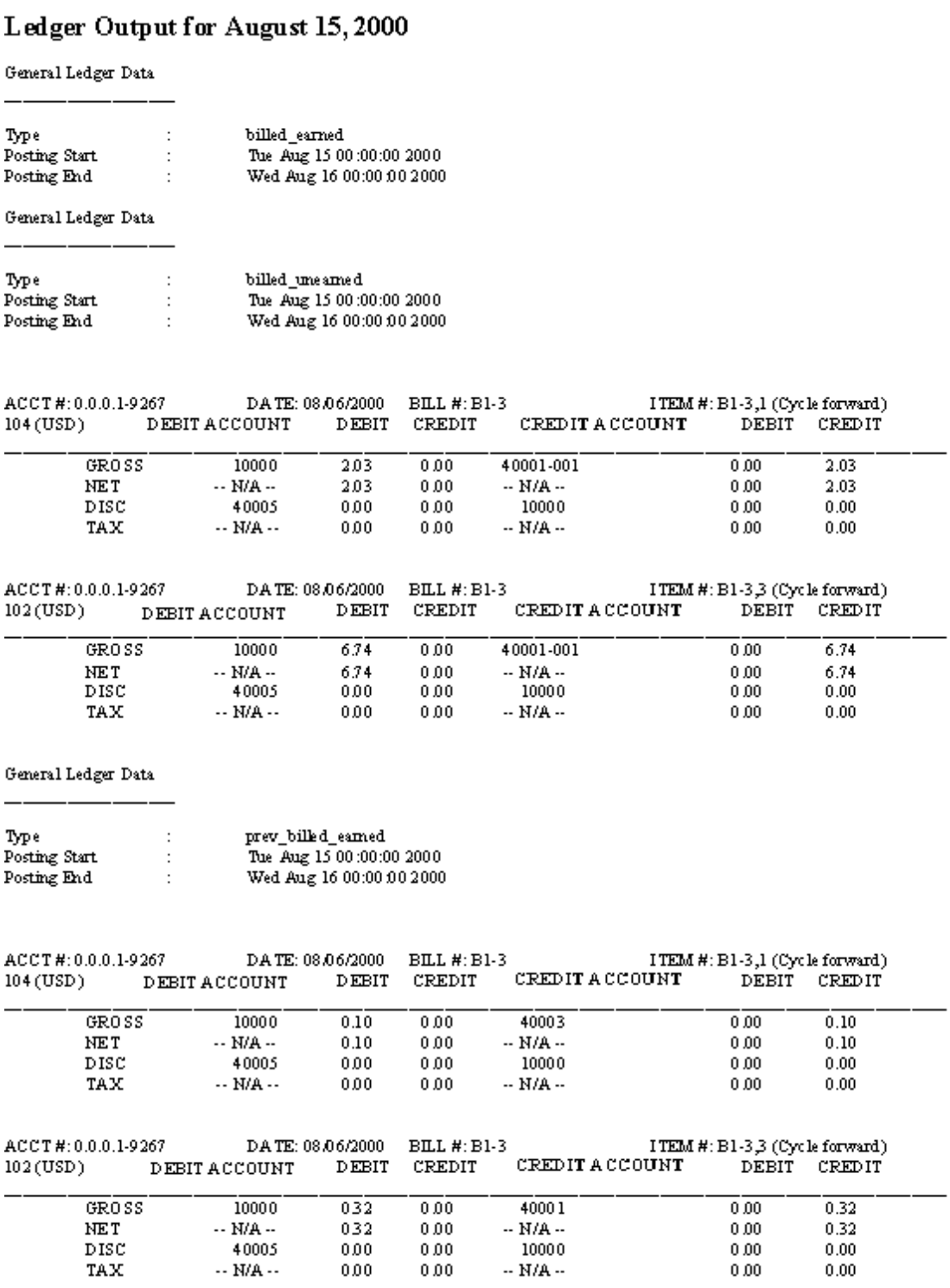
ACCT #: 0.0.0.1-9267 DATE: 08/06/2000 BILL #: B1-3 ITEM #: B1-3,3 (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	0.32	0.00	40001	0.00	0.32
NET	-- N/A --	0.32	0.00	-- N/A --	0.00	0.32
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

Figure 2-10 shows the August 15 G/L report.

Copyright © 2023, 2026, Oracle and/or its affiliates.

Figure 2-10 August 15 G/L Report



Ledger Output for August 15, 2000

General Ledger Data

Type : billed_earned
Posting Start : Tue Aug 15 00:00:00 2000
Posting End : Wed Aug 16 00:00:00 2000

General Ledger Data

Type : billed_unearned
Posting Start : Tue Aug 15 00:00:00 2000
Posting End : Wed Aug 16 00:00:00 2000

ACCT #: 0.0.0.1-9267 DATE: 08/06/2000 BILL #: B1-3 ITEM #: B1-3,1 (Cycle forward)

104 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	2.03	0.00	40001-001	0.00	2.03
NET	-- N/A --	2.03	0.00	-- N/A --	0.00	2.03
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-9267 DATE: 08/06/2000 BILL #: B1-3 ITEM #: B1-3,3 (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	6.74	0.00	40001-001	0.00	6.74
NET	-- N/A --	6.74	0.00	-- N/A --	0.00	6.74
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

General Ledger Data

Type : prev_billed_earned
Posting Start : Tue Aug 15 00:00:00 2000
Posting End : Wed Aug 16 00:00:00 2000

ACCT #: 0.0.0.1-9267 DATE: 08/06/2000 BILL #: B1-3 ITEM #: B1-3,1 (Cycle forward)

104 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	0.10	0.00	40003	0.00	0.10
NET	-- N/A --	0.10	0.00	-- N/A --	0.00	0.10
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-9267 DATE: 08/06/2000 BILL #: B1-3 ITEM #: B1-3,3 (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	0.32	0.00	40001	0.00	0.32
NET	-- N/A --	0.32	0.00	-- N/A --	0.00	0.32
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

Case Study 3

Plan details:

Copyright © 2023, 2026, Oracle and/or its affiliates.

- Customer created: July 6, 2000
- G/L posting: Daily G/L posting
- Purchase: Plan 1 – Measured Web access with discounts from an example **.ipl** file.
  - Monthly email fee: $3.00 (G/L ID 104)
  - Monthly proratable IP fee: $9.95 (G/L ID 102)
- Accrual-based accounting
- On-demand billing: You generate a bill immediately after the deal is purchased. For more information, see "Configuring Bill Now" in *BRM Configuring and Running Billing*.

The G/L report run for July 6 shows that the revenue is earned on the same day that the deal was purchased as seen in Figure 2-11.

Figure 2-11 July 6 G/L Report

General Ledger Data

Type	:	billed_earned
Posting Start	:	Thu Jul 6 00:00:00 2000
Posting End	:	Fri Jul 7 00:00:00 2000

ACCT #: 0.0.0.1-10923 DATE: 07/06/2000 BILL #: B1-2 ITEM #: B1-2,1 (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	0.32	0.00	40001	0.00	0.32
NET	-- N/A --	0.32	0.00	-- N/A --	0.00	0.32
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-10923 DATE: 07/06/2000 BILL #: B1-1 ITEM #: B1-1,1 (Cycle forward)

104 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	0.10	0.00	40003	0.00	0.10
NET	-- N/A --	0.10	0.00	-- N/A --	0.00	0.10
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

General Ledger Data

Type	:	billed_unearned
Posting Start	:	Thu Jul 6 00:00:00 2000
Posting End	:	Fri Jul 7 00:00:00 2000

ACCT #: 0.0.0.1-10923 DATE: 07/06/2000 BILL #: B1-2 ITEM #: B1-2,1 (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	9.63	0.00	40001-001	0.00	9.63
NET	-- N/A --	9.63	0.00	-- N/A --	0.00	9.63
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-10923 DATE: 07/06/2000 BILL #: B1-1 ITEM #: B1-1,1 (Cycle forward)

104 (US Dollar)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	2.90	0.00	40001-001	0.00	2.90
NET	-- N/A --	2.90	0.00	-- N/A --	0.00	2.90
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

In the following example, BRM generates the bill and reports the revenue as earned when the customer purchases a deal.

Figure 2-12 shows the July 7 G/L report.

Copyright © 2023, 2026, Oracle and/or its affiliates.

Figure 2-12 July 7 G/L Report

General Ledger Data

Type : billed_earned
Posting Start : Fri Jul 7 00:00:00 2000
Posting End : Sat Jul 8 00:00:00 2000

General Ledger Data

Type : billed_unearned
Posting Start : Fri Jul 7 00:00:00 2000
Posting End : Sat Jul 8 00:00:00 2000

ACCT #: 0.0.0.1-10923 DATE: 07/06/2000 BILL #: B1-2 ITEM #: B1-2,1 (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	9.31	0.00	40001-001	0.00	9.31
NET	-- N/A --	9.31	0.00	-- N/A --	0.00	9.31
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-10923 DATE: 07/06/2000 BILL #: B1-1 ITEM #: B1-1,1 (Cycle forward)

104 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	2.81	0.00	40001-001	0.00	2.81
NET	-- N/A --	2.81	0.00	-- N/A --	0.00	2.81
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

General Ledger Data

Type : prev_billed_earned
Posting Start : Fri Jul 7 00:00:00 2000
Posting End : Sat Jul 8 00:00:00 2000

ACCT #: 0.0.0.1-10923 DATE: 07/06/2000 BILL #: B1-2 ITEM #: B1-2,1 (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	0.32	0.00	40001	0.00	0.32
NET	-- N/A --	0.32	0.00	-- N/A --	0.00	0.32
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-10923 DATE: 07/06/2000 BILL #: B1-1 ITEM #: B1-1,1 (Cycle forward)

104 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	0.10	0.00	40003	0.00	0.10
NET	-- N/A --	0.10	0.00	-- N/A --	0.00	0.10
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

The following examples show the results of running G/L reports on a bi-monthly cycle with the on-demand billing feature.

Figure 2-13 shows the July 15 G/L report.

Copyright © 2023, 2026, Oracle and/or its affiliates.

Figure 2-13 July 15 G/L Report

General Ledger Data

Type	:	billed_earned
Posting Start	:	Sat Jul 15 00:00:00 2000
Posting End	:	Sun Jul 16 00:00:00 2000

General Ledger Data

Type	:	billed_unearned
Posting Start	:	Sat Jul 15 00:00:00 2000
Posting End	:	Sun Jul 16 00:00:00 2000

ACCT #: 0.0.0.1-10923 DATE: 07/06/2000 BILL #: B1-2 ITEM #: B1-2,1 (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	6.74	0.00	40001-001	0.00	6.74
NET	-- N/A --	6.74	0.00	-- N/A --	0.00	6.74
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-10923 DATE: 07/06/2000 BILL #: B1-1 ITEM #: B1-1,1 (Cycle forward)

104 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	2.03	0.00	40001-001	0.00	2.03
NET	-- N/A --	2.03	0.00	-- N/A --	0.00	2.03
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

General Ledger Data

Type	:	prev_billed_earned
Posting Start	:	Sat Jul 15 00:00:00 2000
Posting End	:	Sun Jul 16 00:00:00 2000

ACCT #: 0.0.0.1-10923 DATE: 07/06/2000 BILL #: B1-2 ITEM #: B1-2,1 (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	0.32	0.00	40001	0.00	0.32
NET	-- N/A --	0.32	0.00	-- N/A --	0.00	0.32
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-10923 DATE: 07/06/2000 BILL #: B1-1 ITEM #: B1-1,1 (Cycle forward)

104 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	0.10	0.00	40003	0.00	0.10
NET	-- N/A --	0.10	0.00	-- N/A --	0.00	0.10
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

Figure 2-14 shows the August 6 G/L report.

Copyright © 2023, 2026, Oracle and/or its affiliates.

Figure 2-14 August 6 G/L Report

General Ledger Data

Type : billed_earned
Posting Start : Sun Aug 6 00:00:00 2000
Posting End : Mon Aug 7 00:00:00 2000

ACCT #: 0.0.0.1-10923 DATE: 08/06/2000 BILL #: B1-4 ITEM #: B1-4,2 (Cycle forward)

104 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	0.10	0.00	40003	0.00	0.10
NET	-- N/A --	0.10	0.00	-- N/A --	0.00	0.10
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-10923 DATE: 08/06/2000 BILL #: B1-4 ITEM #: B1-4,3 (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	0.32	0.00	40001	0.00	0.32
NET	-- N/A --	0.32	0.00	-- N/A --	0.00	0.32
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-10923 DATE: 08/06/2000 BILL #: -- N/A -- ITEM #: P-4 (Payment)

109 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	0.00	12.95	50000	12.95	0.00
NET	-- N/A --	0.00	12.95	-- N/A --	12.95	0.00
DISC	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

General Ledger Data

Type : billed_unearned
Posting Start : Sun Aug 6 00:00:00 2000
Posting End : Mon Aug 7 00:00:00 2000

ACCT #: 0.0.0.1-10923 DATE: 08/06/2000 BILL #: B1-4 ITEM #: B1-4,2 (Cycle forward)

104 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	2.90	0.00	40001-001	0.00	2.90
NET	-- N/A --	2.90	0.00	-- N/A --	0.00	2.90
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-10923 DATE: 08/06/2000 BILL #: B1-4 ITEM #: B1-4,3 (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	9.63	0.00	40001-001	0.00	9.63
NET	-- N/A --	9.63	0.00	-- N/A --	0.00	9.63
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

Figure 2-15 shows the August 7 G/L report.

Copyright © 2023, 2026, Oracle and/or its affiliates.

Figure 2-15 August 7 G/L Report

General Ledger Data

Type : billed_unearned
Posting Start : Mon Aug 7 00:00:00 2000
Posting End : Tue Aug 8 00:00:00 2000

ACCT #: 0.0.0.1-10923 DATE: 08/06/2000 BILL #: B1-4 ITEM #: B1-4,2 (Cycle forward)
104 (USD)

	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	2.81	0.00	40001-001	0.00	2.81
NET	-- N/A --	2.81	0.00	-- N/A --	0.00	2.81
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-10923 DATE: 08/06/2000 BILL #: B1-4 ITEM #: B1-4,3 (Cycle forward)
102 (USD)

	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	9.31	0.00	40001-001	0.00	9.31
NET	-- N/A --	9.31	0.00	-- N/A --	0.00	9.31
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

General Ledger Data

Type : prev_billed_earned
Posting Start : Mon Aug 7 00:00:00 2000
Posting End : Tue Aug 8 00:00:00 2000

ACCT #: 0.0.0.1-10923 DATE: 08/06/2000 BILL #: B1-4 ITEM #: B1-4,2 (Cycle forward)
104 (USD)

	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	0.10	0.00	40003	0.00	0.10
NET	-- N/A --	0.10	0.00	-- N/A --	0.00	0.10
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-10923 DATE: 08/06/2000 BILL #: B1-4 ITEM #: B1-4,3 (Cycle forward)
102 (USD)

	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	0.32	0.00	40001	0.00	0.32
NET	-- N/A --	0.32	0.00	-- N/A --	0.00	0.32
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

Figure 2-16 shows the August 15 G/L report.

Copyright © 2023, 2026, Oracle and/or its affiliates.

Figure 2-16 August 15 G/L Report

General Ledger Data

Type : billed_earned
Posting Start : Tue Aug 15 00:00:00 2000
Posting End : Wed Aug 16 00:00:00 2000

General Ledger Data

Type : billed_unearned
Posting Start : Tue Aug 15 00:00:00 2000
Posting End : Wed Aug 16 00:00:00 2000

ACCT #: 0.0.0.1-10923 DATE: 08/06/2000 BILL #: B1-4 ITEM #: B1-4,2 (Cycle forward)

104 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	2.03	0.00	40001-001	0.00	2.03
NET	-- N/A --	2.03	0.00	-- N/A --	0.00	2.03
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-10923 DATE: 08/06/2000 BILL #: B1-4 ITEM #: B1-4,3 (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	6.74	0.00	40001-001	0.00	6.74
NET	-- N/A --	6.74	0.00	-- N/A --	0.00	6.74
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.0

General Ledger Data

Type : prev_billed_earned
Posting Start : Tue Aug 15 00:00:00 2000
Posting End : Wed Aug 16 00:00:00 2000

ACCT #: 0.0.0.1-10923 DATE: 08/06/2000 BILL #: B1-4 ITEM #: B1-4,2 (Cycle forward)

104 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	0.10	0.00	40003	0.00	0.10
NET	-- N/A --	0.10	0.00	-- N/A --	0.00	0.10
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-10923 DATE: 08/06/2000 BILL #: B1-4 ITEM #: B1-4,3 (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	0.32	0.00	40001	0.00	0.32
NET	-- N/A --	0.32	0.00	-- N/A --	0.00	0.32
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

Case Study 4

Plan details:

- Customer created: July 6, 2003
- G/L posting: Last day of the month
- Purchase: Plan 1 – Unlimited Web access with recurring discounts.
  - Quarterly email fee: $45.00 (G/L ID 102)

Copyright © 2023, 2026, Oracle and/or its affiliates.

- Accrual-based accounting

This case study shows how BRM handles multi-month cycle fees. In this case, the quarterly fee is $45.00. BRM amortizes the fee over the life of the 92-day multi-month cycle. Because the customer's first billing date is August 6, 2003, the revenue for July is considered unbilled in the report as shown in Figure 2-17.

Figure 2-17 July G/L Report

Ledger Output for July 2003

General Ledger Data

Type : unbilled_earned
Posting Start : Tue Jul 1 00:00:00 2003
Posting End : Fri Aug 1 00:00:00 2003

ACCT #: 0.0.0.1-17228 DATE: -- N/A -- BILL #: -- N/A -- ITEM #: -- N/A -- (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	-- N/A --	12.72	0.00	-- N/A --	0.00	12.72
NET	-- N/A --	12.72	0.00	-- N/A --	0.00	12.72
DISC	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

General Ledger Data

Type : unbilled_unearned
Posting Start : Tue Jul 1 00:00:00 2003
Posting End : Fri Aug 1 00:00:00 2003

ACCT #: 0.0.0.1-17228 DATE: -- N/A -- BILL #: -- N/A -- ITEM #: -- N/A -- (Cycle forward)

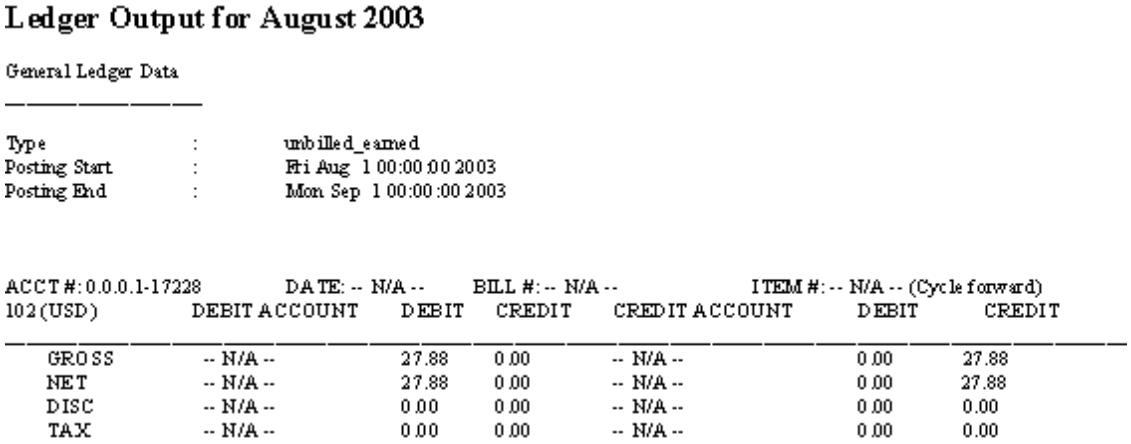
102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	-- N/A --	32.28	0.00	-- N/A --	0.00	32.28
NET	-- N/A --	32.28	0.00	-- N/A --	0.00	32.28
DISC	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

In the August report, you can see that BRM defers revenue based on quarterly fees. Only 26 days of revenue is earned for a quarterly fee that spans 92 days (26/92*45 = 12.71). The remainder is unbilled_unearned revenue.

Figure 2-18 and Figure 2-19 show the G/L reports generated for August.

Copyright © 2023, 2026, Oracle and/or its affiliates.

Figure 2-18 August 1 G/L Report 1



Ledger Output for August 2003

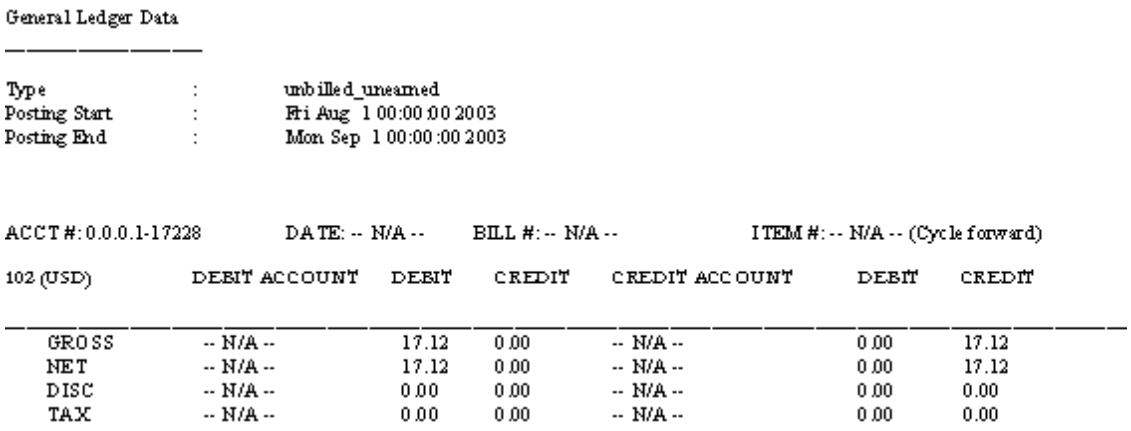
General Ledger Data

Type : unbilled_earned
Posting Start : Fri Aug 1 00:00:00 2003
Posting End : Mon Sep 1 00:00:00 2003

ACCT #: 0.0.0.1-17228 DATE: -- N/A -- BILL #: -- N/A -- ITEM #: -- N/A -- (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	-- N/A --	27.88	0.00	-- N/A --	0.00	27.88
NET	-- N/A --	27.88	0.00	-- N/A --	0.00	27.88
DISC	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

Figure 2-19 August 1 G/L Report 2



General Ledger Data

Type : unbilled_unearned
Posting Start : Fri Aug 1 00:00:00 2003
Posting End : Mon Sep 1 00:00:00 2003

ACCT #: 0.0.0.1-17228 DATE: -- N/A -- BILL #: -- N/A -- ITEM #: -- N/A -- (Cycle forward)

102 (USD)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	-- N/A --	17.12	0.00	-- N/A --	0.00	17.12
NET	-- N/A --	17.12	0.00	-- N/A --	0.00	17.12
DISC	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

In this report, the fee is billed and is reported in the unbilled_earned revenue type. 52 days out of 92 are shown as earned revenue.

Figure 2-20 shows the G/L report generated for September.

Copyright © 2023, 2026, Oracle and/or its affiliates.

Figure 2-20 September G/L Report

Ledger Output for September 2003

General Ledger Data

Type : unbilled_earned
Posting Start : Mon Sep 1 00:00:00 2003
Posting End : Wed Oct 1 00:00:00 2003

ACCT #: 0.0.0.1-17228 DATE: -- N/A -- BILL #: -- N/A -- ITEM #: -- N/A -- (Cycle forward)

102 (US Dollar)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	-- N/A --	42.55	0.00	-- N/A --	0.00	42.55
NET	-- N/A --	42.55	0.00	-- N/A --	0.00	42.55
DISC	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

General Ledger Data

Type : unbilled_unearned
Posting Start : Mon Sep 1 00:00:00 2003
Posting End : Wed Oct 1 00:00:00 2003

ACCT #: 0.0.0.1-17228 DATE: -- N/A -- BILL #: -- N/A -- ITEM #: -- N/A -- (Cycle forward)

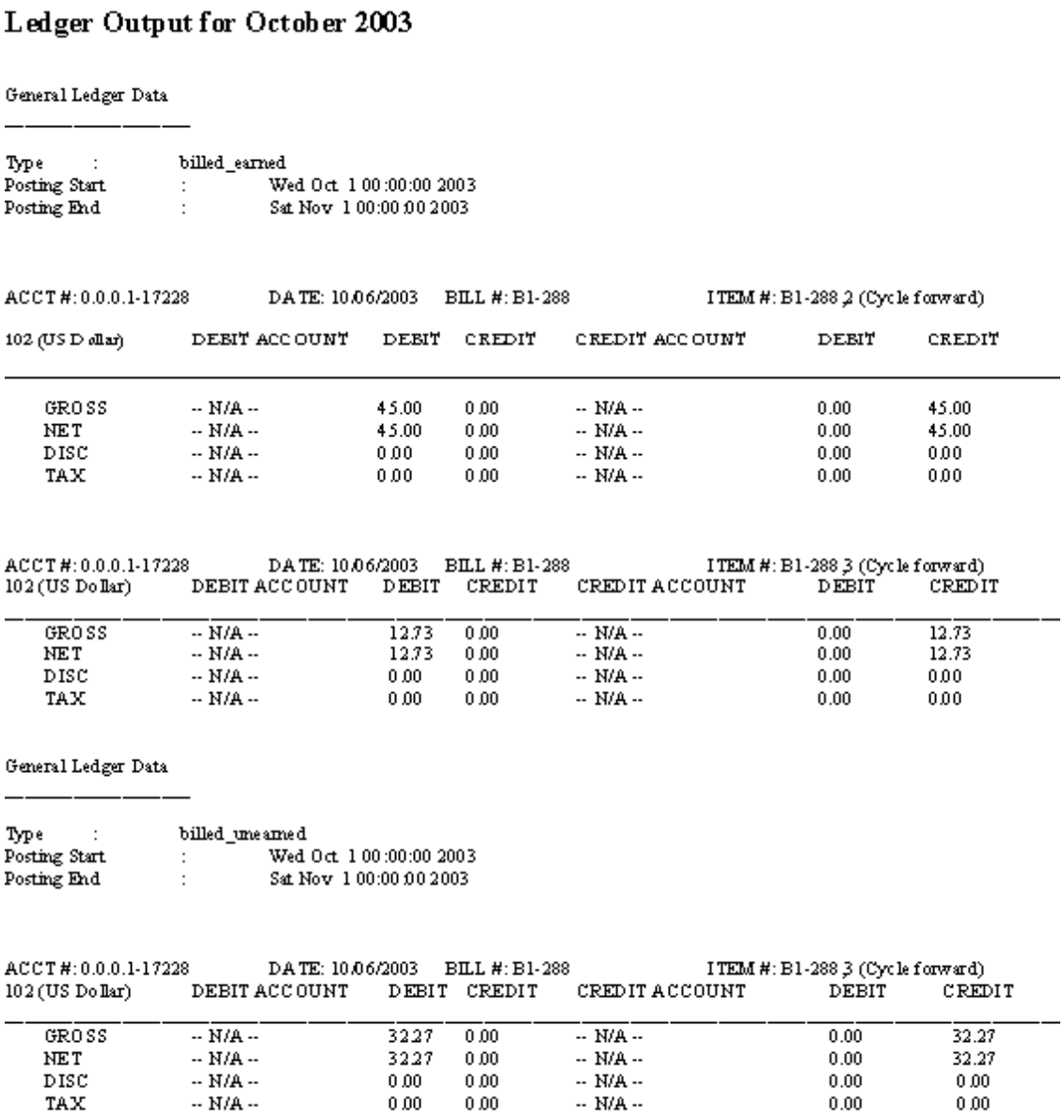
102 (US Dollar)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUN	DEBIT	CREDIT
GROSS	-- N/A --	2.45	0.00	-- N/A --	0.00	2.45
NET	-- N/A --	2.45	0.00	-- N/A --	0.00	2.45
DISC	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

The previously billed date, September 6, 2003, falls outside this report, so the prev_billed_earned option still reports how much is earned during that month. From the 92-day multi-month cycle, the month of October earned 31 days (31/92*45 = 15.18).

Figure 2-21 shows the October G/L report.

Copyright © 2023, 2026, Oracle and/or its affiliates.

Figure 2-21 October G/L Report



Ledger Output for October 2003

General Ledger Data

Type : billed_earned
Posting Start : Wed Oct 1 00:00:00 2003
Posting End : Sat Nov 1 00:00:00 2003

ACCT #: 0.0.0.1-17228 DATE: 10/06/2003 BILL #: B1-288 ITEM #: B1-288 2 (Cycle forward)

102 (US Dollar)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	-- N/A --	45.00	0.00	-- N/A --	0.00	45.00
NET	-- N/A --	45.00	0.00	-- N/A --	0.00	45.00
DISC	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-17228 DATE: 10/06/2003 BILL #: B1-288 ITEM #: B1-288 3 (Cycle forward)

102 (US Dollar)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	-- N/A --	12.73	0.00	-- N/A --	0.00	12.73
NET	-- N/A --	12.73	0.00	-- N/A --	0.00	12.73
DISC	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

General Ledger Data

Type : billed_unearned
Posting Start : Wed Oct 1 00:00:00 2003
Posting End : Sat Nov 1 00:00:00 2003

ACCT #: 0.0.0.1-17228 DATE: 10/06/2003 BILL #: B1-288 ITEM #: B1-288 3 (Cycle forward)

102 (US Dollar)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	-- N/A --	32.27	0.00	-- N/A --	0.00	32.27
NET	-- N/A --	32.27	0.00	-- N/A --	0.00	32.27
DISC	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

Case Study 5

Plan details:

- Customer created: October 6, 2000
- G/L posting: Last day of the month
- Purchase: Plan 1 – Unlimited Web access with discounts. The plan is modified to include a cycle arrears fee instead of a cycle forward fee.
  - Cycle arrears fee: $19.95 (G/L ID 102)
  - Purchase fee: $15.00 (G/L ID 101)

Copyright © 2023, 2026, Oracle and/or its affiliates.

- Accrual-based accounting

The cycle arrears fee, $19.95, is reported as earned when it is billed. The fee was charged on November 6, and the entire amount became billed and earned on that day.

Figure 2-22 shows the October G/L report.

Figure 2-22 October G/L Report

Ledger Output for October 2003

General Ledger Data

Type : billed_earned
Posting Start : Wed Oct 1 00:00:00 2003
Posting End : Sat Nov 1 00:00:00 2003

ACCT #: 0.0.0.1-17228 DATE: 10/06/2003 BILL #: B1-288 ITEM #: B1-288 2 (Cycle forward)

102 (US Dollar)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	-- N/A --	45.00	0.00	-- N/A --	0.00	45.00
NET	-- N/A --	45.00	0.00	-- N/A --	0.00	45.00
DISC	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-17228 DATE: 10/06/2003 BILL #: B1-288 ITEM #: B1-288 3 (Cycle forward)

102 (US Dollar)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	-- N/A --	12.73	0.00	-- N/A --	0.00	12.73
NET	-- N/A --	12.73	0.00	-- N/A --	0.00	12.73
DISC	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

General Ledger Data

Type : billed_unearned
Posting Start : Wed Oct 1 00:00:00 2003
Posting End : Sat Nov 1 00:00:00 2003

ACCT #: 0.0.0.1-17228 DATE: 10/06/2003 BILL #: B1-288 ITEM #: B1-288 3 (Cycle forward)

102 (US Dollar)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	-- N/A --	32.27	0.00	-- N/A --	0.00	32.27
NET	-- N/A --	32.27	0.00	-- N/A --	0.00	32.27
DISC	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

Case Study 6

Plan details:

Copyright © 2023, 2026, Oracle and/or its affiliates.

- Customer created: October 6, 2000
- G/L posting: Monthly G/L posting
- Purchase: Plan 1 – Unlimited Web access with discounts. The plan is modified to have a cycle arrears instead of a cycle forward fee.
  - Monthly email fee: $19.95 (G/L ID 104), with a 10% discount
  - Monthly IP fee: $15.00 (G/L ID 102)
- On-demand billing

Figure 2-23 shows the October G/L report.

Figure 2-23 October G/L Report

Ledger Output for October 2000

General Ledger Data

Type : billed_earned
Posting Start : Sun Oct 1 00:00:00 2000
Posting End : Thu Nov 1 00:00:00 2000

ACCT #: 0.0.0.1-14640 DATE: 10/06/2000 BILL #: B1-24 ITEM #: B1-24,1 (Cycle forward)

102 (US Dollar)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	8.36	0.00	40001	0.00	8.36
NET	-- N/A --	7.52	0.00	-- N/A --	0.00	7.52
DISC	40005	0.84	0.00	10000	0.00	0.84
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-14640 DATE: 10/06/2000 BILL #: B1-23 ITEM #: B1-23,1 (Cycle forward)

104 (US Dollar)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	2.52	0.00	40003	0.00	2.52
NET	-- N/A --	2.52	0.00	-- N/A --	0.00	2.52
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

General Ledger Data

Type : billed_unearned
Posting Start : Sun Oct 1 00:00:00 2000
Posting End : Wed Nov 1 00:00:00 2000

ACCT #: 0.0.0.1-14640 DATE: 10/06/2000 BILL #: B1-24 ITEM #: B1-24,1 (Cycle forward)

102 (US Dollar)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	1.60	0.00	40001-001	0.00	1.60
NET	-- N/A --	1.44	0.00	-- N/A --	0.00	1.44
DISC	40005	0.16	0.00	10000	0.00	0.16
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

ACCT #: 0.0.0.1-14640 DATE: 10/06/2000 BILL #: B1-23 ITEM #: B1-23,1 (Cycle forward)

104 (US Dollar)	DEBIT ACCOUNT	DEBIT	CREDIT	CREDIT ACCOUNT	DEBIT	CREDIT
GROSS	10000	0.48	0.00	40001-001	0.00	0.48
NET	-- N/A --	0.48	0.00	-- N/A --	0.00	0.48
DISC	40005	0.00	0.00	10000	0.00	0.00
TAX	-- N/A --	0.00	0.00	-- N/A --	0.00	0.00

Copyright © 2023, 2026, Oracle and/or its affiliates.

3

Setting Up Your General Ledger

Learn how to set up your general ledger (G/L) in Oracle Communications Billing and Revenue Management (BRM) by creating a chart of accounts (CoA) and G/L IDs.

Topics in this document:

- Setting Up Your General Ledger
- Creating a Chart of Accounts
- Creating G/L IDs
- Assigning G/L IDs to A/R Actions
- Assigning G/L IDs to Prerated Events

Setting Up Your General Ledger

When you set up the general ledger, you do this:

1. Create a list of general ledger (G/L) accounts in the chart of accounts (CoA). See "Creating a Chart of Accounts".
2. Create G/L IDs. See "Creating G/L IDs".
3. Specify the G/L IDs to use for recording financial transactions when a charge or an accounts receivable (A/R) action occurs, such as a write-off. To do so, assign G/L IDs to the following:
   - A/R actions, such as payments and refunds. See "Assigning G/L IDs to A/R Actions".
   - Prerated events. See "Assigning G/L IDs to Prerated Events".
   - Early termination fees, which customers can be charged when they cancel their contract early. See "Creating Subscription Terms" in *PDC Creating Product Offerings*.
   - Tax codes. See "Creating Tax Codes" in *BRM Calculating Taxes*.
   - Charges. See "Configuring Charge Offers" in *PDC Creating Product Offerings*.
   - Discounts. See "Creating Discount Offers" in *PDC Creating Product Offerings*.
   - Deliverables, if you are using deliverable-based revenue recognition. See "Configuring Deliverables" in *PDC Creating Product Offerings*.
4. (For event-based revenue recognition only) Create segments for reporting revenue for groups of customers, such as customers in a geographic region. See "Reporting Revenue for Groups of Customers".
5. 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".
6. Configure how G/L amounts are rounded. See "Rounding General Ledger Data".

Copyright © 2023, 2026, Oracle and/or its affiliates.

Creating a Chart of Accounts

The chart of accounts (CoA) is a list of G/L accounts that your company uses. In BRM, you create a CoA that lists the G/L accounts that are in your company's general ledger on your external financial system.

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".

To create your G/L accounts in BRM:

1. Go to the *BRM_home*/**sys/data/pricing/example** directory.
2. Open the **pin_glchartaccts** file in a text editor. The **pin_glchartaccts** file includes instructions and examples.
3. Add the following information to the file:
   - **coa_id**: The identification number for the CoA. You use this ID when configuring G/L IDs.
   - **coa_name**: A description of the CoA.
4. For each G/L account, add a **gl_coa_acct** line that includes the following information, separated by tabs:
   - The G/L account code or number used in your external financial system.
   - A description of the G/L account.
   - The type of account: asset, equity, expense, liability, or revenue.
   - The status of the account: active or inactive. Inactive accounts cannot be associated with a G/L ID.

   For example:

   ```
   gl_coa_acct    40500     prepaid.A/R     asset      active
   ```
5. Run the following command to load the CoA into a **/config/glid** object in the database:

   load_pin_glchartaccts *pin_glchartaccts_file*

   See "load_pin_glchartaccts".

Sample pin_glchartaccts Entries

The following shows sample entries in a **pin_glchartaccts** file for one CoA with thirteen G/L accounts.

To create multiple CoAs, use multiple sets of the **coa_id**, **coa_name**, and **gl_coa_accts** fields.

```
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    40800     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
```


Copyright © 2023, 2026, Oracle and/or its affiliates.

```
    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
)
```

Creating G/L IDs

To create G/L IDs, you define each G/L ID in the *BRM_home***/sys/data/pricing/example/pin_glid** file and then use the **load_pin_glid** utility to load the G/L IDs into the BRM database.

> **Note**
>
> In a production system, do *not* remove or modify any existing G/L IDs. Doing so could generate inaccurate G/L reports.

To create G/L IDs:

1. Go to the *BRM_home***/sys/data/pricing/example** directory.
2. Open the **pin_glid** file in a text editor. 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.

   > **Note**
   >
   > Segment names are case sensitive and cannot have spaces.

5. For each G/L ID, add the following information:
   - **id**: The identification number for the group of G/L IDs.
   - **taxcode**: (Optional) The tax code used by your tax calculation software.
   - **descr**: A description, such as **Monthly fees**.
   - **type**: The type of G/L ID: standard (**0**), A/R (**1**), Revenue (**2**), or Contract (**3**). See "About G/L IDs" for more information.
6. For each supported revenue type in a G/L ID, add a **gl_acct** line that includes the following information, separated by tabs:
   - The revenue type such as billed, unbilled, and unearned.
   - What the amount represents:
     - **gross**: Reports the total revenue.
     - **disc**: Reports the balance impacts of discounted revenue.

Copyright © 2023, 2026, Oracle and/or its affiliates.

  - **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.

  For example, if you provide a $5 discount on a $30 cycle forward fee: the gross amount is $30, the discounted amount is $5, and the net amount is $25.

- The G/L accounts to use for tracking debits and credits.

For example:

```
gl_acct       billed         gross       purchase.debit        purchase.credit
```

7. Save the **pin_glid** file.
8. 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.

Sample pin_glid Entries

The following shows sample entries in a **pin_glid** file for defining the G/L ID for a purchase fee:

```
#=========================================================================================
# G/L ID for purchase fee
#=========================================================================================
glid
id            10123222
taxcode       PURCHASE_TAXCODE
descr         Purchase Fees
type          2
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       unbilled       gross       purchase.debit        purchase.credit
gl_acct       unbilled       net         purchase.debit        purchase.credit
gl_acct       unbilled       disc        purchase.credit       purchase.debit
```

Assigning G/L IDs to A/R Actions

Some balance impacts are generated by A/R actions, such as payments and refunds, rather than by charges. To have these actions recorded in your general ledger, map each A/R action to a G/L ID.

To map A/R actions to G/L IDs:

1. Go to the *BRM_home***/sys/msgs/reasoncodes** directory.
2. Open the **reasons.en_US** sample file in a text editor.
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 a custom G/L ID.

4. Save the **reasons.en_US** file.
5. Run the following command, which loads this change into the **/config/map_glid** object:

Copyright © 2023, 2026, Oracle and/or its affiliates.

load_localized_strings reasons.*locale*

Assigning G/L IDs to Prerated Events

To assign G/L IDs to prerated events or partially rated events, use the PCM_OP_ACT_POL_SPEC_GLID policy opcode. See "Assigning G/L IDs to Prerated Events" in *BRM Opcode Guide*.

Copyright © 2023, 2026, Oracle and/or its affiliates.

4

Reporting Revenue for Groups of Customers

Learn how to report revenue by customer segments in Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:

- About Reporting Revenue for Groups of Customers
- Changing the Default G/L Segment

About Reporting Revenue for Groups of Customers

> ⚠ **Caution**
>
> Customer segments are supported only for event-based revenue recognition and are included only in individual G/L reports.

By default, individual general ledger (G/L) reports include the revenue data for all customers. You can use *segments* to create individual G/L reports that include data for arbitrary sets of customers, such as for different geographic regions. You specify the segment to run the report on when you generate the individual G/L report.

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 the policy source code in PCM_OP_CUST_POL_PREP_BILLINFO. See "Preparing Bill Unit Data" 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.

Copyright © 2023, 2026, Oracle and/or its affiliates.

By default, reports for parent segments automatically include the data for nested segments underneath them. To prevent a nested segment from being included in a parent report, add the **no_rollup** parameter to the **gl_segment** entry in your **pin_glid** file. For example, the following entries specify to include **.northwest.washington** in parent reports, but to not include **.northwest.oregon**.

```
gl_segment  .
gl_segment  .northwest
gl_segment  .northwest.washington
gl_segment  .northwest.oregon  no_rollup
```

> **Note**
>
> 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

> **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 "Preparing Bill Unit Data" in *BRM Opcode Guide*.

Copyright © 2023, 2026, Oracle and/or its affiliates.

5

General Ledger Configuration Options

Learn how to configure how the Oracle Communications Billing and Revenue Management (BRM) server runs and exports general ledger (G/L) data.

Topics in this document:

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

Disabling G/L Collection in BRM

You can disable G/L collection to improve system performance. When G/L collection is disabled, **/journal** objects are not created.

To disable G/L collection:

1. Go to *BRM_home***/sys/data/config** directory.
2. Use the following command to create an editable XML file from the billing instance of the **/config/business_params** object:

   ```
   pin_bus_params -r BusParamsBilling bus_params_billing.xml
   ```

   This command creates an XML file named **bus_params_billing.xml.out** in your current directory. If you do not want this file in your current directory, specify the path as part of the file name.
3. In **bus_params_billing.xml.out**, set **GeneralLedgerReporting** to **disabled**:

   ```
   <GeneralLedgerReporting>disabled</GeneralLedgerReporting>
   ```

   > ⚠ **Caution**
   >
   > BRM uses the XML in this file to overwrite the existing instance of the **/config/business_params** object. Use care when updating parameters in the file.

4. Save and exit the file.
5. Rename the **bus_params_billing.xml.out** file to **bus_params_billing.xml**.
6. Use the following command to load your changes into the **/config/business_params** object:

   ```
   pin_bus_params bus_params_billing.xml
   ```


Copyright © 2023, 2026, Oracle and/or its affiliates.

You should run this command from the *BRM_home*/**sys/data/config** directory, which includes support files used by the utility. To run it from a different directory, see "pin_bus_params" in *BRM Developer's Guide*.

7. Read the object with the **testnap** utility or the Object Browser to verify that all fields are correct.

   For general instructions on using **testnap**, see "Using the testnap Utility to Test BRM" in *BRM Developer's Guide*. For information on how to use Object Browser, see "Reading Objects" in *BRM Developer's Guide*.

8. Stop and restart the Connection Manager (CM).

   For more information, see "Starting and Stopping the BRM System" in *BRM System Administrator's Guide*.

Disabling Creation of /journal Objects for Noncurrency Balance Impacts

To improve system performance, you can disable creating **/journal** objects for noncurrency balance impacts. To do so, run the **pin_bus_params** utility to change the **NonCurrencyResourceJournaling** business parameter. For more information, see "pin_bus_params" in *BRM Developer's Guide*.

To disable the creation of **/journal** objects for noncurrency balance impacts:

1. Go to *BRM_home*/**sys/data/config**.
2. Use the following command to create an editable XML file from the billing instance of the **/config/business_params** object:

   ```
   pin_bus_params -r BusParamsBilling bus_params_billing.xml
   ```

   This command creates an XML file named **bus_params_billing.xml.out** in your current directory. If you do not want this file in your current directory, specify the path as part of the file name.

3. In **bus_params_billing.xml.out**, set the following entry:

   ```
   <NonCurrencyResourceJournaling>value</NonCurrencyResourceJournaling>
   ```

   where *value* is one of the following:

   - **none**: To disable creation of all **/journal** objects for noncurrency balance impacts
   - **+:***balance_element_ID*,*balance_element_ID*,**...**: To create **/journal** objects for specific noncurrency balance impacts

     Spaces are not allowed. For example:

     ```
     <NonCurrencyResourceJournaling>+:23466,199990,200000</
     NonCurrencyResourceJournaling>
     ```

   - **-:***balance_element_ID*,*balance_element_ID*,**...**: To create **/journal** objects for all *except* specific noncurrency balance impacts

     Spaces are not allowed. For example:

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

   - **all:** To create **/journal** entries for all noncurrency balance impacts. This is the default.

Copyright © 2023, 2026, Oracle and/or its affiliates.

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

> **Caution**
>
> BRM uses the XML in this file to overwrite the existing instance of the **/config/business_params** object. If you delete or modify any other parameters in the file, these changes affect the associated aspects of the BRM configuration.

4. Save and exit the file.
5. Rename the **bus_params_billing.xml.out** file to **bus_params_billing.xml**.
6. Use the following command to load your changes into the **/config/business_params** object:

   ```
   pin_bus_params bus_params_billing.xml
   ```

   You should run this command from the *BRM_home***/sys/data/config** directory, which includes support files used by the utility. To run it from a different directory, see "pin_bus_params" in *BRM Developer's Guide*.
7. Read the object with the **testnap** utility or the Object Browser to verify that all fields are correct.

   For general instructions on using **testnap**, see "Using the testnap Utility to Test BRM" in *BRM Developer's Guide*. For information on how to use Object Browser, see "Reading Objects" in *BRM Developer's Guide*.
8. Stop and restart the CM.

   For more information, see "Starting and Stopping the BRM System" in *BRM System Administrator's Guide*.

Setting the Number of Paying Accounts per G/L Report

To make generating G/L reports more scalable, you can specify the number of paying accounts to include in a G/L report.

To change the default number of paying accounts included in a G/L report:

1. Open the *BRM_home***/apps/pin_billd** file in a text editor.
2. Edit the following entry:

   ```
   pin_ledger_report transaction_grouping 100000
   ```
3. Save and close the file.

Generating G/L Reports Based on the Actual Billing Date

> **Note**
>
> This configuration applies for event-based revenue recognition and individual G/L reports only.

Copyright © 2023, 2026, Oracle and/or its affiliates.

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 G/L reports for the first time, that the G/L 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 G/L report period may again be reported as billed in a new G/L 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 G/L 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 G/L reports, configure the BRM database by using SQL. Specifically, create a view on the ITEM_T table and re-create 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:

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

   where:

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

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

   ```
   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 the ITEM_T table with all of 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:

Copyright © 2023, 2026, Oracle and/or its affiliates.

```
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_REV,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>CREATE OR REPLACE SYNONYM item_t_synonym FOR item_t_gl_view;
   ```

5. Recompile the invalid objects:

   ```
   SQL>exec dbms_utility.compile_schema(USER,false);
   ```

6. Reset the value of OPENED_T of the previously billed items for G/L reports.
7. Exit SQL*Plus.

Segregating Unbilled Revenue by G/L Cycle within a Billing Cycle

> **Note**
>
> This configuration applies for event-based revenue recognition and individual G/L reports only.

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.

Copyright © 2023, 2026, Oracle and/or its affiliates.

> **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** directory.
2. Use the following command to create an editable XML file from the billing instance of the **/config/business_params** object:

   ```
   pin_bus_params -r BusParamsBilling bus_params_billing.xml
   ```

   This command creates an XML file named **bus_params_billing.xml.out** in your current directory. If you do not want this file in your current directory, specify the path as part of the file name.
3. In **bus_params_billing.xml.out**, set **SegregateJournalsByGLPeriod** to **enabled**:

   ```
   <SegregateJournalsByGLPeriod>enabled</SegregateJournalsByGLPeriod>
   ```

   > **Caution**
   >
   > BRM uses the XML in this file to overwrite the existing instance of the **/config/business_params** object. Use care when updating parameters in the file.

4. Save and exit the file.
5. Rename the **bus_params_billing.xml.out** file to **bus_params_billing.xml**.
6. Use the following command to load your changes into the **/config/business_params** object:

   ```
   pin_bus_params bus_params_billing.xml
   ```

   You should run this command from the *BRM_home*/**sys/data/config** directory, which includes support files used by the utility. To run it from a different directory, see "pin_bus_params" in *BRM Developer's Guide*.
7. Read the object with the **testnap** utility or the Object Browser to verify that all fields are correct.

   For general instructions on using **testnap**, see "Using the testnap Utility to Test BRM" in *BRM Developer's Guide*. For information on how to use Object Browser, see "Reading Objects" in *BRM Developer's Guide*.
8. Stop and restart the CM.

   For more information, see "Starting and Stopping the BRM System" in *BRM System Administrator's Guide*.

Copyright © 2023, 2026, Oracle and/or its affiliates.

> **Note**
>
> 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 unchanged.
> - 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_FLD_CALENDAR_DATE             ARRAY [1] allocated 1, used 1
1      PIN_FLD_EFFECTIVE_START_T               TSTAMP [0] (1420099200) 01/01/2015
00:00:00:000 AM
0 PIN_FLD_CALENDAR_DATE           ARRAY [2] allocated 1, used 1
1  PIN_FLD_EFFECTIVE_START_T          TSTAMP [0] (1422777600) 01/02/2015 00:00:00:000 AM
0 PIN_FLD_CALENDAR_DATE             ARRAY [3] allocated 1, used 1
1  PIN_FLD_EFFECTIVE_START_T          TSTAMP [0] (1425196800) 01/03/2015 00:00:00:000 AM
```

After loading, read the configuration object with the **testnap** utility or Object Browser to verify that all fields are correct.

Copyright © 2023, 2026, Oracle and/or its affiliates.

> **Note**
>
> If you want to modify the **/config/gl_calendar** object, delete the existing object using the PCM_OP_DELETE_OBJ opcode and reload it using the PCM_OP_CREATE_OBJ opcode. See "Deleting Objects" in *BRM Developer's Guide*.

Creating Journal Entries for Zero-Balance Events

By default, BRM does not create journal entries for events that result in a zero balance. This means that any event with zero monetary impact is not listed in the JOURNAL_T table, which is used for creating invoices and G/L reports.

For example, emergency calls are free of charge, so they are not displayed in invoices or G/L reports, because they are not picked up by the JOURNAL_T table. Although there is no monetary impact, resources are still consumed which you might want to include in invoices and G/L reports.

You can configure BRM to create journal entries for events that result in a zero balance. You do this by enabling **/journal** objects for zero event balances.

To enable **/journal** objects for zero event balances:

1. Go to *BRM_home***/sys/data/config**.
2. Use the following command to create an editable XML file from the billing instance of the **/config/business_params** object:

   ```
   pin_bus_params -r BusParamsBilling bus_params_billing.xml
   ```

   This command creates an XML file named **bus_params_billing.xml.out** in your current directory. If you do not want this file in your current directory, specify the path as part of the file name.
3. In **bus_params_billing.xml.out**, set **ZeroEventBalances** to **1**:

   ```
   <ZeroEventBalances>1</ZeroEventBalances>
   ```

   > **Caution**
   >
   > BRM uses the XML in this file to overwrite the existing instance of the **/config/business_params** object. Use care when updating parameters in the file.
4. Save and exit the file.
5. Rename the **bus_params_billing.xml.out** file to **bus_params_billing.xml**.
6. Use the following command to load your changes into the **/config/business_params** object:

   ```
   pin_bus_params bus_params_billing.xml
   ```

   You should run this command from the *BRM_home***/sys/data/config** directory, which includes support files used by the utility. To run it from a different directory, see "pin_bus_params" in *BRM Developer's Guide*.
7. Read the object with the **testnap** utility or the Object Browser to verify that all fields are correct.

Copyright © 2023, 2026, Oracle and/or its affiliates.

For general instructions on using **testnap**, see "Using the testnap Utility to Test BRM" in *BRM Developer's Guide*. For information on how to use Object Browser, see "Reading Objects" in *BRM Developer's Guide*.

8. Stop and restart the CM.

   For more information, see "Starting and Stopping the BRM System" in *BRM System Administrator's Guide*.

Copyright © 2023, 2026, Oracle and/or its affiliates.

6

General Ledger Revenue Recognition with Configurable Days Per Month

Learn how to configure Oracle Communications Billing and Revenue Management (BRM) to calculate general ledger (G/L) revenue using a fixed number of days per month for revenue recognition across calendar months.

Topics in this document:

- About G/L Revenue Calculation Using Fixed Monthly Days
- About G/L Revenue Based On Actual Days in the Month
- About G/L Revenue Based On Fixed Days Per Month
- How Fixed Day G/L Revenue Recognition Impacts Journal Entries
- About Rounding G/L Revenue
- Enabling and Using Fixed-Day G/L Revenue Recognition

About G/L Revenue Calculation Using Fixed Monthly Days

By default, BRM calculates General Ledger (G/L) revenue based on the actual number of days in each month. This means the daily revenue amount can vary — for example, February has 28 days, while March has 31. While this variation does not affect full billing cycles (for instance, a $100 monthly fee always resulting in $100 G/L revenue), you may notice inconsistencies when the accounting cycle DOM and G/L cycle DOM are not the same, and the G/L cycle falls between the accounting cycle. This can occur when an account starts, ends, or modifies its subscription mid-cycle.

To address this issue, BRM lets you configure a fixed number of days per month (for example, 30) for calculating G/L revenue. When enabled, the G/L revenue is calculated using the same number of days per month, regardless of the actual calendar month. This approach ensures more consistent revenue recognition for partial, prorated, or non-standard billing cycles.

About G/L Revenue Based On Actual Days in the Month

BRM can calculate G/L revenue using the actual number of days in each month. This method ensures that revenue recognition accurately reflects the specific month's length, such as February having 28 or 29 days, April having 30, and so on.

When applying the actual number of days per month, BRM calculates the G/L revenue for partial billing cycles as follows:

```
Revenue = NumberOfDaysInCycle ÷ NumberOfDaysInMonth x ChargePerMonth
```

For example, assume John's subscription is $100, the G/L DOM is the 10th, and he cancels the subscription on the 18th. In this case, there are 9 days in a month's billing period. Table 6-1 shows the G/L revenue based on which month the subscription ends.

Copyright © 2023, 2026, Oracle and/or its affiliates.

Table 6-1 G/L Revenue Calculation Based on Actual Days per Month

Billing Period	Revenue Calculation	G/L Calculation
Feb 10 - Feb 18	(9 ÷ 28) x $100	$32.14
Mar 10 - Mar 18	(9 ÷ 31) x $100	$29.03
Apr 10 - Apr 18	(9 ÷ 30) x $100	$30.00

Hence, total revenue recognized is $300 but for each month the amount is different.

If the billing cycle spans multiple months, such as in a quarterly cycle BRM calculates the G/L revenue for the first billing period as follows:

```
Revenue = (NumberOfDaysInCycle ÷ NumberOfDaysInMonth) x ChargePerMonth
```

For the last period, BRM calculates the G/L revenue as follows:

```
Revenue = BillingCycleCharge – SumOfPreviousBillingPeriods
```

For example, assume Mary's subscription is $300 per quarter, the G/L DOM is the 1st, and she purchases the subscription on Feb 10. Table 6-2 shows how BRM calculates the G/L revenue for each month in the quarterly billing cycle.

Table 6-2 G/L Revenue Calculation Based on Actual Days per Month for Quarterly Billing Cycle

Billing Period	Revenue Calculation	G/L Calculation
Feb 10 - Feb 18	$ (9/89 x 100)	$64.0449438202
Mar 10 - Mar 18	$ (9/89 x 300 + 22/89 x 300)	$104.4943820225
Apr 10 - Apr 18	$ (9/89 x 300 + 21/89 x 300)	$101.1235955056
May 1 - May 9	$ (9/89 x 300)	$30.3370786517

About G/L Revenue Based On Fixed Days Per Month

To standardize revenue recognition across months of varying month lengths, BRM uses a fixed number of days per month, such as **30.4167**. This value is derived by dividing the total number of days in a year by the number of months: **365 ÷ 12 = 30.4167**. This standardization ensures consistency when calculating G/L revenue for partial billing cycles.

When using a fixed number of days per month, BRM calculates the G/L revenue for partial billing cycles as follows:

```
Revenue = (NumberOfDaysInCycle ÷ FixedDays) x ChargePerMonth
```

For example, assume a customer's subscription is $100 per month, the G/L DOM is the 10th, and they start their subscription on the 18th. In this case, there are 9 days in a month's billing period. Table 6-3 shows that the G/L revenue is the same regardless of the month the subscription starts.

Copyright © 2023, 2026, Oracle and/or its affiliates.

Table 6-3 G/L Revenue Calculation Based on Fixed Days per Month for Partial Billing Cycle

Billing Cycle	Revenue Calculation	G/L Revenue
Feb 10 - Feb 18	(9 ÷ 30.4167) x $100	$29.59
Mar 10 - Mar 18	(9 ÷ 30.4167) x $100	$29.59
Apr 10 - Apr 18	(9 ÷ 30.4167) x $100	$29.59

If the billing cycle spans multiple months, such as in a quarterly cycle, BRM calculates the G/L revenue for the first billing period as follows:

```
Revenue = (NumberOfDaysInCycle ÷ FixedDays) x ChargePerMonth
```

BRM calculates the G/L revenue for the last period as follows:

```
Revenue = BillingCycleCharge – SumOfPreviousBillingPeriods
```

Table 6-4 shows how BRM calculates the G/L revenue for each month, assuming the subscription is purchased on Feb 10, the cost is $300 per quarter, and the G/L DOM is the 1st.

Table 6-4 G/L Revenue Calculation Based on Fixed Days per Month for Quarterly Subscription

Billing Cycle	Revenue Calculation	G/L Revenue
Feb 10 – Feb 28	(19 ÷ 30.4167) x $100	$62.46
Mar 1 – Mar 31	Full cycle	$100.00
Apr 1 – Apr 30	Full cycle	$100.00
May 1 – May 9	$300 - $262.46	$37.54

How Fixed Day G/L Revenue Recognition Impacts Journal Entries

When you enable fixed-day G/L revenue recognition, BRM divides journal entries based on the configured G/L DOM and the account's billing DOM. For example, if the G/L DOM is the 15th, the billing DOM is the 20th, and the subscription purchase date is May 10, BRM creates three journal entries for the account's first billing cycle:

- One entry for revenue from May 10 to May 14
- One entry for revenue from May 15 to May 19
- One entry for revenue from May 20 to June 15

This division is done automatically during cycle event generation using the PCM_OP_BILL_UPDATE_JOURNAL opcode. If necessary, you can implement custom logic through the PCM_OP_GL_POL_PRE_UPDATE_JOURNAL policy opcode, provided that the **CustomJournalUpdate** business parameter is enabled. For more information, see "Customizing G/L Data Stored in /journal Objects" in *BRM Opcode Guide*.

Copyright © 2023, 2026, Oracle and/or its affiliates.

> **Note**
>
> The G/L DOM is read from the **/config/export_gl** object. You must configure the PIN_FLD_GL_SEGMENTS.PIN_FLD_CALENDAR_DOM field for journal splitting to occur.

BRM ensures that the total revenue across the journal entries matches the amount of the billing cycle's G/L revenue. The calculation BRM uses to split the revenue for journal entries varies based on the type of billing cycle.

Example 1: Journal Splitting for a Quarterly Billing Cycle

This example illustrates how revenue is divided into multiple **/journal** objects for a quarterly billing cycle, using a fixed number of days to calculate monthly revenue.

The formula for calculating the G/L revenue for a quarterly billing cycle is as follows:

- The amount recognized for the first period uses this formula:

  ```
  [(FixedDays – MissingDays) ÷ FixedDays] x ChargePerMonth)
  ```

- The amount recognized for the last period uses this formula:

  ```
  BillingCycleCharge – SumOfPreviousBillingPeriods
  ```

where *MissingDays* is the number of days between the start of the previous G/L DOM to the purchase date.

This example assumes the following:

- Subscription: **$300** per quarter
- Purchase date: Feb 10
- Fixed-day value: 30.4167
- G/L DOM: 1
- Billing DOM: 10

In this case, the number of missing days for the first period is **9** (that is, the number of days from Feb 1 through Feb 9). Table 6-5 shows how BRM calculates the G/L revenue for each **/journal** object for this example.

Table 6-5 Journal Splitting for a Quarterly Billing Cycle

Journal Object	Period	Revenue Recognized	Description
1	Feb 10 – Feb 28	$70.41	Calculation: [(30.4167 – 9) ÷ 30.4167] x $100
2	Mar 01 – Mar 31	$100.00	Full month of revenue
3	Apr 01 – Apr 30	$100.00	Full month of revenue
4	May 01 – May 10	$29.59	Calculation: $300 - $270.41 = $29.59

Example 2: Journal Splitting for a Long Cycle

This example illustrates how revenue is divided into multiple **/journal** objects for a long cycle. A long cycle occurs when BRM adds a partial cycle to the next cycle rather than considering it

Copyright © 2023, 2026, Oracle and/or its affiliates.

separate. For more information about long cycles, see "Specifying How to Handle Partial Accounting Cycles" in *BRM Configuring and Running Billing*.

The formula for calculating the G/L revenue for a long billing cycle is as follows:

- The amount recognized for the first period uses this formula:

  ```
  LongBillingCycleCharge – ChargePerMonth
  ```

- The amount recognized for the second period uses this formula:

  ```
  [(FixedDays – MissingDays) ÷ FixedDays] x ChargePerMonth)
  ```

- The amount recognized for the last period uses this formula:

  ```
  ChargePerMonth – SecondBillingPeriodAmount
  ```

This example assumes the following:

- Subscription: **$100** per month
- Long billing cycle charge: $116.13
- Purchase date: Feb 5
- Fixed-day value: 30.4167
- G/L DOM: 1
- Billing DOM: 10

In this case, the long billing cycle goes from Feb 5 thru Mar 10. The number of missing days for the second period is 9 (that is, the number of days from Feb 1 through Feb 9). Table 6-6 shows how BRM calculates the G/L revenue for each **/journal** object for this example.

Table 6-6 Journal Splitting for a Long Billing Cycle

Journal Object	Period	Revenue Recognized	Description
1	Feb 05 – Feb 9	$16.13	$116.13 -- $100
2	Feb 10 – Mar 01	$70.41	Calculation: [(30.4167 – 9) ÷ 30.4167] x $100
3	Mar 01 – Mar 10	$29.59	Calculation: $100 – 70.41

The total recognized revenue is $116.13, distributed across three journal entries when the account creation date falls outside the standard billing cycle.

Example 3: Journal Splitting for a Short Cycle

This example illustrates how revenue is divided into multiple **/journal** objects for a short cycle, which means the billing cycle is less than a month and goes from the purchase date to the billing DOM. For more information about short cycles, see "Specifying How to Handle Partial Accounting Cycles" in *BRM Configuring and Running Billing*.

The formula for calculating the G/L revenue for a short cycle is as follows:

- The amount recognized for the first period uses this formula:

  ```
  [((FixedDays – MissingDays) ÷ FixedDays) x ChargePerMonth)] – (ChargePerMonth – ProratedCharge)
  ```

- The amount recognized for the last period uses this formula:

  ```
  ChargePerMonth – [((FixedDays – MissingDays) ÷ FixedDays) x ChargePerMonth)]
  ```


Copyright © 2023, 2026, Oracle and/or its affiliates.

The example assumes the following:

- Subscription: **$100** per month
- Pro-rated billing charge: $82.14
- Purchase date: Feb 15
- Fixed-day value: 30.4167
- G/L DOM: 1
- Billing DOM: 10

In this case, the short cycle goes from Feb 15 through Mar 10. Table 6-7 shows how BRM calculates the G/L revenue for each **/journal** object for this example.

Table 6-7 Journal Splitting for a Short Billing Cycle

Journal Object	Period	Revenue Recognized	Description
1	Feb 15 – Mar 1	$52.55	Calculation: [((30.4167 – 9) ÷ 30.4167) x $100) – (100-82.14)] = $70.41 - $17.86 = $52.55
2	Mar 01 – Mar 10	$29.59	Calculation: ($100 – [((30.4167 – 9) ÷ 30.4167) x $100))] = $100 -- $70.41 = $29.59

The total recognized revenue is $82.14, distributed across two journal entries.

About Rounding G/L Revenue

BRM rounds G/L revenue amounts and journal entry amounts as part of the revenue calculation and journal entry generation processes.

BRM applies rounding according to the system's currency and rounding configuration. Rounding is applied as follows:

- **Revenue Calculation:**

  When BRM calculates G/L revenue for each billing period or prorated cycle, the revenue amount is rounded based on the defined decimal precision of the account's currency.

- **Journal Entry Creation:**

  When creating each **/journal** object, BRM rounds the revenue amount according to the currency settings and rounding rules. BRM ensures that the total of all journal entries for a billing cycle equals the calculated G/L revenue for that cycle. Any rounding differences are adjusted in the last journal entry for the period.

BRM uses the currency of the account to determine the decimal precision for rounding. BRM supports both zero-decimal and multi-decimal currencies.

For configuration steps and details on balance impact rounding, see "Configuring Balance Impact Rounding" in *PDC Creating Product Offerings*.

Enabling and Using Fixed-Day G/L Revenue Recognition

To enable and use fixed-day G/L revenue recognition in BRM, you must perform a few configuration tasks to ensure that BRM applies a consistent number of days per month when

Copyright © 2023, 2026, Oracle and/or its affiliates.

calculating G/L revenue and splits journal entries correctly for partial or non-standard billing cycles.

The process for enabling and using fixed-day G/L revenue recognition involves these high-level steps:

1. **Set the G/L Day of Month (DOM)**: Define the G/L DOM values that control how journal entries are split across periods.

   See "Configuring the G/L DOM".

2. **Customize how to split journal data**: Optionally, customize how to split the **/journal** objects. To do so, enable the **CustomJournalUpdate** business parameter and customize the PCM_OP_GL_POL_PRE_UPDATE_JOURNAL policy opcode.

   See "Customizing G/L Data Stored in /journal Objects" in *BRM Opcode Guide*.

3. **Enable fixed-day G/L revenue recognition**: Update the appropriate business parameter to activate the feature.

   See "Configuring Fixed-Day G/L Revenue Recognition".

Configuring the G/L DOM

To split journal entries, BRM needs to access each segment's G/L DOM from the PIN_FLD_CALENDAR_DOM field in the **/config/export_gl** object. Ensure that the object contains correct G/L DOM values for each segment supported by your company.

> **Note**
>
> If the G/L DOM is missing, fixed-day calculations are not applied.

If you need to add or change any segment's G/L DOM, update the **pin_config_export_gl.xml** file and then load it into the **/config/export_gl** object using the load utility.

To configure the G/L DOM:

1. Open the *BRM_home***/sys/data/config/pin_config_export_gl.xml** file in an XML editor or a text editor.
2. For each segment, set the **<DayofMonth>** tag to the appropriate value.
3. Save and close the file.
4. Use this command to load the G/L configuration information into the **/config/export_gl** object in the BRM database:

   ```
   load_pin_config_export_gl pin_config_export_gl.xml
   ```

   For more information, see "load_pin_config_export_gl".

5. Validate that your G/L configuration file was loaded into the database correctly. See Validating and Troubleshooting Exported G/L Reports.

Configuring Fixed-Day G/L Revenue Recognition

The **SmoothedGLReportingBasedOnFixedDays** business parameter controls how revenue is recognized for G/L reporting. When enabled, it allows revenue to be calculated using a fixed number of days per month (such as 30) to ensure consistent monthly recognition, regardless of the actual calendar month length.

Copyright © 2023, 2026, Oracle and/or its affiliates.

You can set this parameter to any fixed value, such as 29, 30, 30.5, or 30.4167. For example, you might use 30.4167 because it approximates the average number of days per month in a 365-day year (365 ÷ 12).

By default, this business parameter is set to **0.0**, which means the feature is disabled and revenue is recognized based on the actual number of days in each month.

To enable fixed-day G/L revenue recognition:

1. Go to *BRM_home***/sys/data/config**.
2. Use the following command to create an editable XML file from the **Billing** instance of the **/config/business_params** object:

   ```
   pin_bus_params -r BusParamsBilling bus_params_billing.xml
   ```

   This command creates an XML file named **bus_params_billing.xml.out** in your current directory. If you do not want this file in your current directory, specify the path as part of the file name.
3. In **bus_params_billing.xml.out**, set **SmoothedGLReportingBasedOnFixedDays** to the number of days in each month for calculating the G/L revenue. For example, this specifies to calculate the G/L revenue based on 30 days in each month:

   ```
   <SmoothedGLReportingBasedOnFixedDays>30</SmoothedGLReportingBasedOnFixedDays>
   ```

   > ⚠ **Caution**
   >
   > BRM uses the XML in this file to overwrite the existing instance of the **/config/business_params** object. If you delete or modify any other parameters in the file, these changes affect the associated aspects of the BRM configuration.

4. Save and exit the file.
5. Rename the **bus_params_billing.xml.out** file to **bus_params_billing.xml**.
6. Use the following command to load your changes into the **/config/business_params** object:

   ```
   pin_bus_params bus_params_billing.xml
   ```

   You should run this command from the *BRM_home***/sys/data/config** directory, which includes support files used by the utility. To run it from a different directory, see "pin_bus_params" in *BRM Developer's Guide*.
7. Read the object with the **testnap** utility or the Object Browser to verify that all fields are correct.

   For general instructions on using **testnap**, see "Using the testnap Utility to Test BRM" in *BRM Developer's Guide*. For information on how to use Object Browser, see "Reading Objects" in *BRM Developer's Guide*.
8. Stop and restart the Connection Manager (CM).

   For more information, see "Starting and Stopping the BRM System" in *BRM System Administrator's Guide*.

Copyright © 2023, 2026, Oracle and/or its affiliates.

7

Rounding General Ledger Data

Learn how to configure Oracle Communications Billing and Revenue Management (BRM) to round the balance impacts included in general ledger (G/L) reports.

Topics in this document:

- About Rounding and G/L Reports
- Rounding G/L Report Data After Billing
- Rounding G/L Report Data Prior to Billing

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. See "Specifying Rounding in Balance Impacts" in *PDC Creating Product Offerings*.

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 "Configuring Resource Rounding" in *BRM Setting Up Pipeline Pricing.*

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 7-1 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:

Copyright © 2023, 2026, Oracle and/or its affiliates.

Figure 7-1 Rounding G/L Report Data after Billing

	Item	Journal entry A	Journal entry B
Balance impacts after rounding for rating to a precision of 3:	1.111	1.111	
	2.131	2.132	
	3.141	3.145	
	1.223		1.223
	4.112		4.112
Total	11.723	6.388	5.335
A/R rounded total	11.72	6.39	5.34
	Billing total = 11.72	G/L total = 11.73	

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.

Copyright © 2023, 2026, Oracle and/or its affiliates.

8

Generating General Ledger Reports

Learn how to generate general ledger (G/L) reports in Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:

- About G/L Reports
- Overview of the G/L Report Process
- Running G/L Jobs in Business Operations Center
- Generating G/L Reports Manually

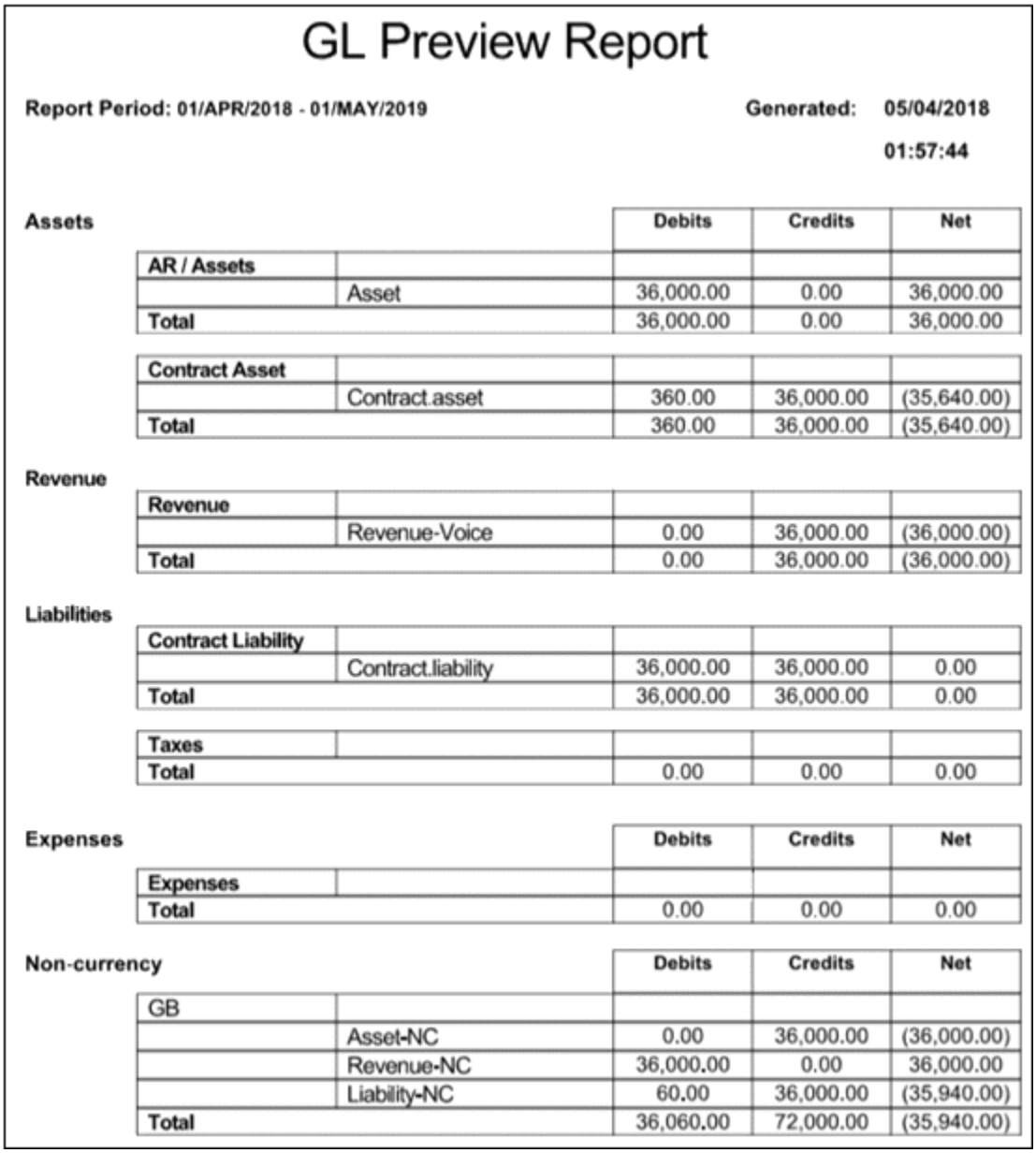
About G/L Reports

G/L reports summarize the debits and credits to your financial accounts during a given period. In BRM, you can generate reports for previewing your G/L data, reports that you can post to your corporate general ledger, and individual reports for each revenue type.

Figure 8-1 shows an example of the data that is included in preview reports.

Copyright © 2023, 2026, Oracle and/or its affiliates.

Figure 8-1 Sample G/L Preview Report



GL Preview Report

Report Period: 01/APR/2018 - 01/MAY/2019 Generated: 05/04/2018 01:57:44

Assets		Debits	Credits	Net
AR / Assets				
	Asset	36,000.00	0.00	36,000.00
Total		36,000.00	0.00	36,000.00
Contract Asset				
	Contract asset	360.00	36,000.00	(35,640.00)
Total		360.00	36,000.00	(35,640.00)
Revenue				
Revenue				
	Revenue-Voice	0.00	36,000.00	(36,000.00)
Total		0.00	36,000.00	(36,000.00)
Liabilities				
Contract Liability				
	Contract liability	36,000.00	36,000.00	0.00
Total		36,000.00	36,000.00	0.00
Taxes				
Total		0.00	0.00	0.00

Expenses		Debits	Credits	Net
Expenses				
Total		0.00	0.00	0.00

Non-currency		Debits	Credits	Net
GB				
	Asset-NC	0.00	36,000.00	(36,000.00)
	Revenue-NC	36,000.00	0.00	36,000.00
	Liability-NC	60.00	36,000.00	(35,940.00)
Total		36,060.00	72,000.00	(35,940.00)

About the Monthly Export Report

If you are using a deliverable-based revenue recognition scheme, the monthly export report is the one that you post to your corporate general ledger on your external financial system. The report includes aggregated data about your unbilled, billed, and earned revenue by general ledger account. The report covers data for one month, starting on a set day of the month such as the 5th.

Copyright © 2023, 2026, Oracle and/or its affiliates.

> **Note**
>
> Because the report aggregates data by general ledger account, the report will always display 0 in the G/L ID field.

When you generate a monthly export report, it is saved to the BRM database and also to an XML output file for posting to your corporate general ledger.

The monthly export report is marked internally as having been generated for that month. This ensures that:

- Only one report XML output file is created for each month. This prevents double-counting in the corporate general ledger on your external financial system. A report XML output file is created only when one does not already exist for a specific month.
- A report XML output file is created for every month. When you generate a monthly export report, a report XML output file is automatically generated for any previous months that you have missed.

  For example, assume your monthly reports start on the 1st, and there are XML files for January and February. When you generate a monthly export report on May 10, two XML output files are created: one for March and one for April.

To ensure that your monthly export reports include the revenue for an entire month, schedule them to run at least 2 days after your G/L reports' starting day of month. For example, if the starting day of month is the 5th, schedule your G/L reports to run on the 7th or later. This provides time for BRM to process transactions that tend to consolidate at the end of the reporting period, such as contract cancellations and expirations.

About the Preview G/L Report

You use the preview report to validate your G/L data before creating the monthly export report. The preview G/L report contains the same data as your monthly export report, but it covers a different period of time and should not be posted to your corporate general ledger.

> **Note**
>
> Because the preview report aggregates data by general ledger account, the report will always display 0 in the G/L ID field.

A preview G/L report covers data from the end date of your last monthly export report to the day the report is generated (exclusive). For example, assume that the last monthly export report that you generated was for Feb 5 through Mar 4. If you generate a preview report on:

- Mar 20th: The report contains data from Mar 5 through Mar 19.
- Mar 30th: The report contains data from Mar 5 through Mar 29.
- Apr 10th: The report contains data from Mar 5 through Apr 9.

When you generate preview reports, they are saved in **/ledger_report** objects in the BRM database and can be viewed in Oracle Analytics Publisher. The reports are also exported to an XML output file, so you can view them in an external reporting system.

Copyright © 2023, 2026, Oracle and/or its affiliates.

> **⚠ Caution**
>
> Do not post preview G/L reports to the corporate general ledger on your external financial system. Doing so could cause double-counting in your corporate general ledger.

About Individual G/L Reports

If you are using an event-based revenue recognition scheme, you generate individual G/L reports for each type of revenue, such as unbilled revenue, unbilled unearned revenue, unbilled earned revenue, and so on. Each report covers data starting and ending on the dates you specify.

When you generate individual G/L reports, they are saved in **/ledger_report** objects in the BRM database and are also exported to an XML output file.

Overview of the G/L Report Process

The process for generating and posting your general ledger data involves these high-level steps:

1. Configuring the file naming convention and location for your G/L report XML output files. See "Configuring General Ledger Report XML Output Files".
2. Generating G/L reports in one of the following ways:
   - Using the Business Operations Center application. See "Running G/L Jobs in Business Operations Center".
   - Manually running the **pin_ledger_report** utility. See "Generating G/L Reports Manually".
   - Using a custom client application. To do so, configure your client to call the PCM_OP_JOB_EXECUTE opcode. See "Running Business Operations Jobs" in *BRM Opcode Guide*.
3. Previewing your reports in Oracle Analytics Publisher to validate that your G/L setup is generating expected results.
4. Retrieving your G/L report XML output files. See the following:
   - About Preview and Monthly Export Report XML Output Files
   - About Individual Report XML Output Files
5. Uploading the report XML output files to your external financial system and then posting them to your corporate general ledger.

Running G/L Jobs in Business Operations Center

You can generate G/L reports by running a general ledger job in Business Operations Center.

You can specify to generate G/L reports based on the purpose for the report:

- **Preview:** Generates a report for previewing your G/L data.
- **Export for Posting:** Generates your monthly export report for deliverable-based revenue recognition.

Copyright © 2023, 2026, Oracle and/or its affiliates.

- **Individual:** Generates a separate G/L report for each type of revenue that you pick. Use this report for event-based revenue recognition.

  You can generate individual reports based on these attributes:

  - **Reporting Period:** Includes revenue from bills that were created between the specified dates.

    For one-time jobs, you choose specific dates for the start and end of the reporting period, such as September 10 through September 30.

    For repeating jobs, you can adjust the reporting period by adding or subtracting days relative to the day the job runs. For example, if the general ledger job runs on the 30th day of the month, you can specify to end the reporting period one day earlier, on the 29th day of the month.

  - **Types of Reports to Generate:** Generates separate reports for one or more of these revenue types: unbilled, unbilled earned, unbilled unearned, previous billed earned, billed, billed earned, and billed unearned.

  - **Include in Report:** Specifies whether to include currency amounts, noncurrency amounts, or both. If you select breakdown by charge type, your reports will include the totals for each type of charge.

  - **Post G/L revenue:** Posts G/L data to prevent backdating events that occurred before the posting date.

    You typically run a G/L report without posting when you want to see the data before the end of the normal G/L cycle. Your regularly scheduled G/L reports should post data.

For more information, see "Generating General Ledger Reports" in Business Operations Center Help.

Generating G/L Reports Manually

You can generate G/L reports manually by running the **pin_ledger_report** utility. You can use the utility to do the following:

- Create a preview report for deliverable-based revenue recognition. See "Creating Preview G/L Reports with pin_ledger_report".
- Create your monthly export report for deliverable-based revenue recognition. See "Creating Monthly Export G/L Reports with pin_ledger_report".
- Create individual reports for event-based revenue recognition. See "Creating Individual G/L Reports with pin_ledger_report".

For more information about the utility, see "pin_ledger_report".

Creating Preview G/L Reports with pin_ledger_report

If you are using deliverable-based revenue recognition, you can create a preview report for validating your G/L data.

To create a preview G/L report, run the following command:

```
pin_ledger_report -mode export -preview
```

The utility saves your preview report to the database, so you can view it in Oracle Analytics Publisher. It also exports the preview report to an XML file using the file name and location you specified in the **pin_config_export_gl.xml** file. See "Configuring Preview and Monthly Report XML Output Files".

Copyright © 2023, 2026, Oracle and/or its affiliates.

Creating Monthly Export G/L Reports with pin_ledger_report

If you are using deliverable-based revenue recognition, you can create a monthly export G/L report that you can post to the corporate general ledger on your external financial system.

To create a monthly export G/L report, run the following command:

```
pin_ledger_report -mode export -aggregate
```

The utility exports the report to an XML file using the file name and location you specified in the **pin_config_export_gl.xml** file. See "Configuring Preview and Monthly Report XML Output Files".

Creating Individual G/L Reports with pin_ledger_report

If you are using event-based revenue recognition, you can create an individual G/L report for each revenue type specified in your **pin_config_export_gl.xml** file. For example, you could create separate G/L reports for billed revenue, unbilled revenue, billed unearned revenue, and so on.

The options for creating individual reports are:

- To create an individual G/L report that you can view in Oracle Analytics Publisher, run **pin_ledger_report** in the **run_report** mode. For example, the following command creates a report on events that occurred from May 25 to the current date:

  ```
  pin_ledger_report -mode run_report -start 05/25/20
  ```

  For more information, see "run_report Mode Syntax".

- To export individual reports to XML files, run **pin_ledger_report** in the **export** mode. For example, the following command creates an XML report file for the EastCoastEnterprises segment:

  ```
  pin_ledger_report -mode export -segment EastCoastEnterprises
  ```

  For more information, see "export Mode Syntax".

- To post G/L data to maintain the validity of revenue data, run **pin_ledger_report** in the **post_only** mode. For example, the following command posts data up to the current date for the root segment:

  ```
  pin_ledger_report -mode post_only
  ```

  Posting G/L data prevents BRM from backdating events that occurred before the posting date; for example, by performing backdated accounts receivable operations.

  For more information, see "post_only Mode Syntax".

- To display a list of previously run reports, run **pin_ledger_report** in the **list_previous** mode. For example, the following command lists the reports that were created from June 25 to the current date:

  ```
  pin_ledger_report -mode list_previous -start 06/25/20
  ```

  For more information, see "list_previous Mode Syntax".

Copyright © 2023, 2026, Oracle and/or its affiliates.

Requirement for Creating a Report for Billed Revenue

If you are creating an individual 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/@databaseAlias
Enter password: password

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.

Copyright © 2023, 2026, Oracle and/or its affiliates.

9

Configuring General Ledger Report XML Output Files

Learn how to configure your general ledger (G/L) XML output files in Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:

- Configuring Preview and Monthly Report XML Output Files
- Configuring Individual Report XML Output Files
- Validating and Troubleshooting Exported G/L Reports

Configuring Preview and Monthly Report XML Output Files

To configure the file name and location of the XML output files for your preview and monthly G/L reports:

1. Open the *BRM_home*/**sys/data/config/pin_config_export_gl.xml** file in an XML editor or a text editor.
2. Edit the following tags:
   - **OutputDirectory**: Specify the directory in which to save your XML output files.
   - **SourceSystemID**: Specify a unique ID for the BRM system containing the G/L data. Use this tag if you export reports from multiple BRM systems and the external system needs to identify the source of the data.
   - **FileNamePrefix**: Optionally, specify a prefix to add to the beginning of the XML output file name. You can use a prefix to identify the type of report.

     If the tag is empty or missing from the file, no prefix is added to the XML file name.
   - **ReportInitialStartDate**: Specify the start date for the first time a preview or monthly export report is generated.

     > **Note**
     >
     > The other XML tags in this file are ignored when you generate a preview or monthly export report.
3. Save and close the file.
4. Use this command to load the G/L configuration information into the **/config/export_gl** object in the BRM database:

   ```
   load_pin_config_export_gl pin_config_export_gl.xml
   ```

   For more information, see "load_pin_config_export_gl".
5. Validate that your G/L configuration file was loaded into the database correctly. See "Validating and Troubleshooting Exported G/L Reports".

Copyright © 2023, 2026, Oracle and/or its affiliates.

About Preview and Monthly Export Report XML Output Files

When you generate a monthly export or preview report, their XML files are saved according to the configuration in the **pin_config_export_gl.xml** file.

Monthly export report files are saved to *outputDirectory* and use the following file naming convention:

*prefix*_**export_aggr**_*endDate_startDate_reportID*.**xml**

Preview report files are saved to *outputDirectory*/**PREVIEW** and use the following file naming convention:

*prefix*_**preview_aggr**_*endDate_startDate_reportID*.**xml**

where:

- *outputDirectory* is the directory specified in the **OutputDirectory** tag of the **pin_config_export_gl.xml** file.
- *Prefix* is an optional prefix that is added to your report XML files. You specify the prefix by using the **FileNamePrefix** tag in the **pin_config_export_gl.xml** file.
- *endDate* is the last day (exclusive) covered by the report in *YYYYMMDD* format.
- *startDate* is the first day covered in the report in *YYYYMMDD* format.
- *reportID* is 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**.

For example, a file named **export_aggr_20200601_20200501_0.0.0.1-73061-4.xml** is a monthly export report that contains data from May 1, 2020 through May 31, 2020.

Sample Configuration File for Preview and Monthly Export Reports

The following shows sample **pin_config_export_gl.xml** file entries for configuring preview and monthly export reports:

```
<GLReportConfiguration>
    <SourceSystemID>India</SourceSystemID>
    <OutputDirectory>
        $PIN_HOME/GL_Output
    </OutputDirectory>
    <FileNamePrefix>
        ASC_GL_Report_
    </FileNamePrefix>
    <ReportInitialStartDate>
        08/30/2020
    </ReportInitialStartDate>
</GLReportConfiguration>
```

In this case, the monthly export report file would be saved to the **$PIN_HOME/GL_Output** directory and have a file name of **ASC_GL_Report_export_aggr_20200918_20200830_0.0.0.1-73061-4.xml**.

Copyright © 2023, 2026, Oracle and/or its affiliates.

The preview report file would be saved to the **$PIN_HOME/GL_Output/PREVIEW** directory and have a file name of **ASC_GL_Report_preview_aggr_20200918_20200830_0.0.0.1-73061-4.xml**.

Configuring Individual Report XML Output Files

To configure the file name, location, and content of the XML output files for your individual G/L reports:

1. Open the *BRM_home***/sys/data/config/pin_config_export_gl.xml** file in an XML editor or a text editor.
2. Edit the following tags for specifying the name and location of your XML output files:
   - **OutputDirectory**: Specify the directory in which to store your XML output files.
   - **SourceSystemID**: Specify a unique ID for the BRM system containing the G/L data. Use this tag if you export reports from multiple BRM systems and the external system needs to identify the source of the data.
   - **FileNamePrefix**: Optionally, specify a prefix to append to the XML file name. You can use a prefix to identify the type of report.
3. Specify the content to include in your reports by editing the tags in Table 9-1:

> **Note**
>
> If you are using Business Operations Center to generate individual reports, the values you enter in Business Operations Center override the values you enter here.

Table 9-1 Configuration Tags for G/L Reports

Use This Tag:	To Do This:
ReportInitialStartDate	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 reports 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="."). All G/L segments to be reported use this date. To set a different start date for a specific G/L segment, use the Segment tag and specify a different initial start date for that G/L segment. This date overrides the root segment start date.
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: **** **Important:** If you specify multiple instances of the same segment, each one must report on a different set of revenue types.

Copyright © 2023, 2026, Oracle and/or its affiliates.

Table 9-1 (Cont.) Configuration Tags for G/L Reports

Use This Tag:	To Do This:
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. See "Setting Up Schedules for Individual G/L Reports".

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:
	`<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:
	`<Frequency>Specific Dates</Frequency>`
	`<Date>`
	`<DayofMonth>-29</DayofMonth>`
	`<Month>-02-</Month>`
	`</Date>`
	`<Date>`
	`<DayofMonth>-31</DayofMonth>`
	`<Month>-05-</Month>`
	`</Date>`

Copyright © 2023, 2026, Oracle and/or its affiliates.

Table 9-1 (Cont.) Configuration Tags for G/L Reports

Use This Tag:	To Do This:
RevenueType	Specify the revenue type for the G/L report: • **Billed** • **Billed earned** • **Prior billed earned** (**Prior billed earned** revenue is also known as **Previously billed earned** revenue.) • **Unbilled** • **Unbilled unearned** • **Unbilled earned** • **Billed unearned** **Important:** If you specify multiple instances of the same G/L segment, each one must report on a different set of revenue types.
ReportLevel	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 "Customizing G/L Reports for Export" in *BRM Opcode Guide*.
ResourceType	Specify the balance element type to include in the G/L report: • **Monetary** • **Non-monetary** • **All**
IncludeNonMonetary	If **ResourceType** is **Non-monetary** or **All**, specify the balance element ID of the nonmonetary balance element to include. Use a separate **ResourceID** 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.
ExcludeNonMonetary	If **ResourceType** is **Non-monetary** or **All**, specify the ID of the nonmonetary balance element to exclude. Use a separate **ResourceID** 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.

4. Save and close the file.
5. Use this command to load the G/L configuration information into the **/config/export_gl** object in the BRM database:

   ```
   load_pin_config_export_gl pin_config_export_gl.xml
   ```

6. Validate that your G/L configuration file was loaded into the database correctly. See "Validating and Troubleshooting Exported G/L Reports".

Setting Up Schedules for Individual G/L Reports

The frequency for your individual 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.

Copyright © 2023, 2026, Oracle and/or its affiliates.

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>
```

About Individual Report XML Output Files

When you generate individual 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 file naming convention:

*RevenueType_ReportEndDate_ReportStartDate_ReportId***.xml**

where:

- *RevenueType* is the abbreviation for the revenue type being reported, such as **ue** for Unbilled earned and **be** for Billed earned.
- *ReportEndDate* is the 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 Table 9-2.

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 do not 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 9-2 describes the XML tags in the G/L export files.

Copyright © 2023, 2026, Oracle and/or its affiliates.

Table 9-2 XML Tags in G/L Reports

XML Element	Description
SourceSystemID	The unique ID for the database containing the G/L data.
ReportId	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**.
RevenueType	Revenue type (billed, unbilled, and so on)
BRM_GL_Segment	The G/L segment being reported on.
ReportCreatedTime	Report run time.
PeriodStartTime	The start date and time of the revenue reporting period.
PeriodStartEnd	The end date and time of the revenue reporting period.
RevenueAmounts	The revenue reported for a specific G/L segment: • The balance element ID and its associated BRM G/L ID. • A list of revenue accounts and their credit and debit amounts. For example, **ARGrossAccount name="monthly.debit"**.

Sample Configuration File for Individual G/L Reports

The following example shows the configuration of **pin_config_export_gl.xml** for the root (**.**) segment in an individual G/L report:

```
<GLReportConfiguration>
  <SourceSystemID>Germany</SourceSystemID>
  <OutputDirectory>
     /$PINHOME/GL_output
  </OutputDirectory>

  <FileNamePrefix>
     test
  </FileNamePrefix>

  <ReportInitialStartDate>
     <Segment name=".">
         <Year>2011</Year>
         <Month>--07--</Month>
         <Day>---01</Day>
     </Segment>
     <Segment name=".east">
         <Year>2011</Year>
         <Month>--07--</Month>
         <Day>---01</Day>
     </Segment>
  </ReportInitialStartDate>

  <SegmentList>
     <Segment name=".">
         <Frequency>Daily</Frequency>
         <RevenueTypeList>
             <RevenueType>Billed earned </RevenueType>
             <RevenueType>Unbilled earned</RevenueType>
             <RevenueType>Billed</RevenueType>
             <RevenueType>Unbilled</RevenueType>
             <RevenueType>Previously billed earned</RevenueType>
```


Copyright © 2023, 2026, Oracle and/or its affiliates.

```
        </RevenueTypeList>
        <ReportLevel>Detailed</ReportLevel>
        <ResourceType>All</ResourceType>
         <!--ExcludeNonMonetary>
            <ResourceID>111111111</ResourceID>
         </ExcludeNonMonetary-->
     </Segment>
   </SegmentList>
  </GLReportConfiguration>
```

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.

```
<BusinessConfiguration>
 <GLReportConfiguration>

   <SourceSystemID>
       California
   </SourceSystemID>

   <OutputDirectory>
      /users/gluser/exported
   </OutputDirectory>

   <FileNamePrefix>
      USA_
   </FileNamePrefix>

   <ReportInitialStartDate>
      <Segment name=".">
          <Year>2006</Year>
          <Month>07</Month>
          <Day>15</Day>
      </Segment>
          <!-- Segment '.westcoast' was introduced on 1/1/2007 -->
      <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>
```


Copyright © 2023, 2026, Oracle and/or its affiliates.

```
            <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

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 that 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 that the data is correct and that the status of the export run is **COMPLETED**. See "Retrieving Audit Data for Exported G/L Reports".

Copyright © 2023, 2026, Oracle and/or its affiliates.

10

Troubleshooting and Correcting G/L Exports

Learn 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 XML Output Files
- Restarting a Failed Export
- Purging G/L Export Data

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.

Each time you generate general ledger reports, BRM writes information to a **/process_audit/export_gl** object in the BRM database. To view the data in the audit object, read the object with the **testnap** utility or the Object Browser. See "Reading an Object and Fields" in *BRM Developer's Guide*.

The **/process_audit/export_gl** object contains the following information:

- The status of the last export run:
  - COMPLETED indicates that the application run has exited normally.
  - IN_PROGRESS indicates that the application is currently running.
  - INCOMPLETE indicates that the application crashed during an earlier application run and the audit entry was marked as such during a subsequent run.
- If the **pin_ledger_report** utility was run in **-export** mode with the **-resend** or **-regenerate** parameter, the POID of the ledger report object last prepared for export.
- A report information array. Each array element contains the following information for one report:
  - The G/L segment for which the report is generated.
  - The type of G/L report.
  - The frequency (or schedule) of exporting the G/L reports.
  - The end time of the reporting period.
  - If non-monetary balance elements were reported on, the balance element IDs that were included in or excluded from the report.

Copyright © 2023, 2026, Oracle and/or its affiliates.

- A list of up to 3 POIDs:

  The first POID is the default report generated by calling the PCM_OP_GL_LEDGER_REPORT opcode.

  If there are 2 POIDs, the second one is either the custom report (Billed, Billed earned, and Previously billed earned) or the cumulative report (Unbilled, Unbilled earned, Unbilled unearned, and Billed unearned) generated after execution of the PCM_OP_GL_POL_EXPORT_GL policy opcode.

  If there are 3 POIDs, the second one is the custom report and the third one is the cumulative report.

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:

- Re-creates the G/L report in the database.
- Reposts the G/L report.
- Re-creates the XML output files.

> **Note**
>
> 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.

> **Note**
>
> 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 following command:

```
pin_ledger_report -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**, enter:

pin_ledger_report -mode export -regenerate 0.0.0.1-123456-4

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

Copyright © 2023, 2026, Oracle and/or its affiliates.

Correcting Incorrect Data or Replacing Lost XML Output Files

When the G/L data in the BRM database has not changed and the XML output 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 re-creates the XML output files. It does *not* re-create or repost the G/L report in the database.

> **Note**
>
> 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 following command:

```
pin_ledger_report -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**, enter:

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

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

> **Note**
>
> 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 **pin_ledger_report** 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:

```
pin_ledger_report -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:

- XML output files on the file system.

Copyright © 2023, 2026, Oracle and/or its affiliates.

Keep the XML output 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 and 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.

  > **Note**
  >
  > 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*.

Copyright © 2023, 2026, Oracle and/or its affiliates.

11

About Deliverable-Based Revenue Recognition

Learn how to set up revenue recognition in Oracle Communications Billing and Revenue Management (BRM) in accordance with the ASC 606 and IFRS 15 accounting standards.

Topics in this document:

- About Revenue Recognition
- About Deliverable-Based Revenue Recognition
- About the Standalone Selling Price
- About the Revenue Earning Schedule
- About Calculating the Revenue Allocation
- About Deliverables
- About Subscription Terms
- About the Revenue from Usage Charges and Administrative Fees
- Sample Revenue Recognition Scenarios
- Setting Up Deliverable-Based Revenue Recognition
- Enabling Deliverable-Based Revenue Recognition

About Revenue Recognition

Revenue recognition is an accounting principle and process for reporting revenue. It specifies when your company can recognize the money received from the products and services that you sell to your customers. BRM supports two types of revenue recognition:

- **Deliverable-based revenue recognition:** Allows you to set up revenue recognition that complies with the Accounting Standards Codification (ASC) 606 and International Financial Reporting Standard (IFRS) 15 accounting standards. See "About Deliverable-Based Revenue Recognition".
- **Event-based revenue recognition:** Allows you to use any revenue recognition scheme, including those supported in previous releases of BRM. See "About Event-Based Revenue Recognition".

The type of revenue recognition that BRM applies depends on whether a package or bundle that your customer purchases contains a deliverable. If the package or bundle:

- Contains a deliverable, deliverable-based revenue recognition is applied to the purchase.
- Does not contain a deliverable, event-based revenue recognition is applied to the purchase.

Copyright © 2023, 2026, Oracle and/or its affiliates.

About Deliverable-Based Revenue Recognition

> **Note**
>
> Deliverable-based revenue recognition must be enabled before you can use it. See "Enabling Deliverable-Based Revenue Recognition".

BRM enables you to set up revenue recognition that complies with the ASC 606 and IFRS 15 standards. The core principle of these standards is that the amount of revenue that is recognized in a contract must match the amount of goods or services that are delivered to customers.

The ASC 606 and IFRS 15 standards have established a five-step process that governs revenue recognition for contracts:

1. Identify the contract with the customer.
2. Identify the performance obligations in the contract.
3. Determine the transaction price.
4. Allocate the transaction price to the performance obligations in the contract.
5. Recognize revenue when the entity satisfies a performance obligation.

To comply with these standards, BRM distributes the selling price of a package or bundle proportionally across all of its goods and services. Prices are distributed according to their standalone selling price, even if a good or service is provided to a customer for free.

BRM determines how much revenue can be recognized by using:

- The standalone selling price of a good or service. See "About the Standalone Selling Price".
- The revenue earning schedule. See "About the Revenue Earning Schedule".
- The proportional price of each good or service. See "About Calculating the Revenue Allocation".

About the Standalone Selling Price

In accounting, the standalone selling price is a rational and unbiased estimate of the potential market price of a good or service, such as $9 for an online movie or $30 per month for unlimited mobile data. Although it is an estimate of the potential market price, the standalone selling price does not always match what you charge your customers. For example, the price of a good or service may be higher due to increased demand or may be lower due to discounts or promotional offers.

BRM uses the standalone selling price to determine how to allocate revenue proportionally across multiple goods or services in a package or bundle.

The standalone selling price is defined in your deliverables. See "About Deliverables".

Copyright © 2023, 2026, Oracle and/or its affiliates.

About the Revenue Earning Schedule

The revenue earning schedule specifies how the revenue is distributed and recognized over time. You define the type of earning schedule to use in your deliverables.

BRM supports these earning schedules:

- **Immediately**: All of the revenue is recognized at once. For example, the revenue from the sale of an eBook is recognized when it is purchased by a customer.

  Use an immediate earning schedule when your company can fulfill a customer's order for the good or service before the purchase is sent to BRM for processing.

- **Linearly over a period of time**: Revenue recognition is distributed across the commitment period, based on the allocation frequency. For example, a package with a one-year commitment term and a monthly frequency will have 1/12th of its revenue recognized each month.

  The commitment period is defined in your subscription terms. See "About Subscription Terms".
  The allocation frequency is defined in your deliverables. See "About Deliverables".

- **Milestone based**: All of the revenue is earned when a milestone is reached, such as when a company technician installs a satellite TV dish at a customer's home or a physical product is shipped from the warehouse to the customer.

  To support a milestone-based earning schedule, you must configure your custom client application to call the contract management opcodes.

About Calculating the Revenue Allocation

BRM determines how to allocate revenue proportionally across multiple goods or services in a package by using:

- The standalone selling price of each good or service in the package.
- The relative value of each good or service in the package. For example, if a package includes goods A and B:
  - The relative value of good A = A/(A+B)
  - The relative value of good B = B/(A+B)
- The contract's total price. The total price includes any flat discounts, but excludes the revenue from early termination fees, usage fees, and taxes.

  If your advertised price includes VAT taxes, BRM calculates the contract's total price without the tax.

To determine the revenue allocation for a good, multiply the relative value and the contract's total price.

For example, if the relative value of good A is 40% and good B is 60%, and the contract's total price is $200, the revenue allocation for good A is $80 (that is, 40% of $200) and the revenue allocation for good B is $120 (that is, 60% of $200).

About Deliverables

Deliverables define the type of good or service you are selling, its standalone selling price, its revenue earning schedule, and the revenue general ledger ID (G/L ID) to associate with

Copyright © 2023, 2026, Oracle and/or its affiliates.

financial transactions. You use deliverables to specify when and how much revenue can be recognized from a customer's purchase of a good or service.

You create deliverables for the following:

- **One-time activities**: These are discrete services that are performed by your company or by your customers, such as a company installing a service at the customer's home. Deliverables for one-time activities have immediate or milestone-based earning schedules. You specify which one to use when you create the deliverable.
- **Physical goods**: These are tangible products that you sell to your customers, such as DVDs, cell phones, and eBooks. Deliverables for physical goods have immediate or milestone-based earning schedules. You specify which one to use when you create the deliverable.
- **Metered services**: These are services with limits on how much customers can use in a cycle, such as 100 free minutes or 20 eBooks a month. Deliverables for metered services have linear earning schedules.
- **Continuous services**: These allow customers to use an unlimited amount of a service over a specified time period, such as unlimited data for a month or unlimited access to a website for a month. Deliverables for continuous services have linear earning schedules.

You do not create deliverables for usage charges.

To create deliverables, see "Configuring Deliverables" in *PDC Creating Product Offerings*.

After you create deliverables, you associate them with your charge offers.

About Subscription Terms

Subscription terms define the commitment period and options for canceling and renewing the goods and services you offer. For example, a subscription term could have a commitment period such as 1 month, 1 year, or 2 years. When you create a subscription term, you specify whether customers can cancel their subscriptions early and whether they incur any fees for doing so.

To create subscription terms, see "Creating Subscription Terms" in *PDC Creating Product Offerings*.

After you create subscription terms, you associate them with your packages and bundles.

About the Revenue from Usage Charges and Administrative Fees

The revenue generated from usage charges and administrative fees such as early termination fees, cannot be predicted ahead of time. These fees are not included when calculating a contract's total price or calculating the proportional price of each good or service in a package. Instead, your company recognizes the revenue from usage charges and administrative fees when customers pay for them.

To set up BRM to recognize revenue from usage charges and administrative fees:

- **Usage charges**: You create a standard G/L ID for usage fees and associate it with a revenue general ledger account for credits.
- **Early termination fees**: To track early termination fees separately, you must associate a revenue G/L ID with the early termination fees in your subscription terms.

Copyright © 2023, 2026, Oracle and/or its affiliates.

Alternatively, BRM can apply revenue from early termination fees across all deliverables in the customer's contract. For example, if a contract has a $200 early termination fee, deliverable A with a relative value of 20%, and deliverable B with a relative value of 80%, BRM applies $40 to deliverable A and $160 to deliverable B.

Sample Revenue Recognition Scenarios

This section describes how revenue would be recognized in the following simple scenarios:

- Scenario 1: One-Year Contract with a Linear Earning Schedule
- Scenario 2: One-Year Contract for a Physical Good with Free Maintenance

Scenario 1: One-Year Contract with a Linear Earning Schedule

Your company offers a 1-year basic TV and internet service subscription for $59 per month with a $10 per month discount. There's also a $100 fee if a customer cancels the subscription after the first month.

Assuming a standalone selling price of $40 for the basic TV service and $25 for the internet service, you'd calculate the revenue allocation for the one-year contract like this:

- The relative value of the basic TV service is 61.5% (that is, $40/($40 + $25)).
- The relative value of the internet service is 38.5% (that is, $25/($40 + $25)).
- The contract's total price is $588 (that is, ($59 * 12 months) – ($10 * 12 months)).
- The revenue allocation is:
  - Basic TV service: $361.62 (that is, 61.5% of $588).
  - Internet service: $226.38 (that is, 38.5% of $588).

Assuming that the revenue is earned linearly over the monthly cycle period, the company would recognize this revenue:

- For the basic TV service: $30.14 per month (that is, $361.62/12 months).
- For the internet service: $18.87 each month (that is, $226.38/12 months).

If a customer cancels the contract after the first month, the company would recognize $100 immediately for the early termination fee.

Scenario 2: One-Year Contract for a Physical Good with Free Maintenance

Your company offers a computer with a year of free maintenance service for a promotional subscription price of $750.

Assuming a standalone selling price of $700 for the computer and $12 per month for the maintenance service, you would calculate the revenue allocation for the computer contract like this:

- The standalone selling price of the contract is $844 (that is, $700 + [$12 * 12 months]).
- The relative value of the computer is 83% (that is, $700/$844).
- The relative value of the maintenance service is 17% (that is, $144/$844).
- The contract's total price is $750.
- The revenue allocation is:

Copyright © 2023, 2026, Oracle and/or its affiliates.

  - Computer: $622.50 (that is, 83% of $750).
  - Maintenance service: $127.50 (that is, 17% of $750).

Assuming that the revenue from the computer is earned when the computer is shipped from the warehouse and the maintenance service is earned linearly over a monthly cycle period, the company would recognize this revenue:

- For the computer: $622.50 when the computer is shipped from the warehouse.
- For the maintenance service: $10.63 each month (that is, $127.50/12 months).

Setting Up Deliverable-Based Revenue Recognition

Setting up deliverable-based revenue recognition involves these high-level steps:

1. Enabling deliverable-based revenue recognition in BRM. See "Enabling Deliverable-Based Revenue Recognition".
2. Setting up your general ledger by:
   - Creating your general ledger accounts.
   - Creating your G/L IDs.
   - Assigning G/L IDs to your charges, discounts, tax codes, and so on.

   See "Setting Up Your General Ledger".
3. Creating deliverables, which define the goods and services that you sell to your customers. See "Configuring Deliverables" in *PDC Creating Product Offerings*.
4. Associating deliverables with your charge offers. See "Configuring Charge Offers" in *PDC Creating Product Offerings*.
5. Creating subscription terms, which define the commitment periods, cancellation options, and renewal options for your customers' contracts. See "Creating Subscription Terms" in *PDC Creating Product Offerings*.
6. Creating bundles and packages for the products and services that you sell. See "Creating Bundles" and "Creating Packages" in *PDC Creating Product Offerings*.
7. Associating your subscription terms with the bundles and packages that you sell to your customers.

You can run reports to view a summary of the totals in your G/L accounts. See "Generating General Ledger Reports".

Enabling Deliverable-Based Revenue Recognition

By default, BRM uses the event-based revenue recognition scheme. To use the deliverable-based revenue recognition scheme, you must enable the **SSPRevenueRecognition** business parameter in BRM and then associate deliverables with your charge offers.

Copyright © 2023, 2026, Oracle and/or its affiliates.

> **⚠ Caution**
>
> - Your customers will not be able to cancel any packages or bundles that they purchased before you enabled the **SSPRevenueRecognition** business parameter. Cancel all of your customers' existing subscriptions before enabling the business parameter.
> - After you enable the **SSPRevenueRecognition** business parameter, you cannot disable it.

After deliverable-based revenue recognition is enabled, the following operations are not supported in BRM:

- Performing future and backdated operations
- Changing deliverable types
- Using multiple G/L segments
- Modifying existing contracts
- Suspending contracts
- Using complex discounts

  Only flat discounts that apply directly to the currency amount of cycle fees or purchase fees is supported. Discount grants on noncurrency resources and discounts based on resource values are not supported.
- Using Conversion Manager
- Using Account Migration Manager
- Performing rerating
- Transferring services
- Performing multi-schema operations
- Using the Web Services Manager opcodes
- Performing accounts receivable (A/R) operations, such as adjustments and write-offs

To enable deliverable-based revenue recognition:

1. Cancel all of your customers' existing subscriptions. See "Working with No-Contract Subscriptions" in Billing Care Help.

   Otherwise, your customers will not be able to cancel any packages or bundles that they purchased before you enabled the **SSPRevenueRecognition** business parameter.
2. Go to *BRM_home*/**sys/data/config**.
3. Create an XML file from the **/config/business_params** object:

   ```
   pin_bus_params -r BusParamsBilling bus_params_billing.xml
   ```
4. In the XML file, change the value for **SSPRevenueRecognition** to enabled:

   ```
   <SSPRevenueRecognition>enabled</SSPRevenueRecognition>
   ```
5. Load the XML file into the BRM database:

   ```
   pin_bus_params bus_params_billing.xml
   ```
6. Stop and restart the CM.

Copyright © 2023, 2026, Oracle and/or its affiliates.

12

About Event-Based Revenue Recognition

Learn how to set up revenue recognition based on the event schedule in Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:

- About Event-Based Revenue Recognition
- Examples of Billed and Unbilled Revenue
- 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

About Event-Based Revenue Recognition

In event-based revenue recognition, BRM recognizes revenue based on the event schedule:

- Revenue from one-time purchases, cancellation charges, and usage charges is earned immediately.
- Revenue from recurring charges is earned over the cycle period, such as over the month or over the quarter, as services are rendered. For example, a monthly cycle charge is considered only partially earned if the customer hasn't used the entire month of service yet.

In event-based revenue recognition, BRM can report these revenue types:

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

Examples of Billed and Unbilled Revenue

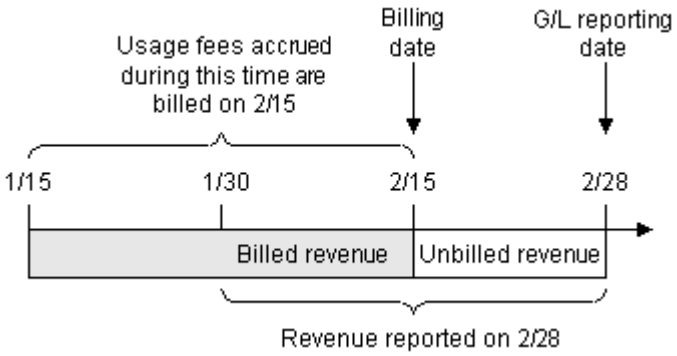
If your company uses immediate revenue recognition, BRM reports all revenue as billed or unbilled.

Copyright © 2023, 2026, Oracle and/or its affiliates.

Billed and Unbilled Usage Fees

In Figure 12-1, 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.

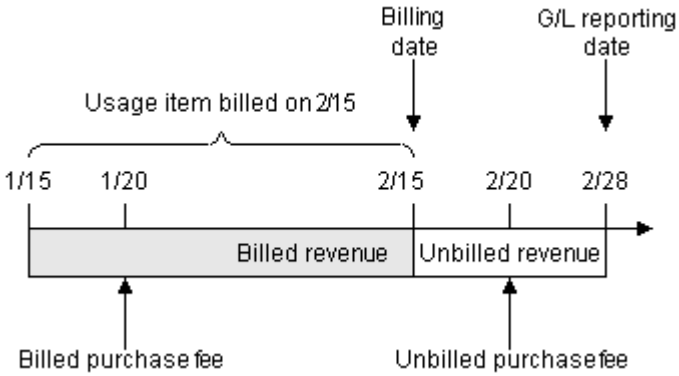
Figure 12-1 Billed and Unbilled Usage Fees



Billed and Unbilled Purchase and Cancellation Fees

In Figure 12-2, 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.

Figure 12-2 Billed and Unbilled Purchase and Cancellation Fees

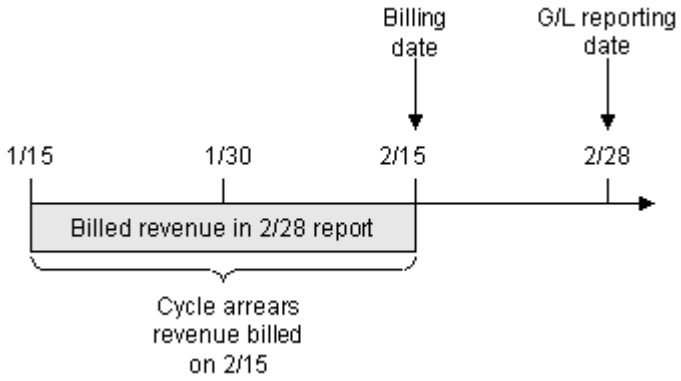


Billed Cycle Arrears Fees

Cycle arrears fees are always billed. In the example illustrated in Figure 12-3, the cycle arrears fee billed on 2/15 is included in the 2/28 report as billed.

Copyright © 2023, 2026, Oracle and/or its affiliates.

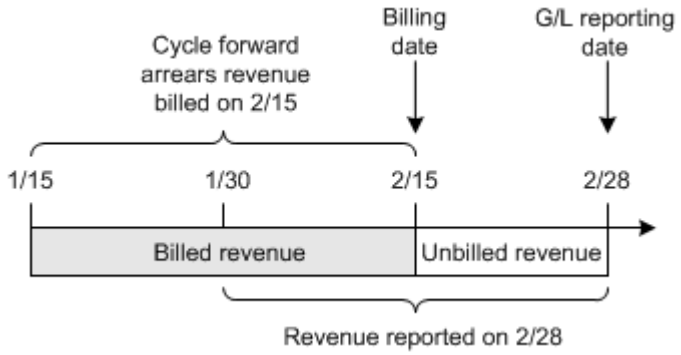
Figure 12-3 Billed Cycle Arrears Fees



Billed and Unbilled Cycle Forward Arrears Fees

In Figure 12-4, 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:

Figure 12-4 Billed and Unbilled Cycle Forward Arrears Fees



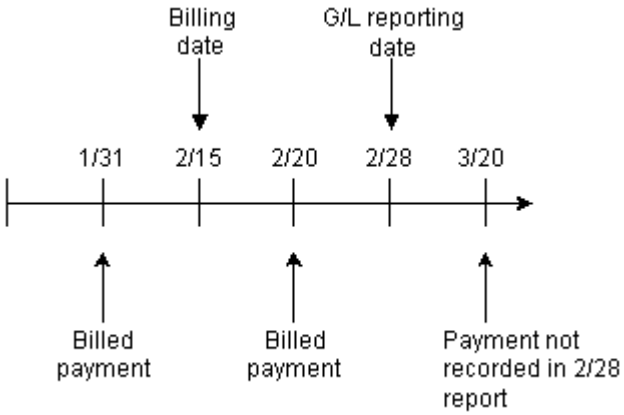
Billed and Unbilled Nonrated Events

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 effect 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 12-5, 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.

Copyright © 2023, 2026, Oracle and/or its affiliates.

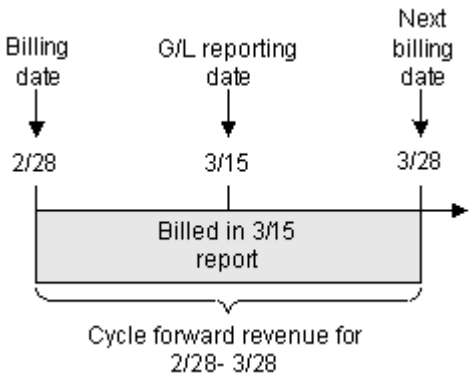
Figure 12-5 Billed and Unbilled Nonrated Events



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 12-6, the cycle forward fee billed on 2/28 is reported as billed in the G/L report on 3/15.

Figure 12-6 Billed and Unbilled Cycle Forward Fees



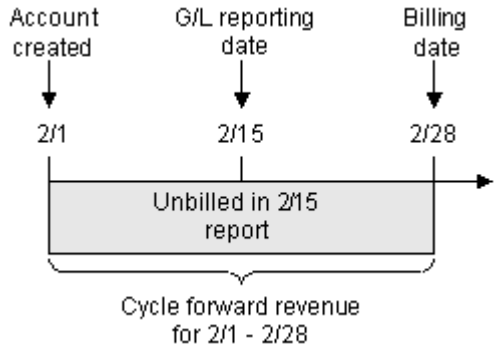
About Unbilled Cycle Forward Fees

Cycle forward fees can be unbilled when the fee is for a new account.

In Figure 12-7, 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.

Copyright © 2023, 2026, Oracle and/or its affiliates.

Figure 12-7 Unbilled Cycle Forward Fees



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:

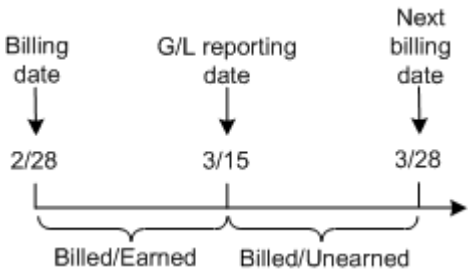
- Billed earned
- Unbilled earned
- Billed unearned
- Unbilled unearned
- Previously billed earned

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 12-8 shows the earned and unearned revenue for a cycle forward fee.

Figure 12-8 Earned and Unearned Revenue in 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 12-9 shows earned and unearned revenue for a cycle forward arrears fee.

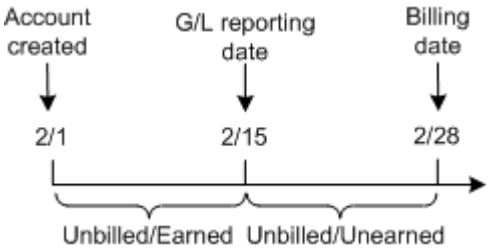
Copyright © 2023, 2026, Oracle and/or its affiliates.

Figure 12-9 Earned and Unearned Cycle Forward Arrears Fees



Figure 12-10 shows the first cycle forward fee when the account is not billed at account creation.

Figure 12-10 Cycle Forward Fee at Account Creation



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

Copyright © 2023, 2026, Oracle and/or its affiliates.

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*.

Copyright © 2023, 2026, Oracle and/or its affiliates.

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 12-1:

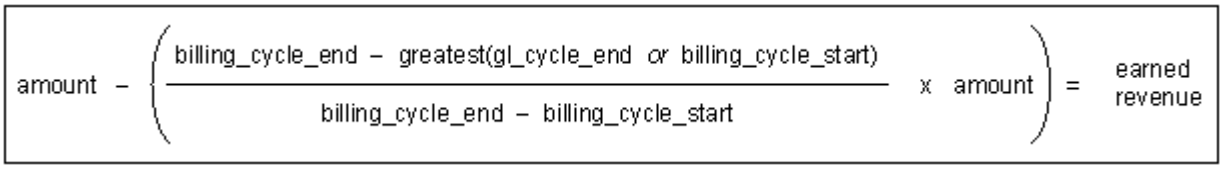
Table 12-1 Fields and Equations

Field Name	Equation Name
PIN_FLD_EARNED_START_T	billing_cycle_start
PIN_FLD_EARNED_END_T	billing_cycle_end

Calculating Billed Earned Revenue

BRM uses the equation in Figure 12-11 to calculate billed earned revenue:

Figure 12-11 Billed Earned Revenue Equation



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 12-12.

Copyright © 2023, 2026, Oracle and/or its affiliates.

Figure 12-12 Calculating Billed Earned Revenue



If the monthly fee is $100, BRM calculates the earned revenue for the period 7/1 to 8/1 as shown in Figure 12-13:

Figure 12-13 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 12-14 to calculate previously billed revenue:

Figure 12-14 Previously Billed Revenue Equation

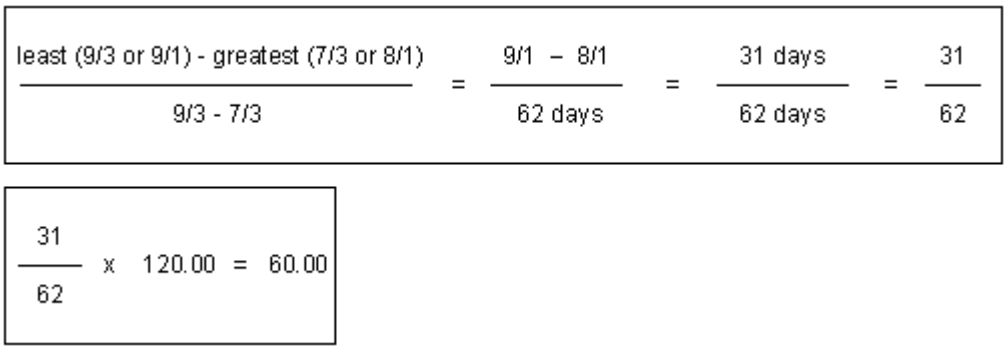
$$\left(\frac{\text{least(billing_cycle_end } \textit{or} \text{ gl_cycle_end) - greatest(billing_cycle_start } \textit{or} \text{ gl_cycle_start)}}{\text{billing_cycle_end - billing_cycle_start}} \right) * \text{amount}$$

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 12-15:

Copyright © 2023, 2026, Oracle and/or its affiliates.

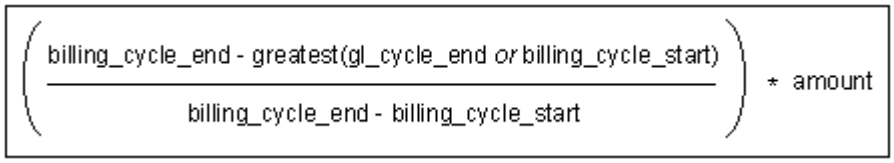
Figure 12-15 Sample Previously Billed Earned Revenue Calculation



Calculating Unearned Revenue

BRM calculates unearned revenue for cycle fees only. BRM uses the equation in Figure 12-16 to calculate unearned revenue:

Figure 12-16 Unearned Revenue Equation



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 12-17. 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:

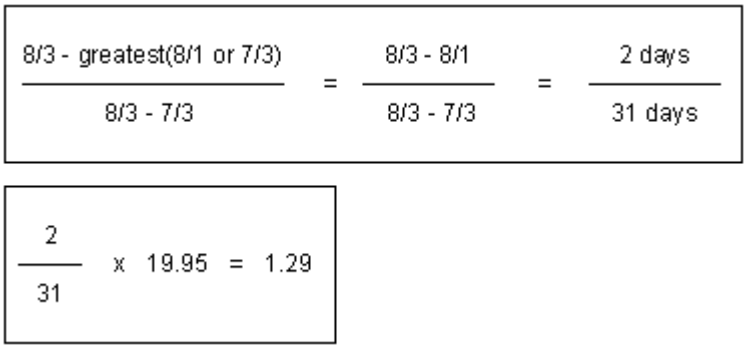
Figure 12-17 Calculating Unearned Revenue



For a monthly cycle fee of $19.95, the calculation for unearned revenue is completed as shown in Figure 12-18:

Copyright © 2023, 2026, Oracle and/or its affiliates.

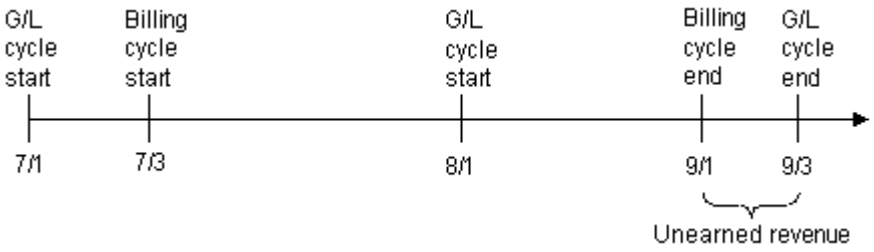
Figure 12-18 Sample Unearned Revenue Calculation



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.

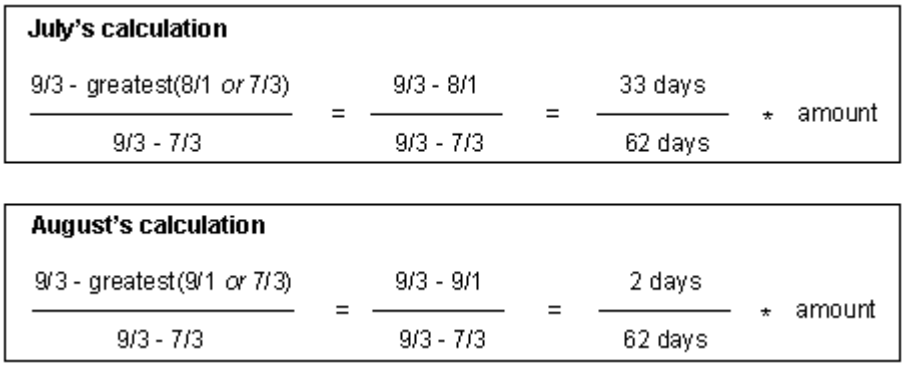
If a customer starts a service on 7/3 with a bimonthly cycle fee, then the timeline is as shown in Figure 12-19:

Figure 12-19 Cumulative Unearned Revenue Timeline



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 12-20:

Figure 12-20 Sample Unearned Revenue Calculations



Copyright © 2023, 2026, Oracle and/or its affiliates.

13

General Ledger Utilities

Learn about the syntax and parameters for the Oracle Communications Billing and Revenue Management (BRM) general ledger (G/L) utilities and scripts.

Topics in this document:

- load_pin_config_export_gl
- load_pin_glchartaccts
- load_pin_glid
- pin_ledger_report

load_pin_config_export_gl

Use this utility to configure general ledger exporting. See "Configuring Preview and Monthly Report XML Output Files" 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 "Setting Up Your General Ledger" for information.

Copyright © 2023, 2026, Oracle and/or its affiliates.

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.

Copyright © 2023, 2026, Oracle and/or its affiliates.

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.

> **Note**
>
> 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.

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

For more information, see the following:

- About Collecting General Ledger Data
- Generating General Ledger Reports
- Configuring General Ledger Report XML Output Files

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

Creates an individual report that you can view in Oracle Analytics Publisher.

See "Creating Individual G/L Reports with pin_ledger_report".

Copyright © 2023, 2026, Oracle and/or its affiliates.

```
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*

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

-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 **billed**. See "About Event-Based Revenue Recognition".

-report *file_name*

Displays the output of the utility. Use the **-report** parameter with the **-test** parameter to display the report without creating the **/ledger_report** object.

-summary | -detail

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

-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.

-test

Runs the utility in calculation mode only.

-verbose

Displays information as the utility runs.

-help

Displays the utility parameters.

Copyright © 2023, 2026, Oracle and/or its affiliates.

-post

Posts the report using the value in the **-end** parameter as the post date. If used, this must be the last parameter in the command line.

post_only Mode Syntax

Posts G/L data to maintain the validity of revenue data.

See "Creating Individual G/L Reports with pin_ledger_report".

```
pin_ledger_report -mode post_only
                  [-segment gl_segment]
                  [-verbose]
                  [-help]
                  [-posted posted_date | -unpost]
```

post_only Mode Parameters

-segment *gl_segment*

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

-verbose

Displays information as the utility runs.

-help

Displays the utility parameters.

-posted *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

Lists previously generated G/L reports.

See "Creating Individual G/L Reports with pin_ledger_report".

```
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.

Copyright © 2023, 2026, Oracle and/or its affiliates.

-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 Event-Based Revenue Recognition".

-verbose
Displays information as the utility runs.

-help
Displays the utility parameters.

export Mode Syntax

Generates and exports reports from the **/config/export_gl** object.

See "Creating Individual G/L Reports with pin_ledger_report".

```
pin_ledger_report  -mode export
                   [-aggregate | -preview]
                   [-segment gl_segment]
                   [-resend report_ID]
                   [-regenerate report_ID]
                   [-restart]
                   [-verbose]
                   [-help]
```

export Mode Parameters

-aggregate
Exports your monthly export report to an XML file. You can then post this XML file to your external financial system. See "About the Monthly Export Report".
Use this report for deliverable-based revenue recognition.

Do not use any other parameters with this option.

-preview
Exports your preview report to an XML file. See "About the Preview G/L Report".
Use this report for deliverable-based revenue recognition.
Do not use any other parameters with this option.

-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".

> **Note**
>
> 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 XML Output Files".

Copyright © 2023, 2026, Oracle and/or its affiliates.

-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.

Copyright © 2023, 2026, Oracle and/or its affiliates.