Oracle® Communications Billing and Revenue Management Designing and Generating Invoices





Oracle Communications Billing and Revenue Management Designing and Generating Invoices, Release 12.0

E51015-06

Copyright © 2017, 2022, 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 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" or "commercial computer software documentation" 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, and MySQL 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.

Contents

Preface

Audience	i>
Documentation Accessibility	i
Diversity and Inclusion	ί
About Invoicing	
About Invoices	1-1
About the Invoicing Process	1-2
Understanding Invoice Layout	1-3
Standard Details	1-3
Adjustment, Payment, Promotion, and Other Charges and Credits Details	1-3
Charge Details	1-4
About Localized Invoices	1-5
Designing Invoices	
About Designing Invoice Templates	2-1
Designing XSLT Invoice Templates	2-1
Turning on XSLT Style Sheet Processing	2-2
Designing HTML Invoice Templates	2-2
Designing Hyperlinked Invoice Templates	2-2
Loading Stored Procedures for Hyperlinked Invoice Templates Reports	2-3
Setting Up the Data Model for Hyperlinked Invoice Templates	2-3
Setting Up the Hyperlinked Invoice Templates	2-4
Modifying the Hyperlink Path Prefix	2-5
Loading Invoice Templates	2-6
Switching between XSLT and HTML Templates	2-7
Customizing Invoice Content	
Grouping Items By Service Type	3-1
Including Payment, A/R, and Tax Details in Invoices	3-2



	A/R Events That Can Be In Invoices	3-2
	Including Shadow Event Adjustment Details in Invoices	3-4
	Including Soft Descriptors in Invoices	3-4
	Aggregating Taxes on Invoices	3-4
	Using Invoice Templates That Display Promotion Names	3-5
	Defining Multiple Invoices for One Customer	3-5
	Using an /invoice Subclass	3-6
4	Invoicing for Hierarchical Bill Units	
	About Invoicing for Hierarchical Bill Units	4-1
	About Viewing Invoices for Accounts That Have Nonpaying Bill Units	4-2
	Setting Defaults for Hierarchical Bill Unit Invoices	4-2
	Sending Child Account Invoices to the Parent Account's Email Address	4-3
5	Adding Siebel CRM Promotion Names to Invoices	
	Adding Siebel CRM Promotion Names to Invoices	5-1
	How BRM Adds Siebel CRM Promotion Names	5-2
	Setting Up BRM Invoices to Display Siebel CRM Promotion Names	5-2
	Specifying Whether BRM Displays Promotion Details on Invoices	5-3
6	Including Invoice Data from Custom Applications	
	Including Invoice Data from Custom Applications	6-1
	Using Data Map Templates	6-1
	Loading the Invoice Data Map Templates	6-2
7	Generating Invoices	
	Generating Invoices Automatically	7-1
	Generating Invoices Manually	7-1
	Generating Detailed Invoices from an External File	7-2
	Generating Invoices for Non-Invoice Payment Methods	7-2
	Exporting Invoices	7-3
8	Sending Invoices to Customers	
	Sending Invoices to Customers	8-1
	Setting the Maximum Invoice Size for Email	8-1
	Sending Invoices to Accounts That Use the Invoice Payment Method	8-1
	Sending Invoices to Accounts That Do Not Use the Invoice Payment Method	8-2



	Configuring the Email Data Manager for Printing	8-3
9	Displaying Invoices	
	About Displaying Invoices	9-1
	Using Web Pages to Display Invoices	9-1
	Using the XML Sample Invoice Viewing Utility	9-2
10	Storing Invoices	
	About Formats for Storing Invoices	10-1
	Specifying the Default Format in Which to Store Invoices in BRM	10-2
	Storing Invoices in a Separate Database Schema	10-3
	Installing the Invoice Data Manager	10-3
	Configuring BRM to Use a Separate Invoice Database Schema	10-4
	Configuring Invoice Applications to Use a Separate Invoice Database Schema	10-5
	Starting the Invoice Data Manager	10-5
	Configuring the Invoice pin.conf for Multiple Database Schemas	10-5
11	Designing and Generating Invoices in Oracle Business Intelligent Publisher About Using BI Publisher for Invoicing	11-1
	Designing Invoices	11-3
	About Invoice Layout Templates for Invoice Document Generation	11-3
	About BI Publisher Reports for Invoice Document Generation	11-4
	Including Late Payment Fees in Invoices	11-4
	Including Finance Charges in Invoices	11-5
	Setting Up Your System to Generate Invoice Documents Using BI Publisher	11-6
	Creating an Invoicing Configuration Business Profile	11-6
	Enabling the BRM-BI Publisher Integration	11-7
	Loading Invoicing Business Profiles into the Database	11-8
	Associating Pre-Existing Bill Units with Business Profiles	11-8
	Viewing Invoices Individually on the BI Publisher Console	11-9
	Enabling the Display of Customer Details on the Web	11-9
	Installing the BRM-BI Publisher Invoicing Integration Package	11-9
	Pre-Installation Requirements	11-10
	Removing Older BRM-BI Publisher Invoicing Integration Layout Template Files	11-10
	Install Billing and Invoicing Utilities	11-11
	Installed Files	11-11
	Upgrading Existing Reports from BI Publisher 11g to BI Publisher 12c	11-13



Configuring the pin_inv_doc_gen others	11-13
Configuring the Infranet.properties File for pin_inv_doc_gen with BI Publisher	11-14
Configuring the docgen.sh Batch Script	11-16
Changing Command-Line Parameters for pin_inv_doc_gen	11-16
Setting Scheduler Configuration	11-17
Configuring the Email Server	11-17
Configuring HTTPS for Secure Communication	11-17
Creating the XMLP_INVOICE_JOB Table in the Scheduler Database	11-18
Configuring the BRM-BI Publisher Invoicing Integration to Support Multischema	
Systems	11-19
Loading Stored Procedures	11-19
Creating New JDBC Data Sources	11-19
Generating BI Publisher Invoice Documents in BI Publisher	11-21
Generating BI Publisher Invoice Documents for a List of Accounts	11-21
Generating BI Publisher Invoice Documents Based on the Account Type	11-22
Generating Duplicate Invoice Documents with BI Publisher	11-23
Generating Invoice Documents for Non-Invoice Accounts	11-23
Generating Invoices for Hierarchical Bill Units with BI Publisher	11-24
Setting Up Invoicing for Hierarchical Bill Units with BI Publisher	11-24
Invoice Template for Hierarchical Bill Units	11-25
Setting Nonpaying Child Bill Unit Processing in BI Publisher and BRM	11-25
Creating Hierarchical Bill Unit Trial Invoices with BI Publisher	11-26
Creating Hierarchical Bill Unit Regular Invoices with BI Publisher	11-27
Regenerating Failed Invoices for Hierarchical Bill Units	11-27
Purging Trial Invoices	11-28
Deferred Taxation Limitations for Hierarchical Bill Units	11-28
Troubleshooting Issues with Hierarchical Bill Unit Invoice Generation	11-29
Storing the Invoice Document	11-31
Sending Invoice Documents to the Customers	11-31
Customizing Invoices	11-31
Customizing Invoice Layout Templates	11-31
Enriching Invoice Data	11-32
Corrective Invoicing	
About Corrective Invoices	12-1
About Content of Corrective Invoices	12-1
About Invoices and Billing Periods	12-2
About Deferred Corrective Invoices	12-2
Configuring to Defer Corrective Invoices	12-3
About Replacement Invoices for Hierarchical Bill Units	12-3
Detailed Replacement Invoice for a Parent Bill	12-3



12

	Summary Replacement invoice for a Parent bill	12-2
	Detailed Replacement Invoice for a Nonpaying Child Bill	12-4
	Summary Replacement Invoice for a Nonpaying Child Bill	12-4
	About Invoice Correction Letters for Hierarchical Bill Units	12-5
	Detailed Invoice Correction Letter for a Parent Bill	12-5
	Summary Invoice Correction Letter for a Parent Bill	12-5
	Detailed Invoice Correction Letter for Nonpaying Child Bills	12-5
	Summary Invoice Correction Letter for Nonpaying Child Bills	12-6
	How BRM Displays Manual Adjustments	12-6
	About Generating Corrective Invoices	12-6
	About the pin_inv_accts Utility Parameters	12-6
	About Corrective Invoice Templates	12-7
	Configuring Corrective Invoice Templates Based on Business Profiles	12-8
	Providing Custom Templates for Corrective Invoices	12-8
	About Corrective Invoices and Collections	12-9
	Corrective Invoices and General Ledger Posts	12-9
	Setting Up Corrective Invoicing in BRM	12-9
	Specifying the Type of Corrective Invoice	12-10
	Providing Custom Templates for Corrective Invoices	12-10
	Generating Corrective Invoices	12-10
	Post-Processing Actions	12-11
	Viewing Corrective Invoices	12-11
	Adjustments and Corrective Invoicing	12-11
13	Improving Invoicing Performance	
	Improving Performance by Customizing Search Results	13-1
	Improving Performance by Customizing Event Caching	13-1
	Improving Performance by Removing Invoice Details You Do Not Need	13-2
14	Setting Up Invoicing on Multiple Database Schemas	
	Setting Up Invoicing on Multiple Database Schemas	14-1
	Running Invoicing on Multiple Database Schemas One at a Time	14-1
	Running Invoicing on Multiple Database Schemas Simultaneously	14-2
15	Invoice Utilities	
	load_pin_invoice_data_map	15-1
	pin_inv_accts	15-1
	pin_inv_doc_gen	15-2
	pin_inv_export	15-6
	hu-un-ovboir	10-0



pin_inv_send	15-10
pin_load_invoice_events	15-12
pin_load_invoice_template	15-13
pin upd assoc bus profile	15-15



Preface

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.

Documentation Accessibility

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

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.



1

About Invoicing

In Oracle Communications Billing and Revenue Management (BRM), an invoice lists the balance information for a customer's bill, and the 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 (*Ibillinfo* 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.



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.



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.

- 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 Business Intelligence Publisher. See "Designing and Generating Invoices in Oracle Business Intelligence Publisher".

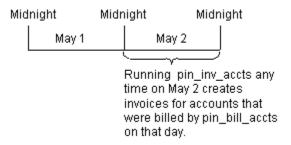
About the Invoicing Process

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, Customer Center, and Self-Care Manager.
 The invoice's format is based on a template you create and load.
- 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-BI Publisher integration package.



By default, the invoice document presents charges in primary currency. However, all the relevant charges details are available in secondary currency in the *linvoice* 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-Business Intelligence 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 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.



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".

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



2

Designing Invoices

In Oracle Communications Billing and Revenue Management (BRM), you design the appearance of an invoice 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_homelsys/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, Customer Center, or a Web browser.



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.

BRM provides two sample style sheets, located in *BRM_homelsys/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. See *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_homelsys/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. See *BRM Opcode Guide* for more information.

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.



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 BI Publisher server.

To configure hyperlinked invoice templates:

- Back up the BI Publisher repository, which has a defaulted location of BIP_Homel user_projects/domains/bifoundation_domain/config/bipublisher/repository/.
- 2. Load the stored procedures. See "Loading Stored Procedures for Hyperlinked Invoice Templates Reports".
- 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".
- 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:



 Go to BIP_home/user_projects/domains/bifoundation_domain/config/ bipublisher/repository/ directory.

where BIP_home is the directory in which BI Publisher is installed.



If you do not use *BIP_home/user_projects/domains/* **bifoundation_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:

- Go to the BIP_homeluser_projects/domains/bifoundation_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:

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



You can also load **HIT.rtf** templates within the BI Publisher server.

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

```
cproperty 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 BI Publisher server.

Modifying the Hyperlink Path Prefix

BI Publisher creates hyperlinks within PDF files. The defaulted directory is in the BI 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:

- 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.
- 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 -usexs1
```

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:

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"



Customizing Invoice Content

In Oracle Communications Billing and Revenue Management (BRM), you can customize the content that is included in your customers' invoices.

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:

- Open the CM configuration file (BRM_homelsys/cm/pin.conf).
- 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.
- Stop and restart the CM.



In the invoicing business profile, if the **EnableInvoicingIntegration** config business parameter is enabled, ensure that in *BRM_homeIsysIcmIpin.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 or Customer Center only, and not in the BI Publisher client.



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).
- 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 *l*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
- /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 *levent/billing/cycle/tax* in the *events.file* file.

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:

- Open the CM configuration file (BRM_homelsys/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_homelsys/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 *BRM Developer's Guide*.

To aggregate taxes on your invoices:

- 1. Go to BRM_homelsys/data/config.
- 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
```

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

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. See 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 or Customer Center 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.
- Use a custom CRM system to pass the invoice name in the input flist of PCM_OP_CUST_SET_PAYINFO. See BRM Opcode Guide.

Using an /invoice Subclass

If you add a subclass to the *linvoice* 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 *lpayinfo* 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_homelsys/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 *linvoice* 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.



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.



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".

About Viewing Invoices for Accounts That Have Nonpaying Bill Units

If you open an account that has a nonpaying bill unit in Billing Care or Customer Center 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, Customer Center, and Self-Care Manager.

In Self-Care Manager, invoices can be viewed only for the account that is logged in.

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 *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.
- 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:

- Go to BRM_home/sys/data/config.
- Create an XML file from the /config/business_params object:
 - pin bus params -r BusParamsInvoicing bus params Invoicing.xml
- 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.



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.



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

- 8. Stop and restart the CM.
- 9. (Multischema systems only) Run the **pin_multidb** script with the **-R CONFIG** parameter. For more information, see *BRM System Administrator's Guide*.

You do not need to restart the 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 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_homelapps/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.



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.

You implement this functionality by customizing Siebel CRM to send data through AIA to the PCM_OP_SUBSCRIPTION_SET_BUNDLE opcode. See *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. See 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 *linvoice* 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.
- 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 *BRM Developer's Guide*.

To specify whether BRM invoices displays promotion names:

- Go to BRM_homelsys/data/config.
- 2. Create an XML file from the **/config/business params** object:

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

In the file, change disabled to enabled:

```
< \verb|PromotionDetailDisplay>| \textbf{enabled} < / \texttt{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.
- (Multischema systems only) Run the pin_multidb script with the -R CONFIG parameter.
 For more information, see BRM System Administrator's Guide.

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. See *BRM Opcode Guide* for more information.



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_homelsysl* 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:

```
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: I



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.



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



7

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 *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*.
- (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.

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:

pin_inv_accts -pay_type 10003



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.

In addition, you can export detailed invoices for accounts that are set up to receive summary invoices, and you can export invoices for a list of bills in an external file.



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:

- Open the invoicing configuration file (BRM_homelapps/pin_inv/pin.conf) and find the pin inv export entries.
- 2. 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.



3. 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
```

- **4.** Change to a directory with a valid invoicing configuration file. See "Connecting BRM Utilities" in *BRM System Administrator's Guide*.
- 5. Run the pin_inv_export utility:

```
pin_inv_export
```

Running **pin_inv_export** with no parameters creates a separate file for each invoice in the database.



You can specify a date range to export only invoices created on certain dates.

To generate trial invoices, run the following command:

```
pin inv export -verbose -trial
```

To generate detailed invoices, 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.

For a description of this utility's syntax and parameters, see "pin inv export".

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_homelapps/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_homelapps/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".

Start the Email Data Manager. See "Sending Email to Customers Automatically" in BRM Managing Customers.

- **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:

```
{\tt pin\_inv\_send -pay\_type} \ \textit{id}
```

For id, enter one payment method ID as shown in Table 8-1:

Table 8-1 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:

- 1. Download from the Internet and install the html2ps converter tool.
- 2. Open the dm_email configuration file (BRM_homelsys/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
```

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



9

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 Web Pages to Display Invoices
- Using the XML Sample Invoice Viewing Utility

About Displaying Invoices

BRM provides these options for displaying invoices:

- The Invoice Viewer in Billing Care or Customer Center.
- Self-Care Manager customer self-care Web pages.
- Sample XML invoice viewing utility (pin_inv_view.pl).
- BI Publisher client, if you use the BRM-Business Intelligence 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*.



- 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 Web Pages to Display Invoices

If you use Self-Care Manager, your customers can display their invoices on your customer self-care Web site. See "Setting Up Customer Self Care with Self-Care Manager" in *BRM Managing Customers*.

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, such as Internet Explorer.

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



10

Storing Invoices

You can 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:

- Self-Care Manager requires pin_flist storage format to display HTML invoices.
- 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-BI Publisher Invoicing Integration package. In addition, you might use this storage format if you are using XML to design and display invoices. See "Designing and Generating Invoices in Oracle Business Intelligence Publisher".

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*.



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.

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



Storing Invoices in a Separate Database Schema



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.
- 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:

- 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



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:

- 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*.
- 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 10-1:

Table 10-1 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.
\$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:

 Open the Connection Manager (CM) configuration file (BRM_homelsyslcml 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.



The invoicing utilities do not use this entry.

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.

Configuring Invoice Applications to Use a Separate Invoice Database Schema

To use a separate database schema for storing invoices:

- Open the invoice configuration file (BRM_homelapps/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_homelapps/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".



11

Designing and Generating Invoices in Oracle Business Intelligence Publisher

You can design, generate, and customize Oracle Communications Billing and Revenue Management (BRM) invoice documents using Oracle Business Intelligence (BI) Publisher.

Topics in this document:

- · About Using BI Publisher for Invoicing
- Designing Invoices
- Setting Up Your System to Generate Invoice Documents Using BI Publisher
- Generating BI Publisher Invoice Documents in BI Publisher
- Generating Invoices for Hierarchical Bill Units with BI Publisher
- Storing the Invoice Document
- Sending Invoice Documents to the Customers
- Customizing Invoices

About Using BI Publisher for Invoicing

BI Publisher enables you to create more stylized and detailed invoices in PDF format. BRM also includes sample RTF templates. You can configure and extend these templates for your specific business needs.

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

With BI Publisher, you use an SQL bursting query, which is the process of splitting data into blocks and generating documents for those blocks. Bursting increases throughput performance in BI Publisher.

To use this feature, you must install BI Publisher and BI Publisher Desktop.

When you use the BRM-BI Publisher integration:

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

Figure 11-1 shows the BRM and BI Publisher invoice generation process:

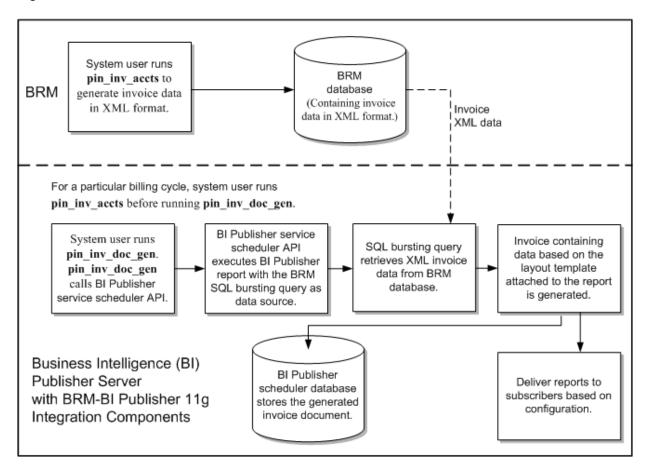


Figure 11-1 BRM and BI Publisher Invoice Generation Process

The BRM-BI Publisher integration generates invoice documents as follows:



The BRM-BI Publisher integration must be enabled. See "Enabling the BRM-BI Publisher Integration".

- BRM associates bill units with a BI Publisher invoice and report. When
 accounts are created, BRM associates each bill unit (/billinfo object) with a BI
 Publisher invoice report and layout template.
- 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.
- 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".

Note:

To generate invoice documents in BI Publisher, you must store the invoice data in XML format. See "About Formats for Storing Invoices".

4. The pin inv doc gen utility:

- Runs BI Publisher reports. This utility, based on the input parameters, starts and runs appropriate BI Publisher reports for invoice documents. It starts a BI Publisher invoicing report, with the SQL bursting data query, through a scheduled job. See "Generating BI Publisher Invoice Documents in BI Publisher".
- 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 BI Publisher database. The utility stores the
 generated invoice documents in the BI Publisher Scheduler database. See "Storing
 the Invoice Document".

The BRM-BI Publisher invoicing integration package contains:

- The SQL bursting query.
- BI Publisher reports for generating invoice documents for consumer and corporate accounts. See "About BI Publisher 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 BI Publisher layout templates or the reports provided with the BRM-BI 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-BI Publisher invoicing integration package contains six layout templates in RTF for the consumer and corporate accounts which can be customized.

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

BRM provides the following default templates for regular invoices:

- BRM_Corporate_Invoice.rtf
- BRM_Consumer_Invoice.rtf



BRM provides the following default **Replacement Invoice** templates:

- 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** templates:

- 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 BI Publisher associates with each of the previously listed default templates for invoices is:

BRM_Invoice_Bursting_Report.xdo

About BI Publisher Reports for Invoice Document Generation

The BRM-BI 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 BI 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.



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-BI 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 *litem* storable class to include *litem/late_fee*. See "Creating Custom Fields and Storable Classes" in *BRM Developer's Guide*.
- 2. Open the *BRM_homelsys/data/pricing/example/config_item_types.xml* file and define a type for the item. For example:

- 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_homelsys/data/pricing/example/config_item_tags.xml* and define a tag for the new service. For example:

- 6. Save the file.
- 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-BI 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:

- Extend the *litem* storable class to include *litem/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:

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_homelsys/data/pricing/example/config_item_tags.xml* and define a tag for the new service. For example:

- 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
```

Setting Up Your System to Generate Invoice Documents Using BI Publisher

To use the BRM-BI Publisher integration framework to generate invoice documents:

- Install BI Publisher. To install BI Publisher, follow the instructions in the BI Publisher documentation.
- 2. In BRM, specify the BI Publisher invoice report and layout templates 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-BI Publisher invoicing integration package on the BI Publisher server. See "Installing the BRM-BI Publisher Invoicing Integration Package".
- 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.

Creating an Invoicing Configuration Business Profile

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

- Reads the invoicing configuration business profile (Iconfig/business_profile object) to determine which BI Publisher invoice report and layout template the bill unit should be associated with.
- Creates an lassociated_bus_profile object for the bill unit. This object specifies
 the names of the BI 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:

- Enable the BRM-BI Publisher integration. See "Enabling the BRM-BI Publisher Integration".
- 2. Specify the requirements for a bill unit to be associated with an invoicing configuration business profile by creating a validation template.
- Specify the validation template, invoice report, and layout template to be associated with the bill unit.
- 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".
- If your system includes accounts that were created before you integrated BRM invoicing with BI Publisher, associate your existing /billinfo objects with /associated_bus_profile objects. See "Associating Pre-Existing Bill Units with Business Profiles".

Enabling the BRM-BI Publisher Integration

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

To enable the BRM-BI Publisher integration for invoices:

- 1. Go to BRM_homelsys/data/config.
- 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:

```
\verb|<EnableInvoicingIntegration>| \textbf{enabled}| < | \texttt{EnableInvoicingIntegration}| \\
```

Store invoices in XML format by setting InvoiceStorageType to 1:

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

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

```
pin bus params bus params Invoicing.xml
```

- Stop and restart the CM.
- 8. (Multischema systems only) Run the **pin_multidb** script with the **-R CONFIG** parameter. For more information, see "pin_multidb" in *BRM System Administrator's Guide*.

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.

Associating Pre-Existing Bill Units with Business Profiles

All **/billinfo** objects created before your BRM-BI Publisher integration are not linked to an **/associated_business_profile** object, which prevents you from creating BI Publisher invoices for those **/billinfo** objects. For example, this occurs if you upgraded to BRM 12.0 from a prior release.

To generate BI Publisher invoice documents for **/billinfo** objects created before the BRM-BI 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:

 Make sure that the BRM-BI Publisher integration is enabled. See "Enabling the BRM-BI Publisher Integration".



The utility fails if you attempt to run it when the BRM-BI 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_homelapps/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*.

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 the BI Publisher Console

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

Configure **ViewInvoiceReport** to point to the data source by editing the data model, configuring the data source, saving changes, and restarting BI 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 BI Publisher.

To enable the display of customer details on the Web:

- Go to the BIP_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:

```
cproperty name="include_rowsettag" value="false"/>
```

- 4. Change false to true.
- Save and close the file.

Installing the BRM-BI Publisher Invoicing Integration Package

This section describes how to install the BRM-BI Publisher invoicing integration package. This package is set up on the BI Publisher server.

To install the BRM-BI Publisher invoicing integration package:

• Ensure that the pre-installation requirements are satisfied. See "Pre-Installation Requirements".



- Manually remove previously installed versions of the BRM-BI Publisher invoicing integration layout template files. See "Removing Older BRM-BI Publisher Invoicing Integration Layout Template Files".
- Run the BRM-BI Publisher invoicing integration package. See "Install Billing and Invoicing Utilities".

Note:

Install the BRM-BI Publisher invoicing integration package on the system on which BRM Reports is installed. If the BRM-BI Publisher invoicing integration package and BRM Reports are not installed on the same system, you cannot view an invoice in BI Publisher 12c.

Pre-Installation Requirements

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

- BI Publisher 12c is installed.
- BI Publisher home environment variable has been successfully set.
- BRM 12.0 is installed. The patch set contains the **pin_inv_doc_gen** utility updates that are required for BI Publisher integration.

For more information on these requirements, see BRM Reports.

Removing Older BRM-BI Publisher Invoicing Integration Layout Template Files

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

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

 Go to BIP_home/user_projects/domains/bifoundation_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:
 - 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 BIP_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 Billing and Invoicing Utilities, see "Installing Individual BRM Components" in *BRM Installation Guide*.

Installed Files

The BRM-BI Publisher invoicing integration program copies the following files listed in Table 11-1 to your computer during installation:

Table 11-1 BI Publisher-Related Files Placed on Your Computer

File	Install Path	Description
docgen.sh	BRM_home/apps/ pin_inv_doc_gen	Batch script for setting environment variables and launching the pin_inv_doc_gen utility.
BRM_Consumer_Correction_Invoice.rtf	BIP_home/user_projects/ domains/ bifoundation_domain/ config/bipublisher/ repository/Reports/ BRM_Invoices/ SchemaNumber	RTF file associated with layout template for consumer corrective invoice letter document.
BRM_Consumer_Invoice.rtf	BIP_home/user_projects/ domains/ bifoundation_domain/ config/bipublisher/ repository/Reports/ BRM_Invoices/ SchemaNumber	RTF file associated with layout template for consumer invoice document.
BRM_Consumer_Replacement_Invoice.rt f	BIP_home/user_projects/ domains/ bifoundation_domain/ config/bipublisher/ repository/Reports/ BRM_Invoices/ SchemaNumber	RTF file associated with layout template for consumer replacement invoice document.



Table 11-1 (Cont.) BI Publisher-Related Files Placed on Your Computer

File	Install Path	Description
BRM_Corporate_Correction_Invoice.rtf	BIP_home/user_projects/ domains/ bifoundation_domain/ config/bipublisher/ repository/Reports/ BRM_Invoices/ SchemaNumber	RTF file associated with layout template for corporate corrective invoice letter document.
BRM_Corporate_Invoice.rtf	BIP_home/user_projects/ domains/ bifoundation_domain/ config/bipublisher/ repository/Reports/ BRM_Invoices/ SchemaNumber	RTF file associated with layout template for corporate invoice.
BRM_Corporate_Replacement_Invoice.rt f	BIP_home/user_projects/ domains/ bifoundation_domain/ config/bipublisher/ repository/Reports/ BRM_Invoices/ SchemaNumber	RTF file associated with layout template for corporate replacement invoice document.
Create_Xmlp_Invoice_Job.sql	BRM_home/apps/ pin_inv_doc_gen	Script for creating XMLP_INVOICE_JOB table in BI Publisher Scheduler database.
invoice_bursting.plb	BRM_home/apps/ pin_inv_doc_gen	Invoice bursting stored procedure.
Infranet.properties.sample	BRM_home/apps/ pin_inv_doc_gen	Infranet.properties.sample file for pin_inv_doc_gen.jar. Contains the configuration entries for the pin_inv_doc_gen utility. Important: Before running the pin_inv_doc_gen utility, save the Infranet.properties.sample file as Infranet.properties.
InvoiceList.xml	BRM_home/apps/ pin_inv_doc_gen	XML file containing the details of accounts and bill units. The InvoiceList.xml file is passed along with –accts_list as an input parameter to pin_inv_doc_gen.
InvoiceList.xsd	BRM_home/apps/ pin_inv_doc_gen	XSD file that validates the InvoiceList.xml.
pin_inv_doc_gen.jar	BRM_home/apps/ pin_inv_doc_gen	JAR file for pin_inv_doc_gen application.
lib/ library	BRM_home/apps/ pin_inv_doc_gen/lib	Directory containing dependent pin_inv_doc_gen libraries.



Upgrading Existing Reports from BI Publisher 11g to BI Publisher 12c

To use BI Publisher 11*g* templates in BI Publisher 12*c*, make the following BI Publisher modifications listed in Table 11-2:

Table 11-2 BI Publisher Template Modifications for BI Publisher 12c

Modification	BI Publisher 11g	BI Publisher 12c
XML Structure	[data source = web service]	<pre>[data source = sql query for bursting]</pre>
	<pre><invoice> <poid>0.0.0.1 /bill 151665 3</poid> </invoice></pre>	<pre><rowset> <invoice_poid>159696</invoice_poid></rowset></pre>
Referencing fields from the root	<pre><? xdoxslt:set_variable(\$_XDOCTX,'currency', /invoice/BILLINFO/ CURRENCY)?></pre>	<pre><? xdoxslt:set_variable(\$_XDOCTX,'curren cy',/ROWSET/ROW/BUFFER_BUF/invoice/ BILLINFO/CURRENCY)?></pre>
'Contains' function	if:contains(EVENT_OBJ ,'adjustment')='true' ?	if:contains(EVENT_OBJ,'adjustment')?
'Doesn't contain' function	<pre>contains(ITEM_OBJ,'wr iteoff_reversal')! ='true'</pre>	<pre>not(contains(ITEM_OBJ,'writeoff_rever sal'))</pre>

Configuring the pin_inv_doc_gen Utility

The <code>pin_inv_doc_gen</code> utility creates, delivers, and stores your invoice documents in the BI Publisher Scheduler database.

Configuring pin_inv_doc_gen involves:

- Configuring the Infranet.properties File for pin_inv_doc_gen with BI Publisher
- 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
- Loading Stored Procedures

Configuring the Infranet.properties File for pin inv doc gen with BI Publisher

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

To configure the Infranet.properties file for pin_inv_doc_gen:

- 1. Go to BRM_homelapps/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.



You must have read-write permissions to access the *linvoice* object. Your invoice data is stored in the *linvoice* 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 11-3.

Table 11-3 Infranet.properties.sample File Entries

Entry	Description
infranet.bip.password	Specifies the password.
	Note: The password can be encrypted with Advanced Encryption Standard (AES) for security.
infranet.bip.username	Specifies the name of the BI Publisher server.
infranet.bip.url	Specifies the URL of BI 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.burst.threadpool.chunksize	Specifies the maximum number of BRM invoices per BI Publishing bursting query.
infranet.burst.threadpool.maxsize	Specifies the maximum number of concurrent threads in the pool.
infranet.burst.threadpool.size	Specifies the number of core threads in the pool.
infranet.connection	Specifies the URL to the BRM service.



Table 11-3 (Cont.) Infranet.properties.sample File Entries

Entry	Description
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.angle	Specifies the angle of the watermark text. The default value is 45 (angle of text is 45 degrees).
	See the BI Publisher Java API documentation.
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.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.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.text	Specifies the text to be displayed as the background watermark in the duplicate invoice.
	The default value is DUPLICATE .
infranet.schedulerdb.url	Specifies the scheduler database URL in the following format:
	jdbc:oracle:thin:hostname:port/service
	where <i>hostname</i> is the hostname of the scheduler database, <i>port</i> is the port number for the scheduler database, and <i>service</i> 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.http.notification.server	Specifies the name of the HTTP server used for notifications.
	Important: Enable HTTP notifications in BI Publisher. See the BI Publisher documentation for details.
infranet.http.notification.userName	Specifies the HTTP server login name.



Table 11-3 (Cont.) Infranet.properties.sample File Entries

Entry	Description
infranet.http.notification.password	Specifies the HTTP server login password.
infranet.log.file	Specifies the log file name for the pin_inv_doc_gen utility.
infranet.log.level	Specifies the level of log for the pin_inv_doc_gen utility.
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 doesn't require a user name or password. Use 1 or 0 .
infranet.notify.http.when.success	Specifies whether to receive HTTP notification when a BI Publisher job is successful. By default, HTTP notification is enabled and set to true . To disable HTTP notification, set this entry to false .
infranet.notify.http.when.warning	Specifies whether to receive HTTP notification when a BI Publisher job contains warnings. By default, HTTP notification is enabled and set to true . To disable HTTP notification, set this entry to false .
infranet.polling.sleeptime	Specifies the sleep time (in milliseconds) between polling calls while waiting for the BI Publisher schedule to complete. The default value is 5000 (5 seconds).
infranet.threadpool.fetchsize	Specifies the number of records that can be fetched by the thread. The default value is 5 .
infranet.threadpool.maxsize	Specifies the maximum number of threads. The default value is 5.
infranet.threadpool.size	Specifies the number of threads. The default value is 3.

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_homelapps/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.



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:



- Go to the BRM_homelapps/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.
- 6. In the command prompt, navigate to the BRM_homelapps/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 BI Publisher documentation.

Configuring the Email Server

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

Configuring HTTPS for Secure Communication

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

To configure HTTPS for secure communication:

- 1. Enable HTTPS in BI Publisher. For more information, see the BI 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 BI 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_homelapps/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:

- Go to BRM_homelapps/pin_inv_doc_gen.
- 2. Open the Create Xmlp Invoice Job.sql file.
- Search for <tablespacename>.
- Replace <tablespacename> with the tablespace name where all the BI Publisher Scheduler tables in the BI Publisher Scheduler database are created.
- 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.
- Click Next.
- i. Select the server on which the BI Publisher software is installed.



- k. Click Finish.
- I. Click Activate Changes.
- m. Stop and restart the BI Publisher server.

Configuring the BRM-BI Publisher Invoicing Integration to Support Multischema Systems

To configure the BRM-BI 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_homelapps/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-BI 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 BI 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://BI_Publisher_host:port/xmlpserver/
```

where:

- BI_Publisher_host is the host name on which the BI Publisher server is installed.
- port is the console port. The default port is 9704.

The BI 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 the schema number in 0.0.0.*n* format. For example, 0.0.0.1 for the primary schema or 0.0.0.2 for the secondary schema.
 - 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.

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 reports directory **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 BIP_home/user_projects/domains/bifoundation_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
```

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.

- 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, you add **-schema 0.0.0.***n* where *n* is the schema number 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.**



Generating BI Publisher Invoice Documents in BI Publisher

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

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

To run the pin_inv_doc_gen utility:

- 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.
- Enter the following command which runs the docgen.sh batch script: docgen.sh

Generating BI Publisher Invoice Documents for a List of Accounts

You can generate BI 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.

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:

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



You can use the *BRM_homelapps/pin_inv_doc_gen/InvoiceList.xml* file as an example.



- 2. Save and close the file.
- 3. Go to the BRM_homelapps/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 step1.

- 7. Save and close the file.
- Go to the BRM_homelapps/pin_inv_doc_gen directory.
- 9. Enter the following command which runs the **docgen.sh** batch script:

docgen.sh

Generating BI Publisher Invoice Documents Based on the Account Type

To generate invoices based on the account type:

- 1. Go to the BRM homelapps/pin inv doc gen directory.
- 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.
- In the command prompt, navigate to the BRM_homelapps/pin_inv_doc_gen directory.
- 7. Enter the following command which runs the **docgen.sh** batch script:

```
docgen.sh
```



Generating Duplicate Invoice Documents with BI Publisher

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

To generate duplicate BI Publisher invoice documents:

- 1. Go to the BRM_homelapps/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 with BI Publisher".
- 3. Open the **docgen.sh** batch script in a text editor.
- 4. Search for the following line:

```
invoicedocgen.pin inv doc gen -status pending
```

Replace with:

invoicedocgen.pin_inv_doc_gen -status generated

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

docgen.sh



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. BI Publisher generates PDFs for any invoice with a non-invoice payment type and stores them in the BI 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 Business Intelligence Publisher.



Generating Invoices for Hierarchical Bill Units with BI 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 BI 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.



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 with BI Publisher

To set up invoicing for hierarchical bill units with BI Publisher:

- Ensure that all the preliminary configurations required for billing are completed. See BRM Configuring and Running Billing.
- 2. Enable the BI Publisher integration. See "Enabling the BRM-BI Publisher Integration".
- 3. Set your system to generate invoice documents using BI Publisher. See "Setting Up Your System to Generate Invoice Documents Using BI Publisher".
- 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 a BI Publisher invoice and report.
- Load the business profile information into the BRM database. See "Loading Invoicing Business Profiles into the Database".
- Configure the pin_inv_doc_gen utility. See "Configuring the pin_inv_doc_gen Utility".
- 8. Verify that both BI Publisher and BRM are set to generate the final invoice either with or without nonpaying child bill unit details. See "Setting Nonpaying Child Bill Unit Processing in BI Publisher and BRM".
- Verify that the BRM_homelapps/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.



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. See BRM Opcode Guide.

Setting Nonpaying Child Bill Unit Processing in BI Publisher and BRM



When generating invoices for hierarchies, both BI Publisher and BRM 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 BI 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 BI Publisher to generate individual invoices.

In BRM and BI 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 in BI Publisher and BRM for hierarchies:

- 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.

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 *linvoice* object of the nonpaying child bill unit.
- 2. In BI 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 BI Publisher.
 - b. From the menu select **Catalog**, and then select **BRM-Invoices**.
 - $\textbf{c.} \quad \text{From the $\tt BRM-Invoices} \ \text{section, click the data model $\tt 0.0.0.1$ 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 BI 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 with BI Publisher

To create hierarchical bill unit trial invoices with BI Publisher:

- 1. Go to the BRM_homelapps/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_homelapps/pin_inv_doc_gen directory.
- Enter the following command to run the docgen.sh batch script: docgen.sh

Creating Hierarchical Bill Unit Regular Invoices with BI Publisher

To create hierarchical bill unt invoices with BI Publisher:

- 1. Go to the BRM_homelapps/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_homelapps/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 homelapps/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_homelapps/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_homelapps/pin_inv_doc_gen directory.
- 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 *linvoice/trial* POID type. If a custom *linvoice* subclass exists, invoices with *linvoice/custom_subclass/trial* POID type are purged.



If the *l*invoice class is partitioned, running the purge utility (**partition_utils - o**) on *l*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:

When 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. In this instance, the cycle_tax_interval entry in BRM_homelsys/cm/pin.conf configuration file is set as follows:



-fm_bill cycle_tax_interval billing



This option is not supported when using trial billing or trial invoicing with BI Publisher for hierarchical bill units.

 When 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. In this instance the cycle_tax_interval entry in BRM Homelsyslcm/pin.conf configuration file is set as follows:

-fm bill cycle tax interval accounting

Note:

This option is supported when you use trial billing or trial invoicing with BI 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.

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 BI Publisher template is configured with a bursting option, which is using the same invoice details twice. Change the BI Publisher template to a nonbursting option.

See "Setting Nonpaying Child Bill Unit Processing in BI Publisher and BRM".

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: BI 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. BI Publisher and BRM are in conflict. To solve the conflict, change the BI Publisher template to a nonbursting option.

See "Setting Nonpaying Child Bill Unit Processing in BI Publisher and BRM".

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:

```
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, BI 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
```



The trial bill number is used to collate invoices in BI Publisher.



Storing the Invoice Document

You can store invoice documents in the BI 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 the Customers

After generating the invoice document, you can email it to the customers. To send the invoice document through email, you must:

- Configure the Scheduler database in BI Publisher. See "Setting Scheduler Configuration".
- · Configure the email server and FTP in BI Publisher. See "Configuring the Email Server".
- Configure the delivery option that is retrieved from the **/payinfo** object. For more information, see *BRM Managing Customers*.

Customizing Invoices



To customize the invoice layout, you should have template-designing experience. See the BI Publisher documentation for instructions on how to create and design layout templates.

You can use the following capabilities of BI 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".
- **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 BI 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 BI Publisher documentation.
- Adding a new layout template file. To do so:
 - 1. Create a new layout template in BI Publisher.
 - 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. BI Publisher provides this functionality by retrieving data from additional data sources.

Note:

The sample layout templates that are available with BRM-BI 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 *linvoice* objects containing data in XML format should be present in the BRM database.
- The additional data source must be configured in BI Publisher to enrich the invoice data. See the discussion of setting up data sources in the BI Publisher documentation.
- Configure data and bursting queries to fetch data from custom data source.

For detailed instructions on creating or updating BI Publisher reports based on multiple data sources, see the discussion of creating BI Publisher reports based on various data sources in the BI Publisher documentation.



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.

Additionally, it supports regenerate 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:

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

pin_bus_params -r BusParamsInvoicing bus_params_Invoicing.xml

In the file, change disabled to enabled:

<ARItemsInCorrectiveInvoice>enabled</ARItemsInCorrectiveInvoice>

- Save the file as bus_params_Invoicing.xml.
- Load the XML file into the BRM database: pin bus params bus params Invoicing.xml
- 6. Stop and restart the CM.
- 7. (Multischema systems only) Run the **pin_multidb** script with the **-R CONFIG** parameter. For more information, see *BRM System Administrator's Guide*.

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:

- 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 showing 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.

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.

About Generating Corrective Invoices

To generate corrective invoices, you include corrective bills when you run the <code>pin_inv_accts</code> utility. In addition, the <code>pin_inv_accts</code> utility automatically runs and generates corrective invoices when you run the <code>pin_bill_day</code> script. The <code>pin_inv_accts</code> 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:

- 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 *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 *BRM Opcode Guide*.

About Corrective Invoice Templates

BRM uses Oracle Business Intelligence (BI) Publisher templates to format corrective invoices. If you plan to use BI Publisher to generate corrective invoices, ensure that you have set up your system appropriately. See "Designing and Generating Invoices in Oracle Business Intelligence Publisher".



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 12-1 lists the default corrective invoice templates for corporate customers.

Table 12-1 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 12-2 lists the 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	BRM_Consumer_Correction_Invoice.rtf	BRM_Consumer_Correction_Invoice_Report.xdo

Table 12-2 Default Corrective Invoice Templates for Noncorporate Customers

Configuring Corrective Invoice Templates Based on Business Profiles

You specify the corrective invoice templates in the *lassociated_bus_profile* object for each bill unit. BRM determines the corrective invoice template type for each bill by using the following fields in the *lassociated_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 hill

After you enable the BRM-BI Publisher invoice integration, BRM associates each bill with an invoice template. When you create a customer account, BRM creates one *I* associated_bus_profile object for each bill unit.

Providing Custom Templates for Corrective Invoices

You can create custom templates in BI Publisher to suit your requirements. For information, see the Oracle Business Intelligence 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:

 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:

Letter

About Corrective Invoices and Collections

Collections Manager processes bills by the POID value in the *Ibill* 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.

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 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 *BRM Configuring and Running Billing*.
- 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 "Setting
 Up Your System to Generate Invoice Documents Using BI Publisher".
 - 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 <code>/event/billing/corrective_bill</code> 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 <code>BRM Configuring and Running Billing</code> for more information.

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 BI 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:

- Create the required corrective invoice template. See the Oracle Business Intelligence Publisher documentation for instructions on how to create and design layout templates.
- Load the custom invoice template into BI Publisher. See "Designing and Generating Invoices in Oracle Business Intelligence Publisher".
- 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 "Designing and Generating Invoices in Oracle Business Intelligence Publisher".
- 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 BI Publisher, see "Designing and Generating Invoices in Oracle Business Intelligence Publisher".

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.
- 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".
- Use pin_inv_doc_gen to generate the corrective invoice documents through BI Publisher. See "Designing and Generating Invoices in Oracle Business Intelligence Publisher".
- 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 BI Publisher client of the BRM-Business Intelligence Publisher Integration framework for displaying corrective invoices.

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**.



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 CM configuration file entries **inv_item_fetch_size** and **inv_event_fetch_size** to change the number of items and events returned in a block of search results.

- 1. Open the CM configuration file (*BRM_homelsys/cm/pin.conf*).
- Change the value of the inv_item_fetch_size entry.
 - The default is 10000.
- Change the value of the inv_event_fetch_size entry.
 - The default is 10000.
- 4. Stop and restart the CM.

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 *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 13-1. To change the fields that are cached, use the PCM_OP_ACT_POL_SPEC_EVENT_CACHE policy opcode. See *BRM Opcode Guide*.

Table 13-1 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), prerated (0x2), taxed (0x4), purchase order (0x8), re-rated(0x20), and reverse_rated(0x40).
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 also disable event caching. To do so:

- 1. Open the CM configuration file (BRM_homelsys/cm/pin.conf).
- 2. Set the **event_cache** entry to **0**:
 - fm inv event cache 0



If this entry is set to any other value or is not present in the file, invoicing assumes there is data in the event cache and produces invoices without event details.

- 3. Save and close the file.
- 4. Stop and restart the CM.

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.



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

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.



14

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_homelapps/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.0.0.2
- 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.

- 1. For each schema you want to run invoicing on, create a subdirectory under BRM homelapps/pin inv.
 - For example, *BRM_homelapps/pin_inv/db1* for schema 1, *BRM_homelapps/pin_inv/db2* for schema 2, and so forth.
- 2. Copy the *BRM_homelapps/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.0.0.2
 - c. Save and close the file.
- **4.** Run the invoicing utilities from the new subdirectories.



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 *lconfig/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 homelbin

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



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 <code>pin_invoice_data_map</code> file is in the <code>BRM_homelsys/data/config</code> directory. If you copy the <code>pin_invoice_data_map</code> file to the same directory from which you run the <code>load_pin_invoice_data_map</code> 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.



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 homelbin

Syntax

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.



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_acctsutility 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 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_acctsutility 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.



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 15-1.

Payment Method	ID
undefined	0
prepaid	10000



Payment Method	ID
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

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



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 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 <code>pin_inv_accts</code> 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.



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 Business Intelligence Publisher (BI Publisher) Scheduler database.

After *linvoice* objects are created by running the **pin_inv_accts** utility, run the **pin_inv_doc_gen** utility on the BI 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 BI Publisher Invoice Documents in BI Publisher".

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 BI Publisher, see "Designing and Generating Invoices in Oracle Business Intelligence Publisher".

Note:

- Before running the pin_inv_doc_gen utility, save the BRM_homelapps/ 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.
- BI Publisher supports multischema systems.

For more information, see "Configuring the Infranet.properties File for pin_inv_doc_gen with BI Publisher".

Location

BRM homelapps/pin inv doc gen

Syntax

Parameters

You must specify at least one parameter. 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 BI Publisher.

-accts list InvoiceList.xml

InvoiceList.xml specifies the list of all the account and bill unit pairs for which an *linvoice* object is generated and stored in the BRM database, and for which an invoice document is not yet generated.

-tvpe

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 *I* invoice objects of its child accounts that have nonpaying bill units (if they have a separate *I* invoice POID) are selected for generating the invoice document. See "Invoicing for Hierarchical Bill Units". However, the *I* 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.

-schema Schema Number

(Valid only for BI Publisher 11g and 12c) Generates the BRM schema number, in the format 0.0.0.n, for processing BRM invoices. For example,

-schema 0.0.0.2

processes the BRM invoices in 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-BI Publisher Invoicing Integration to Support Multischema Systems".

-status

Specifies the status of the *linvoice* 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 with BI Publisher".

-start startDate

-end endDate

Generates invoice documents for accounts or bill units whose *linvoice* 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 *linvoice* objects created until 23:59:59 of the end date are selected by the search query.

Specify the date in *MM/DD/YY*YY format.

-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.

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.



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 homelbin

Syntax

```
pin_inv_export [-start startDate] [-end endDate] [ -trial | -detail
filename] [-verbose] [-help]
```

Parameters

-start startDate

-end endDate

Retrieves 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 (/invoice/trial).



If a custom **/invoice** storable subclass exits, invoices of type **/invoice/.../trial** are exported.



-detail filename

Exports detailed invoices for the bills listed in the specified file. 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:

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.



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_homelapps/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".

Location

BRM_homelbin



Syntax

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 15-2.

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



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 15-3.

Format	ID
HTML	1
DOC1	3

-V

Displays information about successful or failed processing as the utility runs.



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 *l* **config/invoice_events** object.

Location

BRM homelbin

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.

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 Iconfig 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 homelbin

Syntax



Parameters

-brand "0.0.0.1/account 1"

Specifies the root POID.



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".



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 *Iconfig/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 *lconfig/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.

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 *Ibillinfo* objects with an empty PIN_FLD_ASSOC_BUS_PROFILE_OBJ_LIST field and populates the field with an *Iassociated_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-BI Publisher integration framework. This utility fails if you attempt to run
 it when the BRM-BI Publisher integration framework is disabled. See "Enabling the BRMBI Publisher Integration".

Note:

The pin_upd_assoc_bus_profile utility needs the configuration file in *BRM_homel* 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 homelbin

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 *Ibillinfo* 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
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.

