Oracle® Communications Billing and Revenue Management

Designing and Generating Invoices



Release 16.26.0
G54458-01
October 2026



Oracle Communications Billing and Revenue Management Designing and Generating Invoices, Release 16.26.0

G54458-01

Copyright © 2017, 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 Invoicing

About Invoices 1
- About the Invoicing Process 2

Understanding Invoice Layout 3
- Standard Details 3
- Adjustment, Payment, Promotion, and Other Charges and Credits Details 3
- Charge Details 4

About Localized Invoices 4

2 Designing Invoices

About Designing Invoice Templates 1

Designing XSLT Invoice Templates 1
- Turning on XSLT Style Sheet Processing 2

Designing HTML Invoice Templates 2

Designing Hyperlinked Invoice Templates 2
- Loading Stored Procedures for Hyperlinked Invoice Templates Reports 3
- Setting Up the Data Model for Hyperlinked Invoice Templates 3
- Setting Up the Hyperlinked Invoice Templates 4
- Modifying the Hyperlink Path Prefix 5

Loading Invoice Templates 6

Switching between XSLT and HTML Templates 7

3 Customizing Invoice Content

Grouping Items By Service Type 1

Including Payment, A/R, and Tax Details in Invoices 2
- A/R Events That Can Be In Invoices 2

Including Shadow Event Adjustment Details in Invoices 4

Including Soft Descriptors in Invoices 4

Aggregating Taxes on Invoices 4

Copyright © 2017, 2026, Oracle and/or its affiliates.

Using Invoice Templates That Display Promotion Names 5
Defining Multiple Invoices for One Customer 5
Using an /invoice Subclass 6

4 Invoicing for Hierarchical Bill Units

About Invoicing for Hierarchical Bill Units 1
About Viewing Invoices for Accounts That Have Nonpaying Bill Units 2
Setting Defaults for Hierarchical Bill Unit Invoices 2
Sending Child Account Invoices to the Parent Account's Email Address 3

5 Adding Siebel CRM Promotion Names to Invoices

Adding Siebel CRM Promotion Names to Invoices 1
How BRM Adds Siebel CRM Promotion Names 2
Setting Up BRM Invoices to Display Siebel CRM Promotion Names 2
Specifying Whether BRM Displays Promotion Details on Invoices 2

6 Including Invoice Data from Custom Applications

Including Invoice Data from Custom Applications 1
Using Data Map Templates 1
Loading the Invoice Data Map Templates 2

7 Generating Proforma Invoices

About Proforma Invoices 1
About the Customer Process 2
About the Proforma Invoice Process 2
About Generating Proforma Invoices for Wholesale and Hierarchy Accounts 3
Setting Up BRM to Generate Proforma Invoices 3
- Defining the Default Validity Period of Proforma Invoices 3
- Changing the Watermark for the Proforma Invoice 5
- Customizing Your Client Applications to Manage Proforma Invoices 5

Managing Proforma Invoices with pin_bill_accts 5
- Generating Proforma Invoices with pin_bill_accts 5
- Accepting Proforma Invoices with pin_bill_accts 6
- Rejecting Proforma Invoices with pin_bill_accts 6
- Regenerating Proforma Invoices 7
- Configuring the XML File for Proforma Invoices 7
- Configuring the Text File for Proforma Invoices 7

Copyright © 2017, 2026, Oracle and/or its affiliates.

8 Generating Invoices

Generating Invoices Automatically 1
Generating Invoices Manually 1
Generating Detailed Invoices from an External File 2
Generating Invoices for Non-Invoice Payment Methods 2
Exporting Invoices 3

9 Sending Invoices to Customers

Sending Invoices to Customers 1
Setting the Maximum Invoice Size for Email 1
Sending Invoices to Accounts That Use the Invoice Payment Method 1
Sending Invoices to Accounts That Do Not Use the Invoice Payment Method 2
Configuring the Email Data Manager for Printing 2

10 Displaying Invoices

About Displaying Invoices 1
Using the XML Sample Invoice Viewing Utility 1

11 Storing Invoices

About Formats for Storing Invoices 1
Specifying the Default Format in Which to Store Invoices in BRM 2
Storing Invoices in a Separate Database Schema 2
- Installing the Invoice Data Manager 3
- Configuring BRM to Use a Separate Invoice Database Schema 4
- Configuring Invoice Applications to Use a Separate Invoice Database Schema 5
- Starting the Invoice Data Manager 5
- Configuring the Invoice pin.conf for Multiple Database Schemas 5

12 Using Oracle Analytics Publisher for Invoicing

About Using Oracle Analytics Publisher for Invoicing 1
Designing Invoices 3
- About Invoice Layout Templates for Invoice Document Generation 3
- About Reports for Invoice Document Generation 4
- Including Late Payment Fees in Invoices 4
- Including Finance Charges in Invoices 5

Copyright © 2017, 2026, Oracle and/or its affiliates.

13 Setting Up Your System to Generate Invoice Documents

Tasks for Setting Up Invoice Generation 1
- Creating an Invoicing Configuration Business Profile 1
  - Enabling the BRM-Oracle Analytics Publisher Integration 2
  - Loading Invoicing Business Profiles into the Database 2
  - Associating Pre-Existing Bill Units with Business Profiles 3
  - Viewing Invoices Individually on Oracle Analytics Publisher Console 4
  - Enabling the Display of Customer Details on the Web 4
- Installing the BRM-Oracle Analytics Publisher Invoicing Integration Package 4
  - Preinstallation Requirements 5
  - Removing Older BRM-Oracle Analytics Publisher Invoicing Integration Layout Template Files 5
  - Install Billing and Invoicing Utilities 6
  - Installed Template Files 6
- Configuring the pin_inv_doc_gen Utility 7
  - Configuring the Infranet.properties File for pin_inv_doc_gen 7
  - Configuring the docgen.sh Batch Script 10
  - Changing Command-Line Parameters for pin_inv_doc_gen 10
  - Setting Scheduler Configuration 11
  - Configuring the Email Server 11
  - Configuring HTTPS for Secure Communication 11
  - Creating the XMLP_INVOICE_JOB Table in the Scheduler Database 11
- Configuring the BRM-Oracle Analytics Publisher Invoicing Integration to Support Multischema Systems 13
  - Loading Stored Procedures 13
  - Creating New JDBC Data Sources 13

14 Generating Invoices in Oracle Analytics Publisher

Generating Oracle Analytics Publisher Invoice Documents 1
- Generating Invoice Documents for a List of Accounts 1
- Generating Invoice Documents Based on Custom Criteria 2
  - Selecting Invoices Based On Custom Parameters 3
  - Configuring Custom Criteria at the Command Line 7
- Generating Invoice Documents Based on the Account Type 7
- Generating Duplicate Invoice Documents 8
- Generating Invoice Documents for Non-Invoice Accounts 8

Generating Invoices for Hierarchical Bill Units with Oracle Analytics Publisher 9
- Setting Up Invoicing for Hierarchical Bill Units 9
- Invoice Template for Hierarchical Bill Units 10
- Setting Nonpaying Child Bill Unit Processing 10
- Creating Hierarchical Bill Unit Trial Invoices 11

Copyright © 2017, 2026, Oracle and/or its affiliates.

Creating Hierarchical Bill Unit Regular Invoices 12
Regenerating Failed Invoices for Hierarchical Bill Units 12
Purging Trial Invoices 13
Deferred Taxation Limitations for Hierarchical Bill Units 13
Troubleshooting Issues with Hierarchical Bill Unit Invoice Generation 15
Storing the Invoice Document 16
Sending Invoice Documents to Your Customers 16
Customizing Invoices 16
Customizing Invoice Layout Templates 17
Enriching Invoice Data 17

15 Corrective Invoicing

About Corrective Invoices 1
About Content of Corrective Invoices 1
About Invoices and Billing Periods 2
About Deferred Corrective Invoices 2
Configuring to Defer Corrective Invoices 2
About Replacement Invoices for Hierarchical Bill Units 3
Detailed Replacement Invoice for a Parent Bill 3
Summary Replacement Invoice for a Parent Bill 3
Detailed Replacement Invoice for a Nonpaying Child Bill 4
Summary Replacement Invoice for a Nonpaying Child Bill 4
About Invoice Correction Letters for Hierarchical Bill Units 4
Detailed Invoice Correction Letter for a Parent Bill 4
Summary Invoice Correction Letter for a Parent Bill 5
Detailed Invoice Correction Letter for Nonpaying Child Bills 5
Summary Invoice Correction Letter for Nonpaying Child Bills 5
How BRM Displays Manual Adjustments 5
About Generating Corrective Invoices 6
About the pin_inv_accts Utility Parameters 6
About Corrective Invoice Templates 7
Configuring Corrective Invoice Templates Based on Business Profiles 7
Providing Custom Templates for Corrective Invoices 8
About Corrective Invoices and Collections 8
Corrective Invoices and General Ledger Posts 9
Setting Up Corrective Invoicing in BRM 9
Specifying the Type of Corrective Invoice 9
Providing Custom Templates for Corrective Invoices 9
Generating Corrective Invoices 10
Post-Processing Actions 10
Viewing Corrective Invoices 10

Copyright © 2017, 2026, Oracle and/or its affiliates.

Adjustments and Corrective Invoicing 11

16 Improving Invoicing Performance

Improving Performance by Customizing Search Results 1
Improving Performance by Customizing Event Caching 2
Improving Performance by Removing Invoice Details You Do Not Need 4

17 Setting Up Invoicing on Multiple Database Schemas

Setting Up Invoicing on Multiple Database Schemas 1
Running Invoicing on Multiple Database Schemas One at a Time 1
Running Invoicing on Multiple Database Schemas Simultaneously 1

18 Invoice Utilities

load_pin_invoice_data_map 1
pin_inv_accts 2
pin_inv_doc_gen 6
pin_inv_export 8
pin_inv_send 10
pin_load_invoice_events 12
pin_load_invoice_template 13
pin_upd_assoc_bus_profile 15

Copyright © 2017, 2026, Oracle and/or its affiliates.

About This Content

This guide provides an overview of the Oracle Communications Billing and Revenue Management (BRM) invoicing software, including designing and generating invoices, correcting invoicing, and so on.

Audience

This guide is intended for system administrators and developers.

Copyright © 2017, 2026, Oracle and/or its affiliates.

1

About Invoicing

Learn how to generate an invoice in Oracle Communications Billing and Revenue Management (BRM), which provides balance information for a customer's bill along with the corresponding amount due.

Topics in this document:

- About Invoices
- Understanding Invoice Layout
- About Localized Invoices

About Invoices

An invoice lists the balance information for a customer's bill.

When billing runs, a bill is produced for every bill unit (**/billinfo** object) in an account, and invoices are generated for each bill. You can use two invoice types: *detailed* and *summary*.

- A detailed invoice lists the bill items and the events. The detailed invoice mode is the default.

  > **Note**
  >
  > Generating detailed invoices requires more system processing and may slow performance.

- A summary invoice lists only the bill items and not the events.

For bill units in hierarchies, the billing data for each nonpaying bill unit is rolled up to the invoice for the paying bill unit. This invoice is referred to as a *consolidated* invoice because it contains the billing data for the nonpaying bill units in addition to the billing data for the paying bill unit.

> **Note**
>
> If you use detailed invoices, generating the consolidated invoice can be system intensive, and the consolidated invoice can be extremely large. Therefore, you can set invoicing thresholds to control the size of the consolidated invoice and the nonpaying data contained in it. See "Invoicing for Hierarchical Bill Units".

A typical invoice includes this information:

- Charge summary and usage details. The breakdown of usage charges includes the gross usage charges and applicable discounts and taxes. For example, if the net usage charge is $355.08, the invoice document displays the breakdown as: Gross Charge: 348, Discount: (25.20), and Tax: 32.28.

Copyright © 2017, 2026, Oracle and/or its affiliates.

- Service usage details.
- Past due amount details from the previous bill.
- Hierarchical account details.
- Promotion names.
- A/R actions having currency or noncurrency impact, such as adjustment, dispute, settlement, and write-off details.

You can customize the content of invoices; for example, include a customer service phone number. You can also use templates to customize the appearance of the invoice.

You can also customize, enrich, and present BRM invoices by using the capabilities of Oracle Analytics Publisher. See "Using Oracle Analytics Publisher for Invoicing".

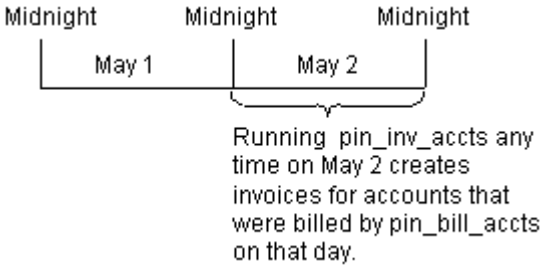
About the Invoicing Process

Use the **pin_inv_accts** utility to generate regular invoices for regular bills and corrective invoices for corrective bills.

For regular bills, run invoicing each day for accounts that had a regular bill created by the **pin_bill_accts** utility.

If you miss any billing days, the **pin_inv_accts** utility still generates invoices for accounts whose billing day was missed. This is because the **pin_bill_accts** utility creates bills for the missed billing days, and the **pin_inv_accts** utility generates invoices for those bills as shown in Figure 1-1.

Figure 1-1 Invoices Created when Running pin_inv_accts



For corrective bills, you run generate a corrective invoice when you create the corrective bill using the **pin_make_corrective_bill** utility.

BRM performs the following operations to generate invoices:

1. BRM creates a bill for each account bill unit when it runs the **pin_bill_accts** utility as part of the daily billing script. BRM needs a bill before generating an invoice.
2. BRM generates an invoice for each bill.
3. BRM stores all invoices either in the BRM database schema that contains the invoiced accounts or in a separate invoice-only schema that you set up.

After an invoice is generated:

- You can display the invoice in Billing Care. The invoice's format is based on a template you create and load.

Copyright © 2017, 2026, Oracle and/or its affiliates.

- You can send the invoice to your customer in email, or as a printed invoice.
- You can export the invoice as an XML, HTML, or DOC1 file.

Understanding Invoice Layout

The invoice examples used here are based on the invoice template for individual accounts provided in the BRM-Oracle Analytics Publisher integration package.

> **Note**
>
> By default, the invoice document presents charges in primary currency. However, all the relevant charges details are available in secondary currency in the **/invoice** object. You can customize the RTF files associated with the layout templates to display the charges details in secondary currency.

The layout of a typical invoice generated using BRM-Oracle Analytics Publisher integration application displays details in the following sections:

Standard Details

This section of the invoice includes the following information:

- Basic bill information, such as invoice date, invoice number, account number, due date, and bill period. The invoice displays this bill-related information on all the pages.
- Customer information, such as name and billing address. For hierarchical accounts, the invoice displays parent account information in this section.
- Amount due, which is a sum of past amount due, current charges, payments, and adjustments.
- Summary of current charges, which is a sum of different types of service usage charges, other charges and credits, and taxes and surcharges. If a customer has used different types of services, such as GSM and ADSL, this section displays the summary of current charges for GSM and ADSL services separately.
- A URL for online payments and a support phone number to resolve customer queries.
- Payment slip, which the customer needs to detach from the invoice and returns when making the payment.

Adjustment, Payment, Promotion, and Other Charges and Credits Details

This section of the invoice includes the following details:

- **Adjustment details:** This section displays the adjustments that the customer service representative (CSR) performs on the customer account. The invoice displays the type of adjustment, date on which the adjustment was done, amount adjusted, and the reason for adjustment.
- **Payment details:** This section displays the payment type, date on which payment is made, payment amount, and the reason for the payment. For example, if the customer has made a cash payment of $15, the **Amount** column displays **(15)**.
- **Promotion details:** This section displays a list of promotions offered by the service provider. It displays promotion-related details, such as promotion name, promotion validity

Copyright © 2017, 2026, Oracle and/or its affiliates.

period, and current status of promotion. For more information on promotions, see "Adding Siebel CRM Promotion Names to Invoices".

- **Other charges and credits details:** This section displays information related to dispute, refund, settlement, write-off, and write-off reversal.

Charge Details

For consumer accounts, this section of the invoice displays the usage charge details for a particular phone number. If the user has multiple phone numbers, different sections for each phone number display the charge details corresponding to that number. For an account that has a paying parent bill unit, the invoice displays the total charges for each nonpaying child bill unit.

This section of the invoice also displays package, service, bundle, and charge offer details used by the customer.

> **Note**
>
> If no device (phone number) is associated with a service, the charge details are displayed under **Non Device details** section.

- **Balance Details:** Includes the details of noncurrency balances available for a service type. For example, it can display the number of free seconds available for the GSM telephony service type.
- **Charge Summary:** Displays total charges, which is a sum of recurring charges, one-time charge, taxes and surcharges, and total discounts.
- **Usage Summary:** Displays the service usage for different services.

  After the summary information section, the invoice displays itemized usage details based on service type. For example, for telephony and SMS services used, the invoice displays two sections for telephony and SMS usage. By default, the invoice displays the rounded-off value of event usage. To display the exact usage duration in the invoice, you can customize the charge offer so that the usage event is populated with the exact net quantity.

> **Note**
>
> The discount and tax break up details are available in the invoice data generated in XML format. However, the invoice document in PDF does not display these details in the Usage Summary section.

About Localized Invoices

You can create localized invoices in BRM, as follows:

- Use UTF8 encoding for hard-coded text strings in XSLT or HTML invoice templates, such as table titles.
- Specify the BRM locale when loading your invoice template. See "Loading Invoice Templates".

Copyright © 2017, 2026, Oracle and/or its affiliates.

You can also use DOC1 from Group 1 Software, Inc., to create localized invoices. For information, see the DOC1 documentation.

Copyright © 2017, 2026, Oracle and/or its affiliates.

2

Designing Invoices

Learn how to design the appearance of an invoice in Oracle Communications Billing and Revenue Management (BRM) by creating templates.

Topics in this document:

- About Designing Invoice Templates
- Designing XSLT Invoice Templates
- Designing HTML Invoice Templates
- Designing Hyperlinked Invoice Templates
- Loading Invoice Templates
- Switching between XSLT and HTML Templates

About Designing Invoice Templates

To design the appearance of an invoice, you create a template.

By default, BRM uses a HTML template (*BRM_home*/**sys/data/config/pin_invoice_template.html**). You can define your own template in one of these formats:

- **XSLT style sheets**: This is the most flexible and powerful option for designing invoices.
- **HTML files**: You can create your own HTML template or modify an HTML template that BRM provides.
- **Hyperlinked Invoice Templates:** Use hyperlinked invoice templates to display invoices for large bill unit hierarchies with better memory efficiency.

If you use a third-party invoice application, you do not need to create an invoice template. You export the invoice data to the external tool. See "Exporting Invoices".

After you create or modify a style sheet, you load it into the BRM database by using the "pin_load_invoice_template" utility. See "Loading Invoice Templates".

Designing XSLT Invoice Templates

BRM uses XSLT style sheets and an XSLT processor to generate HTML invoices that can be displayed in Billing Care, or a Web browser.

> **Note**
>
> By default, BRM supports the Xalan XSLT processor. If you want to use a different XSLT processor, you need to write the implementation for that processor and configure it for BRM. For more information on using custom processors, contact Oracle.

Copyright © 2017, 2026, Oracle and/or its affiliates.

BRM provides two sample style sheets, located in *BRM_home***/sys/data/config/stylesheets**:

- **sample1.xsl**: This is the default BRM invoice template.
- **sample2.xsl**

You can use or edit these style sheets or create your own XSLT style sheet with a third-party XSLT editor.

You can use an XML version of an invoice as the basis for creating the style sheet. To export an invoice to XML, see "Exporting Invoices".

Use the PCM_OP_INV_POL_FORMAT_INVOICE_XSLT opcode to customize XSLT invoices. For more information, see "About Using Opcodes" in *BRM Opcode Guide*.

Turning on XSLT Style Sheet Processing

If you use XSLT style sheet templates, you need to run the BRM invoice formatter to enable invoices to display. The formatter is a Java process that takes XML data and an XSLT style sheet and generates an HTML invoice.

To run the invoice formatter, enter the following command at the command prompt:

```
start_formatter
```

Designing HTML Invoice Templates

You design HTML invoice templates by using an HTML editor. You can use the HTML template provided with BRM as a starting point (*BRM_home***/sys/data/config/pin_invoice_template.html**).

An HTML template needs to include these variables, which are in the default template:

- _INVOICENUM_
- _CORPIMAG_
- _CORPADDR_
- _CUSTADDR_
- _AMOUNTDUE_
- _ACCOUNTSUMMARY_
- _ITEMS_
- _EVENTS_

When invoices are generated, data replaces the variables. To add variables, modify the PCM_OP_INV_POL_FORMAT_INVOICE_HTML policy opcode. For more information, see "About Using Opcodes" in *BRM Opcode Guide*.

Designing Hyperlinked Invoice Templates

BRM uses hyperlinked invoice templates to display invoices for large bill unit hierarchies with better memory efficiency. With hyperlinked invoice templates, you can easily navigate the invoice using a hyperlinked table of contents.

Copyright © 2017, 2026, Oracle and/or its affiliates.

> **Note**
>
> Hyperlinked-invoice-template mode uses FTP delivery to create hyperlink text within PDF files for invoices, because hyperlink text must point to local files and not to links within the Oracle Analytics Publisher server.

To configure hyperlinked invoice templates:

1. Back up the Oracle Analytics Publisher repository, which has a defaulted location of *OAP_Home***/user_projects/domains/OAP_domain/config/bipublisher/repository/**.
2. Load the stored procedures. See "Loading Stored Procedures for Hyperlinked Invoice Templates Reports".
3. Configure the bursting data source and query by setting up the data model configuration for hyperlinked invoice templates. See "Setting Up the Data Model for Hyperlinked Invoice Templates".
4. Set up the hyperlinked invoice templates. See "Setting Up the Hyperlinked Invoice Templates".
5. Add the hyperlink path prefix to *repository***/Reports/BRM_Invoices/0.0.0.1/BRM_Bursting_Invoice_Report.xdo/xdo.cfg**. See "Modifying the Hyperlink Path Prefix".

Loading Stored Procedures for Hyperlinked Invoice Templates Reports

To load the stored procedures:

1. Go to *BRM_home***/apps/pin_inv_doc_gen** directory.
2. Run the following command, which starts 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.*
3. Run the following command:

   ```
   SQL>@invoice_bursting_HIT.plb
   ```

   The stored procedure is loaded.
4. Type **exit** to exit SQL*Plus.

Setting Up the Data Model for Hyperlinked Invoice Templates

To set up the hyperlinked invoice template data model:

1. Go to *OAP_home***/user_projects/domains/OAP_domain/config/bipublisher/repository/** directory.

Copyright © 2017, 2026, Oracle and/or its affiliates.

where *OAP_home* is the directory in which Oracle Analytics Publisher is installed.

> **Note**
>
> If you do not use *OAP_home*/**user_projects/domains/OAP_domain/config/bipublisher/repository/** as your repository, navigate to your new repository at this time.

2. Search for the following:

   `_datamodel.xdm`

   Rename it to the following:

   `_datamodel.xdm.ORIG`

3. Search for the following:

   `_datamodel_HIT.xdm`

   Rename it to the following:

   `_datamodel.xdm`

4. Open the **_datamodel.xdm** file and edit the following:

   `PARAMETER1` to *`__FTP_SERVERNAME__`*

   `PARAMETER4` to *`__FTP_REMOTE_DIRECTORY__`*

   where

   - *`__FTP_SERVERNAME__`* is your FTP server name
   - *`__FTP_REMOTE_DIRECTORY__`* is your FTP destination directory

5. Save and close the file.

Setting Up the Hyperlinked Invoice Templates

To set up the hyperlinked invoice templates:

1. Go to the *OAP_home*/**user_projects/domains/OAP_domain/config/bipublisher/repository/Reports/BRM_Invoices/***schema_number*/**BRM_Bursting_Invoice_Report.xdo** directory.
2. Search for the following:

   `BRM_Corporate_Correction_Invoice.rtf`

   Rename it to the following:

   `BRM_Corporate_Correction_Invoice.rtf.ORIG`

3. Search for the following:

   `BRM_Corporate_Invoice.rtf`

   Rename it to the following:

   `BRM_Corporate_Invoice.rtf.ORIG`

4. Search for the following:

   `BRM_Corporate_Replacement_Invoice.rtf`

   Rename it to the following:

Copyright © 2017, 2026, Oracle and/or its affiliates.

`BRM_Corporate_Replacement_Invoice.rtf.ORIG`

5. Search for the following:

   `BRM_Corporate_Correction_Invoice_HIT.rtf`

   Rename it to the following:

   `BRM_Corporate_Correction_Invoice.rtf`

6. Search for the following:

   `BRM_Corporate_Invoice_HIT.rtf`

   Rename it to the following:

   `BRM_Corporate_Invoice.rtf`

7. Search for the following:

   `BRM_Corporate_Replacement_Invoice_HIT.rtf`

   Rename it to the following:

   `BRM_Corporate_Replacement_Invoice.rtf`

> **Note**
>
> You can also load **HIT.rtf** templates within the Oracle Analytics Publisher server.

8. In the **xdo.cfg** file, edit the following:

   `<property name="fo-external-link-base-url">file:__FTP_REMOTE_DIRECTORY__</property>`

   where *__FTP_REMOTE_DIRECTORY__* is your FTP remote destination directory.

9. Save and close the file.
10. Restart the Oracle Analytics Publisher server.

Modifying the Hyperlink Path Prefix

Oracle Analytics Publisher creates hyperlinks within PDF files. The defaulted directory is in the Oracle Analytics Publisher hierarchy, inaccessible to other users wishing to access the hyperlink. To make these hyperlinks accessible, modify the hyperlink path prefix.

To modify the hyperlink path prefix:

1. Open the *repository*/**Reports/BRM_Invoices/0.0.0.1/BRM_Bursting_Invoice_Report.xdo/xdo.cfg** file in a text editor.
2. In the **xdo.cfg** file, add your desired hyperlink path prefix. For example:

   `file:/tmp/`*`ftpinvoices/filename`***`.pdf`**

   where:

   - *ftpinvoices* is the prefix for the hyperlink path.
   - *filename* is the name of the PDF file.

3. Save and close the file.

Copyright © 2017, 2026, Oracle and/or its affiliates.

4. Copy PDF files that are intended for viewing on another machine in the same directory structure as the source. Alternatively, edit the **xdo.cfg** file so that the final directory path matches the machine intended for viewing.
5. To allow hyperlinked invoice templates to split PDFs and hyperlinks, set thresholds in **/config/business_param** and do the following:
   - For normal invoicing, run **pin_inv_accts** with the **-hierarchy** flag. For example:

     ```
     - pin_inv_accts -verbose -hierarchy
     - pin_inv_accts -verbose -pay_type 10001
     ```

   - For trial invoicing, run **pin_trial_bill_accts** with the **-paytype** parameter. For example:

     ```
     - pin_trial_bill_accts -start xxx -end yyy -pay_type 10007
     - pin_trial_bill_accts -start xxx -end yyy -pay_type 10001
     ```

Loading Invoice Templates

After creating your template, you need to load it into the BRM database by using the **pin_load_invoice_template** utility.

To load an invoice template:

1. Go to a directory that contains a valid configuration file.

   pin_load_invoice_template uses information in the configuration file to connect to the BRM database. See "Connecting BRM Utilities" in *BRM System Administrator's Guide*.
2. To load an XSLT style sheet, enter:

   pin_load_invoice_template -brand "0.0.0.1/account 1"-type text/html -locale *locale_name* **-template** *filename* **-usexsl**

   where:

   - *locale_name* is the BRM locale of the template. See "Locale Names" in *BRM Developer's Guide*.
   - *filename* is the name and full path of the template file.

   For example, this is the command to load the English template **sample1.xsl** for the root account:

   ```
   pin_load_invoice_template -brand "0.0.0.1/account 1" -type text/html -locale en_US -
   template /opt/portal/sys/data/config/sample1.xsl -usexsl
   ```
3. To load an HTML template, enter:

   pin_load_invoice_template -brand "0.0.0.1/account 1"-type HTML -locale *locale_name* **-template** *filename*

   where:

   - *locale_name* is the BRM locale of the template. See "Locale Names" in *BRM Developer's Guide*.
   - *filename* is the name and full path of the template file.

   For example, this is the command to load the English template **pin_invoice_template.html** for the root account:

Copyright © 2017, 2026, Oracle and/or its affiliates.

```
pin_load_invoice_template -brand "0.0.0.1/account 1" -type HTML -locale en_US -
template /opt/portal/sys/data/config/pin_invoice_template.html
```

Switching between XSLT and HTML Templates

You can switch between using an XSLT template and using an HTML template without loading a new template.

To switch to an XSLT template, enter:

```
pin_load_invoice_template -brand "0.0.0.1/account 1"-usexsl
```

To switch to an HTML template, enter:

```
pin_load_invoice_template -brand "0.0.0.1/account 1"
```


Copyright © 2017, 2026, Oracle and/or its affiliates.

3

Customizing Invoice Content

Learn how to customize the content included in your customers' invoices in Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:

- Grouping Items By Service Type
- Including Payment, A/R, and Tax Details in Invoices
- Including Shadow Event Adjustment Details in Invoices
- Including Soft Descriptors in Invoices
- Aggregating Taxes on Invoices
- Using Invoice Templates That Display Promotion Names
- Defining Multiple Invoices for One Customer
- Using an /invoice Subclass

To customize the data displayed in invoices, use the PCM_OP_INV_POL_PREP_INVOICE policy opcode. See "Including Custom Data in Invoices" in *BRM Opcode Guide*.

Grouping Items By Service Type

You can group items by service on invoices. To enable this feature:

1. Open the CM configuration file (*BRM_home***/sys/cm/pin.conf**).
2. To enable service-centric invoicing, set the value of **service_centric_invoice** to **1**, as follows:

   ```
   - fm_inv_pol    service_centric_invoice   1
   ```

   To disable service-centric invoicing, enter **0**.
3. Save and close the file.
4. Stop and restart the CM.

> **Note**
>
> In the invoicing business profile, if the **EnableInvoicingIntegration** config business parameter is enabled, ensure that in *BRM_home***/sys/cm/pin.conf**, the **service_centric_invoice** entry is commented. By default, the **service_centric_invoice** entry is commented. If this entry is not commented, you can view the invoice document in Billing Care only, and not in the Oracle Analytics Publisher client.

Copyright © 2017, 2026, Oracle and/or its affiliates.

Including Payment, A/R, and Tax Details in Invoices

You can specify details about customer's payments, A/R, and tax details. To do so, create a list of the events you want to include, and then use the **pin_load_invoice_events** utility to load that list of event types. See "pin_load_invoice_events" for details.

To include details in the invoice:

1. Open the **events.file** file (*BRM_home*/**sys/data/config/events.file**).
2. Edit the file to add the event details you want to include in the invoice. See "A/R Events That Can Be In Invoices".
3. Load **events.file** by using the following command:

   ```
   pin_load_invoice_events -brand "0.0.0.1/account 1"-eventfile file
   ```

   where *file* is the name of the **events.file** file.
4. To verify that the event types were loaded, you can display the **/config/invoice_events** object by using Object Browser, or use the **robj** command with the **testnap** utility. See "Reading an Object and Writing Its Contents to a File" in *BRM Developer's Guide*.

A/R Events That Can Be In Invoices

To include adjustment details in the invoice, include these events in the **events.file** file (if not already present):

- **/event/billing/adjustment/account**
- **/event/billing/adjustment/event**
- **/event/billing/adjustment/item**
- **/event/billing/adjustment/tax_event**

To include payment details in the invoice, include these events in the **events.file** file (if not already present):

- **/event/billing/payment/cash**
- **/event/billing/payment/cc**
- **/event/billing/payment/check**
- **/event/billing/payment/dd**
- **/event/billing/payment/payorder**
- **/event/billing/payment/postalorder**
- **/event/billing/payment/wtransfer**
- **/event/billing/payment/failed**
- **/event/billing/payment/voucher**
- **/event/billing/payment/voucher_topup**

To include payment reversal details in the invoice, include these events in the **events.file** file (if not already present):

- **/event/billing/reversal/cash**
- **/event/billing/reversal/cc**

Copyright © 2017, 2026, Oracle and/or its affiliates.

- **/event/billing/reversal/check**
- **/event/billing/reversal/dd**
- **/event/billing/reversal/failed**
- **/event/billing/reversal/payorder**
- **/event/billing/reversal/postalorder**
- **/event/billing/reversal/voucher**
- **/event/billing/reversal/wtransfer**

To include write-off-related events, include these events in the **events.file** file (if not already present):

- **/event/billing/writeoff/bill**
- **/event/billing/writeoff/billinfo**
- **/event/billing/writeoff/item**
- **/event/billing/writeoff/tax_bill**
- **/event/billing/writeoff/tax_billinfo**
- **/event/billing/writeoff/tax_item**

To include refund-related events, include these events in the **events.file** file (if not already present):

- **/event/billing/refund/cash**
- **/event/billing/refund/cc**
- **/event/billing/refund/check**
- **/event/billing/refund/dd**
- **/event/billing/refund/payorder**
- **/event/billing/refund/postalorder**
- **/event/billing/refund/wtransfer**

To include write-off reversal events, include these events in the **events.file** file (if not already present):

- **/event/billing/writeoff_reversal**
- **/event/billing/writeoff_reversal/tax**

To include dispute and settlement details in the invoice, include these events in the **events.file** file (if not already present):

- **/event/billing/settlement/event**
- **/event/billing/settlement/item**
- **/event/billing/settlement/tax_event**
- **/event/billing/dispute/event**
- **/event/billing/dispute/item**
- **/event/billing/dispute/tax_event**

To include tax-related information for the summary invoice, include **/event/billing/cycle/tax** in the **events.file** file.

Copyright © 2017, 2026, Oracle and/or its affiliates.

To include any custom event, include the custom event in the **events.file**. For example, you can include **/event/billing/cycle/tax/federal** event to the **events.files**.

Including Shadow Event Adjustment Details in Invoices

Shadow events show the results of an adjustment. When recording a shadow event, you can customize your invoice to either show the adjustment details or show only the result of shadow event adjustments in the end balance.

Shadow events are also created by the BRM rerating process. You use the same configuration entry to show shadow event details for both adjustments and rerating. Therefore, to display adjustment details, you must also display rerating details, and vice versa.

To set the shadow event invoice details entry:

1. Open the CM configuration file (*BRM_home***/sys/cm/pin.conf**).
2. Set the following entry to **1**.

   ```
   - fm_inv_pol show_rerate_details 0
   ```

   The default is **0**.
3. Save and close the file.
4. Stop and restart the CM.

Including Soft Descriptors in Invoices

You can add *soft descriptors*, which contain information to help customers recognize charges to credit card and checking account statements. Soft descriptors are available for Paymentech direct debit and credit card processing. The following soft descriptors can be added to customer statements:

- Your DBA (doing business as) name
- Your product name
- Your customer service phone number (instead of your headquarters city)

There are two ways to add soft descriptor information:

- Configure the *BRM_home***/sys/dm_fusa/pin.conf** configuration file. See "Adding Soft Descriptor Information to Customer Statements" in *BRM Configuring and Collecting Payments*.
- Modify the policy source file for the PCM_OP_PYMT_POL_PRE_COLLECT policy opcode. See "Adding Soft Descriptors to Invoices" in *BRM Opcode Guide*.

Aggregating Taxes on Invoices

You can create invoices that aggregate taxes for each tax supplier and, within each tax supplier, for each tax code. By default, taxes are not aggregated on invoices.

To enable this feature, run the **pin_bus_params** utility to change the **ADSTTaxHandle** business parameter. For information about this utility, see "About Customizing BRM" in *BRM Developer's Guide*.

To aggregate taxes on your invoices:

1. Go to *BRM_home***/sys/data/config**.

Copyright © 2017, 2026, Oracle and/or its affiliates.

2. Create an XML file from the **/config/business_params** object:

   ```
   pin_bus_params -r BusParamsInvoicing bus_params_Invoicing.xml
   ```

3. In the file, change **disabled** to **enabled**:

   ```
   <ADSTTaxHandle>enabled</ADSTTaxHandle>
   ```

4. Save the file as **bus_params_Invoicing.xml**.
5. Load the XML file into the BRM database:

   ```
   pin_bus_params bus_params_Invoicing.xml
   ```

6. Stop and restart the CM.

Using Invoice Templates That Display Promotion Names

To display promotion names and details, your invoices must use a template that includes a space holder for promotion details. The default **sample1.xsl** template is preconfigured to display promotion names and details. You can use this XSL template or create your own template.

To view promotion details on invoices, perform one of the following:

- Use the default **sample1.xsl** invoice template. If your system uses multiple invoice templates, you can switch to the XSL template by using the **pin_load_invoice_template** utility. See "Switching between XSLT and HTML Templates".
- Modify your custom XSL invoice template to display promotion name details, and then load the template into the database. See "Designing XSLT Invoice Templates".
- Customize the **pin_invoice_template.html** template by using the PCM_OP_INV_POL_FORMAT_INVOICE_HTML policy opcode. For more information, see "About Using Opcodes" in *BRM Opcode Guide*.

Defining Multiple Invoices for One Customer

You can define multiple invoices for one customer by identifying each invoice with a unique invoice name. This enables you to assign a specific invoice to a specific bill unit in a customer's account.

For example, a customer wants the invoice for one bill sent to a home address and the invoice for another bill sent to a business address. You can set up two invoices: one with an invoice name of "Home" that contains the customer's home address information, and one with an invoice name of "Work" that contains the customer's work address information.

> **Note**
>
> At any time, you can have only one invoice assigned to a customer account. You can select the invoice type from a list of invoices defined for a customer.

To assign an invoice ID to an invoice, do either of the following:

- Use Billing Care to set the Invoice payment method ID value of an invoice. You access this field from the Payment Options dialog box when setting up a new payment method for an account during account creation or account maintenance.

Copyright © 2017, 2026, Oracle and/or its affiliates.

- Use a custom CRM system to pass the invoice name in the input flist of PCM_OP_CUST_SET_PAYINFO. For more information, see "About Using Opcodes" in *BRM Opcode Guide*.

Using an /invoice Subclass

If you add a subclass to the **/invoice** object and want to use the subclass for invoicing, you must specify the subclass type. You can do this in either of the following ways:

- Specify the invoice object database number and type in the PIN_FLD_INVOICE_OBJ field in the **/payinfo** storable class. Then, using **testnap**, change the **invoice_obj_type** in the **payinfo_t** table to the subclass type.
- Edit the CM configuration file (*BRM_home*/**sys/cm/pin.conf**) by doing the following:
  1. In the **Invoice_db** entry, add the subclass extension name to **/invoice**.

     > **Note**
     >
     > This entry is read during account creation time. Changing it has no effect on existing accounts. If you want the **/invoice** storable subclass to apply to existing accounts, you must use the previous method instead.

  2. Change the following line:

     ```
     -fm_cust_pol  invoice_db  0.0.0.1  /invoice   0
     ```

     To this:

     ```
     -fm_cust_pol  invoice_db  0.0.0.1  /invoice/extension_name  -1
     ```

     > **Note**
     >
     > This entry is not in the CM **pin.conf** file by default unless you install the Invoice Data Manager (**dm_invoice**). You can, however, add the entry yourself.

Copyright © 2017, 2026, Oracle and/or its affiliates.

4

Invoicing for Hierarchical Bill Units

Learn how to use invoicing with hierarchical bill units in Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:

- About Invoicing for Hierarchical Bill Units
- About Viewing Invoices for Accounts That Have Nonpaying Bill Units
- Setting Defaults for Hierarchical Bill Unit Invoices
- Sending Child Account Invoices to the Parent Account's Email Address

About Invoicing for Hierarchical Bill Units

When the **pin_inv_accts** utility processes bills for a bill unit hierarchy, it searches the BRM database for the nonpaying child bill units whose paying parent bill units are members of the hierarchy and generates their invoices automatically.

You can set threshold values to ensure system performance is not decreased while processing invoices for large bill unit hierarchies. The threshold values set the maximum number of nonpaying child bill units that can be rolled up to the paying parent bill unit and processed for invoicing.

If the value of the **/bill** object's PIN_FLD_AR_HIERARCHY_SIZE field is greater than the threshold value, the parent account's invoice includes information from only the paying parent bill unit.

If the number of nonpaying bill units is less than or equal to the threshold, the parent bill unit's invoice includes nonpaying child bill unit information.

> **Note**
>
> If bill suppression is enabled on an account that contains paying parent bill units and the nonpaying child bill units threshold is exceeded, invoicing fails. This occurs because nonpaying bill units, which use the bill number of the account that owns their paying parent bill unit, are generated even when billing for the parent account is suppressed. In such cases, they do not contain bill numbers, which invoices require. To exclude invoicing for these bills, run **pin_inv_accts** with the **-skip_blank_billnos** parameter. (Nonpaying child bill units contain bill numbers only when the account that owns their paying parent bill unit is billed.)

For information on setting the threshold values, see "Setting Defaults for Hierarchical Bill Unit Invoices".

Copyright © 2017, 2026, Oracle and/or its affiliates.

About Viewing Invoices for Accounts That Have Nonpaying Bill Units

If you open an account that has a nonpaying bill unit in Billing Care and then run Bill Now, the account's bill and invoice for the nonpaying child bill unit can be viewed only in the information page of the account that owns the paying parent bill unit.

When **pin_inv_accts** runs, it generates invoices for accounts with nonpaying bill units. You can view these invoices in Billing Care.

Setting Defaults for Hierarchical Bill Unit Invoices

BRM automatically generates invoices for each paying parent bill unit and each nonpaying child bill unit in a bill unit hierarchy. For example, if a bill unit hierarchy contains a paying parent bill unit that has two nonpaying child bill units, BRM generates three invoices: one for the parent bill unit and one for each nonpaying child bill unit.

You can configure whether invoices for paying parent bill units also include information from nonpaying child bill units by running the **pin_bus_params** utility to change the **SubARItemsIncluded** business parameter. For information about this utility, see "About Customizing BRM" in *BRM Developer's Guide*. Specifically, you can configure the following:

- Whether nonpaying A/R items are displayed on a paying parent bill unit's invoice.
- The maximum number of nonpaying child bill units that can be included in a paying parent bill unit's summary invoice.

  When the number of nonpaying child bill units is less than the maximum, all nonpaying bill units are included on the parent's summary invoice. When the number of nonpaying child bill units exceeds the maximum, none of the nonpaying bill units is included on the paying parent's invoice.

- **Note**

  This behavior does not apply to wholesale billing. For wholesale billing, when you generate regular invoices for the wholesale parent bill unit, BRM fetches only the items and events of the parent bill unit because charges from nonpaying child bill units are rolled up to the paying parent bill unit during billing. See "Running Wholesale Billing" in *BRM Configuring and Running Billing*.

- The maximum number of nonpaying bill units that can be included in a paying parent bill unit's detailed invoice.

  When the number of nonpaying child bill units is less than the maximum, the parent's detailed invoice lists the items and events for all nonpaying bill units. When the number of nonpaying child bill units exceeds the maximum, the paying parent's detailed invoice does not include nonpaying bill unit data.

To specify whether nonpaying details are displayed on paying parent invoices:

1. Go to *BRM_home*/**sys/data/config**.
2. Create an XML file from the **/config/business_params** object:

   ```
   pin_bus_params -r BusParamsInvoicing bus_params_Invoicing.xml
   ```


Copyright © 2017, 2026, Oracle and/or its affiliates.

3. Specify whether to display A/R items for each nonpaying bill unit on paying parent invoices:

   Set the **SubArItemsIncluded** entry to **enabled** to display nonpaying A/R items on paying parent invoices and to **disabled** to display only parent A/R items on parent invoices.

   ```
   <SubARItemsIncluded>enabled</SubARItemsIncluded>
   ```

4. Set the threshold for summary invoices:

   Change 0 to the number of nonpaying child bill units allowed in a paying parent's summary invoice:

   ```
   <ThresholdSubordsSummary>0</ThresholdSubordsSummary>
   ```

   If the threshold value is exceeded, the paying parent summary invoice does not contain the nonpaying bill unit data.

   > **Note**
   >
   > A value of **0** means that BRM does not check the size of the account hierarchy, and the paying parent's summary invoice automatically includes all nonpaying bill unit data.

5. Set the threshold for detailed invoices:

   Change 0 to the number of nonpaying child bill units allowed in a paying parent's detailed invoice:

   ```
   <ThresholdSubordsDetail>0</ThresholdSubordsDetail>
   ```

   If the number of nonpaying bill units exceeds the threshold value, the paying parent's detailed invoice includes only the parent bill unit's details.

   > **Note**
   >
   > A value of **0** means that BRM does not check the size of the account hierarchy, and the paying parent's account summary invoice automatically includes all nonpaying bill unit data.

6. Save the file as **bus_params_Invoicing.xml**.
7. Load the XML file into the BRM database:

   ```
   pin_bus_params bus_params_Invoicing.xml
   ```

You do not need to restart the Connection Manager (CM) to enable these entries.

For more information on invoicing thresholds, see "About Invoicing for Hierarchical Bill Units".

Sending Child Account Invoices to the Parent Account's Email Address

By default, invoices for nonpaying child bill units are consolidated into the invoice for their paying parent bill unit and sent to the email address of the parent bill unit's account. To handle cases where the parent invoice is not consolidated and individual invoices exist for each

Copyright © 2017, 2026, Oracle and/or its affiliates.

nonpaying bill unit, you can configure **pin_inv_send** to send all nonpaying bill unit invoices to the email address of the parent invoice's account:

1. Open the invoice configuration file (*BRM_home***/apps/pin_inv/pin.conf**).
2. Change the value of the **send_sub_inv_2_parent** entry to **1** and make sure it is not commented.
3. Save and close the file.

Copyright © 2017, 2026, Oracle and/or its affiliates.

5

Adding Siebel CRM Promotion Names to Invoices

Learn how to add Siebel promotion names to your customers' invoices in Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:

- Adding Siebel CRM Promotion Names to Invoices
- How BRM Adds Siebel CRM Promotion Names
- Setting Up BRM Invoices to Display Siebel CRM Promotion Names
- Specifying Whether BRM Displays Promotion Details on Invoices

Adding Siebel CRM Promotion Names to Invoices

Promotions are marketing bundles that contain multiple Siebel products and services, often at reduced prices. When customers order promotions, their invoices, by default, display the Siebel products and services that come with the promotion, but the invoice does not display the promotion name itself. This could be confusing to customers and make them wonder if they are being billed correctly. If your BRM system uses Siebel CRM and the Application Integration Architecture (AIA), you can set up your system to display Siebel CRM promotion names on customer invoices.

When your system is configured to display promotion names, customer invoices display all promotions that are active, were inactivated, or were canceled during the billing cycle in which the invoice is generated. For example, if a customer switched from promotion A to promotion B in January, the customer's January invoice would display both promotion A and promotion B.

To manage promotions, your customer service representatives (CSRs) must be able to perform the following tasks in Siebel CRM:

- Add a promotion to a customer's account.
- Modify a customer's existing promotion.
- Transfer an existing promotion from one account to another.
- Inactivate a customer's existing promotion.
- Cancel a customer's existing promotion.
- Apply promotions to child accounts in an account hierarchy.

> **Note**
>
> Parent accounts can share their promotions with a child account only if the child account owns the charge offers and discount offers associated with the parent's promotion.

Copyright © 2017, 2026, Oracle and/or its affiliates.

You implement this functionality by customizing Siebel CRM to send data through AIA to the PCM_OP_SUBSCRIPTION_SET_BUNDLE opcode. For more information, see "About Using Opcodes" in *BRM Opcode Guide*.

How BRM Adds Siebel CRM Promotion Names

When a customer orders a promotion, CSRs enter information about the order in Siebel CRM, which passes the information through AIA to BRM. For each promotion ordered by an account, BRM:

- Creates a **/purchased_bundle** object, which stores information about the promotion, such as the promotion's name, validity dates, and status.
- Associates the promotion with the charge offers and discount offers it is bundled with by setting the PIN_FLD_PLAN_OBJ field of the account's **/purchased_product** and **/purchased_discount** objects to the **/purchased_bundle** POID.

When BRM generates invoices, it retrieves all of the account's purchased charge offers and discount offers (**/purchased_product** and **/purchased_discount** objects) and, if they are associated with a promotion that is active, was inactivated, or was canceled during the current billing cycle, the **/purchased_bundle** object as well.

Setting Up BRM Invoices to Display Siebel CRM Promotion Names

To include Siebel CRM promotion names on BRM invoices, perform these tasks:

- Make sure BRM is configured to display promotion details on invoices. It is configured to display promotion details by default. See "Specifying Whether BRM Displays Promotion Details on Invoices".
- Customize Siebel CRM to pass information to the PCM_OP_SUBSCRIPTION_SET_BUNDLE opcode. For more information, see "About Using Opcodes" in *BRM Opcode Guide*.
- Use an invoice template that displays promotion names and details. See "Using Invoice Templates That Display Promotion Names".

> **Note**
>
> If you configure BRM to exclude package, bundle, and charge offer pricing details from invoices, the invoice will not display Siebel CRM promotion names. See "Improving Performance by Removing Invoice Details You Do Not Need".

Specifying Whether BRM Displays Promotion Details on Invoices

When BRM generates invoices, it performs, by default, an additional search for any promotion details associated with each account. All promotion details that are found are stored in the account's **/invoice** object and displayed on the invoice.

You can enable and disable this search:

- When the search for promotion details is enabled, BRM searches for and displays on invoices details about the account's current promotions.

Copyright © 2017, 2026, Oracle and/or its affiliates.

- When the search for promotion details is disabled, promotion details are not displayed on invoices.

To configure this feature, run the **pin_bus_params** utility to change the **PromotionDetailDisplay** business parameter. For information about this utility, see "About Customizing BRM" in *BRM Developer's Guide*.

To specify whether BRM invoices displays promotion names:

1. Go to *BRM_home*/**sys/data/config**.
2. Create an XML file from the **/config/business_params** object:

   ```
   pin_bus_params -r BusParamsInvoicing bus_params_Invoicing.xml
   ```

3. In the file, change **disabled** to **enabled**:

   ```
   <PromotionDetailDisplay>enabled</PromotionDetailDisplay>
   ```

4. Save the file as **bus_params_Invoicing.xml**.
5. Load the XML file into the BRM database:

   ```
   pin_bus_params bus_params_Invoicing.xml
   ```

6. Stop and restart the CM.

To add, modify, inactivate, or cancel promotions, you must customize Siebel CRM to accept the following information and pass it to the PCM_OP_SUBSCRIPTION_SET_BUNDLE opcode. For more information, see "About Using Opcodes" in *BRM Opcode Guide*.

Copyright © 2017, 2026, Oracle and/or its affiliates.

6

Including Invoice Data from Custom Applications

Learn how to include invoice data from custom applications in your customers' invoices in Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:

- Including Invoice Data from Custom Applications
- Using Data Map Templates
- Loading the Invoice Data Map Templates

See also:

- Customizing Invoice Content
- Adding Siebel CRM Promotion Names to Invoices

Including Invoice Data from Custom Applications

If you process events that originate in a custom application, you can create a template to specify the event invoice data so that it can be included in your invoices.

BRM provides a default template file (**pin_invoice_data_map**) located in *BRM_home***/sys/data/config**. This file includes a default template for invoice data. You can modify the template or add new templates to the file.

To use this feature, you must do the following:

- Load the new invoice data template. See "Loading the Invoice Data Map Templates".
- Enable event caching in the CM configuration file. See "Improving Performance by Customizing Event Caching" for information.

Using Data Map Templates

Templates in the **pin_invoice_data_map** file define the fields in invoice data records. If you use different invoice record formats, you can create a template for each record format. The data map file can include any number of templates.

When you define templates, you specify the BRM flist fields that map to the invoice record fields. When the invoice data is processed, the fields are passed in an flist to the invoicing opcodes for processing.

The default template defines the invoice data. You can modify the template or create new templates.

To create or modify templates in the **pin_invoice_data_map** file:

1. Open the *BRM_home***/data/config/pin_invoice_data_map** file.
2. Change the template or add a new template to the end of the file. Use the following syntax:

Copyright © 2017, 2026, Oracle and/or its affiliates.

```
ID template_name
field_level   field_name
field_level   field_name
field_level   field_name
...
```

where:

- *template_name* is the name of the template.
- *field_level* is the level of the field in the flist.
- *field_name* is the associated BRM field.

For example:

```
ID INTEGRATE
0 PIN_FLD_CALLING_NUMBER
0 PIN_FLD_CALLED_NUMBER
. . .
0 PIN_FLD_BAL_IMPACTS
1 PIN_FLD_RATE_TAG
1 PIN_FLD_AMOUNT
```

3. Save and close the file.

The order of the fields in the template must correspond to the order of the fields in the invoice record. Any custom invoice data records that you process must follow these rules:

- Fields must be separated by a pound symbol: #
- Arrays and substructs must be enclosed in angle brackets: **< >**
- Each balance impact element must end with a pipe symbol: **|**

> **Note**
>
> The pipe is optional after the last element.

- The first field in the record must be the name of the corresponding invoice data template and be preceded by the @ symbol.

  For example: @**INTEGRATE**

- The template name in the record must match the template name in the **load_pin_invoice_data_map** file.

Loading the Invoice Data Map Templates

After defining invoice data map templates, you load them into the BRM database by running the **load_pin_invoice_data_map** utility.

Copyright © 2017, 2026, Oracle and/or its affiliates.

> **Note**
>
> - When you run **load_pin_invoice_data_map**, it overwrites the existing invoice data templates. If you are updating a set of templates, you cannot load new templates only. You must load complete sets of invoice data templates each time you run the utility.
> - To connect to the BRM database, **load_pin_invoice_data_map** needs a configuration file in the directory from which you run the utility. See "Connecting BRM Utilities" in *BRM System Administrator's Guide*.
> - If you defined a custom field for invoicing, you must add the full path name of the mapping file to the **load_pin_invoice_data_map** utility's **pin.conf** file. For more information, see "Making Custom Fields Available to Your Applications" in *BRM Developer's Guide*.

To load the data map templates:

1. Go to the directory that contains the utility's configuration file.
2. Run **load_pin_invoice_data_map** to load the data map template by using this syntax:

```
load_pin_invoice_data_map -d -v pin_invoice_data_map
```


Copyright © 2017, 2026, Oracle and/or its affiliates.

7

Generating Proforma Invoices

Learn how to generate a proforma invoice to expedite the invoicing process in Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:

- About Proforma Invoices
- About the Customer Process
- About the Proforma Invoice Process
- About Generating Proforma Invoices for Wholesale and Hierarchy Accounts
- Setting Up BRM to Generate Proforma Invoices
- Managing Proforma Invoices with pin_bill_accts
- Regenerating Proforma Invoices

About Proforma Invoices

Proforma invoice billing is an initial, optional step in the billing process. It is similar to final billing, however, it does not finalize the bill. It creates a pre-invoice that your customers can review for accuracy before the final bill and invoice are generated. Afterward, your customers can manually accept or reject the proforma invoice.

You typically create proforma invoices for customers with very large bills and invoices because they help improve your customers' cash flow.

Proforma invoices can use the same or a different template than your final invoices. Unlike final invoices, they include:

- Details for a specific invoice period
- The current date rather than the billing day of month (DOM)
- A different bill and invoice number
- A watermark indicating that it is a proforma invoice

> **Note**
>
> Proforma bills and invoices should not be included in any external reports.

After you generate a proforma invoice, BRM freezes the events for the invoice period, preventing any adjustments or modifications until the proforma invoice is accepted or rejected. You cannot add delayed events to a proforma invoice. These delayed events are billed in the next open invoice. You can set a validity period for proforma invoices, after which they are automatically accepted. By default, the validity period is 0 days, making them valid indefinitely.

Copyright © 2017, 2026, Oracle and/or its affiliates.

> **Note**
>
> If the validity period is 0, the invoices can never be automatically accepted, since they are valid indefinitely.

About the Customer Process

After your customers receive a proforma invoice, they can do one of the following:

- **Accept the Proforma Invoice:** BRM immediately generates a real invoice with a unique invoice number, an updated invoice date, and a due date. The charges on the real invoice will match those on the proforma invoice.

  > **Note**
  >
  > For PIN_ACTG_TYPE_BALANCE_FORWARD accounting type, you can only accept an invoice if all of the bill unit's previous invoices are accepted. Otherwise, BRM throws a validation error.

- **Modify the Proforma Invoice:** If your customers want to make changes to a proforma invoice, they must first reject it. Then, they can specify to remove or add events or items, or to move items from one bill to another. After the modifications are complete, you must generate a new proforma invoice for the customer.
- **Reject the Proforma Invoice:** All items and events associated with the proforma invoice become available for correction.

About the Proforma Invoice Process

You generate proforma invoices using the **pin_bill_accts** utility or an external client application. For more information, see:

- "Creating bills for Proforma Invoices Using pin_bill_accts" in *BRM Configuring and Running Billing*
- "Managing Proforma Invoices" in *BRM Opcode Guide*

The following outlines the proforma invoice process:

1. **Initial Run:** Proforma invoice billing moves bills from the **InProgress** state to the **PreInvoiceInProgress** state, freezing all items to prevent A/R operations from occurring on them. It updates dates and item statuses, applies taxes and discounts, and generates a proforma invoice.
2. **Approval or Rejection:** Your customer approves or rejects the proforma invoice:
   - If approved, BRM automatically generates a real invoice with all the items from the proforma invoice.
   - If rejected, all items and events associated with the proforma invoice become available for the customer to correct before you regenerate the proforma invoice.
   - If the validity period ends and your customer has not accepted or rejected the invoice, BRM automatically approves the invoice and generates a real invoice. By default, the invoices are valid indefinitely.

Copyright © 2017, 2026, Oracle and/or its affiliates.

3. **Regenerate:** (Optional) If the invoice is rejected, BRM modifies the rejected invoice and its associated items, by removing or adding items to the proforma invoice. You must provide the bill object poid ID to initiate the re-run. It then identifies and processes all pending items related to the specified bill.

About Generating Proforma Invoices for Wholesale and Hierarchy Accounts

You can generate proforma invoices for accounts with hierarchical relationships, such as parent and child accounts.

- **Generating a Proforma Invoice:** Proforma invoices are generated for the child accounts first. The final invoice is generated only for the parent account.
- **Accepting:** Once the proforma invoices for the child accounts are generated, proforma invoices are generated for the parent account. During this process, charges from child account items are rolled up to the parent item, reducing the total bill of the child account to zero.

> **Note**
>
> You must accept the proforma invoices of all child accounts before accepting the parent account's proforma invoice.

- **Rejecting:** The proforma invoice of the parent account is rejected first, then the proforma invoice of the child account is rejected. After this, the specific child bill items are open for A/R operations.
- **Regenerating:** Proforma invoices for the child accounts are regenerated, followed by the proforma invoice for the parent account. Once the proforma invoice for the parent account is regenerated, charges are rolled up to the parent account.

Setting Up BRM to Generate Proforma Invoices

To generate proforma invoices, you must first set up BRM by doing the following:

1. (Optional) Define the default validity for proforma invoices. See "Defining the Default Validity Period of Proforma Invoices".
2. (Optional) Create a custom template for proforma invoices in Oracle Analytics Publisher. For more information on custom templates, see "Designing Invoices".
3. (Optional) Change the watermark in the proforma invoice template. See "Changing the Watermark for the Proforma Invoice".
4. Customize your client application to manage proforma invoices. See "Customizing Your Client Applications to Manage Proforma Invoices".

Afterward, you can start generating proforma invoices.

Defining the Default Validity Period of Proforma Invoices

Customers can accept or reject the proforma invoices within a particular time window after the proforma invoice has been generated. This time window is the validity period of the proforma invoice. You can set the validity period of your proforma invoice by setting the

Copyright © 2017, 2026, Oracle and/or its affiliates.

PreInvoiceValidity business parameter. The default validity period is **0** days, making them valid indefinitely.

After the validity due date, BRM automatically approves the proforma invoices.

To define the default validity period:

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:

   ```
   <PreInvoiceValidity>value</PreInvoiceValidity>
   ```

   Where *value* is the default validity period for proforma invoices. The default validity is **0 days**, making them valid indefinitely.

   > ⚠ **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 © 2017, 2026, Oracle and/or its affiliates.

> **Note**
>
> You can override the validity period for specific proforma invoices based on customer criteria by using the PCM_OP_BILL_POL_CALC_PYMT_DUE policy opcode. For more information, see "Managing Proforma Invoices" in *BRM Opcode Guide*.

Changing the Watermark for the Proforma Invoice

Proforma invoices have a watermark. By default, the watermark text is **PROFORMA INVOICE**.

To change the text of the watermark, set the PIN_FLD_MESSAGE field in the PCM_OP_INV_MAKE_INVOICE opcode to the text you want to see as the watermark. For more information, see "Managing Proforma Invoices" in *BRM Opcode Guide*.

Customizing Your Client Applications to Manage Proforma Invoices

If you use a custom client application to manage invoices, you must customize it to:

- Generate, accept, or reject proforma invoices by calling the PCM_OP_BILL_MAKE_BILL opcode. See "Managing Proforma Invoices" in *BRM Opcode Guide*.
- Move items from one bill to another by calling the PCM_OP_BILL_MOVE_ITEM opcode. See “Moving Bill Items” in *BRM Opcode Guide*.

Managing Proforma Invoices with pin_bill_accts

You can use the **pin_bill_accts** utility to accept, reject, and regenerate proforma invoices:

- Generating Proforma Invoices with pin_bill_accts
- Accepting Proforma Invoices with pin_bill_accts
- Rejecting Proforma Invoices with pin_bill_accts
- Regenerating Proforma Invoices
- Configuring the XML File for Proforma Invoices
- Configuring the Text File for Proforma Invoices

Generating Proforma Invoices with pin_bill_accts

You can generate a proforma invoice using the **pin_bill_accts** utility.

There are three ways in which you can generate the invoice.

- To generate a proforma invoice for all your bills and items, run the following command:

  ```
  pin_bill_accts -preinvoice bill -verbose
  ```

- To generate a proforma invoice for a specific set of accounts using an XML input file, run the command using the **-file** option:

  ```
  pin_bill_accts -preinvoice bill -file fileName.xml
  ```

  For more information on *fileName*, see "Configuring the XML File for Proforma Invoices".

Copyright © 2017, 2026, Oracle and/or its affiliates.

- To generate a proforma invoice for a specific set of accounts in an flist format, you use a properly formatted text file and run the command using the **-from_file** option:

  ```
  pin_bill_accts -preinvoice bill -from_file fileName.txt
  ```

  For more information on *fileName*, see "Configuring the Text File for Proforma Invoices".

Accepting Proforma Invoices with pin_bill_accts

If all of the items in the proforma invoice are correct, you can accept it.

There are three ways in which you can accept the invoice.

- To accept all the proforma invoices generated, run the following command:

  ```
  pin_bill_accts -preinvoice accept -verbose
  ```

- To accept a proforma invoice for a specific set of accounts using an XML input file, run the command using the **-file** option:

  ```
  pin_bill_accts -preinvoice accept -file fileName.xml
  ```

  For more information on *fileName*, see "Configuring the XML File for Proforma Invoices".

- To accept a proforma invoice for a specific set of accounts in an flist format, you use a properly formatted text file and run the command using the **-from_file** option:

  ```
  pin_bill_accts -preinvoice accept -from_file fileName.txt
  ```

  For more information on *fileName*, see "Configuring the Text File for Proforma Invoices".

Rejecting Proforma Invoices with pin_bill_accts

If you are not satisfied with the proforma invoice generated, you can reject it.

There are three ways in which you can reject the invoice.

- To reject all the proforma invoices generated, run the following command:

  ```
  pin_bill_accts -preinvoice reject -verbose
  ```

- To reject a proforma invoice for a specific set of accounts using an XML input file, run the command using the **-file** option:

  ```
  pin_bill_accts -preinvoice reject -file fileName.xml
  ```

  For more information on *fileName*, see "Configuring the XML File for Proforma Invoices".

- To reject a proforma invoice for a specific set of accounts in an flist format, you use a properly formatted text file and run the command using the **-from_file** option:

  ```
  pin_bill_accts -preinvoice reject -from_file fileName.txt
  ```

  For more information on *fileName*, see "Configuring the Text File for Proforma Invoices".

Copyright © 2017, 2026, Oracle and/or its affiliates.

When the proforma invoice is rejected, the items in the proforma invoice are no longer frozen and A/R operations are allowed on them. Then, you can make corrections, including moving the disputed items to another bill in the bill unit. You move bill items by calling the PCM_OP_BILL_MOVE_ITEM opcode. For more information, see "Moving Bill Items" in *BRM Opcode Guide*.

Regenerating Proforma Invoices

You can regenerate a proforma invoice for the rejected items using the **pin_bill_accts** utility or the PCM_OP_BILL_MAKE_BILL opcode. To regenerate a proforma invoice using the opcode, see "Managing Proforma Invoices in *BRM Opcode Guide*".

To regenerate a proforma invoice using the **pin_bill_accts** utility, you generate a new proforma invoice for the rejected items by running the following command:

```
pin_bill_accts -preinvoice bill -file fileName
```

The utility identifies the proforma bill poid ID from the input file, and regenerates the corresponding bill.

Upon regeneration, the corresponding bill and its items are modified and moved back to the **PreInvoiceInProgress** and **PIN_ITEM_STATUS_OPEN** states respectively. Then, the regenerated proforma invoice can be accepted and the final invoice can be generated.

Configuring the XML File for Proforma Invoices

To configure the XML file to generate, accept, and reject proforma invoices, you use the sample file **pin_bill_run_control.xml** in the *BRM_home*/**apps/pin_billd** directory. Use the `<Bill>` tag to accept, reject, and regenerate proforma invoices.

The following is a sample of the XML file to generate, accept, or reject proforma invoices:

```
<BusinessConfiguration>
    <BillRunConfiguration>
        ‹BillingList>
            <Account>123456</Account>
            <Billinfo>123456</Billinfo>
            <Bill>123456789012345678</Bill›
        </BillingList>
    </BillRunConfiguration>
</BusinessConfiguration>
```

Configuring the Text File for Proforma Invoices

You can configure data in an flist format to generate, accept, and reject proforma invoices, by configuring it in a text (**.txt**) file.

The following is a sample of the text file to generate, accept, and reject proforma invoices:

```
0 PIN_FLD_RESULTS       ARRAY [0] allocated 20, used 2
1     PIN_FLD_ACCOUNT_OBJ    POID [0] 0.0.0.1 /account 123456 0
1     PIN_FLD_POID           POID [0] 0.0.0.1 /billinfo 123456 0
1     PIN_FLD_BILL_OBJ       POID [0] 0.0.0.1 /bill 123456789012345678 0
```


Copyright © 2017, 2026, Oracle and/or its affiliates.

> **Note**
>
> The PIN_FLD_BILL_OBJ poid ID is only used to accept, reject, or regenerate a specific proforma invoice.

Copyright © 2017, 2026, Oracle and/or its affiliates.

8

Generating Invoices

Learn how to generate invoices in Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:

- Generating Invoices Automatically
- Generating Invoices Manually
- Generating Detailed Invoices from an External File
- Generating Invoices for Non-Invoice Payment Methods
- Exporting Invoices

To customize how invoices are generated, use the PCM_OP_INV_POL_POST_MAKE_INVOICE and PCM_OP_INV_POL_PREP_INVOICE opcodes. See "Invoice Opcode Workflows" in *BRM Opcode Guide*.

Generating Invoices Automatically

Typically, you generate invoices automatically as part of running the **pin_bill_day** billing script. This script runs several utilities, including **pin_inv_accts**, the invoicing utility. For more information on **pin_bill_day**, see "Running the Billing Scripts" in *BRM Configuring and Running Billing*.

If you configured BRM to use a separate database schema for invoices, **pin_inv_accts** uses **dm_invoice** to store invoices in the separate schema. See "Storing Invoices in a Separate Database Schema".

Generating Invoices Manually

To generate invoices manually:

1. Run the other billing utilities.

   For information about the other utilities, see "About the Billing Utilities" in *BRM Configuring and Running Billing*.

2. (Optional) To store invoices in their own database schema, follow the procedures in "Storing Invoices in a Separate Database Schema".

3. Go to a directory with a valid invoicing configuration file. See "Connecting BRM Utilities" in *BRM System Administrator's Guide*.

4. Run **pin_inv_accts** and specify any necessary parameters. For example, to generate detailed invoices for bills that have the summary flag set, use the **-detail** parameter.

Copyright © 2017, 2026, Oracle and/or its affiliates.

> **Note**
>
> If bill suppression is enabled on an account that has a paying parent bill unit and the nonpaying bill unit threshold of an account that owns the corresponding nonpaying child bill unit is exceeded, invoicing fails. This occurs because nonpaying child bill units, which use the bill number of their paying parent bill unit, are generated even when billing for the parent bill unit is suppressed. In such cases, they do not contain bill numbers, which invoices require. To exclude invoicing for these bills, run **pin_inv_accts** with the **-skip_blank_billnos** parameter. (Nonpaying child bill units contain bill numbers only when their paying parent bill unit is billed.) See "Invoicing for Hierarchical Bill Units".

For more information on the syntax and parameters for **pin_inv_accts**, see "pin_inv_accts".

Generating Detailed Invoices from an External File

To generate detailed invoices from bills contained in an external file, run **pin_inv_accts** with the **-file** parameter:

```
pin_inv_accts -file filename
```

The file contents must be in pin_flist format. For example:

```
0 PIN_FLD_RESULTS      ARRAY [0]
1   PIN_FLD_POID        POID [0] 0.0.0.1 /bill 26011 0
0 PIN_FLD_RESULTS      ARRAY [1]
1   PIN_FLD_POID        POID [0] 0.0.0.1 /bill 26091 0
```

For more information, see "About Formats for Storing Invoices".

This operation overwrites the PIN_FLD_INVOICE_OBJ value in the **/bill** object.

> **Note**
>
> When **pin_inv_accts** processes bills for parent accounts in account hierarchies, it searches BRM for the nonpaying child bill units whose paying parent bill units are owned by the parent accounts and generates their invoices automatically. In general, the file should not contain the bills for nonpaying bill units; however, if they are present, invoices for the nonpaying bill units are generated.

Generating Invoices for Non-Invoice Payment Methods

To generate invoices for non-invoice payment methods, run **pin_inv_accts** with the **-pay_type** parameter:

```
pin_inv_accts -pay_type payment_method_ID
```

where *payment_method_ID* is the element ID of the payment method.

For a list of payment methods and their element IDs, see "Default Payment Methods" in *BRM Configuring and Collecting Payments*.

For example, to generate invoices for credit cards, run the utility with the following command:

Copyright © 2017, 2026, Oracle and/or its affiliates.

```
pin_inv_accts -pay_type 10003
```

> **Note**
>
> This generates *detailed* invoices. To generate *summary* invoices, customize the PCM_OP_INV_POL_SELECT policy opcode to pass the summary value in the PIN_FLD_FLAGS field on its input flist (PIN_FLD_FLAGS = 0x0002). You can also run **pin_inv_accts** with **-summary** parameter to generate summary invoices. See "Invoice Opcode Workflows" in *BRM Opcode Guide* for more information.

Exporting Invoices

You can export the invoices in your database to files in any of these formats: pin_flist, XML, HTML, or DOC1.

> **Note**
>
> If you use DOC1 invoice software, you can export BRM invoices in DOC1 format and then open them in the DOC1 program.

To export invoices to files:

1. Change to a directory with a valid invoicing configuration file, for example*BRM_home*/**apps/pin_inv**. See "Connecting BRM Utilities" in *BRM System Administrator's Guide*.
2. Open the **pin.conf** file and find the **pin_inv_export** entries.
   a. If you are exporting to a format other than **pin_flist**, edit the **invoice_fmt** entry. Enter one of the following that corresponds to the format you want to use for the invoice files:
      - **text/pin_flist**
      - **text/xml**
      - **text/html**
      - **text/doc1**

      For example, to create HTML invoice files, enter:

      ```
      - pin_inv_export  invoice_fmt text/html
      ```

      If you do not edit this entry, **pin_inv_export** creates files in **pin_flist** format.
   b. In the **export_dir** entry, enter the directory in which to store the files. You can enter a complete path or a path relative to the directory where you run **pin_inv_export**:

      ```
      - pin_inv_export  export_dir ./invoice_dir
      ```
3. To export summary invoices for regular bills only, run the **pin_inv_export** utility with no parameters:

   ```
   pin_inv_export
   ```


Copyright © 2017, 2026, Oracle and/or its affiliates.

This creates a separate file for each regular invoice in the database.

4. To export regular invoices for a specified date range:

   ```
   pin_inv_export -start startDate -end endDate
   ```

   See "pin_inv_export" for more information about how to use these parameters.

5. To export summary invoices for trial bills only, run the following command:

   ```
   pin_inv_export -verbose -trial
   ```

6. To export detailed invoices for both trial and regular bills, run the following command:

   ```
   pin_inv_export -verbose -detail filename
   ```

   where *filename* is the name of the file that specifies the bills for which invoices are to be generated.

> **Note**
>
> - When invoices for parent paying accounts are exported, and the invoices for the nonpaying bills were not consolidated into the paying account's invoice, a separate invoice file for each nonpaying bill unit and the paying bill unit is exported.
>
>   The file-naming convention is *accountPOID_billPOID_date.ext*.
>
>   If a nonpaying invoice or parent invoice fails during generation, nonpaying invoices are not exported.
> - To determine which invoice file is the parent account invoice, customize the appropriate invoice format policy opcode to generate another file containing the parent invoice file name for a given bill.
> - You cannot use the **-detail** parameter to export detailed invoices for only regular bills or only trial bills.

For a description of this utility's syntax and parameters, see "pin_inv_export".

Copyright © 2017, 2026, Oracle and/or its affiliates.

9

Sending Invoices to Customers

Learn how to send invoices to customers by using Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:

- Sending Invoices to Customers
- Setting the Maximum Invoice Size for Email
- Sending Invoices to Accounts That Use the Invoice Payment Method
- Sending Invoices to Accounts That Do Not Use the Invoice Payment Method
- Configuring the Email Data Manager for Printing

Sending Invoices to Customers

After generating invoices, you can email them to your customers or print them for postal mailing by using the **pin_inv_send** utility. The utility determines whether to email or print an account's invoice by reading the account's delivery preference, which is set by a CSR during account creation.

By default, the utility emails invoices in HTML format, but you can change the format to DOC1. For more information, see "pin_inv_send".

Setting the Maximum Invoice Size for Email

To set the maximum size of invoices that can be sent by email:

1. Open the invoice configuration file (*BRM_home***/apps/pin_inv/pin.conf**).
2. Set the value of the **inv_send_size** entry in kilobytes and make sure it is not commented.
3. Save and close the file.

Sending Invoices to Accounts That Use the Invoice Payment Method

To email or print invoices for bills that use the Invoice payment method:

1. If you store invoices in a separate database schema, enter the database schema number in the invoicing configuration file (*BRM_home***/apps/pin_inv/pin.conf**). For example:

   ```
   - pin_inv_send  invoice_db 0.0.0.2 /invoice 0
   ```

   Then follow the procedures in "Storing Invoices in a Separate Database Schema".
2. Start the Email Data Manager. See "Sending Email to Customers Automatically" in *BRM Managing Customers*.

Copyright © 2017, 2026, Oracle and/or its affiliates.

3. Go to a directory with a valid invoicing configuration file. See "Connecting BRM Utilities" in *BRM System Administrator's Guide*.
4. Run the **pin_inv_send** utility by entering this command:

   ```
   pin_inv_send
   ```

   For a detailed description of this utility's syntax and parameters, see "pin_inv_send".

Sending Invoices to Accounts That Do Not Use the Invoice Payment Method

To email or print invoices for non-invoice bill units, run the **pin_inv_send** utility with the **pay_type** parameter:

```
pin_inv_send -pay_type id
```

For *id*, enter one payment method ID as shown in Table 9-1:

Table 9-1 Payment Method IDs

Payment method	ID
Undefined	0
Prepaid	10000
Invoice	10001
Debit	10002
Credit card	10003
Direct debit	10005
Smart card	10006
Nonpaying	10007
Beta	10008
Internal	10009
Guest	10010
Cash	10011
Check	10012
Wire transfer	10013
Payorder	10014
Postal order	10015
Voucher	10016
SEPA	10018

Configuring the Email Data Manager for Printing

You can use the third-party tool, HTML to PostScript converter (**html2ps**), to print invoices. Follow these instructions to download the **html2ps** tool and to configure the Email Data Manager (**dm_email**) for printing invoices:

Copyright © 2017, 2026, Oracle and/or its affiliates.

1. Download from the Internet and install the **html2ps** converter tool.
2. Open the **dm_email** configuration file (*BRM_home***/sys/dm_email/pin.conf**).
3. In the **html2ps** entry, enter the path to the **html2ps** tool.

   For example:

   ```
   - dm_email  html2ps  path_to_html2ps_tool
   ```

4. In the **printer** entry, enter the name of your printer:

   ```
   - dm_email  printer  printer_name
   ```

5. In the **tmp_dir** entry, enter the name of the directory where BRM saves temporary HTML and PostScript files when the **send** utility runs. For example:

   ```
   - dm_email  tmp_dir  /tmp
   ```

6. Save and close the file.
7. Stop and restart the Email Data Manager.

Copyright © 2017, 2026, Oracle and/or its affiliates.

10

Displaying Invoices

Learn how to display your customers' invoices in Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:

- About Displaying Invoices
- Using the XML Sample Invoice Viewing Utility

About Displaying Invoices

BRM provides these options for displaying invoices:

- The Invoice Viewer in Billing Care.
- Sample XML invoice viewing utility (**pin_inv_view.pl**).
- Oracle Analytics Publisher client, if you use the BRM-Oracle Analytics Publisher Integration framework to generate invoice documents.

You can also use a third-party viewer to display invoices. For information on how to format and view invoices, refer to the viewer's documentation.

To customize how invoices are displayed, use the PCM_OP_INV_POL_FORMAT_VIEW_INVOICE policy opcode. See "Invoice Opcode Workflows" in *BRM Opcode Guide*.

> **Note**
>
> - You cannot change invoice information in the Invoice Viewer.
> - A physical invoice is generated to request payment from each account that has a payment method of Invoice. If your BRM system is properly configured, you can see the invoice files that are stored in the BRM database for accounts with other payment methods.

Using the XML Sample Invoice Viewing Utility

You can use the **pin_inv_view.pl** utility to view your invoices in XML by using the BRM sample style sheet or any other style sheet you specify. This utility uses Perl and CGI to display invoices you select in any XML-capable browser.

To use this utility, go to the *BRM_home***/source/apps/sampleviewer** directory. You need to run **make** to build **pin_inv_view.pl**.

Copyright © 2017, 2026, Oracle and/or its affiliates.

11

Storing Invoices

Learn how to configure the format in which invoices are stored and the schema in which they are stored in the Oracle Communications Billing and Revenue Management (BRM) database.

Topics in this document:

- About Formats for Storing Invoices
- Specifying the Default Format in Which to Store Invoices in BRM
- Storing Invoices in a Separate Database Schema

About Formats for Storing Invoices

BRM stores each invoice as an object in the database. By default, invoices are stored in the same database schema as the accounts you are invoicing, but you can configure BRM to store invoices in a separate schema. For more information, see "Storing Invoices in a Separate Database Schema".

Invoices are stored in at least one of these formats:

- **pin_flist**: The internal BRM data structure. By default, BRM stores invoice data in this format and then reformats it into HTML format to display in the Invoice Viewer.

  This is the most useful format for these reasons:

  - All the available display options work with invoices stored in pin_flist format.
  - You can write a program that saves the pin_flist invoices in any other format. This is useful when providing invoice data to third-party invoice viewers. For information about the pin_flist format, see "Understanding Flists and Storable Classes" in *BRM Developer's Guide*.

- **XML**: You must use XML format if you are generating the final invoice document by using the BRM-Oracle Analytics Publisher Invoicing Integration package. In addition, you might use this storage format if you are using XML to design and display invoices. See "Using Oracle Analytics Publisher for Invoicing".

  To use XML as an alternative or additional storage format, modify the PCM_OP_INV_POL_FORMAT_INVOICE and PCM_OP_INV_POL_FORMAT_INVOICE_XML opcodes. See "Customizing the Format for Printed Invoices" and "Customizing the Format for XML Invoices" in *BRM Opcode Guide*.

  > **Note**
  >
  > You can store invoice data in both pin_flist and XML formats at the same time. To do so, modify the PCM_OP_INV_POL_FORMAT_INVOICE policy opcode.

You have the option of also storing invoices in HTML or DOC1 format. These are primarily display formats and, in most cases, storing invoices in these formats is not useful. However, you might need to store invoices in the same format you use to display them.

Copyright © 2017, 2026, Oracle and/or its affiliates.

Storing invoices in HTML or DOC1 format requires extensive programming. Use the PCM_OP_INV_POL_FORMAT_INVOICE_HTML and PCM_OP_INV_POL_FORMAT_INVOICE_DOC1 policy opcodes. See "Customizing the Format for HTML Invoices" and "Customizing the Format for DOC1 Invoices" in *BRM Opcode Guide*.

Specifying the Default Format in Which to Store Invoices in BRM

By default, BRM stores invoices in flist format. You can configure BRM to store invoices in XML format by modifying a field in the **invoicing** instance of the **/config/business_params** object.

To enable this feature, run the **pin_bus_params** utility to change the **InvoiceStorageType** business parameter. For information about this utility, see "pin_bus_params" in *BRM Developer's Guide*.

To specify the default format in which to store invoices:

1. Go to *BRM_home***/sys/data/config**.
2. Create an XML file from the **/config/business_params** object:

   ```
   pin_bus_params -r BusParamsInvoicing bus_params_Invoicing.xml
   ```

3. Do one of the following:
   - To specify flist as the output format, set **InvoiceStorageType** to **0**. This is the default.
   - To specify XML as the output format, set **InvoiceStorageType** to **1**.

   For example, to specify XML format:

   ```
   <InvoiceStorageType>1</InvoiceStorageType>
   ```

4. Save the file as **bus_params_Invoicing.xml**.
5. Load the XML file into the BRM database:

   ```
   pin_bus_params bus_params_Invoicing.xml
   ```

6. Stop and restart the CM.

Storing Invoices in a Separate Database Schema

> **Note**
>
> Storing invoices in a separate database schema is supported only on Oracle databases.

By default, your invoices are stored in the same database schema that contains the invoiced accounts. However, using a separate schema to store invoices provides the following benefits:

- Speeds up the invoicing process.
- Potentially stores a large number of invoices.

Copyright © 2017, 2026, Oracle and/or its affiliates.

- Enables you to view, email, and print invoices without affecting performance of the main schema.

The procedures for storing invoices in a separate schema include the following:

1. Installing the Invoice Data Manager
2. Configuring BRM to Use a Separate Invoice Database Schema
3. Configuring Invoice Applications to Use a Separate Invoice Database Schema
4. Starting the Invoice Data Manager

Installing the Invoice Data Manager

> **Note**
>
> The Invoice Data Manager (DM) is supported only on Oracle databases.

The Invoice DM is not installed by default when you install BRM. To install the Invoice DM:

1. Make sure you have a separate database schema installed and available to use for storing invoices.

   See "Installing a Multischema System" in *BRM Installation Guide* and "Adding Database Schemas to a Multischema System" in *BRM System Administrator's Guide*.

2. On the machine containing BRM, open the *BRM_home*/**setup/pin_setup.values** file in a text editor.
3. Add **"dm_invoice"** to the **@COMPONENT_LIST** line:

   ```
   @COMPONENT_LIST = ("dm_oracle", "dm_invoice", "cm");
   ```

   > **Note**
   >
   > Make sure **"dm_invoice"** appears after **"dm_oracle"**.

4. Modify the entries in the **pin_setup.values** file shown in Table 11-1:

Table 11-1 Entries to be Modified in pin_setup File

Entry	Description
$INVOICE_DB{'user'}	Must contain the user name to log in to your invoice schema. This user name must be different from the one used for any other schema in the database.
$INVOICE_DB{'password'}	Must contain the password to log in to your invoice schema.
$INVOICE_DB{'alias'}	Must contain the database alias of the invoice schema.
$INVOICE_DB{'Host'}	Must contain the host name of the machine running the invoice schema.
$INVOICE_DB{'tables_group'}	Must contain the name of the data tablespace.
$INVOICE_DB{'indexes_group'}	Must contain the name of the index tablespace.
$DM_INVOICE{'port'}	Must contain the port number of the Invoice DM.

Copyright © 2017, 2026, Oracle and/or its affiliates.

Table 11-1 (Cont.) Entries to be Modified in pin_setup File

Entry	Description
$DM_INVOICE{'db_num'}	Must contain the database schema number of the invoice database.

5. Install **dm_invoice** by using one of these methods:
   - To install all BRM components, run this script:

     BRM_home`/setup/pin_setup`
   - To install only the Invoice DM, run this script:

     BRM_home`/setup/scripts/pin_cmp_dm_invoice.pl`

   For the complete installation procedure, see "Installing BRM" in *BRM Installation Guide*.

Configuring BRM to Use a Separate Invoice Database Schema

You must configure the CM to use the invoice DM and separate invoice database schema. To configure the CM, perform the following procedure on the machine containing the CM:

1. Open the Connection Manager (CM) configuration file (*BRM_home***/sys/cm/pin.conf**).

   For information about configuration files, see "Using Configuration Files to Connect and Configure Components" in *BRM System Administrator's Guide*.
2. In the **invoice_db** entry, enter the invoice database schema number.

   For example:

   ```
   - fm_cust_pol  invoice_db 0.0.6.1 /invoice 0
   ```

   0.0.6.1 is the BRM default database schema number for the invoice schema.

   The customer Facilities Module (FM) uses this entry to store the invoice database schema numbers for new customers. BRM uses that information to store the invoices each time you generate invoices.

   > **Note**
   >
   > The invoicing utilities do not use this entry.

3. Make sure you have a **dm_pointer** entry for the invoice schema referenced in the **invoice_db** entry. The **pin_inv_accts** utility uses this entry.

   For more information, see the documentation in the CM configuration file.
4. Make sure you have a **dm_attributes** entry for the invoice schema:

   ```
   - cm dm_attributes 0.0.6.1 assign_account_obj,searchable
   ```

   For more information, see the documentation in the CM configuration file.
5. Save and close the file.
6. Stop and restart the CM.

Copyright © 2017, 2026, Oracle and/or its affiliates.

Configuring Invoice Applications to Use a Separate Invoice Database Schema

To use a separate database schema for storing invoices:

1. Open the invoice configuration file (*BRM_home*/**apps/pin_inv/pin.conf**) in a text editor.
2. In the Invoice DB number section for the **pin_inv_send** and **pin_inv_export** applications, change the **invoice_db** parameter value to your invoice database schema number. The following examples use the BRM default invoice database schema number, **0.0.6.1**:

   ```
   # - pin_inv_send invoice_db 0.0.6.1 /invoice 0
   # - pin_inv_export invoice_db 0.0.6.1 /invoice 0
   ```

3. Save and close the file.

Starting the Invoice Data Manager

To start the Invoice DM, enter the **pin_ctl start dm_invoice** command.

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

Configuring the Invoice pin.conf for Multiple Database Schemas

In a multischema environment, to store invoices in the database schema 0.0.0.2:

1. Open the invoice configuration file (*BRM_home*/**apps/pin_inv/pin.conf**).
2. Change the login name in the following entry to your login name:

   ```
   - nap login_name name
   ```

3. Change the **database** parameter value to **0.0.0.2** in the following entries:

   ```
   - pin_inv_accts database 0.0.0.2 /search 0
   - pin_inv_send database 0.0.0.2 /search 0
   - pin_inv_export database 0.0.0.2 /search 0
   ```

4. Save and close the file.

For information on running invoice utilities in a multischema environment, see "Setting Up Invoicing on Multiple Database Schemas".

Copyright © 2017, 2026, Oracle and/or its affiliates.

12

Using Oracle Analytics Publisher for Invoicing

Learn how to utilize Oracle Analytics Publisher to design, generate, and customize invoice documents in Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:

- About Using Oracle Analytics Publisher for Invoicing
- Designing Invoices

About Using Oracle Analytics Publisher for Invoicing

Oracle Analytics Publisher enables you to create more stylized and detailed invoices in PDF format.

Oracle Analytics Publisher lets you choose from various desktop tools, such as Microsoft Word and Adobe Acrobat, to create document templates. While it supports data sources such as databases, Web services, and files, the BRM-Oracle Analytics Publisher invoicing integration only uses SQL queries directly on the database as the data source for better performance. By default, the BRM-Oracle Analytics Publisher invoicing integration supports PDF as an output format and delivery options such as email, FTP, and SFTP.

With Oracle Analytics Publisher, you use an SQL bursting query, which splits data into blocks and generates documents for those blocks. Bursting increases throughput performance in Oracle Analytics Publisher. You must install Oracle Analytics Publisher and Oracle Analytics Publisher Desktop to use this feature.

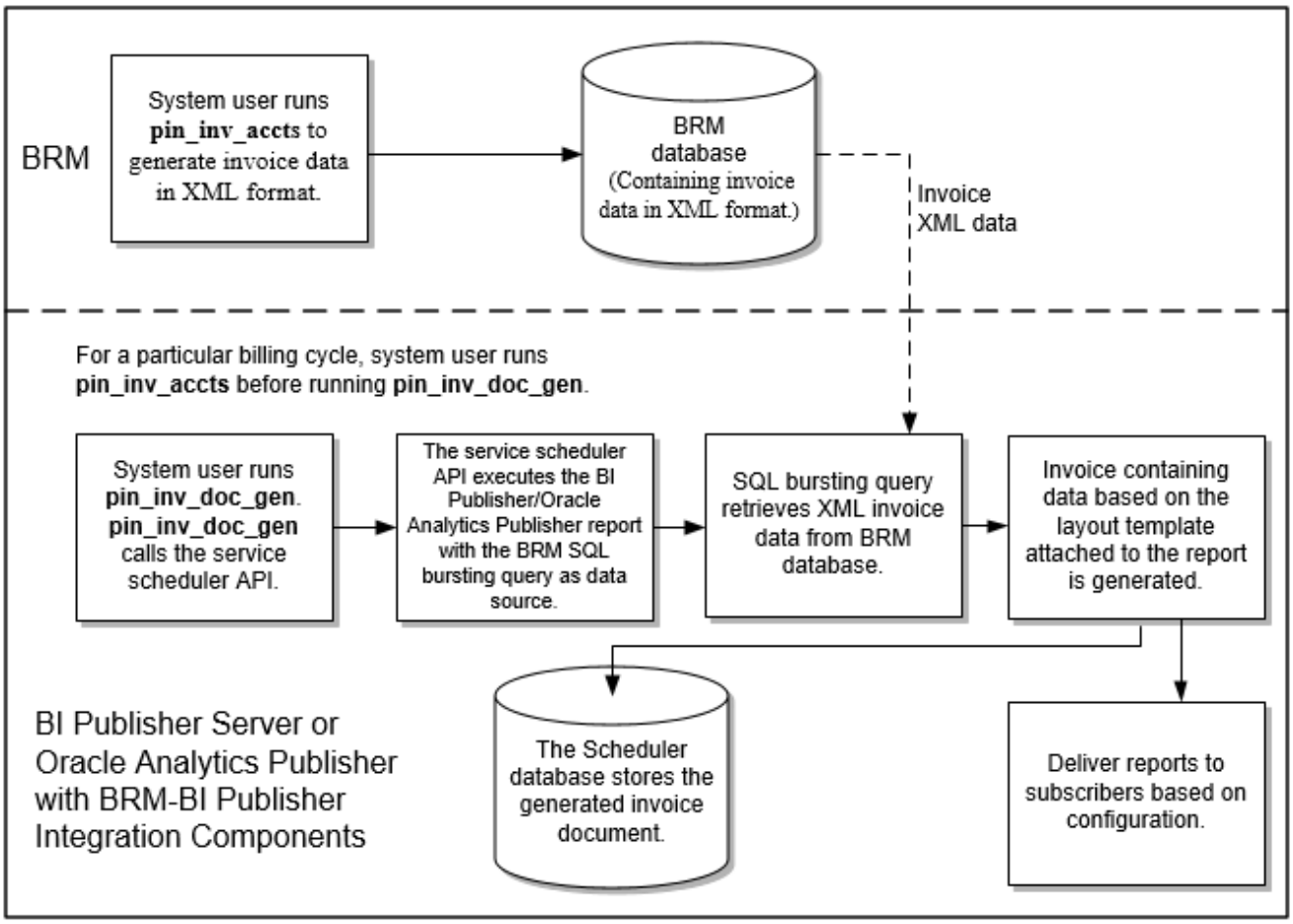
When you use the BRM-Oracle Analytics Publisher integration:

- BRM generates the invoicing data in XML format and then passes the data to Oracle Analytics Publisher.
- Oracle Analytics Publisher formats the invoice, delivers it, and then stores it in the Oracle Analytics Publisher database.

Figure 12-1 shows the BRM and Oracle Analytics Publisher invoice generation process:

Copyright © 2017, 2026, Oracle and/or its affiliates.

Figure 12-1 BRM and Oracle Analytics Publisher Invoice Generation Process



The BRM-Oracle Analytics Publisher integration generates invoice documents as follows:

> **Note**
>
> The BRM-Oracle Analytics Publisher integration must be enabled. See "Enabling the BRM-Oracle Analytics Publisher Integration".

1. **BRM associates bill units with an Oracle Analytics Publisher invoice and report**. When accounts are created, BRM associates each bill unit (**/billinfo** object) with an Oracle Analytics Publisher invoice report and layout template.
2. **BRM generates a bill.** BRM creates a bill for each account bill unit when it runs the **pin_bill_accts** utility as part of the daily billing script. BRM needs a bill before generating an invoice. See "Running Billing Utilities" in *BRM Configuring and Running Billing*.
3. **BRM generates the invoice and stores it in the BRM database.** When the **pin_inv_accts** utility is run, it stores the invoice data in an **/invoice** object in the BRM database. See "Generating Invoices".

Copyright © 2017, 2026, Oracle and/or its affiliates.

> **Note**
>
> To generate invoice documents in Oracle Analytics Publisher, you must store the invoice data in XML format. See "About Formats for Storing Invoices".

4. **The pin_inv_doc_gen utility**:
   - **Runs reports**. This utility, based on the input parameters, starts and runs appropriate Oracle Analytics Publisher reports for invoice documents. It starts an Oracle Analytics Publisher invoicing report, with the SQL bursting data query, through a scheduled job. See "Generating Oracle Analytics Publisher Invoice Documents".
   - **Calls the SQL bursting data query to pull the invoice data from the BRM database**. The SQL bursting data query pulls the invoice data in XML format from the BRM database and returns the XML data necessary to generate invoice documents by the **pin_inv_doc_gen** utility.
   - **Generates the invoice in the preferred format and sends it using the configured information**. The delivery information is stored in the **/payinfo/invoice** object of the account **/billinfo** object.
   - **Stores the invoice document in the Oracle Analytics Publisher database.** The utility stores the generated invoice documents in the Scheduler database. See "Storing the Invoice Document".

The BRM-Oracle Analytics Publisher invoicing integration package contains:

- The SQL bursting query.
- Reports for generating invoice documents for consumer and corporate accounts. See "About Reports for Invoice Document Generation".
- Sample layout templates in rich text format (RTF) for consumer and corporate accounts. See "About Invoice Layout Templates for Invoice Document Generation".

Designing Invoices

To help you design invoices, you can use the Oracle Analytics Publisher layout templates or the reports provided with the BRM-Oracle Analytics Publisher invoicing integration package. You can also modify the templates to include late payment fees and finance charges.

About Invoice Layout Templates for Invoice Document Generation

The BRM-Oracle Analytics Publisher invoicing integration package contains six layout templates in RTF for the consumer and corporate accounts which can be customized.

If the BRM-Oracle Analytics Publisher integration framework is enabled, when you create an account, the bill units of the account are associated with an **/associated_bus_profile** object. The **/associated_bus_profile** object contains the report and template names that are used for generating invoice documents for this account on the Oracle Analytics Publisher server. See "Creating an Invoicing Configuration Business Profile".

BRM provides the following default template files for regular invoices:

- **BRM_Corporate_Invoice.rtf**
- **BRM_Consumer_Invoice.rtf**

BRM provides the following default **Replacement Invoice** template files:

Copyright © 2017, 2026, Oracle and/or its affiliates.

- **BRM_Corporate_Replacement_Invoice.rtf**
- **BRM_Consumer_Replacement_Invoice.rtf**

When you use the default template for a replacement invoice, the resulting document contains all the line items and information from the previous bill or invoice. In addition, it lists the previous bill number, the previous bill date, the previous bill totals and the corrections totals. The item summary section lists the previous totals, the correction totals and the current totals for each item. The events details section lists the previous total, correction amount and current total for each event.

BRM provides the following default **Invoice Correction Letter** template files:

- **BRM_Corporate_Correction_Invoice.rtf**
- **BRM_Consumer_Correction_Invoice.rtf**

The invoice correction letter is in a letter format. When you use the default template for an invoice correction letter, the resulting document contains a brief explanation of the correction, a bill summary section and the details of only the corrections made to the previous bill or invoice. The invoice correction letter displays the previous bill number, the previous bill date, the previous bill total, the total correction amount and the current total.

The report file that Oracle Analytics Publisher associates with each of the previously listed default template files for invoices is:

- **BRM_Invoice_Bursting_Report.xdo**

About Reports for Invoice Document Generation

The BRM-Oracle Analytics Publisher invoicing integration package contains two reports for consumer and corporate accounts. Details of the construction of these reports are:

- The data source of the sample Oracle Analytics Publisher reports is the SQL bursting query.
- Each report has a layout template in RTF associated with it that decides the look and feel of the invoice document.
- The output formats supported for the invoice document are PDF, XML, RTF, PPT, Excel, and HTML. The default output format specified in the reports is PDF.

> **Note**
>
> For the invoice documents to support data from various data sources, you need to configure additional data sources in the sample reports and customize the sample layout template.

Including Late Payment Fees in Invoices

If you are generating the final invoice document by using the BRM-Oracle Analytics Publisher Invoicing Integration framework, you can customize the invoice data to include the late payment fee.

To include late payment fees in the invoice data generated in XML format:

1. Extend the **/item** storable class to include **/item/late_fee**. See "Creating Custom Fields and Storable Classes" in *BRM Developer's Guide*.

Copyright © 2017, 2026, Oracle and/or its affiliates.

2. Open the *BRM_home*/sys/data/pricing/example/config_item_types.xml file and define a type for the item. For example:

```
<ItemTypeElement>
    <ItemTag>LateFee</ItemTag>
    <ItemDescription>Late_Fee</ItemDescription>
    <ItemType precreate="false" type="cumulative">/item/late_fee</
ItemType>
</ItemTypeElement>
```

3. Save the file.
4. Load the configuration item types by using the **load_config_item_types** utility:

```
load_config_item_types config_item_types.xml
```

5. Open the *BRM_home*/sys/data/pricing/example/config_item_tags.xml and define a tag for the new service. For example:

```
<ItemTagElement>
    <ItemTag>LateFee</ItemTag>
    <EventType>/event/billing/late_fee</EventType>
    <ServiceType>/account</ServiceType>
</ItemTagElement>
```

6. Save the file.
7. Load the configuration item tags by using the **load_config_item_tags** utility:

```
load_config_item_tags config_item_tags.xml
```

Including Finance Charges in Invoices

If you are generating the final invoice document by using the BRM-Oracle Analytics Publisher Invoicing Integration framework, you can customize the invoice data generated in XML format to include the finance charges.

To include finance charges in the invoice data generated in XML format:

1. Extend the **/item** storable class to include **/item/finance_charges**. See "Creating Custom Fields and Storable Classes" in *BRM Developer's Guide*.
2. Open the *BRM_home*/sys/data/pricing/example/config_item_types.xml file and define a type for the item. For example:

```
<ItemTypeElement>
    <ItemTag>FinanceCharges</ItemTag>
    <ItemDescription>Finance_Charges</ItemDescription>
    <ItemType precreate="false" type="cumulative">/item/finance_charges</
ItemType>
</ItemTypeElement>
```

3. Save the file.

Copyright © 2017, 2026, Oracle and/or its affiliates.

4. Load the configuration item types by using the **load_config_item_types** utility:

   ```
   load_config_item_types config_item_types.xml
   ```

5. Open the *BRM_home*/**sys/data/pricing/example/config_item_tags.xml** and define a tag for the new service. For example:

   ```
   <ItemTagElement>
       <ItemTag>FinanceCharges</ItemTag>
       <EventType>/event/billing/finance_charges</EventType>
       <ServiceType>/account</ServiceType>
   </ItemTagElement>
   ```

6. Save the file.
7. Load the configuration item tags by using the **load_config_item_tags** utility.

   ```
   load_config_item_tags config_item_tags.xml
   ```


Copyright © 2017, 2026, Oracle and/or its affiliates.

13

Setting Up Your System to Generate Invoice Documents

Learn how to configure your Oracle Communications Billing and Revenue Management (BRM) system to generate invoices through Oracle Analytics Publisher.

Topics in this document:

- Tasks for Setting Up Invoice Generation

Tasks for Setting Up Invoice Generation

To set up your BRM system for generating invoices using Oracle Analytics Publisher:

1. Install Oracle Analytics Publisher. To install Oracle Analytics Publisher, follow the instructions in the Oracle Analytics Publisher documentation.
2. In BRM, specify the Oracle Analytics Publisher invoice report and layout template names. See "Creating an Invoicing Configuration Business Profile".
3. In BRM, specify the events that you want included in your invoices by editing the **events.file** file and then loading it into the BRM database with the **pin_load_invoice_events** utility. See "Including Payment, A/R, and Tax Details in Invoices".
4. Install the BRM-Oracle Analytics Publisher invoicing integration package on the Oracle Analytics Publisher server. In the BRM installer, this package is installed by selecting the Billing and Invoicing Utilities component, which installs the BRM Reports and BRM Invoices files required for Oracle Analytics Publisher invoice document generation. See "Installing the BRM-Oracle Analytics Publisher Invoicing Integration Package".
5. Configure the **pin_inv_doc_gen** utility. See "Configuring the pin_inv_doc_gen Utility".
6. Configure the data source and data model.
7. Configure for multischema. Store and modify copies of the report in each schema. See "Configuring the BRM-Oracle Analytics Publisher Invoicing Integration to Support Multischema Systems".

Creating an Invoicing Configuration Business Profile

When the BRM-Oracle Analytics Publisher integration is enabled, BRM automatically performs the following during the bill unit creation process:

- Reads the invoicing configuration business profile (**/config/business_profile** object) to determine which invoice report and layout template the bill unit should be associated with.
- Creates an **/associated_bus_profile** object for the bill unit. This object specifies the names of the Oracle Analytics Publisher invoice report and layout template to use.
- Links the bill unit to the **/associated_bus_profile** object it created.

To create an invoicing configuration business profile:

Copyright © 2017, 2026, Oracle and/or its affiliates.

1. Enable the BRM-Oracle Analytics Publisher integration. See "Enabling the BRM-Oracle Analytics Publisher Integration".
2. Specify the requirements for a bill unit to be associated with an invoicing configuration business profile by creating a validation template.
3. Specify the validation template, invoice report, and layout template to be associated with the bill unit.
4. Load the business profile information into the BRM database by using the **load_pin_business_profile** utility. See "Loading Invoicing Business Profiles into the Database".
5. If your system includes accounts that were created before you integrated BRM invoicing with Oracle Analytics Publisher, associate your existing **/billinfo** objects with **/associated_bus_profile** objects. See "Associating Pre-Existing Bill Units with Business Profiles".

Enabling the BRM-Oracle Analytics Publisher Integration

By default, BRM-Oracle Analytics Publisher Integration is disabled and invoices are stored in flist format. To use the BRM-Oracle Analytics Publisher integration, you must enable BRM-Oracle Analytics Publisher Integration and configure BRM to store invoices in XML format.

To enable the BRM-Oracle Analytics Publisher integration for invoices:

1. Go to *BRM_home***/sys/data/config**.
2. Create an XML file from the **/config/business_params** object:

   ```
   pin_bus_params -r BusParamsInvoicing bus_params_Invoicing.xml
   ```

3. Enable integration by setting **EnableInvoicingIntegration** to **enabled**:

   ```
   <EnableInvoicingIntegration>enabled</EnableInvoicingIntegration>
   ```

4. Store invoices in XML format by setting **InvoiceStorageType** to **1**:

   ```
   <InvoiceStorageType>1</InvoiceStorageType>
   ```

5. Save the file as **bus_params_Invoicing.xml**.
6. Load the XML file into the BRM database:

   ```
   pin_bus_params bus_params_Invoicing.xml
   ```

7. Stop and restart the CM.

For more information about **pin_bus_params**, see "pin_bus_params" in *BRM Developer's Guide*.

Loading Invoicing Business Profiles into the Database

After editing the **pin_business_profile.xml** file, use the **load_pin_business_profile** utility to load the contents of the file into **/config/business_profile** objects in the BRM database. For more information, see "Managing Business Profiles" in *BRM Managing Customers*.

Copyright © 2017, 2026, Oracle and/or its affiliates.

Associating Pre-Existing Bill Units with Business Profiles

All **/billinfo** objects created before your BRM-Oracle Analytics Publisher integration are not linked to an **/associated_business_profile** object, which prevents you from creating Oracle Analytics Publisher invoices for those **/billinfo** objects.

To generate Oracle Analytics Publisher invoice documents for **/billinfo** objects created before the BRM-Oracle Analytics Publisher integration, you must run the **pin_upd_assoc_bus_profile** utility.

The **pin_upd_assoc_bus_profile** utility is a standalone multithreaded application (MTA) that performs the following operations:

- Searches for **/billinfo** objects in the BRM database that have an empty PIN_FLD_ASSOC_BUS_PROFILE_OBJ_LIST field.
- Creates an **/associated_bus_profile** object for each **/billinfo** object with an empty field.
- Links the **/billinfo** object to its **/associated_bus_profile** object by populating the **/billinfo** object's PIN_FLD_ASSOC_BUS_PROFILE_OBJ_LIST field.

To associate pre-existing bill units with business profiles:

1. Make sure that the BRM-Oracle Analytics Publisher integration is enabled. See "Enabling the BRM-Oracle Analytics Publisher Integration".

   > **Note**
   >
   > The utility fails if you attempt to run it when the BRM-Oracle Analytics Publisher integration is disabled.

2. (Optional) To associate only a subset of your pre-existing bill units with business profiles, create a text file that specifies the **/billinfo** objects to update. The file must use the following format:

   ```
   0 PIN_FLD_RESULTS     ARRAY [0] allocated 20, used 1
   1    PIN_FLD_POID      POID [0] 0.0.0.1 /billinfo 37395 1
   0 PIN_FLD_RESULTS     ARRAY [1] allocated 20, used 1
   1    PIN_FLD_POID      POID [0] 0.0.0.1 /billinfo 37488 3
   ```

3. Go to the *BRM_home*/**apps/pin_billd** directory.
4. Enter the following command:

   pin_upd_assoc_bus_profile [**-file** *filename*]

   where *filename* specifies the name and location of the file that lists the **/billinfo** objects to update.

   > **Note**
   >
   > **pin_upd_assoc_bus_profile** requires a configuration file in the directory from which you run the utility. The configuration file requires the standard CM connection and MTA entries. See "Configuring Your Multithreaded Application" in *BRM Developer's Guide*.

Copyright © 2017, 2026, Oracle and/or its affiliates.

If you run the utility without any parameters, the utility searches all **/billinfo** objects in the BRM database. If you run the utility with the **-file** parameter, the utility searches only those **/billinfo** objects specified in the file.

Viewing Invoices Individually on Oracle Analytics Publisher Console

In the BRM_Invoices reports directory (*OAP_home***/user_projects/domains/bifoundation_domain/config/bipublisher/repository/Reports/BRM_Invoices/0.0.0***n*), the **ViewInvoiceReport** report is generated to view individually each invoice on the Oracle Analytics Publisher console.

Configure **ViewInvoiceReport** to point to the data source by editing the data model, configuring the data source, saving changes, and restarting Oracle Analytics Publisher. Enter the invoice POID for the input parameter and click **Submit** to see the report.

View the invoice on the console, by choosing the applicable template, such as Consumer Invoice, Corporate Invoice, Consumer Correction Invoice, Corporate Correction Invoice, Consumer Replacement Invoice, or Corporate Replacement Invoice.

Enabling the Display of Customer Details on the Web

By default, displaying contact details, account and bill numbers when using the Web is disabled in Oracle Analytics Publisher.

To enable the display of customer details on the Web:

1. Go to the *OAP_home***/user_projects/domains/bifoundation_domain/config/bipublisher/repository/Reports/BRM_Invoices/schema_number/schema_number.xdm** directory.
2. Open the **_datamodel.xdm** file in a text editor.
3. Search for the following entry:

   ```
   <property name="include_rowsettag" value="false"/>
   ```

4. Change **false** to **true**.
5. Save and close the file.

Installing the BRM-Oracle Analytics Publisher Invoicing Integration Package

The BRM-Oracle Analytics Publisher invoicing integration package is installed through the BRM installer as part of the Billing and Invoicing Utilities component. It includes the BRM invoice reports, data model, SQL bursting query, and sample Oracle Analytics Publisher layout templates required to generate invoice documents for consumer and corporate accounts.

To install the BRM-Oracle Analytics Publisher invoicing integration package:

- Ensure that the preinstallation requirements are satisfied. See "Preinstallation Requirements".
- Manually remove previously installed versions of the BRM-Oracle Analytics Publisher invoicing integration layout template files. See "Removing Older BRM-Oracle Analytics Publisher Invoicing Integration Layout Template Files".
- Run the BRM-Oracle Analytics Publisher invoicing integration package. See "Install Billing and Invoicing Utilities".

Copyright © 2017, 2026, Oracle and/or its affiliates.

> **Note**
>
> Install the BRM-Oracle Analytics Publisher invoicing integration package on the system on which BRM Reports is installed. If the BRM-Oracle Analytics Publisher invoicing integration package and BRM Reports are not installed on the same system, you cannot view an invoice in Oracle Analytics Publisher.
>
> If the Oracle Analytics Publisher invoice reports or layout templates are not available after a Complete BRM installation, verify that Oracle Analytics Publisher was installed and that the Oracle Analytics Publisher home environment variable was set before the Billing and Invoicing Utilities component was installed. If they were not, rerun the BRM installer and install the Billing and Invoicing Utilities component.

Preinstallation Requirements

The following must be completed before you run the BRM-Oracle Analytics Publisher invoicing integration package application:

- Oracle Analytics Publisher is installed.
- Oracle Analytics Publisher home environment variable has been successfully set.

> **Note**
>
> Install Oracle Analytics Publisher and set the Oracle Analytics Publisher home environment variable before installing the Billing and Invoicing Utilities component. If you install BRM before installing Oracle Analytics Publisher, or if the Oracle Analytics Publisher home environment variable is not set, the BRM Reports and BRM Invoices files required for Oracle Analytics Publisher invoice document generation are not installed. After installing Oracle Analytics Publisher and setting the home environment variable, rerun the BRM installer and install the Billing and Invoicing Utilities component.

For more information on these requirements, see "About BRM Reports" in *BRM Reports*.

Removing Older BRM-Oracle Analytics Publisher Invoicing Integration Layout Template Files

When you install the BRM-Oracle Analytics Publisher invoicing integration package, the older versions of BRM-Oracle Analytics Publisher invoicing integration layout template files are not automatically overwritten. Before installing the BRM-Oracle Analytics Publisher invoicing integration package, you must manually remove all previous versions of BRM-Oracle Analytics Publisher invoicing integration layout template files.

To remove older BRM-Oracle Analytics Publisher invoicing integration layout template files:

1. Go to *OAP_home*/**user_projects/domains/OAP_domain/config/bipublisher/repository/Reports/BRM_Invoices/***SchemaNumber*/**BRM_Bursting_Invoice_Report.xdo/**.

   where *SchemaNumber* is the directory for the BRM schema number to process BRM invoices; for example, 0.0.0.1.

2. Manually remove the following layout template files:

Copyright © 2017, 2026, Oracle and/or its affiliates.

- BRM_Consumer_Correction_Invoice.rtf*
- BRM_Consumer_Invoice.rtf*
- BRM_Consumer_Replacement_Invoice.rtf*
- BRM_Corporate_Correction_Invoice.rtf*
- BRM_Corporate_Invoice.rtf*
- BRM_Corporate_Replacement_Invoice.rtf*

3. Go to *OAP_home*/**user_projects/domains/bifoundation_domain/config/bipublisher/repository/Reports/BRM_Invoices/***SchemaNumber*/**ViewInvoiceReport.xdo/**.
4. Manually remove the following layout template files:
   - BRM_Consumer_Correction_Invoice_en_US.rtf*
   - BRM_Consumer_Invoice_en_US.rtf*
   - BRM_Consumer_Replacement_Invoice_en_US.rtf*
   - BRM_Corporate_Correction_Invoice_en_US.rtf*
   - BRM_Corporate_Invoice_en_US.rtf*
   - BRM_Corporate_Replacement_Invoice_en_US.rtf*

Install Billing and Invoicing Utilities

To install the BRM-Oracle Analytics Publisher invoicing integration package, install the Billing and Invoicing Utilities component. This component installs the BRM Reports and BRM Invoices files used by Oracle Analytics Publisher invoice document generation.

To install Billing and Invoicing Utilities in GUI mode, see "Installing Individual BRM Components" in *BRM Installation Guide*.

For silent installation, create a silent installation response file that includes the Billing and Invoicing Utilities component. Use the BRM Installation Guide instructions for silent installation to generate and run the response file.

Installed Template Files

The BRM-Oracle Analytics Publisher invoicing integration program copies the templates listed in Table 13-1 to your computer's *OAP_home*/**user_projects**/**domains**/**bifoundation_domain**/**config**/**bipublisher**/**repository**/**Reports**/**BRM_Invoices**/*SchemaNumber* directory.

Table 13-1 Oracle Analytics Publisher-Related Files Placed on Your Computer

Templates	Description
BRM_Consumer_Correction_Invoice.rtf	Layout template for consumer corrective invoice letter documents.
BRM_Consumer_Invoice.rtf	Layout template for consumer invoice documents.
BRM_Consumer_Replacement_Invoice.rtf	Layout template for consumer replacement invoice documents.
BRM_Corporate_Correction_Invoice.rtf	Layout template for corporate corrective invoice letter documents.
BRM_Corporate_Invoice.rtf	Layout template for corporate invoices.
BRM_Corporate_Replacement_Invoice.rtf	Layout template for corporate replacement invoice documents.

Copyright © 2017, 2026, Oracle and/or its affiliates.

Configuring the pin_inv_doc_gen Utility

The **pin_inv_doc_gen** utility creates, delivers, and stores your invoice documents in the Oracle Analytics Publisher Scheduler database.

Configuring **pin_inv_doc_gen** involves:

- Configuring the Infranet.properties File for pin_inv_doc_gen
- Configuring the docgen.sh Batch Script
- Changing Command-Line Parameters for pin_inv_doc_gen
- Setting Scheduler Configuration
- Configuring the Email Server
- Configuring HTTPS for Secure Communication
- Creating the XMLP_INVOICE_JOB Table in the Scheduler Database

Configuring the Infranet.properties File for pin_inv_doc_gen

The **Infranet.properties** file stores your network configuration entries for the **pin_inv_doc_gen** utility.

To configure the **Infranet.properties** file for Oracle Analytics Publisher:

1. Go to *BRM_home*/**apps/pin_inv_doc_gen**.
2. Save the **Infranet.properties.sample** file as **Infranet.properties**.

   The **Infranet.properties.sample** file for the **pin_inv_doc_gen** utility includes the standard configuration entries. See "Using Configuration Files to Connect and Configure Components" in *BRM System Administrator's Guide*.

   > **Note**
   >
   > You must have read-write permissions to access the **/invoice** object. Your invoice data is stored in the **/invoice** object. The BRM system administrator creates this user and grants the required permissions.

3. Open the **Infranet.properties** file in a text editor, and add your network entries and their values, as listed in Table 13-2.

Table 13-2 Infranet.properties.sample File Entries

Entry	Description
infranet.login.type	Specifies the type of login. A type 1 login requires the application to provide a user name and password. A type 0 login is a trusted login that comes through a Connection Manager (CM) proxy. It does not require a user name or password. Use **1** or **0**.
infranet.log.level	Specifies the level of logging for the **pin_inv_doc_gen** utility.
infranet.log.file	Specifies the log file name for the **pin_inv_doc_gen** utility.

Copyright © 2017, 2026, Oracle and/or its affiliates.

Table 13-2 (Cont.) Infranet.properties.sample File Entries

Entry	Description
infranet.bip.url	Specifies the URL of Oracle Analytics Publisher in the following format: *http://BIP_Hostname:Port/xmlpserver* **Note**: Enable HTTPS and the SSL port for secure communication. See "Configuring HTTPS for Secure Communication".
infranet.bip.username	Specifies the name of the Oracle Analytics Publisher server.
infranet.bip.password	Specifies the password. **Note:** The password can be encrypted with Advanced Encryption Standard (AES) for security.
infranet.threadpool.size	Specifies the number of threads. The default value is **3**.
infranet.threadpool.maxsize	Specifies the maximum number of threads. The default value is **5**.
infranet.threadpool.fetchsize	Specifies the number of records that can be fetched by the thread. The default value is **5**.
infranet.burst.threadpool.size	Specifies the number of core threads in the pool.
infranet.burst.threadpool.maxsize	Specifies the maximum number of concurrent threads in the pool.
infranet.burst.threadpool.chunksize	Specifies the maximum number of BRM invoices per Oracle Analytics Publishing bursting query.
infranet.polling.sleeptime	Specifies the sleep time (in milliseconds) between polling calls while waiting for the Oracle Analytics Publisher schedule to complete. The default value is **5000** (5 seconds).
infranet.dmsearchargs.size	Specifies the number of account and bill unit pairs the utility should process at one time. The maximum value is **14**.
infranet.dupinvdir.path	Specifies the directory path for storing duplicate invoice documents.
infranet.dupinvoice.watermark.text	Specifies the text to display as the background watermark in the duplicate invoice. The default value is **DUPLICATE**.
infranet.dupinvoice.watermark.coordx	Specifies the x-coordinate of the start of the watermark text. The default value is **200f** (starting coordinate lower left x).
infranet.dupinvoice.watermark.coordy	Specifies the y-coordinate of the start of the watermark text. The default value is **250f** (starting coordinate lower left y).
infranet.dupinvoice.watermark.colorr	Specifies the red component of the watermark text color. The default value is **0.7f**.
infranet.dupinvoice.watermark.colorg	Specifies the green component of the watermark text color. The default value is **0.7f**.
infranet.dupinvoice.watermark.colorb	Specifies the blue component of the watermark text color. The default value is **0.7f**.
infranet.dupinvoice.watermark.font	Specifies the font of the watermark text. The default value is **Arial**.
infranet.dupinvoice.watermark.fontsize	Specifies the size of the watermark text. The default value is **48**.
infranet.dupinvoice.watermark.angle	Specifies the angle of the watermark text. The default value is **45** (angle of text is 45 degrees). See the Oracle Analytics Publisher Java API documentation.

Copyright © 2017, 2026, Oracle and/or its affiliates.

Table 13-2 (Cont.) Infranet.properties.sample File Entries

Entry	Description
infranet.schedulerdb.url	Specifies the scheduler database URL in the following format: `jdbc:oracle:thin:@protocol://hostname:port/service` where *protocol* is either **tcp** or **tcps**, *hostname* is the hostname of the scheduler database, *port* is the port number for the scheduler database, and *service* is the service name of the scheduler database.
infranet.schedulerdb.user	Specifies the user name for the scheduler database.
infranet.schedulerdb.credentials	Specifies the security credentials for connecting to the scheduler database.
infranet.jdbcpool.size	Specifies the initial number of connections maintained in the pool. The default is set to the same as **burst.threadpool.size**.
infranet.jdbcpool.maxsize	Specifies the maximum number of connections that can be created. The default is set to the same as **burst.threadpool.maxsize**.
infranet.SECURITY_CREDENTIALS	Specifies the KeyStore security credentials.
infranet.saveXMLData	Specifies whether to save a copy of the XML data to the Oracle Analytics Publisher database. The default is **false**.
infranet.http.notification.server	Specifies the name of the HTTP server used for notifications. **Important:** Enable HTTP notifications in Oracle Analytics Publisher. See the Oracle Analytics Publisher documentation for details.
infranet.http.notification.userName	Specifies the HTTP server login name.
infranet.http.notification.password	Specifies the HTTP server login password.
infranet.notify.http.when.success	Specifies whether to receive HTTP notification when an Oracle Analytics Publisher job is successful. By default, HTTP notification is enabled and set to **true**.
infranet.notify.http.when.warning	Specifies whether to receive HTTP notification when an Oracle Analytics Publisher job contains warnings. By default, HTTP notification is enabled and set to **true**.
infranet.report.parameters	Specifies the custom criteria that an account must meet in a set of key-value pairs. Enter key-value pairs in the following format: `name1:value1;name2:value2;...nameN:valueN` where: • *nameN* is the name of the custom parameter you added in the report's data model prepended with **pin_Inv_**. For example: **p_Inv_Status**, **p_Inv_ReportName**, or **p_Inv_EffectiveT**. • *valueN* is the required value of the specified field. These values are overridden by the **pin_inv_doc_gen -params** option. See "Generating Invoice Documents Based on Custom Criteria".
infranet.report.parameter.separator	Specifies the character for separating the key-value pairs in **infranet.report.parameters**. The default is a colon (**:**).
infranet.report.parameter.delimiter	Specifies the character for delimiting a list of key-value pairs in **infranet.report.parameters**. The default is a semicolon (**;**).

Copyright © 2017, 2026, Oracle and/or its affiliates.

Table 13-2 (Cont.) Infranet.properties.sample File Entries

Entry	Description
infranet.report.parameter.procedure.name	Specifies the name of the custom procedure to run, in this format: *pkgName*.*procedureName*. where: • *pkgName* is the name of the package. • *procedureName* is the name of the custom procedure you create. See "Selecting Invoices Based On Custom Parameters" for more information.
infranet.pcp.ssl.enabled	Specifies whether to enable SSL between BRM and the Oracle Analytics Publisher. The default is **true**.
infranet.pcp.ssl.wallet.location	Specifies the path to the client wallet file. The default is *BRM_home*/**wallet/client**.
infranet.pcp.ssl.wallet.filename	Specifies the name of the client wallet file. The default is **cwallet.sso**.

4. Save and close the file.

Configuring the docgen.sh Batch Script

The **docgen.sh** batch script is used for setting environment variables and launching the **pin_inv_doc_gen** utility.

To configure the **docgen.sh** batch script:

1. Go to the *BRM_home*/**apps/pin_inv_doc_gen** directory.
2. Open the **docgen.sh** batch script in a text editor.
3. Edit the default values available in the file to the values specific to your system environment.

> **Note**
>
> The **pin_inv_doc_gen** utility runs only in the 32-bit mode. Set the Java executable in the **docgen.sh** file to the 32-bit Java version.

4. Save and close the file.

Changing Command-Line Parameters for pin_inv_doc_gen

To change the command-line parameters for **pin_inv_doc_gen** utility, you update the **docgen.sh** batch script:

1. Go to the *BRM_home*/**apps/pin_inv_doc_gen** directory.
2. Open the **docgen.sh** batch script in a text editor.
3. Search for the following line:

```
invoicedocgen.pin_inv_doc_gen -status pending
```

4. Add or replace the **pin_inv_doc_gen** command-line parameter with the required parameter. For syntax and parameter usage, see "pin_inv_doc_gen".
5. Save and close the file.

Copyright © 2017, 2026, Oracle and/or its affiliates.

6. In the command prompt, navigate to the *BRM_home*/**apps/pin_inv_doc_gen** directory.
7. Enter the following command which runs the **docgen.sh** batch script:

   ```
   docgen.sh
   ```

   The **pin_inv_doc_gen** utility runs with the updated command-line parameter.

Setting Scheduler Configuration

To set scheduler configuration, see the discussion of setting scheduler configuration options in the Oracle Analytics Publisher documentation.

Configuring the Email Server

To configure the email server in Oracle Analytics Publisher, see the discussion of setting up delivery options in the Oracle Analytics Publisher documentation.

Configuring HTTPS for Secure Communication

Oracle Analytics Publisher can be configured to support HTTPS for secure communication. In addition to setting up HTTPS in Oracle Analytics Publisher, you also configure the **Infranet.properties** file and **pin_inv_doc_gen** utility.

To configure HTTPS for secure communication:

1. Enable HTTPS in Oracle Analytics Publisher. For more information, see the Oracle Analytics Publisher documentation.
2. Open the **Infranet.properties** file in a text editor.
3. Add the following line:

   ```
   infranet.bip.url=https://BIP_Hostname:SSL_port/xmlpserver
   ```

   where:

   - *BIP_Hostname* is the host name on which the Oracle Analytics Publisher server is installed.
   - *SSL_port* is your secure sockets layer port number.
4. Save and close the file.
5. In the command prompt, navigate to the *BRM_home*/**apps/pin_inv_doc_gen** directory.
6. Enter the following command which runs the **docgen.sh** batch script:

   ```
   docgen.sh
   ```

   The **pin_inv_doc_gen** utility runs with the updated URLs.
7. Enter **yes** for:

   ```
   Is it a trusted certificate?
   ```

Creating the XMLP_INVOICE_JOB Table in the Scheduler Database

The XMLP_INVOICE_JOB table stores the mapping of the job ID to the Invoice_POID for which an invoice document is generated.

To create the XMLP_INVOICE_JOB table in the Scheduler database:

Copyright © 2017, 2026, Oracle and/or its affiliates.

1. Go to *BRM_home*/**apps/pin_inv_doc_gen**.
2. Open the **Create_Xmlp_Invoice_Job.sql** file.
3. Search for **<tablespacename>**.
4. Replace **<tablespacename>** with the tablespace name where all the Scheduler tables in the Oracle Analytics Publisher Scheduler database are created.
5. Save and close the file.
6. In a terminal, enter the following command to open SQL*Plus:

   ```
   sqlplus user_name@Database_Name
   Enter password: password
   ```

   where:

   - *user_name* is the user name for the Scheduler database.
   - *password* is the password for the Scheduler database.
   - *Database_Name* is the SID of the Scheduler database.
7. At the SQL prompt, enter the following command:

   ```
   SQL> @Create_Xmlp_Invoice_Job.sql
   ```

8. Configure the Java Naming and Directory Interface (JNDI) for invoiceXREF to allow the **pin_inv_doc_gen** utility to connect to the Scheduler database:
   a. Log on to the Oracle WebLogic Console.
   b. Click **Services**.
   c. Click **Data Sources**.
   d. Click the **Lock and Edit** button.
   e. Create a new data source.
   f. Enter JNDI details:

      Name=invoiceXREF

      JNDI name= jdbc/OracleBRM/invoiceXREF
   g. Click **Next**.
   h. Enter the Scheduler database details that you used when you installed the XMLP_INVOICE_JOB table.
   i. Click **Next**.
   j. Select the server on which the Oracle Analytics Publisher software is installed.
   k. Click **Finish**.
   l. Click **Activate Changes**.
   m. Stop and restart the Oracle Analytics Publisher server.

Copyright © 2017, 2026, Oracle and/or its affiliates.

Configuring the BRM-Oracle Analytics Publisher Invoicing Integration to Support Multischema Systems

To configure the BRM-Oracle Analytics Publisher invoicing integration to support multischema systems, do the following:

- Configure the bursting data source and query by loading stored procedures. See "Loading Stored Procedures".
- Create new JDBC data sources. See "Creating New JDBC Data Sources".

Loading Stored Procedures

To load stored procedures, do the following:

1. Go to *BRM_home*/**apps/pin_inv_doc_gen**
2. Open SQL*Plus at the command line:

   ```
   sqlplus user_name@Database_Name
   Enter password: password
   ```

3. Enter the following command:

   ```
   SQL>@invoice_bursting.plb
   ```

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

Creating New JDBC Data Sources

The BRM-Oracle Analytics Publisher invoicing integration packages invoicing reports and the data model for BRM primary schema. To create new JDBC sources to support BRM multischema, you follow these procedures. For more information, refer to the Oracle Analytics Publisher documentation.

Creating Data Source for Each BRM Schema

To create a data source for each BRM schema:

1. In a Web browser, enter the following URL:

   ```
   http://OAP_host:port/xmlpserver/
   ```

   where:

   - *OAP_host* is the host name on which the Oracle Analytics Publisher server is installed.
   - *port* is the console port. The default port is **9704**.

     The Oracle Analytics Publisher Enterprise client opens.
2. On the global header, click **Administration**.
3. Under Data Sources, click **JDBC Connection**.
4. On the JDBC page, click the **Add Data Source** button.
5. Enter the following database connection details for the BRM schema:
   - In the **Data Source Name** field, enter a name, for example, **BRM_DATA_SOURCE**.

Copyright © 2017, 2026, Oracle and/or its affiliates.

- In the **Driver Type** field, enter **Oracle 12c**.
- In the **Database Driver Class** field, enter **oracle.jdbc.OracleDriver**.
- In the **Connection String** field, enter your database connection details. For example:

  jdbc:oracle:thin:@*Hostname*:port:SID

  where *Hostname* is the host name for the database that contains the secondary schema.
- In the **Username** field, enter the BRM schema user name.
- In the **Password** field, enter the password for the BRM schema user.

6. Click **Test Connection**, which verifies the connection to BRM schema.

   A confirmation message is displayed.
7. Click **Apply**, which saves your changes.
8. Repeat the previous steps for each schema.

Setting Up the Data Model Configuration for Multischema

To configure invoices for multischema:

1. In the **BRM_Invoices** reports directory, copy the **Reports** directory from **0.0.0.1** to **0.0.0.*n***.
2. Rename the data model directory within **0.0.0**.*n*. For example, to set up the data model configuration for **0.0.0.2**, do the following:

   ```
   cd OAP_home/user_projects/domains/OAP_domain/config/bipublisher/repository/Reports/
   BRM_Invoices
   cp -r 0.0.0.1 0.0.0.2
   cd 0.0.0.2
   mv 0.0.0.1.xdm 0.0.0.2.xdm
   ```
3. Edit the data source name specified in "Creating Data Source for Each BRM Schema".

   For example, in **0.0.0.2/0.0.0.2.xdm/_datamodel.xdm**, enter the data source name in **defaultDataSourceRef** and the BRM schema number in **CDATA**.
4. Set the URL of the BRM bursting report data model to the secondary schema by modifying **dataModel URL** in **0.0.0.2/BRM_Bursting_Invoice_Report.xdo/_report.xdo**.
5. To configure **pin_inv_doc_gen** for multischema changes, edit the **docgen.sh** batch script. For example, add **-schema 0.0.0.*n*** after the **-status pending** flag of each schema.

> **Note**
>
> The **-schema 0.0.0.*n*** parameter is optional. If it is not specified, schema information comes from the **infranet.connection** value in the **Infranet.properties** configuration file. The default is **0.0.0.1.**

Copyright © 2017, 2026, Oracle and/or its affiliates.

14

Generating Invoices in Oracle Analytics Publisher

Learn how to generate and customize Oracle Communications Billing and Revenue Management (BRM) invoice documents with Oracle Analytics Publisher.

Topics in this document:

- Generating Oracle Analytics Publisher Invoice Documents
- Generating Invoices for Hierarchical Bill Units with Oracle Analytics Publisher
- Storing the Invoice Document
- Sending Invoice Documents to Your Customers
- Customizing Invoices

Generating Oracle Analytics Publisher Invoice Documents

You use the **pin_inv_doc_gen** utility to generate invoice documents by integrating the capabilities of BRM and Oracle Analytics Publisher. This utility takes input parameters, generates the invoice document, and stores the instance of the invoice document generated in the Scheduler database. Based on the input parameters you pass to **pin_inv_doc_gen**, the utility fetches the corresponding invoice POID, Oracle Analytics Publisher report name, and layout template name of an **/invoice** object.

To configure the **pin_inv_doc_gen** utility with the **Infranet.properties** file, see "Configuring the Infranet.properties File for pin_inv_doc_gen".

To run the **pin_inv_doc_gen** utility:

1. Configure the **docgen.sh** batch script to run the appropriate **pin_inv_doc_gen** command. To view the utility's syntax and parameters, see "pin_inv_doc_gen".
2. Go to the *BRM_home***/apps/pin_inv_doc_gen** directory.
3. Enter the following command which runs the **docgen.sh** batch script:

   ```
   docgen.sh
   ```

Generating Invoice Documents for a List of Accounts

You can generate Oracle Analytics Publisher invoice documents for a list of accounts by using the **pin_inv_doc_gen** utility **-accts_list InvoiceList.xml** parameter. The number of data units (accounts and bill units) in the **InvoiceList.xml** file should be within the permissible range for the Data Manager (DM) to search the **/invoice** objects from the BRM database.

You can specify the number of account and bill unit pairs the **pin_inv_doc_gen** utility should process at one time by using the **infranet.dmsearchargs.size** parameter in the **Infranet.properties** file. The maximum value is 14. For example, if the **InvoiceList.xml** file lists 20 account and bill unit pairs and the **infranet.dmsearchargs.size value** is set to 6, the **pin_inv_doc_gen** utility processes six account and bill unit pairs at a time.

Copyright © 2017, 2026, Oracle and/or its affiliates.

In the **InvoiceList.xml** file, if you specify the account and bill unit pair of a corporate account, the invoice document is generated only for the specified account and bill unit. If the specified corporate account has child accounts that own only nonpaying bill units, the **/invoice** objects of those child accounts are not selected for generating the invoice document.

To generate invoice documents for a list of accounts and their bill units:

1. Create an XML file that lists all of the accounts and bill units for which to create Oracle Analytics Publisher invoices. The XML file must be in the following format:

   > **Note**
   >
   > You can use the *BRM_home*/**apps/pin_inv_doc_gen/InvoiceList.xml** file as an example.

   ```
   <InvoiceDocGenConfig>
      <InvoicingList>
         <Account>101304</Account>
         <Billinfo>402678</Billinfo>
      </InvoicingList>
      <InvoicingList>
         <Account>405304</Account>
         <Billinfo>522678</Billinfo>
      </InvoicingList>
   <InvoiceDocGenConfig>
   ```

2. Save and close the file.
3. Go to the *BRM_home*/**apps/pin_inv_doc_gen** directory.
4. Open the **docgen.sh** batch script in a text editor.
5. Search for the following line:

   ```
   invoicedocgen.pin_inv_doc_gen -status pending
   ```

6. Replace with:

   ```
   invoicedocgen.pin_inv_doc_gen -accts_list InvoiceList
   ```

   where *InvoiceList* specifies the name and location of the file that includes a list of accounts and bill units for which to create invoice documents. This is the file you created in step 1.
7. Save and close the file.
8. Go to the *BRM_home*/**apps/pin_inv_doc_gen** directory.
9. Enter the following command which runs the **docgen.sh** batch script:

   ```
   docgen.sh
   ```

Generating Invoice Documents Based on Custom Criteria

You can generate invoice documents for invoices that meet your custom criteria, such as only for Canadian customers with an invoice delivery type of email. You specify the criteria using a list of key-value pairs, which correspond to parameters you add to the data model defining the requirements for selecting invoices.

Copyright © 2017, 2026, Oracle and/or its affiliates.

> **Note**
>
> With custom criteria, the selected invoices are split into batches, and one job is scheduled for each batch using the bursting mode.

Selecting Invoices Based On Custom Parameters

To configure BRM and Oracle Analytics Publisher to select invoices based on your custom parameters:

1. Do the following in Oracle Analytics Publisher:
   a. Add your custom parameters to the report's data model.
   b. Customize the data set query to use your custom parameters.
   c. Optionally, customize the SQL bursting data query to use your custom parameters.
2. Configure BRM to call your custom procedure rather than the default **inv.get_invoice_chunk()** procedure. Your custom procedure must split the selected invoices into batches so that each batch is scheduled as a job.

   To do so, set this parameter in the *BRM_home***/apps/pin_inv_doc_gen/infranet.properties** file:

   infranet.report.parameter.procedure.name=*pkgName.procedureName*

   where:

   - *pkgName* is the name of the package.
   - *procedureName* is the name of the custom procedure you create.
3. Create a custom procedure that parses the key-value strings and uses them in queries to generate chunks from the selected invoices.

   Use the following procedure interface, which selects invoices based on the IN parameters and returns the details specified in the OUT parameter:

```
PROCEDURE custom_get_invoice_chunk(
              batch_size IN NUMBER,
              start_t IN NUMBER,
              end_t IN NUMBER,
              invoice_type IN INTEGER,
              params IN VARCHAR2,
              row_count  OUT NUMBER,
              ID_RECORD IN OUT InvTypesTest.InvoiceIdCursor
              );
```

   The following describes the IN parameters:

   - **batch_size**: The number of invoices to put in each batch or chunk.
   - **start_t**: The start date to use for selecting **/invoice** objects.
   - **end_t**: The end date to use for selecting **/invoice** objects.
   - **invoice_type**: The type of invoices to select for document generation: **0** means consumer invoices only, **1** means corporate invoices only, and **2** means all invoices.

Copyright © 2017, 2026, Oracle and/or its affiliates.

- **params**: A list of additional key-value pairs passed at the command line or to the **docgen.sh** script.

The following describes the OUT parameters:

- **row_count**: The total number of **/invoice** objects that were selected for document generation.
- **ID_RECORD**: A reference cursor of type **InvoiceId**, which should be defined as follows. Each cursor row holds information about each batch.

  InvoiceId is defined as follows in the **INVTYPES** package:

```
TYPE InvoiceId IS RECORD(
              first invoice_t.poid_id0%TYPE,
              last invoice_t.poid_id0%TYPE,
              count NUMBER);
TYPE InvoiceIdCursor IS ref cursor RETURN InvoiceId;
```

4. Add your custom stored procedure to the **/config/stored_procedure** object. BRM only runs the stored procedures listed in **/config/stored_procedure**.
5. Pass your custom parameters and their required values as key-value pairs to BRM in one of the following ways:
   - Using the **infranet.report.parameters** entry in the **pin_inv_doc_gen** utility's **Infranet.properties** file. See "Configuring the Infranet.properties File for pin_inv_doc_gen".
   - Using a **pin_inv_doc_gen** command. Custom criteria provided through a utility command override any custom criteria in the **Infranet.properties** file. See "Configuring Custom Criteria at the Command Line".

The following shows a sample procedure:

```
PROCEDURE custom_get_invoice_chunk(
                batch_size IN NUMBER,
                start_t IN NUMBER,
                end_t IN NUMBER,
                invoice_type IN INTEGER,
                params IN VARCHAR2,
                row_count  OUT NUMBER,
                ID_RECORD IN OUT InvTypes.InvoiceIdCursor
                ) AS

          row_cursor      InvTypes.InvoiceRowCursor;
          poid_id         NUMBER;
          invoice_id      NUMBER;
          chunk_count     NUMBER := 0;
          chunk_min       NUMBER;
          chunk_max       NUMBER;
          stmt VARCHAR2(2000);
BEGIN
          row_count := 0;

          -- Open cursor to get all invoice IDS, depending on the input
parameters
          CASE invoice_type
          WHEN 2 THEN
                  -- ALL invoice types
```


Copyright © 2017, 2026, Oracle and/or its affiliates.

```
                    stmt := 'SELECT invoice_t.poid_id0 FROM invoice_t,
billinfo_t ' ||
                            ' WHERE invoice_t.status = 0 '
||
                            ' and invoice_t.created_t >= ' || start_t
||
                            ' and invoice_t.created_t < ' || end_t
||
                            ' and invoice_t.billinfo_obj_id0 =
billinfo_t.poid_id0 ' ||
                            ' and ( (billinfo_t.parent_flags > 0 and
billinfo_t.ar_billinfo_obj_id0 = billinfo_t.poid_id0 ) '
||
                           -- parse params and add criteria for additional
parameters
                            ' or  invoice_t.check_split_flags = 0) '
||
                            ' ORDER BY invoice_t.poid_id0';

          WHEN 0 THEN
                    -- CONSUMER invoice type only
                    stmt := 'SELECT invoice_t.poid_id0 FROM invoice_t,
billinfo_t ' ||
                            ' WHERE invoice_t.status = 0 '
||
                            ' and invoice_t.created_t >= ' || start_t
||
                            ' and invoice_t.created_t < ' || end_t
||
                            ' and invoice_t.billinfo_obj_id0 =
billinfo_t.poid_id0 ' ||
                            ' and billinfo_t.parent_flags = 0 '
||
                            ' and invoice_t.check_split_flags = 0 '
||
                           --parse params and add criteria for additional
parameters
                            ' ORDER BY invoice_t.poid_id0';

          WHEN 1 THEN
                    -- CORPORATE invoice type only
                    stmt := 'SELECT invoice_t.poid_id0 FROM invoice_t,
billinfo_t ' ||
                            ' WHERE invoice_t.status = 0 '
||
                            ' and invoice_t.created_t >= ' || start_t
||
                            ' and invoice_t.created_t < ' || end_t
||
                            ' and invoice_t.billinfo_obj_id0 =
billinfo_t.poid_id0 ' ||
                            ' and billinfo_t.parent_flags > 0 '
||
                            ' and billinfo_t.ar_billinfo_obj_id0 =
billinfo_t.poid_id0 ' ||
                           --parse params and add criteria for additional
```


Copyright © 2017, 2026, Oracle and/or its affiliates.

```
parameters
                          ' ORDER BY invoice_t.poid_id0';
          END CASE;

          OPEN row_cursor FOR stmt;

             LOOP
                -- fetch each row
                FETCH row_cursor INTO poid_id;
                EXIT WHEN row_cursor%NOTFOUND;

                row_count := row_count + 1;
                chunk_count := chunk_count + 1;
                chunk_max := poid_id;

                IF chunk_count = 1 THEN
                        chunk_min := poid_id;
                END IF;
                IF chunk_count = batch_size THEN
                        -- Insert the batch into temp
table
                        INSERT INTO invoice_batches ( first, last, count)
VALUES (chunk_min, chunk_max, chunk_count);
                        -- Start the next chunk
                        chunk_count := 0;
                END IF;

             END LOOP;

          CLOSE row_cursor;

          -- Insert the leftovers
          IF chunk_count > 0 THEN
                  INSERT INTO invoice_batches ( first, last, count) VALUES
(chunk_min, chunk_max, chunk_count);
          END IF;

          OPEN ID_RECORD for
          select * from invoice_batches
          ORDER BY first;

EXCEPTION
          WHEN OTHERS THEN
              row_count := 0;
```

> **Note**
>
> By default, the invoices selected in each chunk returned by **custom_get_invoice_chunk()** must match the invoices selected by the data set and burst query. Oracle Analytics Publisher throws an exception if the invoices selected by **custom_get_invoice_chunk()** and the data set query do not match.

Copyright © 2017, 2026, Oracle and/or its affiliates.

Configuring Custom Criteria at the Command Line

To configure custom criteria at the command line:

1. Go to the *BRM_home*/apps/pin_inv_doc_gen directory.
2. Open the **docgen.sh** batch script in a text editor.
3. Search for the following line:

   ```
   invoicedocgen.pin_inv_doc_gen -status pending
   ```

4. Replace the line with the following:

   ```
   invoicedocgen.pin_inv_doc_gen -status pending -params
   name1:value1;name2:value2;...nameN:valueN
   ```

   where:

   - *nameN* is the name of the custom parameter you added in the report's data model prepended with **pin_Inv_**. For example: **p_Inv_Status**, **p_Inv_ReportName**, or **p_Inv_EffectiveT**.
   - *valueN* is the required value of the specified field.

   For example, this command generates invoice documents whose billing segment and billing state are both 1. That is, the following fields in the **/billinfo** object are set to **1**: PIN_FLD_BILLING_SEGMENT and PIN_FLD_BILLING_STATE:

   ```
   invoicedocgen.pin_inv_doc_gen -status pending -params
   p_Inv_BillSegment:1;p_Inv_BillingState:1
   ```

5. Save and close the file.
6. In the command prompt, navigate to the *BRM_home*/apps/pin_inv_doc_gen directory.
7. Enter the following command, which runs the **docgen.sh** batch script:

   ```
   docgen.sh
   ```

Generating Invoice Documents Based on the Account Type

To generate invoices based on the account type:

1. Go to the *BRM_home*/apps/pin_inv_doc_gen directory.
2. Open the **docgen.sh** batch script in a text editor.
3. Search for the following line:

   ```
   invoicedocgen.pin_inv_doc_gen -status pending
   ```

4. Do one of the following:
   - To generate invoice document for consumer account types, replace with:

     ```
     invoicedocgen.pin_inv_doc_gen -type consumer
     ```

   - To generate invoice document for corporate account types, replace with:

     ```
     invoicedocgen.pin_inv_doc_gen -type corporate
     ```

5. Save and close the file.
6. In the command prompt, navigate to the *BRM_home*/apps/pin_inv_doc_gen directory.
7. Enter the following command which runs the **docgen.sh** batch script:

Copyright © 2017, 2026, Oracle and/or its affiliates.

```
docgen.sh
```

Generating Duplicate Invoice Documents

You can generate duplicate Oracle Analytics Publisher invoice documents for invoices that you have already generated by using the **pin_inv_doc_gen** utility with the **-status generated** parameter.

To generate duplicate Oracle Analytics Publisher invoice documents:

1. Go to the *BRM_home***/apps/pin_inv_doc_gen** directory.
2. Make sure the **infranet.dupinvdir.path** entry in the utility's **Infranet.properties** file specifies the directory in which to write the file. See "Configuring the Infranet.properties File for pin_inv_doc_gen".
3. Open the **docgen.sh** batch script in a text editor.
4. Search for the following line:

   ```
   invoicedocgen.pin_inv_doc_gen -status pending
   ```

5. Replace with:

   ```
   invoicedocgen.pin_inv_doc_gen -status generated
   ```

6. Save and close the file.
7. In the command prompt, navigate to the *BRM_home***/apps/pin_inv_doc_gen** directory.
8. Enter the following command to run the **docgen.sh** batch script:

   ```
   docgen.sh
   ```

> **Note**
>
> Oracle recommends **DocMerger** as the utility to generate a duplicate invoice with a DUPLICATE watermark. However, you can use this utility only for PDF documents. Hence, you can generate an invoice document with a DUPLICATE watermark only if the original invoice document was generated in PDF.

Generating Invoice Documents for Non-Invoice Accounts

Regardless of the payment type or delivery method, any invoice in pending status can be printed. Therefore, you can generate invoice documents for non-invoice accounts such as credit cards. Oracle Analytics Publisher generates PDFs for any invoice with a non-invoice payment type and stores them in the Oracle Analytics Publisher database.

To deliver invoice documents for non-invoice accounts, configure the delivery method of the bursting query by adding a delivery type (such as FTP) to the DEL_CHANNEL parameter and by adding the appropriate values to PARAMETER 1 through PARAMETER 8 as described in the *Oracle Fusion Middleware Report Designer's Guide for Oracle Analytics Publisher*.

Copyright © 2017, 2026, Oracle and/or its affiliates.

Generating Invoices for Hierarchical Bill Units with Oracle Analytics Publisher

Bill unit hierarchies enable customers to pay other customers' bills. For more information, see discussions about account and bill unit hierarchies in *BRM Managing Accounts Receivable* and *BRM Configuring and Running Billing*.

Generating trial and regular invoices with Oracle Analytics Publisher is suited for large bill unit hierarchies for the following reasons:

- Each bill unit is processed within its own transaction, which is later used in the processing cycle.
- Each bill unit is not loaded into RAM, which reduces RAM utilization.
- Collation is run after the generation of each bill unit within the hierarchy. Each bill unit is arranged in a predetermined sequence and by default is rendered in PDF.

> **Note**
>
> By default, if any bill unit in the hierarchy fails to generate, collation is halted, and an invoice is not produced.

Setting Up Invoicing for Hierarchical Bill Units

To set up invoicing for hierarchical bill units with Oracle Analytics Publisher:

1. Ensure that all the preliminary configurations required for billing are completed. See *BRM Configuring and Running Billing*.
2. Enable the Oracle Analytics Publisher integration. See "Enabling the BRM-Oracle Analytics Publisher Integration".
3. Set your system to generate invoice documents using Oracle Analytics Publisher. See "Tasks for Setting Up Invoice Generation".
4. Verify that BRM is set up with the required templates to generate hierarchical invoices. See "Invoice Template for Hierarchical Bill Units".
5. Associate bill units with an Oracle Analytics Publisher invoice and report.
6. Load the business profile information into the BRM database. See "Loading Invoicing Business Profiles into the Database".
7. Configure the **pin_inv_doc_gen** utility. See "Configuring the pin_inv_doc_gen Utility".
8. Verify that BRM and Oracle Analytics Publisher are set to generate the final invoice either with or without nonpaying child bill unit details. See "Setting Nonpaying Child Bill Unit Processing".
9. Verify that the *BRM_home*/**apps/pin_trial_bill/pin.conf** file's **Threshold** entry is commented out. Setting thresholds limits the number of nonpaying child bill units that are invoiced, which leads to errors when the parent account is trial billed.

Copyright © 2017, 2026, Oracle and/or its affiliates.

Invoice Template for Hierarchical Bill Units

Use the **BRM_Corporate_Invoice.rtf** template for generating hierarchical bill unit invoices. The **BRM_Corporate_Invoice.rtf** template includes the following layout and formatting:

- Depending on how the hierarchical bill unit tax is calculated, the template adds one of the following labels:
  - If the tax is calculated at the nonpaying child bill unit level, a **Subord Total (Tax Included)** label is added on the first page of the invoice.
  - If the tax is calculated at the paying parent bill unit level, a **Subord Total (Tax Not Included)** label is added on the first page of the invoice.
- To distinguish the trial invoice from the regular invoice, a **Trial Invoice** watermark is added to all the pages after the first page. You can change the watermark text by customizing the PCM_OP_INV_POL_PREP_INVOICE policy opcode. For more information, see "About Using Opcodes" in *BRM Opcode Guide*.

Setting Nonpaying Child Bill Unit Processing

> **Note**
>
> When generating invoices for hierarchies, BRM and Oracle Analytics Publisher must be set to process the final invoice either with or without nonpaying child bill units:
>
> - If you require an invoice that includes nonpaying details, set BRM to process nonpaying child bill units and Oracle Analytics Publisher to generate a single invoice from a group of invoices.
> - If you require an invoice that does not include nonpaying details, set BRM not to process nonpaying child bill units and Oracle Analytics Publisher to generate individual invoices.

In BRM and Oracle Analytics Publisher, you can choose whether to include the full nonpaying bill unit details in the parent bill unit's invoice.

- When you include the full nonpaying bill unit details, a separate invoice is generated for each nonpaying child bill unit and the totals rolled up into the paying parent's invoice. When the final invoice is generated, each nonpaying invoice is collated and merged into the final invoice.
- When you do not include the full nonpaying details for hierarchical bill units, only one parent invoice is created, which contains totals from the nonpaying bill units.

To set nonpaying bill unit processing for hierarchies:

1. In BRM, set the threshold for the maximum number of nonpaying child bill units allowed in the parent invoice with the **ThresholdSubordsSummary** and **ThresholdSubordsDetail** business parameters.
   - To generate an invoice for each nonpaying child bill unit and include all of them in the final invoice, set the threshold value higher than the number of nonpaying child bill units.
   - To generate an invoice without nonpaying child bill unit details, set the threshold value to less than or equal to the number of nonpaying child bill units.

Copyright © 2017, 2026, Oracle and/or its affiliates.

For more information on how to set the **ThresholdSubordsSummary** and **ThresholdSubordsDetail** business parameters, see "Invoicing for Hierarchical Bill Units".

> **Note**
>
> - A value of **0** disables the threshold parameter. BRM does not create separate invoices for nonpaying child bill units.
> - If nonpaying details are not captured in the final invoice, they are still available in the **/invoice** object of the nonpaying child bill unit.

2. In Oracle Analytics Publisher, set the **BRM_Bursting_Invoice_Report** to generate individual invoices or a single invoice from a group of invoices as follows:
   a. Log in to Oracle Analytics Publisher.
   b. From the menu select **Catalog**, and then select **BRM-Invoices**.
   c. From the **BRM-Invoices** section, click the data model **0.0.0.1** link.

      The **0.0.0.1** page appears.
   d. In the **Data Model** section, expand the **Bursting** list and select **BRM_Bursting_Invoice_Report_Bursting_Query**.
   e. From the **Split By** list in the **BRM_Bursting_Invoice_Report_Bursting_Query** section, select one of the following:
      - To generate a single invoice from a group of invoices, select **/ROW_SET/LIST_ROW/ROW/BILL_NO**.
      - To generate individual invoices, select **/ROW_SET/LIST_ROW/ROW/OBJ_ID0**.
   f. Save the data model **0.0.0.1**.
   g. Exit Oracle Analytics Publisher.

   By default the BRM report **BRM_Bursting_Invoice_Report** is set to create the final invoice from a group of invoices for hierarchical accounts. Invoices with the same bill number are generated and then collated and merged into the parent invoice as a single PDF.

Creating Hierarchical Bill Unit Trial Invoices

To create hierarchical bill unit trial invoices with Oracle Analytics Publisher:

1. Go to the *BRM_home***/apps/pin_trial_bill** directory.
2. Run the following command, which creates trial invoices for nonpaying child bill units:

   ```
   pin_trial_bill_accts -pay_type 10007
   ```
3. Run the following command, which creates trial invoices for the paying parent bill units:

   ```
   pin_trial_bill_accts -pay_type ID
   ```

   where *ID* is the element ID of the payment method for the parent bill unit. For more information, see "Default Payment Methods" in *BRM Configuring and Collecting Payments*.
4. Go to the *BRM_home***/apps/pin_inv_doc_gen** directory.
5. Enter the following command to run the **docgen.sh** batch script:

   ```
   docgen.sh
   ```


Copyright © 2017, 2026, Oracle and/or its affiliates.

Creating Hierarchical Bill Unit Regular Invoices

To create hierarchical bill unit invoices with Oracle Analytics Publisher:

1. Go to the *BRM_home***/apps/pin_billd** directory.
2. Run the following command, which runs billing for nonpaying child bill units:

   ```
   pin_bill_accts -pay_type 10007
   ```

3. Run the following command, which runs billing for the paying parent bill units:

   ```
   pin_bill_accts -pay_type ID
   ```

   where *ID* is the element ID of the payment method for the parent bill unit. For more information, see "Understanding Payment Methods" in *BRM Managing Customers*.
4. Go to the *BRM_home***/apps/pin_inv_accts** directory.
5. Run the following command, which creates regular invoices for nonpaying child bill units:

   ```
   pin_inv_accts -hierarchy
   ```

6. Run the following command, which creates regular invoices for the paying parent bill units:

   ```
   pin_inv_accts -pay_type ID
   ```

7. Go to the *BRM_home***/apps/pin_inv_doc_gen** directory.
8. Enter the following command to run the **docgen.sh** batch script:

   ```
   docgen.sh
   ```

Regenerating Failed Invoices for Hierarchical Bill Units

When an error occurs in trial billing, generation of the bill unit's invoice stops. You must rerun trial billing on the nonpaying child bill units before rerunning trial billing on the paying parent bill units. You must also regenerate trial invoices on the paying parent bill units that did not contain an error.

To regenerate invoices for hierarchical bill units that failed the last trial billing run:

1. Fix the errors.
2. Go to the *BRM_home***/apps/pin_trial_bill** directory.
3. Run the following command, which reruns trial billing on nonpaying child bill units that contained errors in the last billing run:

   ```
   pin_trial_bill_accts -pay_type 10007 -retry
   ```

4. Run the following command, which reruns trial billing on the paying parent bill units that contained an error in the last billing run:

   ```
   pin_trial_bill_accts -pay_type ID -retry
   ```

   where *ID* is the element ID of the payment method for the parent account. For more information, see "Understanding Payment Methods" in *BRM Managing Customers*.
5. Run the following command, which generates trial invoicing for the paying parent bill units that did not contain an error in the last billing run:

   ```
   pin_trial_bill_accts -pay_type ID
   ```

6. Go to the *BRM_home***/apps/pin_inv_doc_gen** directory.

Copyright © 2017, 2026, Oracle and/or its affiliates.

7. Enter the following command to run the **docgen.sh** batch script:

   ```
   docgen.sh
   ```

Purging Trial Invoices

To purge trial invoices, run the **pin_trial_bill_purge** utility.

By default, **pin_trial_bill_purge** purges invoices with the **/invoice/trial** POID type. If a custom **/invoice** subclass exists, invoices with **/invoice/***custom_subclass***/trial** POID type are purged.

> **Note**
>
> If the **/invoice** class is partitioned, running the purge utility (**partition_utils -o**) on **/invoice** purges both regular invoices and trial invoices.

For a description of the syntax and parameters of this utility, see "pin_trial_bill_purge" in *BRM Configuring and Running Billing*.

For more information about partitioning and purging database tables, see the following in *BRM System Administrator's Guide*:

- Partitioning Database Tables
- About Purging Data

Deferred Taxation Limitations for Hierarchical Bill Units

BRM offers two options for deferred taxation: taxes can be consolidated into a single item for both the paying parent and nonpaying child bill units, or taxes can be calculated for each individual nonpaying child bill unit, which are listed as separate items on the parent bill.

To configure deferred taxes:

1. Go to *BRM_home***/sys/data/config**.
2. Use the following command to create an editable XML file from the accounts receivable 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:

   ```
   <CycleTaxInterval>value</CycleTaxInterval>
   ```

   where *value* is one of the following:

   - **billing**: Specifies that taxes are consolidated into a single item for both the paying parent and nonpaying child bill unit. The total tax is calculated at the parent level for the entire hierarchy using an aggregated total due as its basis.

Copyright © 2017, 2026, Oracle and/or its affiliates.

> **Note**
>
> This option is not supported when using trial billing or trial invoicing with Oracle Analytics Publisher for hierarchical bill units.

- **accounting**: Specifies that taxes are calculated for each individual nonpaying child bill unit. The total tax is a summation of taxes that were calculated for each bill unit.

> **Note**
>
> This option is supported when you use trial billing or trial invoicing with Oracle Analytics Publisher for hierarchical bill units, with the following limitations:
>
> – The total tax value can contain small rounding errors caused by the system calculating taxes on smaller totals of each bill unit rather than the grand total.
>
> – The basis for determining the tax rate is with the individual nonpaying child's bill unit rather than the parent. To avoid tax rate discrepancies between trial and production runs, all tax-affecting attributes (for example, address information and tax exemptions) for all bill units within the hierarchy must be synchronized to those of the parent bill unit.

> **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 © 2017, 2026, Oracle and/or its affiliates.

Troubleshooting Issues with Hierarchical Bill Unit Invoice Generation

This section lists problems you might encounter while trying to generate hierarchical bill unit invoices and provides information on how to fix them.

Nonpaying child invoices not generated

Scenario: A parent account has a paying parent bill unit. Each of the parent account's two child accounts has a nonpaying child bill unit whose paying parent bill unit is owned by the parent account. The **ThresholdSubordsSummary** and **ThresholdSubordsDetail** business parameters are set to **1**. The following command is run:

```
pin_inv_accts -pay_type 10018 -verbose
```

Solution: To create nonpaying child invoices, the **pin_inv_accts** utility requires the **-hierarchy** parameter. Run the following commands:

```
pin_inv_accts –hierarchy -verbose
pin_inv_accts –pay_type 10018 -verbose
```

Invoice generation is successful, but the billing amounts for the parent bill unit in the final invoice are incorrect

Scenario: One parent account has a paying parent bill unit. Each of the parent account's two child accounts has a nonpaying child bill unit whose paying parent bill unit is owned by the parent account. The **ThresholdSubordsSummary** and **ThresholdSubordsDetail** business parameters are set to **1**. The following commands are run:

```
pin_inv_accts -pay_type 10007 -verbose
pin_inv_accts -pay_type 10018 -verbose
```

Solution: The Oracle Analytics Publisher template is configured with a bursting option, which is using the same invoice details twice. Change the template to a nonbursting option.

See "Setting Nonpaying Child Bill Unit Processing".

Multiple /invoice objects are created instead of one, and after running the docgen.sh script, the invoice is generated, but the amounts are incorrect

Scenario: One parent account has a paying parent bill unit. Each of the parent account's two child accounts has a nonpaying child bill unit whose paying parent bill unit is owned by the parent account. The **ThresholdSubordsSummary** and **ThresholdSubordsDetail** business parameters are set to **0**. The following commands are run:

```
pin_inv_accts -pay_type 10007 -verbose
pin_inv_accts -pay_type 10018 -verbose
```

Solution: Oracle Analytics Publisher is set to create invoices for nonpaying child bill units. Whereas the threshold business parameters setting in BRM is set *not* to create invoices for nonpaying child bill units. Oracle Analytics Publisher and BRM are in conflict. To solve the conflict, change the template to a nonbursting option.

See "Setting Nonpaying Child Bill Unit Processing".

Error when running multiple trial invoicing runs for standalone accounts

Scenario: Unable to run multiple instances of trial billing on a standalone account. Invoices are not created for the second run. The following commands are run:

Copyright © 2017, 2026, Oracle and/or its affiliates.

```
pin_trial_bill_accts -pay_type 10001 -verbose
pin_trial_bill_accts -pay_type 10001 -verbose
```

Solution: As the trial bill number is the same for both trial invoices, Oracle Analytics Publisher is attempting to collate each generated invoice for the final invoice. Include the **docgen.sh** script after each run to create a final invoice for each instance as follows:

```
pin_trial_bill_accts -pay_type 10001 -verbose
doc_gen.sh
pin_trial_bill_accts -pay_type 10001 -verbose
doc_gen.sh
```

> **Note**
>
> The trial bill number is used to collate invoices in Oracle Analytics Publisher.

Storing the Invoice Document

You can store invoice documents in the Oracle Analytics Publisher Scheduler database. Oracle recommends configuring a separate database, other than the BRM database, to store invoice documents. See "Setting Scheduler Configuration".

Sending Invoice Documents to Your Customers

After generating invoice documents, you can email them to your customers. To send the invoice document through email, you must:

- Configure the Scheduler database in Oracle Analytics Publisher. See "Setting Scheduler Configuration".
- Configure the email server and FTP in Oracle Analytics Publisher. See "Configuring the Email Server".
- Configure the delivery option that is retrieved from the **/payinfo** object. For more information, see "About Customer Information Collected When an Account is Created" in *BRM Managing Customers*.

Customizing Invoices

> **Note**
>
> To customize the invoice layout, you should have template-designing experience. See the Oracle Analytics Publisher documentation for instructions on how to create and design layout templates.

You can use the following capabilities of Oracle Analytics Publisher to customize invoice documents:

- **Customizing invoice format**: You can format the sample invoice layout templates or create new invoice templates. See "Customizing Invoice Layout Templates".

Copyright © 2017, 2026, Oracle and/or its affiliates.

- **Enriching invoice data**: You can enrich the data that is displayed in the invoice by getting data from various sources. See "Enriching Invoice Data".

Customizing Invoice Layout Templates

You can customize the invoice layout template by:

- Updating the existing layout template file by using Oracle Analytics Publisher Desktop (Template Builder) and Microsoft Word. For more information about updating the layout template in RTF, see the discussion about creating an RTF template in the Oracle Analytics Publisher documentation.
- Adding a new layout template file. To do so:
  1. Create a new layout template in Oracle Analytics Publisher.
  2. Create a new invoicing **/config/business_profile** object in BRM to reflect the addition of the new layout template. See "Creating an Invoicing Configuration Business Profile".

Enriching Invoice Data

After receiving invoice data from the BRM database, you might want to enrich the invoice data with marketing messages, customer information available in a customer relationship management (CRM) application, and special messages from the accounting department. Oracle Analytics Publisher provides this functionality by retrieving data from additional data sources.

> **Note**
>
> The sample layout templates that are available with BRM-Oracle Analytics Publisher invoicing integration package display data fetched from only the BRM database. If you want to enrich your invoice document with additional data sources, you need to customize the sample layout templates. See "Customizing Invoice Layout Templates".

To enrich BRM invoice data:

- The **/invoice** objects containing data in XML format should be present in the BRM database.
- The additional data source must be configured in Oracle Analytics Publisher to enrich the invoice data. See the discussion of setting up data sources in the Oracle Analytics Publisher documentation.
- Configure data and bursting queries to fetch data from custom data source.

  For detailed instructions on creating or updating Oracle Analytics Publisher reports based on multiple data sources, see the discussion of creating reports based on various data sources in the Oracle Analytics Publisher documentation.

Copyright © 2017, 2026, Oracle and/or its affiliates.

15

Corrective Invoicing

Learn how to create corrective invoices and bills in Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:

- About Corrective Invoices
- About Generating Corrective Invoices
- Setting Up Corrective Invoicing in BRM
- Generating Corrective Invoices
- Adjustments and Corrective Invoicing

See also:

- About Invoicing
- "About Corrective Billing" in *BRM Configuring and Running Billing*

About Corrective Invoices

A corrective invoice is the invoice associated with a corrective bill. You generate a corrective invoice after you make corrections to a bill.

Corrective invoices can be the result of simple corrections or corrections to charges.

- Simple corrections to a bill consist of changes that do not affect the amount owed by the customer, such as changes to the invoice address or to the language in the customer profile.
- Corrections to the charges result from corrections to product offerings and accounts receivable (A/R) actions.

About Content of Corrective Invoices

Corrective invoices include the following:

- Customer information, such as a name and address.
- Previous bill information, such as the previous bill number, billing period, and previous billing date.
- Corrective bill information, such as the corrective bill number and the new payment due date.
- Bill totals.

BRM supports the following types of corrective invoices:

- Replacement invoices, which show the regular (unchanged) portion of a regular invoice and all the corrections made to the invoice. The replacement invoice includes all bill items and events from the previous invoice and any corrections to those bill items and events.

Copyright © 2017, 2026, Oracle and/or its affiliates.

Additionally, it supports regenerating invoices (also known as duplicate invoices), which are previously generated replacement invoices with no changes to the data on that invoice.

- Invoice correction letters, which list only the corrections as a letter for the customer whose bill you corrected.

BRM can generate summary and detailed versions of replacement invoices and invoice correction letters.

About Invoices and Billing Periods

You can issue multiple corrective invoices for a billing period. After BRM issues the corrective invoice for a corrective bill, it cancels the original invoice for the bill and processes subsequent payments against the corrective bills for that billed period. Payments become due on the new due dates associated with the corrective invoice for the corrected bill.

BRM does not carry over A/R actions that were included in a corrective invoice to the next regular invoice. For example, an adjustment of $10 allocated to the November bill period results in a corrective bill for November. BRM does not re-list the $10 adjustment details in the regular bill for December. BRM would list the adjustment details in the regular bill for December only if the corrective bill for November did not include the adjustment.

It is recommended that, when you generate a corrective bill, you should also generate the corrective invoice for that corrective bill. A/R actions that occur between the time a corrective bill and its corrective invoice are generated may lead to incorrect invoices.

If a payment is allocated to specific bill items on the original invoice, BRM processes that payment against the latest corrected balance for the bill items. Any payment that is in excess is automatically credited to the balance on the account.

When BRM processes payments for a bill *before* it issues the corrective invoice for the bill, the corrective bill includes all the payment details for payments already processed against that bill.

About Deferred Corrective Invoices

Whenever you make an adjustment to a bill, you also generate a corrective bill and a corrective invoice. However, you can defer generating the corrective bill and corrective invoice after the next regular billing and invoicing period. In this case, if you enable the **ARItemsInCorrectiveInvoice** business parameter, adjustments are allocated to the corrective bill and displayed only in the corrective invoice whenever it is generated. You can generate the corrective bill and invoice in any billing period.

> **Note**
>
> - Regular invoices do not display any adjustments allocated to other bills.
> - If you make an adjustment and if you have not generated the corrective bill and corrective invoice for the bills, the adjustment will appear in the **PREV_TOTAL** field of the next regular invoice.
> - Deferring corrective invoices is applicable only for accounts of open item account type.

Configuring to Defer Corrective Invoices

To configure deferring corrective invoices:

Copyright © 2017, 2026, Oracle and/or its affiliates.

1. Go to *BRM_home*/**sys/data/config**.
2. Create an XML file from the **/config/business_params** object:

   pin_bus_params -r BusParamsInvoicing bus_params_Invoicing.xml
3. In the file, change **disabled** to **enabled**:

   ```
   <ARItemsInCorrectiveInvoice>enabled</ARItemsInCorrectiveInvoice>
   ```
4. Save the file as **bus_params_Invoicing.xml**.
5. Load the XML file into the BRM database:

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

About Replacement Invoices for Hierarchical Bill Units

You can configure the replacement invoices for parent bills to include information from nonpaying bill units (**/billinfo** objects) by setting the **SubARItemsIncluded** business parameter. For information on **SubARItemsIncluded**, see the description for pricing and rating business parameter entries in *BRM System Administrator's Guide*.

Detailed Replacement Invoice for a Parent Bill

If you include information from nonpaying child bill units, the detailed replacement invoice for the parent bill contains the following:

- The bill total and the bill due amount (or the balance) on the parent bill updated for corrections made to the parent bill or to any nonpaying child bill.
- Adjustment details of the parent and nonpaying child bills.
- For corrected bill items associated with a nonpaying child bill, the corrected fields such as previous amount and corrected amount.
- A list of all events for the parent bill and the nonpaying child bills.

If you do not include information from nonpaying child bill units, the detailed replacement invoice for the parent bill contains the following:

- The bill total and the bill due amount (or the balance) on the parent bill updated for corrections made to the parent bill or any nonpaying child bill.
- All bill items for the parent bill only.
- Adjustment details for the parent bill only.
- A list of all events for the parent bill only.

Summary Replacement Invoice for a Parent Bill

If you include information from nonpaying bill units, the summary replacement invoice for the parent bill contains the following:

- The bill total and the bill due amount (or the balance) on the parent bill updated for corrections made to the parent bill or to any nonpaying child bill.
- Adjustment items.

If you do not include information from nonpaying bill units, the summary replacement invoice for the parent bill contains the following:

Copyright © 2017, 2026, Oracle and/or its affiliates.

- All bill items for the parent bill only.
- Adjustment items for the parent bill only.

Detailed Replacement Invoice for a Nonpaying Child Bill

Whether you include or exclude items from the nonpaying child bill in the parent bill's invoice, the detailed replacement invoice for the nonpaying child bill contains the following:

- The adjustment details, if a nonpaying child bill had a correction.
- For the corrected bill items associated with a nonpaying child bill, the correction fields such as the previous amount and the corrected amount.
- Content identical to the original invoice, if a nonpaying child bill had no charge corrections.
- All the events for the nonpaying child bill.

Summary Replacement Invoice for a Nonpaying Child Bill

Whether you include or exclude items from the nonpaying child bill in the parent bill's invoice, the summary replacement invoice for the nonpaying child bill contains the following:

- All bill items for the nonpaying child bill, whether or not the nonpaying child bill had a correction.
- The adjustment details, if a nonpaying child bill had a correction.
- For the corrected bill items associated with a nonpaying child bill, the correction fields such as the previous amount and corrected amount.
- Content identical to the original invoice, if a nonpaying child bill had no charge correction.

About Invoice Correction Letters for Hierarchical Bill Units

You can configure the invoice correction letters for parent bills to include information from nonpaying child bill units by setting the **SubARItemsIncluded** business parameter. For information on **SubARItemsIncluded**, see the description for pricing and rating business parameter entries in *BRM System Administrator's Guide*.

Detailed Invoice Correction Letter for a Parent Bill

If you include information from nonpaying bill units, the detailed invoice correction letter for the parent bill contains the following:

- The bill total and the bill due amount (or the balance) on the parent bill updated for corrections made to the parent bill or to any nonpaying child bill.
- For event-level adjustments only, the corrected events show the original and corrected amounts.

If you do not include information from nonpaying bill units, the detailed invoice correction letter for the parent bill contains the following:

- The bill total and the bill due amount or the balance on the parent bill updated for corrections made to any nonpaying child bill.
- Adjustment details for the parent bill only.

Copyright © 2017, 2026, Oracle and/or its affiliates.

Summary Invoice Correction Letter for a Parent Bill

If you include information from nonpaying bill units, the summary invoice correction letter for the parent bill contains the following:

- The bill total and the bill due amount (or the balance) on the parent bill updated for corrections made to the parent bill or to any nonpaying child bill.
- Adjustment items.
- Only the corrected bill items associated with a nonpaying child bill that has a correction. For each correction, the invoice correction letter displays the previous amount and corrected amount.

If you do not include information from nonpaying bill units, the summary invoice correction letter for the parent bill contains the following:

- The bill total and the bill due amount (or the balance) on the parent bill updated for corrections made to the parent bill or to any nonpaying child bill.
- Adjustment items for the parent bill only.
- No bill items if no adjustment was made to the parent bill.

Detailed Invoice Correction Letter for Nonpaying Child Bills

If a nonpaying child bill does not have any corrections, BRM does not generate an invoice correction letter.

Whether you include or exclude items from the nonpaying child bill in the parent bill's invoice, the detailed invoice correction letter for the nonpaying child bill contains only the corrected bill items. For each correction, the invoice correction letter displays the previous amount and corrected amount. All adjusted events are shown.

Summary Invoice Correction Letter for Nonpaying Child Bills

If a nonpaying child bill does not have any correction, BRM does not generate an invoice correction letter.

For each nonpaying child bill with a correction, whether you include or exclude items from the nonpaying child bill in the parent bill's invoice, the summary invoice correction letter for the nonpaying child bill contains the following:

- Adjustment items.
- Only the corrected bill item associated with a nonpaying child bill that has a correction. Each such corrected bill item displays the correction fields, such as the previous amount and corrected amount.

How BRM Displays Manual Adjustments

BRM displays the adjustment details and corrected bill items for the manual adjustments that you enter at the account, service, bill, item, and event level. For adjustments at the event level, BRM displays the corrected events.

Copyright © 2017, 2026, Oracle and/or its affiliates.

About Generating Corrective Invoices

To generate corrective invoices, you include corrective bills when you run the **pin_inv_accts** utility. In addition, the **pin_inv_accts** utility automatically runs and generates corrective invoices when you run the **pin_bill_day** script. The **pin_inv_accts** utility can generate summary and detailed invoices for regular bills and summary and detailed corrective invoices for corrective bills.

When creating a corrective invoice, the **pin_inv_accts** utility does the following:

1. Validates the input parameters to ensure that a corrective invoice can be generated for the corrective bill. For example, it checks that there is a corrective bill.
2. Generates the corrective invoice with all the invoice data from the bill, with the account attributes and any additional data elements (for example new aggregations) required to be displayed on the customer invoice.

For a complete description of how invoices are generated in BRM, see "About Using Opcodes" in *BRM Opcode Guide*. *BRM Opcode Guide*.

About the pin_inv_accts Utility Parameters

The **pin_inv_accts** utility is used to generate regular and corrective invoices. For a complete description of the parameters associated with the **pin_inv_accts** utility, see "pin_inv_accts".

To generate corrective invoices, run the **pin_inv_accts** utility, provide values for the parameters that are common to both regular and corrective invoices, and provide values for parameters that are specific to corrective invoices.

Use the following input parameters when you generate corrective invoices:

- **-corr_bills**. The **-corr_bills** parameter creates corrective invoices.
- **-corr_type** *type*. The **-corr_type** parameter indicates the type of corrective invoice you require:
  - **replacement** generates a replacement invoice.
  - **correction** generates an invoice correction letter.

  The **pin_inv_accts** utility uses the value input for **-corr_type** as the required type of corrective invoice for the corrective bill. This value overrides any previous value for the corrective invoice type.
- **-summary | -detail**. BRM generates a summary corrective invoice if **-summary** parameter is set. It generates a detailed invoice if the **-detail** parameter is set.

  For both regular and corrective invoices, use **-summary** or **-detail** to generate the respective summary or detailed versions of the invoice. If you do not specify either of these parameters, **pin_inv_accts** creates a detailed corrective invoice.

For example:

```
pin_inv_accts -corr_bills -corr_type replacement -summary
```

The **pin_inv_accts** utility calls the PCM_OP_INV_MAKE_INVOICE opcode to generate the invoice. For more information on the PCM_OP_INV_MAKE_INVOICE opcode, see "About Using Opcodes" in *BRM Opcode Guide*.

Copyright © 2017, 2026, Oracle and/or its affiliates.

About Corrective Invoice Templates

BRM uses Oracle Analytics Publisher templates to format corrective invoices. If you plan to use Oracle Analytics Publisher to generate corrective invoices, ensure that you have set up your system appropriately. See "Using Oracle Analytics Publisher for Invoicing".

> **Note**
>
> BRM does not support versioning for corrective invoice templates. If you modify the template for replacement invoices or invoice correction letters, BRM uses the modified template.

Table 15-1 lists the default corrective invoice templates for corporate customers.

Table 15-1 Default Corrective Invoice Templates for Corporate Customers

Corrective Invoice Type	Default Template	Report File
Replacement Invoice	BRM_Corporate_Replacement_Invoice.rtf	BRM_Corporate_Replacement_Invoice_Report.xdo
Invoice Correction Letter	BRM_Corporate_Correction_Invoice.rtf	BRM_Corporate_Correction_Invoice_Report.xdo

Table 15-2 lists the default corrective invoice templates for noncorporate customers.

Table 15-2 Default Corrective Invoice Templates for Noncorporate Customers

Corrective Invoice Type	Default Template	Report File
Replacement Invoice	BRM_Consumer_Replacement_Invoice.rtf	BRM_Consumer_Replacement_Invoice_Report.xdo
Invoice Correction Letter	BRM_Consumer_Correction_Invoice.rtf	BRM_Consumer_Correction_Invoice_Report.xdo

Configuring Corrective Invoice Templates Based on Business Profiles

You specify the corrective invoice templates in the **/associated_bus_profile** object for each bill unit. BRM determines the corrective invoice template type for each bill by using the following fields in the **/associated_bus_profile** object.

- PIN_FLD_TEMPLATE_NAME specifies the template name.
- PIN_FLD_REPORT_NAME specifies the report name.
- PIN_INV_TYPE specifies whether the invoice is for a regular bill or a corrective bill.

After you enable the BRM-Oracle Analytics Publisher invoice integration, BRM associates each bill with an invoice template. When you create a customer account, BRM creates one **/associated_bus_profile** object for each bill unit.

Copyright © 2017, 2026, Oracle and/or its affiliates.

Providing Custom Templates for Corrective Invoices

You can create custom templates in Oracle Analytics Publisher to suit your requirements. For information, see the Oracle Analytics Publisher documentation.

To be able to generate a corrective invoice using a custom template for an account's corrective invoices, specify the custom template name in the business profile for the account in the **pin_business_profile.xml** file. If you use both replacement invoices and invoice correction letters for an account, you need to define both sets of templates for the business profile associated with that account.

- **Consumer Invoices**: Specify the custom corrective invoice templates for consumer invoices in place of the following default entries in the **pin_business_profile.xml** file:

```
          <InvoiceType type="RepInvoice" />
          <NameValue key="RepInvoiceReport_Name"
value="BRM_Consumer_Repl_Invoice_Report"/>
          <NameValue key="RepInvoiceTemplate_Name"
value="BRM_Consumer_Repl_Invoice"/>
          <InvoiceType type="InvoiceCorrLetter" />
          <NameValue key="InvCorrLetterReport_Name"
value="BRM_Consumer_Invoice_CorrLetter_Report"/>
          <NameValue key="InvCorrLetterTemplate_Name"
          value="BRM_Consumer_Invoice_CorrLetter"/>
```

- **Corporate Invoices**: Specify the custom corrective invoice templates for corporate invoices in place of the following default entries in the **pin_business_profile.xml** file:

```
          <InvoiceType type="RepInvoice" />
          <NameValue key="RepInvoiceReport_Name"
value="BRM_Corporate_Repl_Invoice_Report"/>
          <NameValue key="RepInvoiceTemplate_Name"
value="BRM_Corporate_Repl_Invoice"/>
          <InvoiceType type="InvoiceCorrLetter" />
          <NameValue key="InvCorrLetterReport_Name"
value="BRM_Corporate_Invoice_CorrLetter_Report"/>
          <NameValue key="InvCorrLetterTemplate_Name"
value="BRM_Corporate_Invoice_CorrLetter"/>
```

About Corrective Invoices and Collections

Collections Manager processes bills by the POID value in the **/bill** object and not the bill number, which differs for the regular bill and its corrective bill. Collections Manager uses the POID value to keep track of the payments and the resulting effect on a bill (that is, whether the bill enters, stays in, or exits collections).

When a corrective bill has been created for a regular bill not in collections, Collections Manager checks the new bill balance against the threshold you defined for collection eligibility. If the bill balance after adjustments goes above that threshold, BRM automatically moves the corrected bill into collections.

When a prior bill is already in collections, BRM suspends actions on that prior invoice until the corrections are entered and processed. After the corrections are processed, BRM generates the corrective invoice and checks the new bill balance against the threshold you defined for collection eligibility. If the new bill balance goes below that threshold, BRM automatically moves the corrected bill out of collections.

Copyright © 2017, 2026, Oracle and/or its affiliates.

For more information on Collections Manager and its **pin_collections_process** utility, see the discussion on how Collections Manager processes bill units in *BRM Collections Manager*.

Corrective Invoices and General Ledger Posts

BRM posts the difference between the original and corrected amounts to the open general ledger (G/L) period only. It does not post such adjustment amounts to the prior G/L period to which the original amounts were posted.

For more information on general ledger, see "About Collecting G/L Data" in *BRM Collecting General Ledger Data*.

Setting Up Corrective Invoicing in BRM

To set up the corrective invoicing process in BRM:

1. Ensure that all the preliminary configurations required for regular and corrective billing are completed. See "Running the Billing Scripts" in *BRM Configuring and Running Billing*.
2. Ensure that BRM is set up according to your requirements:
   - To use the default templates for corrective invoices, verify that BRM is set up with the required templates to generate the required type of corrective invoice. See "Tasks for Setting Up Invoice Generation".
   - To use custom templates for corrective invoices, verify that BRM is set up with the required templates to generate the required type of corrective invoice. See "Providing Custom Templates for Corrective Invoices".
3. Specify the required type of corrective invoice, if necessary. See "Specifying the Type of Corrective Invoice".

Specifying the Type of Corrective Invoice

The **/event/billing/corrective_bill** object created by BRM contains the settings for the corrective invoice type (replacement invoice or invoice correction letter) and the format for the invoice (summary or detail). See "Running the Billing Scripts" in *BRM Configuring and Running Billing*.

You can override these settings when you run the **pin_inv_accts** utility. See "About the pin_inv_accts Utility Parameters" and "pin_inv_accts".

Providing Custom Templates for Corrective Invoices

The default Oracle Analytics Publisher templates for replacement invoices or invoice correction letters have been described in "About Corrective Invoice Templates". You can select an appropriate corrective invoice template for customer accounts from the default corrective invoice layout templates that BRM provides.

Alternatively, you can design a corrective invoice template to suit your business requirements and associate that template with specific accounts.

If you plan to use custom templates for corrective invoices, do the following:

1. Create the required corrective invoice template. See the Oracle Analytics Publisher documentation for instructions on how to create and design layout templates.

Copyright © 2017, 2026, Oracle and/or its affiliates.

2. Load the custom invoice template into Oracle Analytics Publisher. See "Using Oracle Analytics Publisher for Invoicing".
3. Modify the **pin_business_profile.xml** file to associate the name of this custom invoice template with the default replacement invoice or invoice correction letter template. See "Providing Custom Templates for Corrective Invoices".

   For more information on **pin_business_profile.xml**, see "Using Oracle Analytics Publisher for Invoicing".
4. Load the **pin_business_profile.xml** file into **/config/business_profile** objects in the BRM database by using the **load_pin_business_profile** utility. See "Creating and Managing Business Profiles " and "load_pin_business_profile" in *BRM Managing Customers*.
5. Update the **/associated_bus_profile** objects for the modified **/config/business_profile** objects by using the PCM_OP_CUST_SET_ASSOCIATED_BUS_PROFILE opcode. See *BRM Opcode Flist Reference* for more information about this opcode.
6. Complete the steps listed under "Generating Corrective Invoices". When you submit the corrective bill, set the corrective invoice type to the required type.

For information on designing and generating invoices in Oracle Analytics Publisher, see "Using Oracle Analytics Publisher for Invoicing".

Generating Corrective Invoices

To generate corrective invoices:

1. Verify that the required corrective bills have been generated:
   a. Open Billing Care.
   b. Review the contents of the corrective bills.
2. The required templates should be available. Verify that the templates are successfully stored in the business profiles. See "Providing Custom Templates for Corrective Invoices".
3. Run **pin_inv_accts** with the required input parameters to generate the corrective invoice either automatically as part of daily billing or separately. See "About the pin_inv_accts Utility Parameters". For a description of the **pin_inv_accts** utility, see "pin_inv_accts".
4. Use **pin_inv_doc_gen** to generate the corrective invoice documents through Oracle Analytics Publisher. See "Using Oracle Analytics Publisher for Invoicing".
5. (Optional) If you plan to use custom programs (such as DOC1) to generate and publish corrective invoice documents, run **pin_inv_export** with the required input parameters and export corrective invoices to a file in a format you can use with other programs. See **pin_inv_export** in *BRM Developer's Reference*.

Post-Processing Actions

After the corrective invoice is generated:

- View the Corrective Invoice. See "Viewing Corrective Invoices".
- Send the corrective invoices to your customer. See "Sending Invoices to Customers".

Viewing Corrective Invoices

BRM supports the Oracle Analytics Publisher client of the BRM-Oracle Analytics Publisher Integration framework for displaying corrective invoices.

Copyright © 2017, 2026, Oracle and/or its affiliates.

For more information, see the description for displaying invoices in "Displaying Invoices".

Adjustments and Corrective Invoicing

Any account adjustments or service adjustments for a past bill must be allocated to the prior bill *before* BRM generates the corrective bill for that bill. To ensure that the totals and balances for the bill item and the bill correctly reflect the corrections, enter all adjustments either manually or automatically through the rerating process before you generate the corrective invoice for such a bill.

When it generates the corrective bill for such a prior bill, BRM includes all the A/R actions applied or allocated to that bill. It does not enable you to select some allocated A/R items and exclude others to be included in the next bill or the bill in progress.

The **AllocateReratingAdjustments** business parameter is used to specify whether BRM should allocate automatic adjustments from rerating to original bills. When you enable **AllocateReratingAdjustments** business parameter, BRM allocates adjustments to the corrected bill items in the following manner:

- For open item accounting, the adjustment items are allocated to each bill that was corrected. And allocation is made to each of the original bills which included events or items that were corrected by these adjustments.
- For balance forward accounting, corrections are posted to the last bill only. Therefore, the rerating allocation is made to the final bill only. This final bill carries over the balances for all prior bill periods.

The value of the **subARItemsIncluded** business parameter determines how BRM displays the adjustments in the summary and detailed versions of a **Replacement Invoice** or an **Invoice Correction Letter**.

Copyright © 2017, 2026, Oracle and/or its affiliates.

16

Improving Invoicing Performance

Learn how to improve invoicing performance in Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:

- Improving Performance by Customizing Search Results
- Improving Performance by Customizing Event Caching
- Improving Performance by Removing Invoice Details You Do Not Need

Improving Performance by Customizing Search Results

By default, when generating invoices, BRM searches for all items and events in the database, regardless of whether they are included in the invoice. BRM performs the following search operations in steps, which means that it returns search results in blocks instead of returning all search results at one time:

- Searches for the **/bill** object to retrieve the account summary information such as the bill number, billing cycle details, payment due date, and amount due.
- Searches for items and events associated with the **/bill** object to retrieve billing information such as purchase fees, cycle fees, and usage fees.
- Searches for all A/R items and events associated with the **/billinfo** object to retrieve A/R information such as adjustments, disputes, and refunds.

After the search operations are complete, BRM then categorizes the results and displays only the specified events and items on the invoice.

You can improve performance by using the **pin_bus_params** utility to set the **InvEventFetchSize** and **InvItemFetchSize** business parameters to change the number of events and items returned in a block of results. For information about this utility, see "pin_bus_params" in *BRM Developer's Guide*.

To improve performance by changing the number of events and items returned:

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

   ```
   pin_bus_params -r BusParamsInvoicing bus_params_Invoicing.xml
   ```

   This command creates an XML file named **bus_params_Invoicing.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_Invoicing.xml.out**, do the following:
   - Set **InvEventFetchSize** to the maximum number of events allowed:

     ```
     <InvEventFetchSize>value</InvEventFetchSize>
     ```
   - Set **InvItemFetchSize** to the maximum number of items allowed:

Copyright © 2017, 2026, Oracle and/or its affiliates.

```
<InvItemFetchSize>value</InvItemFetchSize>
```

The default value for both business parameters is **10000**.

> ⚠ **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_Invoicing.xml.out** file to **bus_params_Invoicing.xml**.
6. Use the following command to load your changes into the **/config/business_params** object:

   ```
   pin_bus_params bus_params_Invoicing.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*.

To customize the search operation (for example, to search for specific items or to omit a step), use the PCM_OP_INV_POL_SELECT policy opcode. See "Customizing Invoice Search Operations" in *BRM Opcode Guide*.

Improving Performance by Customizing Event Caching

You can improve performance by limiting the amount of information cached. However, when retrieving information, it is quicker to read from a cached field than from the event table. To customize event caching, you can change the fields that are cached, or disable event caching.

By default, BRM caches the fields shown in Table 16-1. To change the fields that are cached, use the PCM_OP_ACT_POL_SPEC_EVENT_CACHE policy opcode. See "Specifying Event Fields to Cache for Invoicing" in *BRM Opcode Guide*.

Table 16-1 Cached Invoicing Fields

Field	Description
PIN_FLD_AMOUNT	The account balance impact. The value can be positive or negative.
PIN_FLD_DISCOUNT	The discount applied to the balance impact.
PIN_FLD_IMPACT_TYPE	Balance impact type - rated by BRM rated-engine (**0x1**), pre-rated (**0x2**), taxed (**0x4**), purchase order (**0x8**), re-rated (**0x20**), and reverse_rated (**0x40**).

Copyright © 2017, 2026, Oracle and/or its affiliates.

Table 16-1 (Cont.) Cached Invoicing Fields

Field	Description
PIN_FLD_ITEM_OBJ	Link to the item object affected by this event. Applies only to the balance array element that impacts currency balances. (This may be different from the PIN_FLD_ITEM_OBJ field in the base **/event** object.)
PIN_FLD_QUANTITY	The quantity applied. The number of units actually applied using this charge.
PIN_FLD_RATE_TAG	Description of the charge used. Same as PIN_FLD_DESCR in the **/rate** object.
PIN_FLD_RESOURCE_ID	Numeric value of the balance element that is impacted.
PIN_FLD_TAX_CODE	Tax code for the charge used. When taxes do not apply, this field is set to **0**.

You can disable event caching by using the **pin_bus_params** utility. For information about this utility, see "pin_bus_params" in *BRM Developer's Guide*.

To disable event caching:

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

   ```
   pin_bus_params -r BusParamsInvoicing bus_params_Invoicing.xml
   ```

   This command creates an XML file named **bus_params_Invoicing.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_Invoicing.xml.out**, set **EventCache** to **disabled**:

   ```
   <EventCache>disabled</EventCache>
   ```

   > ⚠ **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_Invoicing.xml.out** file to **bus_params_Invoicing.xml**.
6. Use the following command to load your changes into the **/config/business_params** object:

   ```
   pin_bus_params bus_params_Invoicing.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).

Copyright © 2017, 2026, Oracle and/or its affiliates.

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

Improving Performance by Removing Invoice Details You Do Not Need

By default, BRM invoices include billing details, such as usage items, device details, and tax totals, which helps customers understand the charges on their invoices. If your company does not offer telephony services or if your company does not want to publish certain details on invoices, you can improve invoicing performance by configuring BRM to retrieve only the invoice details needed.

To do so, you use a CM configuration file entry to specify the details to exclude from your invoices:

- **Exclude device details from invoices**. Improves performance by configuring BRM to skip the device search and the totaling of balance impacts per device.

  > **Note**
  >
  > Do not exclude device details if any of the accounts in your system contain devices; otherwise, accounts with devices will have incorrect invoices.

- **Exclude tax totals from invoices**. Improves performance by configuring BRM to skip the aggregation of tax totals. Exclude tax totals only if you do not have tax in the entire system and you will not perform A/R actions with taxes.

  > **Note**
  >
  > When you exclude tax totals from invoices, the **Taxes and Surcharges** line in the invoice Summary of Current Charges section is set to **0.00**.

- **Exclude package, bundle, and charge offer pricing details from invoices**. Improves performance by configuring BRM to skip the retrieval of package, bundle, and charge offer pricing details from the BRM database. When you exclude that information, invoices do not include package names, detail names, or Siebel CRM promotion names.
- **Exclude balance group details**. Improves performance by configuring BRM to skip the search for balance group details.

  > **Note**
  >
  > When you exclude balance group details, invoices do not include noncurrency balances.

To improve invoicing performance:

1. Open the CM configuration file (*BRM_home***/sys/cm/pin.conf**).
2. Set the **inv_perf_features** entry to the appropriate value:

   ```
   - fm_inv inv_perf_features FlagValue
   ```


Copyright © 2017, 2026, Oracle and/or its affiliates.

where *FlagValue* is a bitmap flag that specifies the details to exclude from your invoices. Each bit position controls a specific feature. To use multiple features, you must OR the bitmaps. For example, to exclude balance impact details, device details, and tax total details from your invoices, set *FlagValue* to **0x00003010**.

- **0x00000010** includes balance impact details from invoices.
- **0x00000800** improves search performance.
- **0x00001000** excludes device details from invoices.
- **0x00002000** excludes tax totals from invoices.
- **0x00004000** excludes package, bundle, charge offer pricing details from invoices.
- **0x00008000** excludes balance group details from invoices.

3. Save and close the file.
4. Stop and restart the CM.

Copyright © 2017, 2026, Oracle and/or its affiliates.

17

Setting Up Invoicing on Multiple Database Schemas

Learn how to set up invoicing on multiple database schemas in Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:

- Setting Up Invoicing on Multiple Database Schemas
- Running Invoicing on Multiple Database Schemas One at a Time
- Running Invoicing on Multiple Database Schemas Simultaneously

Setting Up Invoicing on Multiple Database Schemas

You can run invoicing in a multischema system in either of the following ways:

- Run invoicing on one schema at a time by using one instance of the invoicing utilities. See "Running Invoicing on Multiple Database Schemas One at a Time".
- Run invoicing on multiple schemas simultaneously by using multiple instances of the invoicing utilities. See "Running Invoicing on Multiple Database Schemas Simultaneously".

Running Invoicing on Multiple Database Schemas One at a Time

Running invoicing utilities on multiple database schemas one at a time requires that you edit the invoicing configuration file each time you run the invoicing utilities. Perform the following procedure before you run invoicing:

1. Open the invoicing configuration file (*BRM_home*/**apps/pin_inv/pin.conf**).
2. Change the value of the **login_name** entry to a database account in the schema against which you want to run invoicing.

   For example, to run invoicing using the account **root.0.0.0.2**, change the **login_name** entry as follows:

   ```
   - nap login_name root.0.0.0.2
   ```
3. Save and close the file.
4. Run the invoicing utilities.

Running Invoicing on Multiple Database Schemas Simultaneously

Running invoicing on multiple database schemas simultaneously requires that you create parallel instances of the invoicing configuration files, each of which is configured for a particular schema. Then, you run all instances of your invoicing utilities.

Copyright © 2017, 2026, Oracle and/or its affiliates.

1. For each schema you want to run invoicing on, create a subdirectory under *BRM_home*/**apps/pin_inv**.

   For example, *BRM_home*/**apps/pin_inv/db1** for schema 1, *BRM_home*/**apps/pin_inv/db2** for schema 2, and so forth.

2. Copy the *BRM_home*/**apps/pin_inv/pin.conf** file into each new subdirectory.

3. In each invoicing subdirectory, do the following:

   a. Open the **pin.conf** file.

   b. Change the database number in the **login_name** entry to a database account that resides in the schema against which you want to run invoicing.

      For example, to run invoicing using the account **root.0.0.0.2**, change the **login_name** entry as follows:

      ```
      - nap login_name root.0.0.0.2
      ```

   c. Save and close the file.

4. Run the invoicing utilities from the new subdirectories.

Copyright © 2017, 2026, Oracle and/or its affiliates.

18

Invoice Utilities

Learn about the invoice utilities provided with Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:

- load_pin_invoice_data_map
- pin_inv_accts
- pin_inv_doc_gen
- pin_inv_export
- pin_inv_send
- pin_load_invoice_events
- pin_load_invoice_template
- pin_upd_assoc_bus_profile

load_pin_invoice_data_map

Use this utility to load invoice data templates into the **/config/invoice_data_map** object in the BRM database. You define the invoice data template in the *BRM_home***/data/config/pin_invoice_data_map** file.

For more information, see "Including Invoice Data from Custom Applications".

Location

*BRM_home***/bin**

Syntax

```
load_pin_invoice_data_map [-d] [-v] [-h] invoice_data_map_file
```

Parameters

-d
Creates a log file for debugging purposes. Use this parameter when the utility appears to have run with no errors but the invoice data templates have not been loaded into the database.

-v
Displays information about successful or failed processing as the utility runs.
To redirect the output to a log file, use the following command. Replace *filename***.log** with the name of the log file:
load_pin_invoice_data_map -v > *filename***.log**

Copyright © 2017, 2026, Oracle and/or its affiliates.

> **Note**
>
> If a file with the same name exists, it is overwritten.

-h
Displays the syntax and parameters for this utility.

invoice_data_map_file
The name and location of the file that defines invoice data templates. The default **pin_invoice_data_map** file is in the *BRM_home***/sys/data/config** directory.
If you copy the **pin_invoice_data_map** file to the same directory from which you run the **load_pin_invoice_data_map** utility, you do not need to specify the path or the file name.
If you use the default **pin_invoice_data_map** file name, you do not need to specify it on the command line when it is located in the same directory as this utility.

Results

The **load_pin_invoice_data_map** utility notifies you only if it encounters errors.

> **Note**
>
> You must restart the Connection Manager (CM) to make new invoice data templates available. See "Starting and Stopping the BRM System" in *BRM System Administrator's Guide*.

pin_inv_accts

Use this utility to generate regular invoices and corrective invoices. For regular invoices, run this utility as part of your daily billing to create an invoice for each account that is billed on that day.

> **Note**
>
> - For regular invoices, you typically run **pin_inv_accts** as part of **pin_bill_day**, the daily billing script. If you run invoicing separately from **pin_bill_day**, you must run **pin_bill_accts** before you run **pin_inv_accts**. For more information on generating regular invoices, see "About Invoicing".
> - For corrective invoices, run this utility after you have generated the necessary corrective bills. For more information, see "About Corrective Invoices".

For information on **pin_bill_day**, see "About Running the Billing Scripts" in *BRM Configuring and Running Billing*.

Location

*BRM_home***/bin**

Copyright © 2017, 2026, Oracle and/or its affiliates.

Syntax

```
pin_inv_accts [-test] [-start startDate] [-end endDate] [-regen]
              [-corr_bills -reg_bills -corr_type replacement|correction]

              [-summary|-detail] [-pay_type id|-hierarchy|-file filename]

              [-skip_blank_billnos] [-verbose] [-help]
```

Parameters

-test
Runs a test to find out how many accounts meet the criteria without performing the action. The test has no effect on the accounts. This is most useful when run with the **-verbose** option.

-start *startDate*
-end *endDate*
Generates invoices for accounts whose billing cycle ended during the time period defined by *startDate* and *endDate*, including the *endDate*.
If you do not specify an *endDate*, **pin_inv_accts** uses the current date. If you do not specify a *startDate*, **pin_inv_accts** searches for all invoices that you have not already sent.
You can specify the date in either of two formats:

- *MM*/*DD*/*YY*
- -*N*, where *N* is the number of days before the current date.

-regen
Regenerates previously created invoices in a different format.
To regenerate invoices within a specified date range, use the **-start** and **-end** parameters. If you do not include start and end dates, **pin_inv_accts** regenerates all invoices in the database that are not already stored in the specified format.
To regenerate invoices for nonpaying child bills in a bill unit hierarchy, run the **pin_inv_accts** utility with the **-regen** and **-hierarchy** parameters. By default, invoices are regenerated only for parent bills in a bill unit hierarchy and for paying nonhierarchical bills.

> **Note**
>
> You can regenerate invoices only if you still have all the original **/event** and **/bill** objects for the months you want to regenerate. If the balances are maintained correctly in the **/bill** objects and are not updated, you can regenerate invoices.

-corr_bills
Generates corrective invoices for corrective bills. The type of corrective invoice it generates depends on the input for **-corr_type**. Additionally, whether the utility generates a summary or detailed corrective invoice document depends on whether **-summary** or **-detail** parameter is set in the input.

- If the **-summary** parameter is set, the **pin_inv_accts**utility generates a summary **Replacement Invoice** when the input parameter is **-corr_type replacement** and a summary **Invoice Correction Letter** when the input parameter is **-corr_type correction**. (As stated earlier, if you did not provide any input for the **-corr_type** parameter, the

Copyright © 2017, 2026, Oracle and/or its affiliates.

pin_inv_accts utility uses the current value for the corrective invoice type stored with the corrective bill.)

- If the **-detail** parameter is set, the **pin_inv_accts**utility generates a detailed **Replacement Invoice** when the input parameter is **-corr_type replacement** and a detailed **Invoice Correction Letter** when the input parameter is **-corr_type correction**. (As stated earlier, if you did not provide any input for the **-corr_type** parameter, the **pin_inv_accts** utility uses the current value for the corrective invoice type stored with the corrective bill.)

-reg_bills
Generates regular invoices for regular bills (that is, bills that have no corrections). Generates summary invoice if **-summary** parameter is set; generates detailed invoice if **-detail** parameter is set. If you do not specify any of these parameters, **pin_inv_accts** uses **-detail** parameter.

-corr_type replacement|correction
Generates replacement invoices if the **-corr_type replacement** parameter is set, and generates invoice correction letters if the **-corr_type correction** parameter is set.
If you do not specify any of these parameters, **pin_inv_accts** uses the current value for the corrective invoice type stored with the corrective bill.

-summary|-detail
Generates summary invoices if the **-summary** parameter is set; generates detailed invoices if the **-detail** parameter is set.

- For regular invoices, if you do not specify any of these parameters, **pin_inv_accts** uses **-detail** parameter.
- For corrective invoices, if you do not specify any of these parameters, **pin_inv_accts** uses the current value for the corrective invoice type stored with the corrective bill.

> **Note**
>
> If you generate both regular and corrective invoices in the same run, BRM generates the same type of invoices (summary or detailed).
> If you need to generate one set of invoices as summaries and another set of invoices in detail, run **pin_inv_accts** twice, once for regular invoices and once for corrective invoices.

-pay_type *id*
Generates invoices only for accounts with the specified payment method ID. If you do not specify this parameter, **pin_inv_accts** generates invoices for all payment methods.
Use one payment method ID from Table 18-1.

Payment Method	ID
undefined	0
prepaid	10000
invoice	10001
debit	10002
credit card	10003
direct debit	10005
smart card	10006
nonpaying	10007

Copyright © 2017, 2026, Oracle and/or its affiliates.

Payment Method	ID
beta	10008
internal	10009
guest	10010
cash	10011
check	10012
wire transfer	10013
payorder	10014
postal order	10015
voucher	10016
SEPA	10018

For example, to generate invoices for all bills paid by the check payment method, use this syntax:

```
pin_inv_accts -paytype 10012
```

For more information, see "About Payment Methods" in *BRM Configuring and Collecting Payments*.

-hierarchy
Retrieves bills for paying bill units in which the bill's PIN_FLD_AR_HIERARCHY_SIZE value exceeds the invoicing threshold and generates a separate invoice for each nonpaying bill unit (**/billinfo** object).
See "Invoicing for Hierarchical Bill Units".

-file *filename*
Generates an invoice for each bill in the specified file. For example:

```
pin_inv_accts -file bills_020207
```

> **Note**
>
> The file must be in **pin_flist** format, which is the internal BRM data structure. For example:

```
0 PIN_FLD_RESULTS     ARRAY [0]
1 PIN_FLD_POID        POID [0] 0.0.0.1 /bill 26011 0
0 PIN_FLD_RESULTS     ARRAY [1]
1 PIN_FLD_POID        POID [0] 0.0.0.1 /bill 26091 0
```

For more information on the **pin_flist** format, see "About Formats for Storing Invoices".

-skip_blank_billnos
Excludes invoicing for nonpaying bill units that do not contain bill numbers.
Nonpaying bill units, which use the bill number of their paying parent bill unit, are generated even when billing for the parent bill unit is suppressed. In such cases, they will not contain bill numbers, which invoices require. Therefore, when bill suppression is enabled on an account

Copyright © 2017, 2026, Oracle and/or its affiliates.

that owns a paying parent bill unit, and the nonpaying bill unit threshold is exceeded (multiple threads are used to process the invoices), invoicing fails because the parent bill number is not available. This parameter enables **pin_inv_accts** to proceed without processing those bill units.
(Nonpaying bill units contain bill numbers only when their paying parent bill unit is billed.)

-verbose
Displays information about successful or failed processing as the utility runs.

> **Note**
>
> This parameter is always used in conjunction with other parameters and commands. It is not position dependent. For example, you can enter **-verbose** at the beginning or end of a command to initiate the verbose parameter. To redirect the output to a log file, use the following syntax with the verbose parameter. Replace *filename***.log** with the name of the log file:
> **pin_inv_accts** *any_other_parameter* **-verbose >** *filename***.log**

-help
Displays the syntax and parameters for this utility.

Results

When it is called internally by the **pin_bill_day** script, the **pin_inv_accts** utility logs error information in the **pin_inv.pinlog** file.

pin_inv_doc_gen

Use this utility to create, deliver, and store invoice documents in the Oracle Analytics Publisher Scheduler database.

After **/invoice** objects are created by running the **pin_inv_accts** utility, run the **pin_inv_doc_gen** utility on the Oracle Analytics Publisher server to generate the invoice document.

Start the **pin_inv_doc_gen** utility by running the **docgen.sh** script on the server running Oracle WebLogic Server. For information on the **docgen.sh** script, see "Generating Oracle Analytics Publisher Invoice Documents".

Set the **infranet.dmsearchargs.size** parameter in the **Infranet.properties** configuration file to the number of account and bill unit pairs the utility processes at one time. The maximum value for **infranet.dmsearchargs.size** is 14. For example, if the **InvoiceList.xml** file lists 20 account and bill unit pairs, and **infranet.dmsearchargs.size** is set to **6**, the **pin_inv_doc_gen** utility processes six account and bill unit pairs at a time.

For more information on generating invoices with Oracle Analytics Publisher, see "Using Oracle Analytics Publisher for Invoicing".

Copyright © 2017, 2026, Oracle and/or its affiliates.

> **Note**
>
> - Before running the **pin_inv_doc_gen** utility, save the *BRM_home***/apps/pin_inv_doc_gen/Infranet.properties.sample** configuration file as **Infranet.properties**.
> - To connect to the BRM database, run **pin_inv_doc_gen** from the directory that contains the **Infranet.properties** configuration file.
> - Oracle Analytics Publisher supports multischema systems.
>
> For more information, see "Configuring the Infranet.properties File for pin_inv_doc_gen".

Location

*BRM_home***/apps/pin_inv_doc_gen**

Syntax

```
pin_inv_doc_gen [-accts_list InvoiceList.xml][-type [-consumer|-corporate]
                [-schema SchemaNumber]
                [-status pending|generated]
                [-start startDate][-end endDate]
                [-params name1:value1;name2:value;...nameN:valueN]
                [-h|-help]
```

Parameters

If you do not include parameters at the command line, the utility generates invoice documents for all invoices in a pending status. The parameters **-accts_list** and **-type** are mutually exclusive. The parameters **-status** and **-start**, **-end** are valid only if included along with other parameters. The parameter **-schema** is valid only with Oracle Analytics Publisher.

-accts_list *InvoiceList*.xml
*InvoiceList***.xml** specifies the list of all account and bill unit pairs for which an **/invoice** object is generated and stored in the BRM database, and for which an invoice document is not yet generated.

-type
Specifies the type of account for which the invoice document is to be generated:

- **-type consumer** generates a consumer account.
- **-type corporate** generates a corporate account. For a corporate account, all the **/invoice** objects of its child accounts that have nonpaying bill units (if they have a separate **/invoice** POID) are selected for generating the invoice document. See "Invoicing for Hierarchical Bill Units". However, the **/invoice** object of the child account that has a paying bill unit is not selected for generating invoice documents of a corporate type.

The **-type** parameter maps to the PIN_FLD_PARENT_FLAGS field of the **/billinfo** object of the **/invoice** object. For consumer accounts, the PIN_FLD_PARENT_FLAGS field value is **0.** For corporate accounts, which own the paying parent bill unit in the bill unit hierarchy, the PIN_FLD_PARENT_FLAGS field value is greater than zero.

Copyright © 2017, 2026, Oracle and/or its affiliates.

-schema *SchemaNumber*
(Valid only for Oracle Analytics Publisher in the appropriate version) Generates the BRM schema number, in the format **0.0.0**.*n*, for processing BRM invoices. For example, the following parameter specifies to process BRM invoices in schema 0.0.0.2:

```
-schema 0.0.0.2
```

If the schema number is not specified, it defaults to **0.0.0.1**.
For more information, see "Configuring the BRM-Oracle Analytics Publisher Invoicing Integration to Support Multischema Systems".

-status pending|generated
Specifies the status of the **/invoice** object for which the invoice is to be generated:

- **-status pending** selects the **/invoice** objects for a document that has not been generated. It selects **/invoice** objects with a status of **0** and generates documents for them.
- **-status generated** generates duplicate documents for invoices for a document that has been generated. The duplicate document displays a **Duplicate** watermark on the document. For more information, see "Generating Duplicate Invoice Documents".

-start *startDate*
-end *endDate*
Generates invoice documents for accounts or bill units whose **/invoice** objects were created during the time period defined by *startDate* and *endDate*.
If you do not specify a start date, **pin_inv_doc_gen** fetches all **/invoice** objects with a status of PIN_INV_STATUS_PENDING.
If you do not specify an end date, **pin_inv_doc_gen** uses the current date. The end date cannot be greater than the current date. The end date specified is inclusive, which means all **/invoice** objects created until 23:59:59 of the end date are selected by the search query.
Specify the date in *MM*/*DD*/*YYYY* format.

-params *nameN*:*valueN*
Specifies to generate invoice documents based on accounts that meet custom criteria. The criteria consists of a list of key-value pairs, where:

- *nameN* is the name of the custom parameter you added in the report's data model prepended with **pin_Inv_**. For example: **p_Inv_Status**, **p_Inv_ReportName**, or **p_Inv_EffectiveT**.
- *valueN* is the required value of the specified field.

By default, the separator between *nameN* and *paramN* is a colon (:), and the delimiter is a semicolon (**;**). You can change the separator and delimiter using the **infranet.report.parameter.separator** and **infranet.report.parameter.delimiter** entries in the *BRM_home***/apps/pin_inv_doc_gen/Infranet.properties** file. See "Configuring the Infranet.properties File for pin_inv_doc_gen".

-h|-help
Displays the syntax and parameters for this utility.

pin_inv_export

Use this utility to export invoices to a format you can use with other programs, such as DOC1. This command-line utility retrieves invoices from the BRM database and saves them as files.

Copyright © 2017, 2026, Oracle and/or its affiliates.

You specify the format and directory for the invoice files in the invoice utility configuration file. For more information, see "Exporting Invoices" and the instructions in the invoicing configuration file.

The **pin_inv_export** utility generates file names automatically with an **inv** prefix.

If you run the **pin_inv_export** utility without any parameters, it exports summary invoices for regular bills to separate files.

> **Note**
>
> For nonpaying child bill invoices that are *not* consolidated into the parent invoice, **pin_inv_export** creates a separate invoice file for each nonpaying child bill. The file naming convention is:
>
> **inv**_*BillPOID_BillDate_NonpayingChildAccountPOID_ARaccountPOID*.*ext*
>
> For nonpaying child bill invoices that are consolidated into the parent invoice and are exported as an individual invoice, the file name will *not* contain the parent account POID. Instead, it will be in the format:
>
> **inv**_*BillPOID_BillDate_NonpayingChildAccountPOID*.*ext*

Location

BRM_home/**bin**

Syntax

```
pin_inv_export [-start startDate] [-end endDate] [ -trial | -detail filename]
[-verbose] [-help]
```

Parameters

-start *startDate*
-end *endDate*
Exports invoices for accounts whose billing cycle ended during the time period defined by *startDate* and *endDate*, including the *endDate*.
If you do not specify an end date, **pin_inv_export** uses the current date. If you do not specify a start date and an end date, **pin_inv_export** creates a file for every invoice in the database. You can specify the date in either of two formats:

- *MM*/*DD*/*YY*
- -*N*, where *N* is the number of days before the current date.

For example:
pin_inv_export -verbose -start 01/01/09 -end 01/10/09
pin_inv_export -verbose -start -10, sets the start date to 10 days before the current date and the end date to the current date.

-trial
Exports trial invoices only (**/invoice/trial**).

Copyright © 2017, 2026, Oracle and/or its affiliates.

> **Note**
>
> If a custom **/invoice** storable subclass exists, invoices of type **/invoice/.../trial** are exported.

-detail *filename*
Exports detailed invoices for both trial and regular bills listed in the specified file. You cannot use the **-detail** parameter to export detailed invoices for only regular bills or only trial bills. This parameter overrides the PIN_FLD_INV_TYPE value specified in the **/payinfo** object associated with the bill.
The file must be in pin_flist format and can contain any number of bills. For example:

```
0 PIN_FLD_RESULTS     ARRAY [0]
1 PIN_FLD_POID        POID [0] 0.0.0.1 /bill 26011 0
0 PIN_FLD_RESULTS     ARRAY [1]
1 PIN_FLD_POID        POID [0] 0.0.0.1 /bill 26091 0
```

For more information on generating a pin_flist file of bills, see "pin_inv_accts".

-verbose
Displays information about successful or failed processing as the utility runs.

> **Note**
>
> This parameter is always used in conjunction with other parameters and commands. It is not position dependent. For example, you can enter **-verbose** at the beginning or end of a command to initiate the verbose parameter. To redirect the output to a log file, use the following syntax with the **verbose** parameter.
> **pin_inv_export** *any_other_parameter* **–verbose >** *filename***.log**
> where *filename* is the name of the log file.

-help
Displays the syntax and parameters for this utility.

pin_inv_send

Use this utility to email or print invoices depending on the delivery method specified in a customer's account (**Email** or **Postal**).

You can configure the following behavior by editing the **pin_inv_send** utility's configuration file (*BRM_home***/apps/pin_inv/pin.conf**).

- To set the maximum size in kilobytes of invoices that can be sent by email, set the **inv_send_size** entry. This entry is used to restrict sending large invoices to the Email Data Manager (**dm_email**).
- To send individual nonpaying child bill invoices to the email address of the parent A/R bill, set the **send_sub_inv_2_parent** entry.

The Email Data Manager must be running when you run this utility, even for printing. For information, see "Sending Email to Customers Automatically" in *BRM Managing Customers* and "Configuring the Email Data Manager for Printing".

For more information about sending invoices, see "Sending Invoices to Customers".

Copyright © 2017, 2026, Oracle and/or its affiliates.

Location

BRM_home/**bin**

Syntax

```
pin_inv_send    [-start startDate] [-end endDate] [-pay_type id]
                [-format_id id] [-v] [-h]
```

Parameters

-start *startDate*
-end *endDate*
Emails or prints invoices for accounts with invoices whose billing cycle ended during the time period defined by the *startDate* and *endDate*, including the *endDate*.
If you do not specify an end date, **pin_inv_send** uses the current date. If you do not specify a start date, **pin_inv_send** searches for all invoices that you have not already sent.
You can specify the date in either of two ways:

- *MM/DD/YY*
- *-N*, where *N* is the number of days before the current date.

-pay_type *id*
Emails or prints invoices only for accounts with the specified payment method. If you do not specify this parameter, **pin_inv_send** emails or prints invoices for invoice payment methods only.
Use one payment method ID from Table 18-2.

Payment Method	ID
undefined	0
prepaid	10000
invoice	10001
debit	10002
credit card	10003
direct debit	10005
smart card	10006
nonpaying	10007
beta	10008
internal	10009
guest	10010
cash	10011
check	10012
wire transfer	10013
payorder	10014
postal order	10015
voucher	10016
SEPA	10018

Copyright © 2017, 2026, Oracle and/or its affiliates.

For example, to email or print invoices for all bills paid by the check payment method, enter this command:

```
pin_inv_send -pay_type 10012
```

For more information on payment methods, see "About Payment Methods" in *BRM Configuring and Collecting Payments*.

-format_id *id*
Specifies the invoice's format. By default, **pin_inv_send** mails invoices in HTML format. Use this parameter to use DOC1 format instead. The format ID can have the values shown in Table 18-3.

Format	ID
HTML	1
DOC1	3

-v
Displays information about successful or failed processing as the utility runs.

> **Note**
>
> This parameter is always used in conjunction with other parameters and commands. It is not position dependent. For example, you can enter **-v** at the beginning or end of a command to initiate the verbose parameter. To redirect the output to a log file, use the following syntax with the verbose parameter. Replace *filename***.log** with the name of the log file:
> **pin_inv_send** *any_other_parameter* **–v >** *filename***.log**

-h
Displays the syntax and parameters for this utility.

pin_load_invoice_events

Use this utility to load a list of the events that you want to include on invoices into a **/config/invoice_events** object.

Location

*BRM_home***/bin**

Syntax

```
pin_load_invoice_events  [-brand "0.0.0.1/account 1"] [-eventfile file]
                         [-logfile logfile] [-reload] [-debug] [-help]
```

Parameters

-brand "0.0.0.1/account 1"
Specifies the root POID.

Copyright © 2017, 2026, Oracle and/or its affiliates.

> **Note**
>
> The quotation marks are required.

-eventfile *file*
Specifies the path name of the event file. The events in the file should be listed one per line. For example:
/event/billing/adjustment/account
/event/billing/adjustment/event
/event/billing/adjustment/item
/event/billing/adjustment/tax_event
/event/billing/cycle/tax
/event/billing/payment/cash
/event/billing/payment/cc
...

-logfile *logfile*
Specifies the full path name of the log file.

-reload
Reloads an existing **/config** object.

-debug
Logs the flist and detailed messages in the log file.

-help
Displays the syntax and parameters for this utility.

Results

The **pin_load_invoice_events** utility should notify you that it was successful. If errors are returned, look in the utility log file **(default.pinlog)** for detailed error messages. The log file is located either in the directory from which the utility was started or in a directory specified in the configuration file.

pin_load_invoice_template

Use this utility to load the invoice template into the BRM database or to change the flag that specifies whether to use an HTML template or an XSL style sheet for an existing template object.

For information about creating invoice templates, see "Designing Invoices".

Location

*BRM_home***/bin**

Syntax

```
pin_load_invoice_template  [-brand "0.0.0.1/account 1"] [-type template_type]
                           [-locale locale_string] [-template template_file]

                           [-usexsl] [-debug] [-logfile logfile] [-help]
```


Copyright © 2017, 2026, Oracle and/or its affiliates.

Parameters

-brand "0.0.0.1/account 1"
Specifies the root POID.

> **Note**
>
> The quotation marks are required.

-type *template_type*
Specifies the template type. To use an HTML template, specify "HTML". To use an XSL style sheet, specify a mime type, for example, "text/html".

> **Note**
>
> The **-template** option must specify a file containing a template of the same type you specify in this option.

-locale *locale_string*
Specifies the locale such as, en_US.

-template *template_file*
Specifies the full pathname of the template file.

-usexsl
Uses the XSL invoice formatting opcode. This option sets the **PIN_FLD_FLAG** field in the **/config/invoice_templates** object to XSL style sheets.

-debug
Logs the flist and detailed messages in the log file.

-logfile *logfile*
Specifies the full path name of the log file.

-help
Displays the syntax and parameters for this utility.

Changing the Template Flag

To change the flag that specifies whether to use an HTML template or an XSL style sheet, set the PIN_FLD_FLAGS field in the **/config/invoice_templates** object by using this syntax:

- For an HTML template:

  ```
  pin_load_invoice_template -brand "0.0.0.1/account 1"
  ```

- For an XSL style sheet:

  ```
  pin_load_invoice_template -brand "0.0.0.1/account 1" [-usexsl]
  ```

  When the **-usexsl** option is not specified, an HTML template will be used.

Copyright © 2017, 2026, Oracle and/or its affiliates.

Results

The **pin_load_invoice_template** utility should notify you that it was successful. If errors are returned, look in the utility log file (**default.pinlog**) for detailed error messages. The log file is located either in the directory from which the utility was started or in a directory specified in the configuration file.

pin_upd_assoc_bus_profile

Use this utility to associate bill units with an invoicing business profile. It searches the BRM database for **/billinfo** objects with an empty PIN_FLD_ASSOC_BUS_PROFILE_OBJ_LIST field and populates the field with an **/associated_bus_profile** object. See "Associating Pre-Existing Bill Units with Business Profiles".

Before running **pin_upd_assoc_bus_profile**, you must:

- Load invoicing business profiles into the database. See "Loading Invoicing Business Profiles into the Database".
- Enable the BRM-Oracle Analytics Publisher integration framework. This utility fails if you attempt to run it when the BRM-Oracle Analytics Publisher integration framework is disabled. See "Enabling the BRM-Oracle Analytics Publisher Integration".

> **Note**
>
> The **pin_upd_assoc_bus_profile** utility needs the configuration file in *BRM_home*/**apps/pin_billd**. This configuration file must be present in the directory from which you run the utility.
>
> Because **pin_upd_assoc_bus_profile** is based on the BRM multithreaded application (MTA) framework, the configuration file also requires some performance-related configuration entries. For more information, see "Configuring Your Multithreaded Application" in *BRM Developer's Guide*.

Location

BRM_home/**bin**

Syntax

```
pin_upd_assoc_bus_profile  [-file filename] [-verbose] [-help]
```

Parameters

-file *filename*
Specifies the name and location of the input file that lists the **/billinfo** objects to update. This is an optional parameter.
The input file should be in the following format:

```
0 PIN_FLD_RESULTS       ARRAY [0] allocated 20, used 1
1     PIN_FLD_POID       POID [0] 0.0.0.1 /billinfo 37395 1
```


Copyright © 2017, 2026, Oracle and/or its affiliates.

```
0 PIN_FLD_RESULTS       ARRAY [1] allocated 20, used 1
1     PIN_FLD_POID       POID [0] 0.0.0.1 /billinfo 37488 3
```

-verbose
Displays information about successful or failed processing as the utility runs.

-help
Displays the syntax and parameters for this utility.

Copyright © 2017, 2026, Oracle and/or its affiliates.