Oracle® Communications
Billing and Revenue Management
Designing and Generating Invoices
Release 7.4
E25090-03

February 2014
## Preface

- Audience ......................................................................................................................................................... ix
- Downloading Oracle Communications Documentation ......................................................................................... ix
- Documentation Accessibility .................................................................................................................................. ix
- Document Revision History .................................................................................................................................... ix

## 1 Designing and Generating Invoices

### About Invoices .................................................................................................................................................. 1-1
- BRM Invoice Features ......................................................................................................................................... 1-2
- About the Invoicing Process ................................................................................................................................. 1-2
- About Invoicing for Hierarchical Account Groups ............................................................................................. 1-3
  - About Viewing Invoices for Subordinate Accounts ......................................................................................... 1-3
- About Brand-Specific Invoices ............................................................................................................................. 1-4
- About Localized Invoices ..................................................................................................................................... 1-4

### Defining the Invoice Type .................................................................................................................................. 1-4

### Defining Multiple Invoices for One Customer .................................................................................................... 1-5

### Designing Invoice Templates .............................................................................................................................. 1-5
- Using XSLT Invoice Templates ............................................................................................................................... 1-6
  - Turning On XSLT Style Sheet Processing ........................................................................................................ 1-6
- Using HTML Invoice Templates .......................................................................................................................... 1-6
- Loading Invoice Templates ................................................................................................................................... 1-7
- Disabling the Default HTML Template ............................................................................................................. 1-8
- Switching Between XSLT and HTML Templates ................................................................................................. 1-8
- How Invoices are Formatted ................................................................................................................................ 1-8

### Generating Invoices ......................................................................................................................................... 1-9
- Generating Invoices Automatically .......................................................................................................................... 1-9
- Generating Invoices Manually ................................................................................................................................. 1-10
- Generating Detailed Invoices from an External File ............................................................................................. 1-11
- Generating Invoices for Non-Invoice Payment Methods ..................................................................................... 1-11
- Improving Performance When Generating Invoices .......................................................................................... 1-11
  - Specifying Event Fields to Cache for Invoicing .............................................................................................. 1-11
  - Decoding Cached Event Data for Invoicing ....................................................................................................... 1-12
- How Invoices are Generated ................................................................................................................................. 1-14
  - How PCM_OP_INV_MAKE_INVOICE Works ................................................................................................. 1-14

### Customizing Invoice Search Operations .......................................................................................................... 1-15
Improving Performance by Removing Invoice Details You Do Not Need .............................................. 1-18

Customizing the Information Included in Invoices .............................................................................. 1-19
- Customizing the Layout of Bill Items in Invoices .............................................................................. 1-19
- Creating Service-centric Invoices ...................................................................................................... 1-20
- Including Payment, A/R, and Taxation Details in Invoices ................................................................. 1-20
  - Including Invoice Event Details ......................................................................................................... 1-20
  - Including the Time Zone in Invoices ................................................................................................. 1-22
  - Including Shadow Event Adjustment Details in Invoices ............................................................... 1-23
- Including Soft Descriptors in Invoices ................................................................................................. 1-23
  - Customizing the Policy Source File for Soft Descriptors ................................................................. 1-24
- Including Custom Data in Invoices ...................................................................................................... 1-25
- Including Late Payment Fees in Invoices ............................................................................................ 1-25
- Including Finance Charges in Invoices ............................................................................................... 1-26
- Aggregating Taxes on Invoices ........................................................................................................... 1-27

Specifying Invoice Data from Pipeline Manager and Custom Applications ........................................... 1-28
- Using Data Map Templates .................................................................................................................. 1-28
- Loading the Invoice Data Map Templates ........................................................................................... 1-29
- Enabling Event Caching ...................................................................................................................... 1-30
- Adding Invoice Data to Pipeline Output .............................................................................................. 1-30

About Formats for Storing Invoices ...................................................................................................... 1-30

Storing Invoices in a Separate Database ................................................................................................ 1-31
- Installing the Invoice Data Manager .................................................................................................... 1-31
- Configuring BRM to Use a Separate Invoice Database ....................................................................... 1-32
- Configuring Invoice Applications to Use a Separate Invoice Database ............................................. 1-33
- Starting the Invoice Data Manager ..................................................................................................... 1-33
- Configuring pin.conf for Multiple Databases ...................................................................................... 1-33

Displaying Invoices ............................................................................................................................... 1-34
- Using Web Pages .................................................................................................................................. 1-34
- Using the XML Sample Invoice Viewing Utility .................................................................................. 1-34
- Customizing the Format for Online Invoices ....................................................................................... 1-34
  - Customizing the Format for Printed Invoices ................................................................................... 1-35
  - Customizing the Format for HTML Invoices .................................................................................... 1-36
  - Customizing the Format for XML Invoices ....................................................................................... 1-36
  - Customizing the Invoice Format by Using an XSL Style Sheet ......................................................... 1-36
  - Customizing the Format for DOC1 Invoices ..................................................................................... 1-36
  - Displaying an Invoice on Demand .................................................................................................... 1-36
- Understanding Invoice Layout ............................................................................................................. 1-37
  - Standard Details ................................................................................................................................. 1-37
  - Adjustment, Payment, Promotion, and Other Charges and Credits Details .................................... 1-38
  - Charge Details .................................................................................................................................. 1-39

Sending Invoices to Customers .............................................................................................................. 1-41
- Setting the Maximum Invoice Size for Email ......................................................................................... 1-42
- Sending Child Account Invoices to the Parent Account’s Email Address ............................................ 1-42
- Configuring the Email Data Manager for Printing ................................................................................ 1-42
2 Designing and Generating Invoices in Oracle Business Intelligence Publisher 11g

About BRM Integration with BI Publisher for Invoicing .................................................... 2-1
Overview of BRM and BI Publisher Invoice Generation ..................................................... 2-2
About the Invoicing Process ............................................................................................... 2-3
About Associating Bill Units with a BI Publisher Invoice and Report ................................ 2-3
About BRM-Business Intelligence Publisher Invoicing Integration Package Components.... 2-4
Designing Invoices ........................................................................................................... 2-4
About Invoice Layout Templates for Invoice Document Generation ............................... 2-5
Default Templates for Regular Invoices ........................................................................... 2-5
Report File for Default Invoice Templates ........................................................................ 2-5
About BI Publisher Reports for Invoice Document Generation ......................................... 2-5
Setting Up Your System to Generate Invoice Documents Using BI Publisher ................ 2-5
Specifying BI Publisher Invoice Report and Template Names in BRM ............................. 2-6
Enabling the BRM-BI Publisher Integration ...................................................................... 2-6
Creating Bill Unit Validation Templates for Invoices ......................................................... 2-7
Specifying the BI Publisher Report and Template .............................................................. 2-9
Loading Invoicing Business Profiles into the Database ....................................................... 2-9
Associating Pre-Existing Bill Units with Business Profiles ............................................... 2-10
About the /associated_bus_profile Object ..................................................................... 2-11
Creating /associated_bus_profile Objects ......................................................................... 2-11
Viewing Invoices Individually on the BI Publisher Console ............................................... 2-11
Creating /associated_bus_profile Objects for Accounts in a Branded Environment ..... 2-11
Creating Reports for Accounts in a Branded Environment in BI Publisher ....................... 2-12
Installing the BRM-BI Publisher Invoicing Integration Package ....................................... 2-13
Pre-Installation Requirements ......................................................................................... 2-13
Setting the PORTAL_JRE Environment Variable .............................................................. 2-14
Removing Older BRM-BI Publisher 11g Invoicing Integration Layout Template Files ...... 2-14
Enabling the BRM-BI Publisher Invoicing Integration Package Application ........................................... 2-15
Installed Files ........................................................................................................................................... 2-15
Upgrading Existing Reports from BI Publisher 10g to BI Publisher 11g ..................................................... 2-16
Configuring the pin_inv_doc_gen Utility .................................................................................................. 2-17
Setting the JAVA_HOME Environment Variable .......................................................................................... 2-17
Configuring the Infranet.properties File for pin_inv_doc_gen with BI Publisher 11g ................................. 2-18
Configuring the docgen.sh File .................................................................................................................. 2-19
Changing Command-Line Parameters for pin_inv_doc_gen ...................................................................... 2-20
Setting Scheduler Configuration ................................................................................................................ 2-20
Configuring the Email Server ................................................................................................................... 2-20
Configuring HTTPS for Secure Communication ....................................................................................... 2-20
Creating the XMLP_INVOICE_JOB Table in the Scheduler Database ..................................................... 2-21
Configuring the BRM-BI Publisher Invoicing Integration to Support Multischema Systems ........................ 2-22
Loading Stored Procedures ....................................................................................................................... 2-22
Creating New JDBC Data Sources ............................................................................................................ 2-22
Creating Data Source for Each BRM Schema ............................................................................................ 2-22
Setting Up the Data Model Configuration for Multischema ...................................................................... 2-23
Generating BI Publisher Invoice Documents in BI Publisher 11g ................................................................ 2-23
Generating BI Publisher Invoice Documents for a List of Accounts .......................................................... 2-24
Generating BI Publisher Invoice Documents Based on the Account Type ................................................ 2-25
Generating Duplicate Invoice Documents with BI Publisher 11g ............................................................. 2-25
Generating Invoice Documents for Non-Invoice Accounts ....................................................................... 2-26
Storing the Invoice Document .................................................................................................................. 2-26
Sending Invoice Documents to the Customers .......................................................................................... 2-26
Customizing Invoices .................................................................................................................................. 2-26
Customizing Invoice Layout Templates ..................................................................................................... 2-26
Updating the Invoice Layout Template File .............................................................................................. 2-27
Adding a New Invoice Layout Template .................................................................................................. 2-27
Enriching Invoice Data ............................................................................................................................. 2-27
Specifying the Default Format in Which to Store Invoices in BRM ............................................................. 2-28

3 Designing and Generating Invoices in Oracle Business Intelligence Publisher 10g

About BRM Integration with BI Publisher for Invoicing ........................................................................... 3-1
Overview of BRM and BI Publisher Invoice Generation ............................................................................ 3-1
About the Invoicing Process ........................................................................................................................ 3-2
About Associating Bill Units with a BI Publisher Invoice and Report ........................................................... 3-3
About BRM-Business Intelligence Publisher Invoicing Integration Package Components........................ 3-4
Designing Invoices ....................................................................................................................................... 3-4
About Invoice Layout Templates for Invoice Document Generation .......................................................... 3-4
Default Templates for Regular Invoices .................................................................................................... 3-5
Report Files for Default Invoice Templates .............................................................................................. 3-5
About BI Publisher Reports for Invoice Document Generation .................................................................. 3-5
Setting Up Your System to Generate Invoice Documents Using BI Publisher ......................................... 3-5
Specifying BI Publisher Invoice Report and Template Names in BRM ..................................................... 3-6
Enabling the BRM-BI Publisher Integration ............................................................................................... 3-6
Generating BI Publisher Invoice Documents ........................................................................... 3-22
Configuring the BRM-BI Publisher Invoicing Integration to Support Multischema Systems .... 3-12
Installing BRM-BI Publisher Invoicing Integration Package ..................................................... 3-13
Windows Installation ..................................................................................................................... 3-13
Pre-Installation Requirements ........................................................................................................ 3-13
Setting the BI Publisher System Environment Variable ................................................................. 3-13
Installing the BRM-BI Publisher Invoicing Integration Package Application ................................. 3-13
Setting the LD_LIBRARY_PATH Environment Variable ................................................................. 3-14
Linux Installation .......................................................................................................................... 3-14
Pre-Installation Requirements ........................................................................................................ 3-14
Enabling the BRM-BI Publisher Invoicing Integration Package Application .................................. 3-14
Setting the LD_LIBRARY_PATH Environment Variable ................................................................. 3-15
Installed Files .................................................................................................................................. 3-15
Configuring the pin_inv_doc_gen Utility ....................................................................................... 3-16
Configuring the Infranet.properties File for pin_inv_doc_gen ....................................................... 3-16
Specifying the Utility Output Format ............................................................................................... 3-17
Configuring the BipEnv.bat File ...................................................................................................... 3-18
Changing Command-Line Parameters for pin_inv_doc_gen ......................................................... 3-18
Setting Scheduler Configuration ..................................................................................................... 3-18
Configuring the Email Server ......................................................................................................... 3-18
Creating the XMLP_INVOICE_JOB Table in the Scheduler Database ............................................ 3-19
Deploying the BRM Web Service .................................................................................................... 3-19
Creating Shared Library in OC4J Server ......................................................................................... 3-21
Testing the Newly Deployed Web Service ....................................................................................... 3-21
Configuring How PublicBRMfetchInvoiceXMLService Connects to the BRM Database ............. 3-22
Configuring the BRM-BI Publisher Invoicing Integration to Support Multischema Systems ....... 3-22
Configuring the Connection Pool and the Data Source ................................................................. 3-22
Adding JNDI Entries in the Infranet.properties File of pin_inv_doc_gen ....................................... 3-24
Generating BI Publisher Invoice Documents .................................................................................. 3-24
Generating BI Publisher Invoice Documents for a List of Accounts ........................................... 3-25
Generating BI Publisher Invoice Documents Based on the Account Type ..................................... 3-26
Generating Duplicate BI Publisher Invoice Documents .................................................................. 3-26
Regenerating BI Publisher Invoice Documents ............................................................................. 3-27
Storing the Invoice Document ....................................................................................................... 3-28
Sending Invoice Documents to the Customers ............................................................................. 3-28
Customizing Invoices .................................................................................................................... 3-28
Customizing Invoice Layout Templates .......................................................................................... 3-28
Updating the Invoice Layout Template File .................................................................................... 3-28
Adding a New Invoice Layout Template .......................................................................................... 3-28
Enriching Invoice Data .................................................................................................................. 3-29
4 Invoice Utilities

load_pin_invoice_data_map ................................................................. 4-2
pin_inv_accts .................................................................................. 4-4
pin_inv_doc_gen ............................................................................. 4-8
pin_inv_export ............................................................................... 4-11
pin_inv_send ................................................................................. 4-13
pin_load_invoice_events ................................................................. 4-16
pin_load_invoice_template .............................................................. 4-18
pin_upd_assoc_bus_profile .............................................................. 4-20
Preface

This book 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 document is intended for system administrators and developers involved in invoicing.

Downloading Oracle Communications Documentation

Product documentation is located on Oracle Technology Network:

http://docs.oracle.com

Additional Oracle Communications documentation is available from Oracle E-Delivery:

https://edelivery.oracle.com

Documentation Accessibility

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


Access to Oracle Support

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

Document Revision History

The following table lists the revision history for this book.

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E25090-01</td>
<td>March 2013</td>
<td>Initial release.</td>
</tr>
<tr>
<td>Version</td>
<td>Date</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| E25090-02   | July 2013  | Documentation updates for BRM 7.4 Patch Set 18.  
  ■ Made minor formatting and text changes.  
  ■ Added "Designing and Generating Invoices in Oracle Business Intelligence Publisher 11g" chapter.  
  ■ Updated the "pin_inv_doc_gen" utility for BI Publisher 11g. |
| E25090-03   | February 2014 | Documentation updates for BRM 7.4 Patch Set 20.  
  ■ Made minor formatting and text changes.  
  ■ Updated the following:  
    "Designing and Generating Invoices in Oracle Business Intelligence Publisher 11g" chapter.  
    "Invoice Utilities" chapter. |
Designing and Generating Invoices

This chapter describes how to design and generate Oracle Communications Billing and Revenue Management (BRM) invoices and how to send them to customers.

About Invoices

You can use BRM to generate invoices and provide them to customers. An invoice lists the balance information for a customer’s bill. When you use Customer Center to display the invoices, the invoices show the total balance in both primary and secondary currencies (if two currencies are configured with appropriate exchange rates), but show only the primary currency for individual events.

When billing runs, a bill is produced for every bill unit in an account and invoices are generated for each bill. There are two invoice types provided: detailed and summary.

■ A detailed invoice lists the bill items for the bill unit and the events that have currency balance impacts greater than zero. The detailed invoice mode is the default.

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

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

For accounts in a hierarchical group, the billing data for each subordinate bill unit in a child account is rolled up to the invoice for the parent A/R account. This invoice is referred to as a consolidated invoice because it contains the billing data for the subordinate bill units of each child account in addition to the billing data for the accounts receivable (A/R) account’s bill unit. For information about hierarchical account groups, see “Creating and Managing Hierarchical Account Groups” in BRM Managing Accounts Receivable.

Note: If you use detailed invoices, generating the consolidated invoice can be system intensive, and the consolidated invoice can be extremely large. Therefore, you can set invoicing thresholds to control the size of the consolidated invoice and the subordinate data contained in it. For more information, see “About Invoicing for Hierarchical Account Groups”.

BRM Invoice Features

BRM invoices present a set of data related to service usage, charges, discounts, promotions, taxes, and surcharges.

Important features of BRM invoices are:

- Supports customization. You can customize the information included in an invoice in several ways. See "Customizing the Information Included in Invoices".
- Supports brand-specific invoices. See "About Brand-Specific Invoices".
- Supports localized invoices. See "About Localized Invoices".
- Supports multiple invoices for one customer. See "Defining Multiple Invoices for One Customer".
- Supports different ways to send the invoice to customers. See "Sending Invoices to Customers".
- Includes the following details:
  - 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.
  - Wireless service usage details (if the customer uses wireless service)
  - Past due amount details from the past bill
  - Usage details of accounts with multiple phone numbers
  - Hierarchical account details
  - Promotion names
  - A/R actions having currency or non-currency impact, such as adjustment, dispute, settlement, and write-off details

If you use the BRM-Business Intelligence Publisher Integration framework to generate invoice documents, you can add marketing messages, customer information from a customer relationship management (CRM) system, and messages from accounting department to the invoice. See "Designing and Generating Invoices in Oracle Business Intelligence Publisher 10g".

For a sample invoice, see "Understanding Invoice Layout".

About the Invoicing Process

BRM performs the following operations to generate invoices:

1. When you run the daily billing script, BRM finds accounts at the end of their billing cycle. For a description of this script, see "About Billing Your Customers" in BRM Configuring and Running Billing.

2. 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.

3. BRM generates an invoice for each bill. See "Generating Invoices".

4. BRM stores all invoices in either the BRM database or a separate database you set up. For more information, see "Storing Invoices in a Separate Database".
After an invoice is generated:

- You can display the invoice in Customer Center and Self-Care Manager. The format of the invoice is based on a template you create and load. See "Designing Invoice Templates".
- You can send the invoice to your customer. See "Sending Invoices to Customers".

For information about generating invoices, see "Generating Invoices".

You can also customize, enrich, and present BRM invoices by using the capabilities of Oracle Business Intelligence Publisher. For information about generating invoice documents by integrating BRM and Business Intelligence Publisher capabilities, see "Designing and Generating Invoices in Oracle Business Intelligence Publisher 10g".

**About Invoicing for Hierarchical Account Groups**

When the pin_inv_accts utility processes bills for parent accounts in an account hierarchy, it searches the BRM database for the subordinate bill units and generates their invoices automatically.

You can set threshold values to ensure system performance is not decreased while processing invoices for large hierarchical account groups. The threshold values set the maximum number of child accounts containing subordinate bill units that can be rolled up to the parent A/R account 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 parent bill unit.

If the number of subordinate bill units is less than or equal to the threshold, the parent account’s invoice includes subordinate bill unit information.

**Important:** If bill suppression is enabled on a parent (A/R) account and the subordinate account threshold is exceeded, invoicing fails. This occurs because subordinate bill units, which use the bill number of their parent account, are generated even when billing for the parent account is suppressed. In such cases, they will not contain bill numbers, which invoices require. To exclude invoicing for these bills, run pin_inv_accts with the -skip_blank_billnos parameter. (Subordinate bill units contain bill numbers only when their parent accounts are billed.)

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

**About Viewing Invoices for Subordinate Accounts**

If you open a subordinate account in Customer Center and run Bill Now for the account, the account’s bill and invoice can be viewed only in the parent account’s information page.
When `pin_inv_accts` is run, it generates invoices for subordinate accounts. You can view these invoices in Customer Center and Self-Care Manager when viewing the subordinate account’s information.

In Self-Care Manager, invoices can be viewed only for the account that is logged in. To view an invoice for a subordinate account, the subordinate account must be logged in.

**About Brand-Specific Invoices**

If you support multiple brands, you can use a separate invoice design for each brand. Create a separate template for each brand and then specify the brand when loading each template. See "Designing Invoice Templates" and "Loading Invoice Templates".

If a brand does not have a template associated with it, BRM uses the system template.

You can also define a different list of event types for each brand’s invoices with the `pin_load_invoice_events` utility. See "pin_load_invoice_events".

**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.

**Defining the Invoice Type**

The PIN_FLD_INV_TYPE field in the `/payinfo` object defines which type of invoice to generate for a given bill unit, detailed (0 or NULL) or summary (1). By default, BRM generates detailed invoices for bills that use the Invoice payment method.

To generate summary invoices, change the value of the PIN_FLD_INV_TYPE field in the `/payinfo` object to 1.

- To set this value when creating a customer account, pass it in the input flist of the PCM_OP_CUST_COMMIT_CUSTOMER opcode.
- To set this value when adding or changing a payment method, pass it in the input flist of the PCM_OP_CUST_SET_PAYINFO opcode.

When this value is set, the PCM_OP_INV_MAKE_INVOICE opcode uses it in conjunction with the position of the bill, in relation to other bills, to set the PIN_FLD_INV_FLAGS value in the `/invoice` object. Possible values are listed in Table 1–1:

<table>
<thead>
<tr>
<th>Type of Invoice</th>
<th>PIN_FLD_INV_FLAGS Hex Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary invoice for a non-hierarchical account.</td>
<td>0x0005</td>
</tr>
</tbody>
</table>
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 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. For more information, see Customer Center Help.

- Use a custom CRM system to pass the invoice name in the input flist of PCM_OP_CUST_SET_PAYINFO. Use the PIN_FLD_NAME field in the PIN_FLD_PAYINFO array.

Designing Invoice Templates

The invoice’s format is based on a template you create and load. BRM uses the template to display invoices in Customer Center and customer self-care Web pages.
You can use one invoice template for all accounts, or, if you support multiple brands, you can use a separate template for each brand.

By default, BRM uses a simple XSLT template (BRM_Home/sys/data/config/stylesheets/sample1.xsl). 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. See "Using XSLT Invoice Templates".
- **HTML files**: You can create your own HTML template or modify an HTML template that BRM provides. See "Using HTML Invoice Templates".

After you create a template, you load it into BRM. See "Loading Invoice Templates".

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".

### Using XSLT Invoice Templates

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

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

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

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

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

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

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

### 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
```

### Using HTML Invoice Templates

You design HTML invoice templates by using an HTML editor or other application. You can use the HTML template provided with BRM as a starting point (BRM_Home/sys/data/config/pin_invoice_template.html).
An HTML template needs to include these variables, which you will find 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 "Customizing the Format for HTML Invoices".

After you create an HTML template, you load it into the BRM database by using `pin_load_invoice_template`. See "Loading Invoice Templates".

**Loading Invoice Templates**

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

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 "Creating Configuration Files for BRM Utilities" in BRM System Administrator’s Guide.

2. To load an XSLT style sheet, enter:

   `pin_load_invoice_template -brand "brand\_POID" -type text/html -locale locale\_name -template filename -usexsl`

   where:
   
   - `brand\_POID` is the POID of a brand account or the root account. Use a brand account POID to assign the template to a specific brand. Use the root account POID if your system does not use brands or if you want the template to be available to multiple brands.
   
   - `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 "brand\_POID" -type HTML -locale locale\_name -template filename`
Designing Invoice Templates

where:

- **brand_POID** is the POID of a brand account or the root account. Use a brand account POID to assign the template to a specific brand. Use the root account POID if your system does not use brands or if you want the template to be available to multiple brands.

- **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
```

Disabling the Default HTML Template

If you load brand-specific templates into the BRM database, you need to disable the system template that BRM uses by default:

1. Open the Connection Manager (CM) configuration file (`BRM_Home/sys/cm/pin.conf`).
2. Disable the default template that belongs to the root account by entering a crosshatch (#) at the beginning of this entry:
   ```
   - fm_inv_pol html_template $DM{'db_num'} /config/invoice_templates 101
   ```
3. Save and close the file.
4. Stop and restart the CM. See "Starting and Stopping the BRM System" in BRM System Administrator’s Guide.

For information about configuration files, see "Using Configuration Files to Connect and Configure Components" in BRM System Administrator’s Guide.

Switching Between XSLT and HTML Templates

You can switch between using an XSLT template and using an HTML template without loading a new template. You do this for a specific brand or for root.

To switch to an XSLT template, enter:

```
pin_load_invoice_template -brand "brand_POID" -usexsl
```

To switch to an HTML template, enter:

```
pin_load_invoice_template -brand "brand_POID"
```

where **brand_POID** is the POID of a brand account or the root account.

How Invoices are Formatted

To format invoices, use the PCM_OP_INV_FORMAT_INVOICE opcode. This opcode performs XSL transformation on an invoice.

This opcode is called by the PCM_OP_INV_POL_FORMAT_INVOICE_XSLT policy opcode to apply an XSL style sheet to an invoice. It receives as input an XML
formatted invoice and an XSL style sheet. It applies the style sheet to the invoice and returns the formatted invoice.

---

**Note:** The XSL transformation is done in the Java Server with the help of XSLT software.

---

PCM_OP_INV_FORMAT_INVOICE uses the POID of the /bill object or /invoice object to locate and retrieve a specific invoice. Specify the output format of the invoice as a mime type in the PIN_FLD_TYPE_STR field in the input flist.

PCM_OP_INV_FORMAT_INVOICE performs the following tasks:

- Checks the /invoice object to see if the format specified in the PIN_FLD_TYPE_STR field is stored in the object. Invoices can be stored in the database as pin_flist, XML, HTML, or DOC1 format.
- If the specified format is stored, the invoice is retrieved and returned in the format specified.
- If the specified format is not stored, the PCM_OP_INV_POL_FORMAT_VIEW_INVOICE policy opcode is called to attempt to format the invoice. Invoices can use HTML, XML, or DOC1 format. To apply an XSL style sheet to the invoice, specify the XSL mime type in the PIN_FLD_TYPE_STR field.

If successful, PCM_OP_INV_FORMAT_INVOICE returns a buffer containing the formatted invoice. The PIN_FLD_RESULT field in the output flist is set to 1.

If unsuccessful, no invoice is returned. The PIN_FLD_RESULT field is set to 0.

### Generating Invoices

Use the **pin_inv_accts** utility to generate invoices either automatically as part of daily billing or separately.

For information on how to generate invoices during trial billing, see "About Trial Billing" in *BRM Configuring and Running Billing*.

### 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 "About Running the Billing Scripts" in *BRM Configuring and Running Billing*.

**pin_inv_accts** runs twice. In the first run, it performs the following tasks to handle hierarchy groups:

1. Searches for all bills that have reached the end of their billing cycle and for which invoices have not yet been generated.

2. Checks the subordinate threshold values in the /config/business_params object to determine the maximum number of child accounts having subordinate bill units that are allowed in the hierarchy group. See "About Invoicing for Hierarchical Account Groups".

3. For each /bill object retrieved, checks the PIN_FLD_AR_HIERARCHY_SIZE value to determine if it exceeds the threshold value.

4. If the threshold is exceeded, uses multiple threads to retrieve the subordinate bill units and to generate an invoice for each one.
If no errors occurred, `pin_inv_accts` runs a second time to generate invoices for parent accounts in hierarchy groups and for non-hierarchical accounts:

5. Searches for all bills that have reached the end of their billing cycle and for which invoices have not yet been generated.

6. For each `/bill` object retrieved, generates an invoice.

If you configured BRM to use a separate invoice database, `pin_inv_accts` uses `dm_invoice` to store invoices in a separate database. See "Storing Invoices in a Separate Database".

Generating Invoices Manually

To generate invoices manually:

1. Run the other billing utilities.
   
   For the names of the other utilities and the order in which to run them, see "Running Billing Utilities Manually" in *BRM Configuring and Running Billing*.

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


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.

   **Important:** If bill suppression is enabled on a parent (A/R) account and the subordinate account threshold is exceeded, invoicing fails. This occurs because subordinate bill units, which use the bill number of their parent account, are generated even when billing for the parent account is suppressed. In such cases, they will not contain bill numbers, which invoices require. To exclude invoicing for these bills, run `pin_inv_accts` with the `-skip_blank_billnos` parameter. (Subordinate bill units contain bill numbers only when their parent accounts are billed.) For information on invoicing for account hierarchies, see "About Invoicing for Hierarchical Account Groups".

For more information on the syntax and parameters for `pin_inv_accts`, see "`pin_inv_accts`".
Generating Invoices

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 an account hierarchy, it searches BRM for the subordinate bill units and generates their invoices automatically. In general, the file should not contain the bills for subordinate bill units; however, if they are present, invoices for the subordinate 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 "Understanding Payment Methods" in *BRM Managing Customers*.

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

```
pin_inv_accts -pay_type 10003
```

**Note:** This generates detailed invoices. To generate summary invoices, customize the PCM_OP_INV_POL_SELECT policy opcode to pass the summary value in the PIN_FLD_FLAGS field on its input flist (PIN_FLD_FLAGS = 0x0002). You can also run `pin_inv_accts` with `-summary` parameter to generate summary invoices.

---

Improving Performance When Generating Invoices

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_Home/sys/cm/pin.conf).
2. Change the value of the inv_item_fetch_size entry.
   The default is 10000.
3. Change the value of the inv_event_fetch_size entry.
   The default is 10000.
4. Stop and restart the CM. See "Starting and Stopping the BRM System" in BRM System Administrator’s Guide.

To customize the search operation (for example, to search for specific items or to omit a step), see "Customizing Invoice Search Operations".

### Specifying Event Fields to Cache for Invoicing

The PIN_FLD_INVOICE_DATA field in the /event object contains a cache of all fields that need to be handled in invoicing. Use the PCM_OP_ACT_POL_SPEC_EVENT_CACHE policy opcode to define which balance impact fields to cache for invoicing.

You can improve performance by limiting the amount of information cached. However, if you need the information, it is quicker to cache a field than have it read from the event table.

By default, the policy opcode caches the following PIN_FLD_BAL_IMPACTS array fields shown in Table 1–2 in the base table field PIN_FLD_INVOICE_DATA.

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIN_FLD_AMOUNT</td>
<td>The account balance impact. The value can be positive or negative.</td>
</tr>
<tr>
<td>PIN_FLD_DISCOUNT</td>
<td>The discount applied to the balance impact.</td>
</tr>
<tr>
<td>PIN_FLD_IMPACT_TYPE</td>
<td>Balance impact type - rated by BRM rated-engine (0x1), pre-rated (0x2), taxed (0x4), purchase order (0x8), re-rated(0x20), and reverse_rated(0x40).</td>
</tr>
<tr>
<td>PIN_FLD_ITEM_OBJ</td>
<td>Link to the item object affected by this event. Applies only to the balance array element that impacts currency resources. (This may be different from the PIN_FLD_ITEM_OBJ field in the base /event object.)</td>
</tr>
</tbody>
</table>

Note: The PIN_FLD_INVOICE_DATA field is limited to 4000 bytes. If the event cache size of the PIN_FLD_INVOICE_DATA field is greater than 4000 bytes, it is ignored and the invoice displays a 0 amount. For Oracle databases, you can increase the size of the invoice_data column to work around this limitation.
Designing and Generating Invoices

If you remove these fields from the PCM_OP_ACT_POL_SPEC_EVENT_CACHE policy opcode and leave the event cache turned on, there will be no event details in the invoices.

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIN_FLD_QUANTITY</td>
<td>The quantity applied; the number of units that were actually applied using this rate.</td>
</tr>
<tr>
<td>PIN_FLD_RATE_TAG</td>
<td>Description of the rate used. Same as PIN_FLD_DESCR in the rate object.</td>
</tr>
<tr>
<td>PIN_FLD_RESOURCE_ID</td>
<td>Numeric value of the resource that is impacted.</td>
</tr>
<tr>
<td>PIN_FLD_TAX_CODE</td>
<td>Tax code for the rate used. When taxes do not apply, this field is set to 0.</td>
</tr>
</tbody>
</table>

If you remove these fields from the PCM_OP_ACT_POL_SPEC_EVENT_CACHE policy opcode and leave the event cache turned on, there will be no event details in the invoices.

---

**Note:** If you do not use event caching, return a NULL pointer to the caller of the PCM_OP_ACT_POL_SPEC_EVENT_CACHE policy opcode.

---

You can customize the PCM_OP_ACT_POL_SPEC_EVENT_CACHE policy opcode to cache additional balance impact array fields by modifying the `BRM_Home/source/sys/fm_act_pol/fm_act_pol_spec_event_cache.c` file.

If you turn off caching in the CM configuration file, these fields are read directly from the event table, which slows performance.

---

**Important:**

- If you remove the default fields of the PIN_FLD_BAL_IMPACTS array from the PCM_OP_ACT_POL_SPEC_EVENT_CACHE policy opcode and leave the event cache turned on, there will be no event details in the invoices.
- The event cache must not exceed 4000 bytes. If you have a large number of elements in the PIN_FLD_BAL_IMPACTS array, you must disable the `event_cache` flag in the CM configuration file.

---

To enable or disable caching of the PIN_FLD_BAL_IMPACTS array, edit the CM configuration file (`BRM_Home/source/sys/cm/pin.conf`). The default is caching on. Any number except zero enables caching.

```bash
-fm_inv event_cache 1
```

**Decoding Cached Event Data for Invoicing**

The PCM_OP_INV_DECODE_INVOICE_DATA opcode parses the event data being retrieved for an invoice. This opcode retrieves the contents of the PIN_FLD_INVOICE_DATA field, parses the data, and returns the decoded data in the output flist.

The input flist contains the PIN_FLD_INVOICE_DATA field, which is a cached string that needs to be decoded. It is limited to 4000 bytes. If the cache size is greater than 4000 bytes, it is ignored.

The output flist contains the PIN_FLD_EXTENDED_INFO substruct, which contains the fields decoded from the PIN_FLD_INVOICE_DATA field.
Note: If you defined custom event caching for Pipeline Manager in the /config/event_data_map file, you must configure PCM_OP_INV_DECODE_INVOICE_DATA to decode the data.

For information on defining fields to cache for invoicing, see "Specifying Event Fields to Cache for Invoicing".

How Invoices are Generated

pin_inv_accts runs as part of your daily billing to create a regular invoice for each account that is billed on that day. See "pin_inv_accts".

pin_inv_accts runs twice to create these invoices: in the first run, invoices are generated for subordinate bill units in a hierarchy; in the second run, invoices are generated for parent accounts and all non-hierarchical accounts. In both runs, this utility calls the PCM_OP_INV_MAKE_INVOICE opcode to create an invoice. This is the first opcode that gets called to create an invoice for a designated /bill object.

How PCM_OP_INV_MAKE_INVOICE Works

PCM_OP_INV_MAKE_INVOICE takes the POID of the bill object and calls the PCM_OP_INV_POL_SELECT policy opcode to determine whether custom processing or default processing is used to retrieve the events and items for billing. If custom processing is used, control is handed to the PCM_OP_INV_POL_SELECT policy opcode, which returns the results after the search operations are complete.

If default processing is used, PCM_OP_INV_MAKE_INVOICE performs the following steps:

- Retrieves the invoicing threshold value in the /config/business_params object to determine whether invoices for subordinate bill units should be generated or consolidated into the parent account invoice. If generated, the parent account’s invoice will not contain the subordinate bill unit’s invoice data. See "About Invoicing for Hierarchical Account Groups".

- Retrieves all required information from the /bill, /account, /billinfo, and /payinfo objects.
  - It uses the PIN_FLD_INV_TYPE field value from the /payinfo object to determine whether to generate a detailed invoice or a summary invoice. If the value is 1, a summary invoice is created. If the value is 0 or NULL, a detailed invoice is created. See "About the Invoicing Process".
  - It uses the PIN_FLD_AR_HIERARCHY_SIZE value to determine if multithreaded processing should be used. This value defines the number of subordinate bill units for the parent account of the hierarchy group. If the PIN_FLD_AR_HIERARCHY_SIZE value for a bill exceeds the invoicing threshold, and the bill is for a parent A/R account, PCM_OP_INV_MAKE_INVOICE retrieves the subordinate bill units in that hierarchy by using multiple threads and processes them first.

Note: The -detail and -summary parameters of pin_inv_accts override the PIN_FLD_INV_TYPE field value in the /payinfo object.

- Retrieves all events belonging to the bill items based on the /config/invoice_events object for a particular brand.
Customizing Invoice Search Operations

You can customize the search templates during the invoicing operation so not all search operations are performed. To omit any of the search operations, or to modify the search for specific events and items, you use the PCM_OP_INV_POL_SELECT policy opcode, which is called by the PCM_OP_INV_MAKE_INVOICE opcode when generating invoices.

The PCM_OP_INV_POL_SELECT policy opcode enables you to create custom search templates to select which bill items and events are retrieved for invoices. This improves performance during invoicing because BRM performs only selective searches and does not have to process all accounts and events in the defined invoicing period.

BRM uses the PIN_FLD_BOOLEAN field in the output flist of this policy opcode to determine whether the invoices should be generated using the default item and event processing in PCM_OP_INV_MAKE_INVOICE or the custom item and event processing defined in the PCM_OP_INV_POL_SELECT policy opcode.

- PIN_BOOLEAN_FALSE indicates the PCM_OP_INV_POL_SELECT policy opcode is ignored and the output flist to PCM_OP_INV_MAKE_INVOICE is the same as the flist PCM_OP_INV_POL_SELECT received as input. This is the default.
- PIN_BOOLEAN_TRUE indicates the PCM_OP_INV_POL_SELECT policy opcode performs the processing. The output flist to PCM_OP_INV_MAKE_INVOICE contains the input flist and the results arrays generated by the custom processing. The output flist generally contains an array of items specified for the invoice, an array of events pertaining to each item returned, and an array of information retrieved from the event’s invoice data cache.

Important: Bill Now and on-demand billing will not generate invoices for subordinate bill units even when the threshold value is exceeded. The consolidated invoice for the parent (paying) bill unit always gets generated.
PCM_OP_INV_MAKE_INVOICE uses the return flist from the PCM_OP_INV_POL_SELECT policy opcode to format the invoice and store it in the database. If the PCM_OP_INV_POL_SELECT policy opcode was not used for processing, it returns the input flist without changes.

If errors occur during the invoicing operations, use the PCM_OP_INV_POL_POST_MAKE_INVOICE policy opcode to handle them. For example, you can customize this policy opcode to return the event that caused the invoice failure. When PCM_OP_INV_MAKE_INVOICE encounters errors returned by the PCM_OP_INV_POL_SELECT policy opcode, it sends the error buffer as a string to the PCM_OP_INV_POL_POST_MAKE_INVOICE policy opcode.

For information on changing the events displayed on an invoice, see "Including Payment, A/R, and Taxation Details in Invoices".

For information on handling errors, see "Configuring Error Checking for Customized Invoicing".

Creating Custom Search Templates

To create a custom search template for invoicing, use the PCM_OP_INV_POL_SELECT policy opcode. BRM uses the PIN_FLD_BOOLEAN value in the output flist of the PCM_OP_INV_POL_SELECT policy opcode to determine whether to use the default search functionality in PCM_OP_INV_MAKE_INVOICE or the customized search functionality in this policy opcode.

To customize the invoicing search operations, set up your policy opcode with the following information:

1. Specify the invoice template to use.

   **Note:** The invoice templates and style sheets that define the appearance of invoices are defined in the /config/invoice_templates object. You can define a different appearance for invoices. To load custom style sheets, run the "pin_load_invoice_template" utility.

2. Copy the input flist to the output flist.

3. Define the search criteria for the search template. For example, create the conditions under which A/R items and events are retrieved.

4. Search for the items defined in the search template.

5. Compile the results flist.

6. Append the retrieved items to the output flist.

7. Search for events corresponding to the retrieved items.

8. Append the retrieved events to the output flist.

9. Call the PCM_OP_INV_DECODE_INVOICE_DATA opcode to decode the PIN_FLD_INVOICE_DATA value of the /event object.

10. Send the result back to PCM_OP_INV_MAKE_INVOICE.
Customizing Invoice Search Operations

Example: Generating Invoices Based on Event Types

You can customize your database so that invoices contain a specific set of data based only on events you want. For example, you can generate invoices for your system that contain only adjustment and dispute information.

To do this:

1. In the PCM_OP_INV_POL_SELECT policy opcode, create the following search templates. In each template, customize the policy opcode to retrieve the items and events listed.

   **Template 1**
   - /item/adjustment
   - /event/billing/adjustment/event
   - /event/billing/adjustment/item
   - /event/billing/adjustment/tax_event

   **Template 2**
   - /item/dispute
   - /event/billing/dispute/event
   - /event/billing/dispute/item
   - /event/billing/dispute/tax_event

2. Append the search results from Template 1 and Template 2 to the output flist.

3. Customize the invoicing policy opcodes to display and print the A/R information retrieved by the search templates.

**Note:** The events that are recorded on an invoice are defined in the /config/invoice_events object. By default, this object includes A/R events, such as payments and refunds, as well as cycle and usage events. You can create a /config/invoice_events object for custom events so that the search operation retrieves them and the specified attributes get displayed on the invoice. To specify the events recorded on an invoice, run the pin_load_invoice_events utility.

Configuring Error Checking for Customized Invoicing

To perform error checking for customized invoicing, use the PCM_OP_INV_POL_POST_MAKE_INVOICE policy opcode. This policy opcode is called after the invoice commit transaction to troubleshoot the errors detected by PCM_OP_INV_MAKE_INVOICE. If the /invoice object POID value in the input flist of the PCM_OP_INV_POL_POST_MAKE_INVOICE policy opcode is type only, the invoice object was not created because errors occurred.
The PIN_FLD_ERROR_INFO array in the input and output flists contain information about the error.

You can configure the PCM_OP_INV_POL_POST_MAKE_INVOICE policy opcode to capture the event that caused the error and return it in the output flist.

For more information about error handling, see "Understanding API Error Handling and Logging" in BRM Developer’s Guide.

**Improving Performance by Removing Invoice Details You Do Not Need**

By default, BRM invoices include billing details, such as usage items, device details, and tax totals, which helps customers understand the charges on their invoices. If your company does not offer telephony services or if your company does not want to publish certain details on invoices, you can improve invoicing performance by configuring BRM to retrieve only the invoice details needed.

To do so, you use a CM configuration file entry to specify the details to exclude from your invoices:

- **Exclude device details from invoices.** Improves performance by configuring BRM to skip the device search and the totaling of resources per device.

  **Caution:** 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 plan, deal, and product pricing details from invoices.** Improves performance by configuring BRM to skip the retrieval of plan, deal, and product pricing details from the BRM database. When you exclude plan, deal, and product pricing details, invoices do not include plan 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 non-currency resources.

- **Consolidate the searches for real-time events and batch events into one search.** Improves performance by configuring BRM to retrieve BRM and pipeline event details from the BRM cache.

  **Note:** To consolidate the searches, the **event_cache** entry in the CM configuration file must be enabled. See "Enabling Event Caching".


---

1-18 BRM Designing and Generating Invoices
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:
   
   \[ \text{fm_inv} \text{ inv_perf_features FlagValue} \]
   
   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 excludes balance impact details from invoices.
   - 0x00000800 improves search performance.
   - 0x00001000 excludes device details from invoices.
   - 0x00002000 excludes tax totals from invoices.
   - 0x00004000 excludes plan, deal, product pricing details from invoices.
   - 0x00008000 excludes balance group details from invoices.
3. Save and close the file.
4. Stop and restart the CM. See "Starting and Stopping the BRM System" in BRM System Administrator’s Guide.

Customizing the Information Included in Invoices

You can customize the information included in an invoice in several ways:

- Customize the layout of bill items. See "Customizing the Layout of Bill Items in Invoices".
- Group items by service type. See "Creating Service-centric Invoices".
- Include customer payment, A/R, and taxation details. See "Including Payment, A/R, and Taxation Details in Invoices".
- Include the customer’s time zone. See "Including the Time Zone in Invoices".
- Include shadow events. See "Including Shadow Event Adjustment Details in Invoices".
- Include soft descriptors. See "Including Soft Descriptors in Invoices".
- Include custom data. See "Including Custom Data in Invoices".
- Include late payment fees. See "Including Late Payment Fees in Invoices".
- Include finance charges. See "Including Finance Charges in Invoices".
- Aggregate taxes by tax supplier. See "Aggregating Taxes on Invoices".

Customizing the Layout of Bill Items in Invoices

You can customize the layout of bill items on an invoice.

To list bill items in a different order, use the PCM_OP_INV_POL_PREP_INVOICE policy opcode.

To make other changes to the layout:
Customizing the Information Included in Invoices

- If you are using an HTML invoice template, use the PCM_OP_INV_POL_FORMAT_INVOICE_HTML policy opcode.
- If you are using an XSLT style sheet for the invoice template, you can perform this level of customization in the style sheet.

Creating Service-centric Invoices

You can create invoices that group items by service. To do this, you need to enable your invoices to show bill items grouped by service. By default, service-centric invoicing is disabled.

1. Open the CM configuration file (BRM_Home/sys/cm/pin.conf).
2. Do one of the following:
   - 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, set the value of service_centric_invoice to 0.
3. Save and close the file.
4. Stop and restart the CM. See "Starting and Stopping the BRM System" in BRM System Administrator’s Guide.

For information about configuration files, see "Using Configuration Files to Connect and Configure Components" in BRM System Administrator’s Guide.

---

Note: In the invoicing business profile, if the EnableInvoicingIntegration config business parameter is enabled, ensure that in BRM_Home/sys/cm/pin.conf, the service_centric_invoice entry is commented. By default, the service_centric_invoice entry is commented. If this entry is not commented, you can view the invoice document only in Customer Center and not in the BI Publisher client.

---

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

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

For information on customizing the events retrieved during invoicing, see "Customizing Invoice Search Operations".

Including Invoice Event Details

To include details in the invoice:

1. Open the events.file file (BRM_Home/sys/data/config/events.file).
2. Edit the file to add the event details you want to include in the invoice. See the list of events you can include following this procedure.
3. Load events.file by using the following command:

   pin_load_invoice_events -brand "brand_poid" -eventfile file
where \textit{brand\_poid} is the POID of the brand account and \textit{file} is the name of the \texttt{events.file} file.

If your system is not brand enabled, \textit{brand\_poid} must be the POID of the root account, \texttt{0.0.0.1 /account 1}.

4. To verify that the event types were loaded, you can display the \texttt{/config/invoice\_events} object by using Object Browser, or use the \texttt{robj} command with the \texttt{testnap} utility. See "Reading an Object and Writing its Contents to a File" in \textit{BRM Developer's Guide}.

\textbf{Events that can be included in the events.file file}

To include adjustment details in the invoice, include these events in the \texttt{events.file} file (if not already present):

- \texttt{/event/billing/adjustment/account}
- \texttt{/event/billing/adjustment/event}
- \texttt{/event/billing/adjustment/item}
- \texttt{/event/billing/adjustment/tax\_event}

To include payment details in the invoice, include these events in the \texttt{events.file} file (if not already present):

- \texttt{/event/billing/payment/cash}
- \texttt{/event/billing/payment/cc}
- \texttt{/event/billing/payment/check}
- \texttt{/event/billing/payment/dd}
- \texttt{/event/billing/payment/payorder}
- \texttt{/event/billing/payment/postalorder}
- \texttt{/event/billing/payment/wtransfer}
- \texttt{/event/billing/payment/failed}
- \texttt{/event/billing/payment/voucher}
- \texttt{/event/billing/payment/voucher\_topup}

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

- \texttt{/event/billing/reversal/cash}
- \texttt{/event/billing/reversal/cc}
- \texttt{/event/billing/reversal/check}
- \texttt{/event/billing/reversal/dd}
- \texttt{/event/billing/reversal/payorder}
- \texttt{/event/billing/reversal/postalorder}
- \texttt{/event/billing/reversal/voucher}
- \texttt{/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`

The invoice displays tax-related information of the account whose invoice is generated.

To include tax-related information for the summary invoice, include `/event/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 the Time Zone in Invoices

By default, invoices do not show the customer’s time zone. To show the time zone on an invoice, do one of the following:
Customizing the Information Included in Invoices

■ If you are using an HTML invoice template, use the PCM_OP_INV_POL_FORMAT_INVOICE_HTML policy opcode.
■ If you are using an XSLT invoice template, edit the style sheet to use the PIN_FLD_TIMEZONE_ADJ_END_T and PIN_FLD_RATED_TIMEZONE_ID fields instead of the PIN_FLD_END_T field for the events.

Including Shadow Event Adjustment Details in Invoices

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.

In CM configuration file (BRM_Home/sys/cm/pin.conf), by default, the show_rerate_details entry is set to 0. This configuration controls whether to include shadow adjustment details due to rerating in the invoice details. By enabling this entry, rerated events, which have RERATE_OBJ set, are dropped from the invoice details. This means any shadow adjustment, (for example event adjustment before billing) will create shadow event and remain in the invoice details.

The invoice document generated by using BRM-BI Publisher Integration framework displays all the event details including all shadow events to justify the charges being displayed. Oracle recommends keeping the default value of the show_rerate_details entry for the invoice document generated by using BRM-BI Publisher Integration framework.

To set the shadow event invoice details entry:
1. Open the CM configuration file (BRM_Home/sys/cm/pin.conf).
2. Set the following entry to 1.
   - fm_inv_pol show_rerate_details 0
   The default is 0.
3. Save and close the file.
4. Stop and restart the CM. See “Starting and Stopping the BRM System” in BRM System Administrator’s Guide.

Including Soft Descriptors in Invoices

You can add soft descriptors, which contain information to help customers recognize charges, to credit card and checking account statements. Soft descriptors are available for Paymentech direct debit and credit card processing. The following soft descriptors can be added to customer statements:
■ Your DBA (doing business as) name
■ Your product name
■ Your customer service phone number (instead of your headquarters city)

There are two ways to add soft descriptor information:
■ Configure the BRM_Home/sys/dm_fusa/pin.conf configuration file. See “About Paymentech Soft Descriptor Credit-card and Checking Statement Information” in
Modify the policy source file for the PCM_OP_PYMT_POL_PRE_COLLECT policy opcode. See "Customizing the Policy Source File for Soft Descriptors".

Customizing the Policy Source File for Soft Descriptors
You can customize the policy source file for the PCM_OP_PYMT_POL_PRE_COLLECT policy opcode to programmatically retrieve the soft descriptor information.

```
Note: You can also customize the PCM_OP_PYMT_POL_PRE COLLECT policy opcode to set a minimum amount to charge. See "Setting the Minimum Amount to Charge" in BRM Configuring and Running Billing.
```

This example shows how to retrieve the merchant name and a plan descriptor. The maximum entry is 22 characters including spaces. If the information is longer than 22 characters, it is truncated on the statement.

```c
pin_flist_t*sub_flistp = NULL;
void*vp = NULL;

/*
 * For each element in the PIN_FLD_CHARGES array add the soft
 * descriptors.
 */

/*
 * Add the merchant info substruct for the soft descriptors.
 */

sub_flistp = PIN_FLIST_SUBSTR_ADD(flistp, PIN_FLD_MERCHANT_INFO, ebufp);

/*
 * Add the merchant 'doing business as' to the soft descriptors.
 */

vp = (void *) "Portal Internet Service";
PIN_FLIST_FLD_SET(sub_flistp, PIN_FLD_MERCHANT, vp, ebufp);

/*
 * Read the Product information from the account product array to pick up the plan object.
 * (Exercise left to reader.)
 */

/*
 * Read Plan description from the plan object
 */

vp = (void *) NULL;

s_flistp = PIN_FLIST_CREATE(&ebuf);
vp = PIN_FLIST_FLD_GET(prod_flistp, PIN_FLD_PLAN_OBJ, 0, ebufp);
PIN_FLIST_FLD_SET(s_flistp, PIN_FLD_POID, vp, ebufp);
PIN_FLIST_FLD_SET(s_flistp, PIN_FLD_DESCR, NULL, ebufp);

/*
 * Read plan info.
 */
```
Customizing the Information Included in Invoices

Including Custom Data in Invoices

To customize the data displayed in invoices, use the PCM_OP_INV_POL_PREP_INVOICE policy opcode.

This policy opcode searches the invoice database for information about the account and creates an flist of relevant fields to include at invoicing time.

You can customize invoice information by performing additional data searches, modifying large invoice flists, or adding additional brand-specific content to branded invoices. Customization done by the PCM_OP_INV_POL_PREP_INVOICE policy opcode is passed to the PCM_OP_INV_POL_FORMAT_INVOICE policy opcode, where the storage format is specified.

You can customize the events listed on an invoice as follows:

- **Include non-currency events**. By default, BRM invoices list all events with a currency balance impact greater than zero. Use the PCM_OP_INV_POL_PREP_INVOICE policy opcode to include non-currency events. For example, you can list hours of usage if you track hours as part of a promotion.

- **List fewer or more events**. Use the PCM_OP_INV_POL_PREP_INVOICE policy opcode.

**Important**: Using the PCM_OP_INV_POL_PREP_INVOICE policy opcode to control which event information is displayed on invoices at invoicing time can affect performance if you access information normally not stored in the PIN_FLD_INVOICE_DATA field of the event. You will get better performance if you define the event information to be stored for invoicing when the event is created by using the PCM_OP_ACT_POL_SPEC_EVENT_CACHE policy opcode.

You must modify this policy opcode to enable messaging through the Universal Messaging Service (UMS).

Including Late Payment Fees in Invoices

If you are generating the final invoice document by using BRM-BI Publisher Invoicing Integration framework, you can customize the invoice data generated in XML format to include the late payment fee. For more information, see "Designing and Generating
Customizing the Information Included in Invoices

Invoices in Oracle Business Intelligence Publisher 10g

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

1. Extend the /item storable class to include /item/late_fee. For more information, 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:

   ```xml
   <ItemTypeElement>
       <ItemTag>late_fee</ItemTag>
   </ItemTypeElement>
   ```

3. Save the file.

4. Load the configuration item types by using the load_config_item_types utility:
   ```bash
   load_config_item_types config_item_types.xml
   ```

Including Finance Charges in Invoices

If you are generating the final invoice document by using BRM-BI Publisher Invoicing Integration framework, you can customize the invoice data generated in XML format to include the finance charges. For more information, see "Designing and Generating Invoices in Oracle Business Intelligence Publisher 10g".

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

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

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

   ```xml
   <ItemTypeElement>
       <ItemTag>FinanceCharges</ItemTag>
   </ItemTypeElement>
   ```

3. Save the file.

4. Load the configuration item types by using the load_config_item_types utility:
   ```bash
   load_config_item_types config_item_types.xml
   ```
5. Open the BRM_Home/sys/data/pricing/example/config_item_tags.xml and define a tag for the new service. For example:

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

6. Save the file.

7. Load the configuration item tags by using the `load_config_item_tags` utility.

   `load_config_item_tags config_item_tags.xml`

### 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 aggregate taxes on your invoices, modify a field in the `invoicing` instance of the `/config/business_params` object:

1. Open the BRM_Home/sys/data/config/bus_params_Invoicing.xml file in an XML editor.

2. Find the following line the XML file:

   ```xml
   <ADSTTaxHandle>disabled</ADSTTaxHandle>
   ```

3. Change `disabled` to `enabled`.

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

4. Save the file and rename it `bus_params_Invoicing.xml`.

5. Use the following command to load this change into the `/config/business_params` object:

   `pin_bus_params bus_params_Invoicing.xml`

Execute this command from the BRM_Home/sys/data/config directory, which includes support files used by the utility. To execute it from a different directory, see "pin_bus_params" in BRM Developer’s Guide.

6. Read the object with the `testnap` utility or Object Browser to verify that all fields are correct. See "Reading an Object and Fields" in BRM Developer’s Guide.

   **Note:** See "Using testnap" in BRM Developer’s Guide for general instructions on using `testnap`. See "Reading Objects by Using Object Browser" in BRM Developer’s Guide for information on how to use Object Browser.

7. Stop and restart the CM. For more information, see "Starting and Stopping the BRM System" in BRM System Administrator’s Guide.
8. For multiple databases, run the `pin_multidb` script with the `-R CONFIG` parameter. For more information on this script, see "pin_multidb" in BRM System Administrator’s Guide.

Specifying Invoice Data from Pipeline Manager and Custom Applications

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

BRM provides a default template file (`pin_invoice_data_map`) located in `BRM_Home/sys/data/config`. This file includes a default INTEGRATE template for Pipeline Manager invoice data. You can modify the INTEGRATE 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 "Enabling Event Caching".
- Configure the output module to add invoice data. See "Adding Invoice Data to Pipeline Output".

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 INTEGRATE template defines Pipeline Manager invoice data. You can modify the INTEGRATE 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 INTEGRATE 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
   ```
3. Save and close the file.

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

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

---

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

---

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

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

### Loading the Invoice Data Map Templates

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

---

**Caution:** 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 `load_pin_invoice_data_map`.

---

**Important:** 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 "Creating Configuration Files for 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
   ```
Enabling Event Caching

Pipeline Manager caches invoice data before sending it to BRM for processing. Therefore, you must enable event caching to include Pipeline Manager invoice data in your invoices.

1. Open the CM configuration file (`BRM_Home/sys/cm/pin.conf`).
2. Make sure the `event_cache` entry is set to 1:
   ```
   - fm_inv event_cache 1
   ```
3. Save and close the file.

Adding Invoice Data to Pipeline Output

To add data from a pipeline to your invoices, you need to configure the OUT_GenericStream module to add the data to the BRM billing record.

To output invoice data, add the following registry parameter to all OUT_GenericStream modules:

```
AddInvoiceData = TRUE
```

This parameter is read by the output grammar. When set to `TRUE`, the output module adds invoice data to each BRM billing record.

For more information about OUT_GenericStream module registry entries, see "OUT_GenericStream" in BRM Configuring Pipeline Rating and Discounting.

About Formats for Storing Invoices

BRM stores each invoice as an object in a database. By default, BRM stores invoices in the same database it stores the accounts you are invoicing. But BRM can also store invoices in a separate database. For information on how to store invoices in a separate database, see "Storing Invoices in a Separate Database".

You must store invoices in at least one of two 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 the following reasons:
  - Customer Center and Self-Care Manager require 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 to provide invoice data to third-party invoice viewers. For information on the pin_flist format, see "Understanding Flists and Storable Classes" in BRM Developer’s Guide.

- **XML:** You might use this storage format if you are using XML to design and display invoices. You use this format if you are generating the final invoice document by using BRM-BI Publisher Invoicing Integration package. See "Designing and Generating Invoices in Oracle Business Intelligence Publisher 11g" or "Designing and Generating Invoices in Oracle Business Intelligence Publisher 10g".
To use XML as an alternative or additional storage format, modify the PCM_OP_INV_POL_FORMAT_INVOICE policy opcode.

**Note:** You can store invoice data in pin_flist and XML formats at the same time. To store invoice data in multiple formats, modify the PCM_OP_INV_POL_FORMAT_INVOICE policy opcode. If you use XSL style sheet, you can store the invoice data in pin_flist format and use Customer Center to display invoices.

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. But 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 policy opcode.

**Storing Invoices in a Separate Database**

**Important:** Storing invoices in a separate database is supported only on Oracle databases.

By default, your invoices are stored in the main BRM database. However, using a separate database 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 BRM database.

The procedures for storing invoices in a separate database include:

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

**Installing the Invoice Data Manager**

**Important:** The Invoice Data Manager (DM) is supported only on Oracle databases.

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

1. Make sure you have a separate Oracle database installed and available to use for storing invoices.
2. On the machine containing BRM, open the BRM_Home/setup/pin_setup.values file in a text editor.
3. Add "dm_invoice" to the @COMPONENT_LIST line:

@COMPONENT_LIST = ("dm_taxware", "dm_oracle", "dm_invoice", 'cm');

Important: Make sure "dm_invoice" appears after "dm_oracle".

4. Modify the following entries in the pin_setup.values file as described in Table 1–3:

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$INVOICE_DB['user']</td>
<td>Must contain the user name to log in to your invoice database. This user name must be different from the one used for the main BRM database.</td>
</tr>
<tr>
<td>$INVOICE_DB['password']</td>
<td>Must contain the password to log in to your invoice database.</td>
</tr>
<tr>
<td>$INVOICE_DB['alias']</td>
<td>Must contain the database alias of the invoice database.</td>
</tr>
<tr>
<td>$INVOICE_DB['Host']</td>
<td>Must contain the hostname of the machine running the invoice database.</td>
</tr>
<tr>
<td>$INVOICE_DB['tables_group']</td>
<td>Must contain the name of the data tablespace.</td>
</tr>
<tr>
<td>$INVOICE_DB['indexes_group']</td>
<td>Must contain the name of the index tablespace.</td>
</tr>
<tr>
<td>$DM_INVOICE['port']</td>
<td>Must contain the port number of the Invoice DM.</td>
</tr>
<tr>
<td>$DM_INVOICE['db_num']</td>
<td>Must contain the database number for the invoice database.</td>
</tr>
</tbody>
</table>

5. Install dm_invoice by using one of these methods:

- To install all BRM components, run the following script:
  
  BRM_Home/setup/pin_setup

- To install only the Invoice DM, run the following 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

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

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

   For information about configuration files, see "Using Configuration Files to Connect and Configure Components" in BRM System Administrator's Guide.

2. In the invoice_db entry, enter the invoice database number.

   For example:

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

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

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

3. Make sure you have a `dm_pointer` entry for the invoice database 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 database:

   ```
   - 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.


### Configuring Invoice Applications to Use a Separate Invoice Database

To use a separate database for storing invoices:

1. Open the invoice configuration file (`BRM_Home/apps/pin_inv/pin.conf`) in a text editor.

2. In the Invoice DB number section for the `pin_inv_send` and `pin_inv_export` applications, change the `invoice_db` parameter value to your invoice DB number. These examples use the BRM default invoice database 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 pin.conf for Multiple Databases

In a multiple database environment, to store invoices in the second database:

1. Open the invoice configuration file (`BRM_Home/apps/pin_inv/pin.conf`).

2. Change the login name in the following entry to your login name:

   ```
   - nap login_name name
   ```

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

   ```
   - pin_inv_accts database 0.0.0.2 /search 0
   - pin_inv_send database 0.0.0.2 /search 0
   - pin_inv_export database 0.0.0.2 /search 0
   ```
4. Save and close the file.

For information on running invoice utilities with multiple databases, see "Setting Up Invoicing on Multiple Databases".

Displaying Invoices

BRM provides these options for displaying invoices:

- The Invoice Viewer in Customer Center. See the discussion of reviewing an invoice in the Customer Center Help.
- Self-Care Manager customer self-care Web pages.
- Sample XML invoice viewing utility (\texttt{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.

\begin{itemize}
  \item You cannot change invoice information in the Invoice Viewer.
  \item 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.
\end{itemize}

Using Web Pages

If you use Self-Care Manager, your customers can also display their invoices on your customer self-care Web site. See "Setting Up Customer Self Care With Self-Care Manager" in \textit{BRM Managing Customers}.

Using the XML Sample Invoice Viewing Utility

You can use the \texttt{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 5.

To use this utility, go to the \texttt{BRM_Home/source/apps/sampleviewer} directory. You need to run \texttt{make} to build \texttt{pin_inv_view.pl}.

Customizing the Format for Online Invoices

To display invoices, use the PCM\_OP\_INV\_VIEW\_INVOICE opcode.

This opcode uses the POID of the \texttt{/bill} object or \texttt{/invoice} object to locate and retrieve a specific invoice. Specify the output format of the invoice as a mime type in the PIN\_FLD\_TYPE\_STR field in the input flist. If you provide the bill number in PIN\_FLD\_BILL\_NO, the opcode searches the bill database and, if necessary, the \texttt{history_bills} objects.

PCM\_OP\_INV\_VIEW\_INVOICE performs the following tasks:
Uses the PIN_FLD_THRESHOLD_UPPER value in the input flist to determine the maximum size (KB) of the invoice to be viewed. If the invoice is larger than the size specified, an error message and the invoice size, rather than the invoice itself, is displayed.

Checks the following fields in the /invoice object:

- PIN_FLD_INV_SIZE, to determine the size of the invoice.
  
  If the invoice size is greater than the invoicing threshold value, PCM_OP_INV_VIEW_INVOICE does not try to format the invoice.
  
  If the invoice size is less than the invoicing threshold, PCM_OP_INV_VIEW_INVOICE continues processing.

- PIN_FLD_INV_FLAGS, to determine the type of invoice to view. See "Defining the Invoice Type".

- PIN_FLD_TYPE_STR, to determine if the format specified is stored in the object. Invoices can be stored in the database in pin_flist, XML, HTML, or DOC1 format.
  
  If the specified format is stored, the invoice is retrieved and returned in the format specified. If the specified format is not stored, the PCM_OP_INV_POL_FORMAT_VIEW_INVOICE policy opcode is called to attempt to format the invoice. Invoices can use HTML, XML, or DOC1 format. To apply an XSL style sheet to the invoice, specify the XSL mime type in the PIN_FLD_TYPE_STR field.

If successful, PCM_OP_INV_VIEW_INVOICE returns a buffer containing the formatted invoice. The PIN_FLD_RESULT value in the output flist is 1.

If not successful due to system errors, no invoice is returned. The PIN_FLD_RESULT value is 0.

If not successful because the invoice exceeded the threshold size, the PIN_FLD_RESULT value is 2.

Customizing the Format for Printed Invoices

To format invoices for printing, use the PCM_OP_INV_POL_FORMAT_INVOICE policy opcode.

This policy opcode is called when invoices are generated to specify if the invoices are to be stored in XML or pin_flist format in the /invoice object. The default is XML.

- If the storage format is specified as XML or pin_flist, the return flist contains only the specified format type.

- If you customize this policy opcode to generate other formats, the return flist contains the format type along with a buffer containing the formatted invoices.

Any customization of invoice content done by the PCM_OP_INV_POL_PREP_INVOICE policy opcode is passed to the PCM_OP_INV_POL_FORMAT_INVOICE policy opcode prior to storage.

You can customize the PCM_OP_INV_POL_FORMAT_INVOICE policy opcode to generate other storage formats. For example, if you want to store invoices in HTML format, you can add code to call the PCM_OP_INV_POL_FORMAT_INVOICE_HTML policy opcode, then add the formatted invoice to the buffer in the output flist.
**Customizing the Format for HTML Invoices**

To format invoices in HTML, use the `PCM_OP_INV_POL_FORMAT_INVOICE_ HTML` policy opcode.

This policy opcode is called by the `PCM_OP_INV_POL_FORMAT_INVOICE` policy opcode when the invoice format requested is HTML.

If your system has the invoicing-by-service feature enabled, the `PCM_OP_INV_POL_FORMAT_INVOICE_HTML` policy opcode will display the invoice items by service instance. If your system is brand enabled, this policy opcode caches all `/config/invoice_events` objects associated with the specific brand. Otherwise, only the `/config/invoice_events` object belonging to the root account is cached.

**Customizing the Format for XML Invoices**

To format invoices in XML, use the `PCM_OP_INV_POL_FORMAT_INVOICE_XML` policy opcode.

This opcode is called by the `PCM_OP_INV_POL_FORMAT_VIEW_INVOICE` policy opcode when the invoice format requested is XML.

**Customizing the Invoice Format by Using an XSL Style Sheet**

To format invoices using an XSL style sheet, use the `PCM_OP_INV_POL_FORMAT_INVOICE_XSLT` policy opcode.

You can brand invoices by using brand-specific XSL style sheets.

To specify the use of XSL style sheets, set the PIN_FLD_FLAGS field in the `/config/invoice_templates` object to 1.

This opcode is called by the `PCM_OP_INV_POL_FORMAT_INVOICE` policy opcode when the `/config/invoice_templates` object specifies an XSL style sheet.

For more information about loading and editing XSL style sheets, see “Using XSLT Invoice Templates”.

**Customizing the Format for DOC1 Invoices**

To format invoices for DOC1, use the `PCM_OP_INV_POL_FORMAT_INVOICE_DOC1` policy opcode.

---

**Important:** For DOC1 format, you must have the DOC1 software.

---

This opcode also checks the `pin.conf` file to see if your system has service-centric invoicing turned on. If invoicing by service is enabled, this opcode reorganizes the flist so that it displays the invoice items by service instance.

If your system has the invoicing-by-service feature enabled, the `PCM_OP_INV_POL_FORMAT_INVOICE_DOC1` policy opcode displays the invoice items by service instance.

**Displaying an Invoice on Demand**

To display an invoice on demand, use the `PCM_OP_INV_POL_FORMAT_VIEW_INVOICE` policy opcode. This policy opcode generates an invoice in the specified format.
BRM can store the invoice in the database in either pin_flist or XML format. The default is pin_flist. The storage format is specified by the PCM_OP_INV_FORMAT_VIEW_INVOICE opcode.

The PCM_OP_INV_POL_FORMAT_VIEW_INVOICE policy opcode is called when PCM_OP_INV_VIEW_INVOICE requests an invoice in a format that is not stored on the /invoice object. This opcode attempts to generate the invoice in the requested format. Invoices may be formatted as HTML or DOC1. An XML format is also available, but it displays as HTML format.

The PCM_OP_INV_POL_FORMAT_VIEW_INVOICE policy opcode calls one of the following policy opcodes, depending on the requested format:

- PCM_OP_INV_POL_FORMAT_INVOICE_HTML
- PCM_OP_INV_POL_FORMAT_INVOICE_DOC1
- PCM_OP_INV_POL_FORMAT_INVOICE_XML
- PCM_OP_INV_POL_FORMAT_INVOICE_XSLT

The PCM_OP_INV_POL_FORMAT_VIEW_INVOICE policy opcode checks the PIN_FLD_FLAG field in the/config/invoice_templates object to see if an XSL style sheet should be applied to the invoice. If the flag is set, the PCM_OP_INV_POL_FORMAT_INVOICE_XSLT policy opcode is called to format the invoice.

The PCM_OP_INV_POL_FORMAT_VIEW_INVOICE policy opcode also checks the pin.conf file to see if your system has service-centric invoicing turned on. If invoicing by service is enabled, this opcode reorganizes the flist so that it displays the invoice items by service instance. See "Creating Service-centric Invoices".

---

**Note:** Because the flist is reorganized, if you apply an XSL style sheet to the invoice, you may need to change the style sheet to reflect the change in output created by the service instance organization.

---

**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.

---

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

---

The layout of a typical invoice generated using BRM-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 hierarchy 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.

Figure 1–1 shows the standard data in a typical invoice:

**Figure 1–1 Typical Invoice**

<table>
<thead>
<tr>
<th>INVOICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Doe</td>
</tr>
<tr>
<td>CA</td>
</tr>
<tr>
<td>Cupertino</td>
</tr>
<tr>
<td>CA, 95014</td>
</tr>
<tr>
<td>Invoice Date: 06/18/2009</td>
</tr>
<tr>
<td>Invoice No.: B1-2</td>
</tr>
<tr>
<td>Account No.: 0.0.0.1-147722</td>
</tr>
<tr>
<td>Due Date: 07/17/2009</td>
</tr>
<tr>
<td>Bill Period: 06/17/2009 To 06/17/2009</td>
</tr>
<tr>
<td>Past Due Amt</td>
</tr>
<tr>
<td>478.04</td>
</tr>
</tbody>
</table>

Summary of Current Charges | Amount ($) |
| Service: SMS | 15.50 |
| Service: TELEPHONY | 94.88 |
| Taxes & Surcharges | 6.90 |
| Total Current Charges | 116.38 |
| Other Charges and Credits | 3.55 |

---

**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.

Figure 1–2 shows the adjustment, payment, promotion, and other charges and credits details in a typical invoice:

**Figure 1–2  Adjustment Details and Other Charges and Credits on Invoice**

<table>
<thead>
<tr>
<th>Adjustment Details</th>
<th>Date</th>
<th>Amount (E)</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item</td>
<td>05/18/2009</td>
<td>(10.00)</td>
<td>Item Adjustment</td>
</tr>
<tr>
<td>Account</td>
<td>05/18/2009</td>
<td>(2.00)</td>
<td>Event for Account Adjustment</td>
</tr>
<tr>
<td>Item</td>
<td>05/19/2009</td>
<td>(40.00)</td>
<td>Item Adjustment</td>
</tr>
<tr>
<td>Account</td>
<td>05/25/2009</td>
<td>(10.00)</td>
<td>Event for Account Adjustment</td>
</tr>
<tr>
<td>Account</td>
<td>05/25/2009</td>
<td>(20.00)</td>
<td>Event for Account Adjustment</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Other charges and Credits</th>
<th>Ref No.</th>
<th>Name</th>
<th>Date</th>
<th>Amount (E)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1-1 Dispute</td>
<td>05/18/2009</td>
<td>0.50</td>
<td>Event Dispute</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1-1 Dispute</td>
<td>05/18/2009</td>
<td>(5.00)</td>
<td>Event Dispute</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1-1 Settlement</td>
<td>05/25/2009</td>
<td>0.00</td>
<td>Event Settlement</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1-1 Settlement</td>
<td>05/25/2009</td>
<td>0.00</td>
<td>Event Settlement</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W1-1 Write-off</td>
<td>05/25/2009</td>
<td>(0.85)</td>
<td>Item Writeoff</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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, there are different sections for each phone number that displays the charge details corresponding to that number. In a hierarchy, for a parent account, the invoice displays the total charges for each subordinate account.

This section of the invoice also displays plan, service, deal, and product details used by the customer.

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

- Resource Details: Includes the details of non-currency resources 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 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 data record (EDR) usage. To display the exact usage duration in the invoice, you can customize the product 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.

Figure 1–3 shows the charge details for a particular number in a typical consumer invoice:
After generating invoices, you email or print them by using a separate application. You can also make invoices available on your customer self-care Web site for customers to view. See "Setting Up Customer Self Care With Self-Care Manager" in BRM Managing Customers.

Use the `pin_inv_send` utility to email electronic invoices or to send printed invoices to your customers. This utility determines whether to email or print an invoice by checking an account’s delivery method. The delivery method is set by a CSR in the Payment Setup panel of Customer Center. There are two possible delivery methods:

- **Email**: `pin_inv_send` emails the invoice.
Postal: `pin_inv_send` prints the invoice.

By default, this utility sends invoices in HTML format. You can also specify DOC1 format. For more information, see "`pin_inv_send`".

**Setting the Maximum Invoice Size for Email**

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

1. Open the invoice configuration file (`BRM_Home/apps/pin_inv/pin.conf`).
2. Set the value of the `inv_send_size` entry in kilobytes and make sure it is not commented.
   
   This entry is used by PCM_OP_INV_VIEW_INVOICE to restrict sending large invoices to the Email Data Manager (`dm_email`).
3. Save and close the file.

**Sending Child Account Invoices to the Parent Account’s Email Address**

By default, invoices for the subordinate bill units in an account hierarchy are consolidated into the invoice for the paying bill unit of the parent account and sent to the parent account’s email address. To handle cases where the parent account’s invoice is not consolidated and individual invoices exist for each subordinate bill unit, you can configure `pin_inv_send` to send all subordinate bill unit invoices to the email address of parent account’s invoice:

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

**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_Home/sys/dm_email/pin.conf`).
3. In the `html2ps` entry, enter the path to the `html2ps` tool.
   
   For example:
   
   ```bash
dm_email html2ps path_to_html2ps_tool
   ```
4. In the `printer` entry, enter the name of your printer:
   
   ```bash
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:
   
   ```bash
dm_email tmp_dir /tmp
   ```
6. Save and close the file.
7. Stop and restart the Email Data Manager. See "Starting and Stopping the BRM System" in *BRM System Administrator’s Guide*. 
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, enter the database number in the invoicing configuration file (BRM_Home/apps/pin_inv/pin.conf). For example:

   - pin_inv_send invoice_db 0.0.0.2 /invoice 0

   Then follow the procedures in "Storing Invoices in a Separate Database".

2. 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 "Creating Configuration Files for BRM Utilities" in BRM System Administrator’s Guide.

4. Run the pin_inv_send utility by entering this command:

   \[pin_inv_send\]

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

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

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

\[pin_inv_send \ -pay_type \ id\]

For \(id\), enter one payment method ID as shown in Table 1–4:

<table>
<thead>
<tr>
<th>Payment Method</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Undefined</td>
<td>0</td>
</tr>
<tr>
<td>Prepaid</td>
<td>10000</td>
</tr>
<tr>
<td>Invoice</td>
<td>10001</td>
</tr>
<tr>
<td>Debit</td>
<td>10002</td>
</tr>
<tr>
<td>Credit card</td>
<td>10003</td>
</tr>
<tr>
<td>Direct debit</td>
<td>10005</td>
</tr>
<tr>
<td>Smart card</td>
<td>10006</td>
</tr>
<tr>
<td>Subordinate</td>
<td>10007</td>
</tr>
<tr>
<td>In Customer Center this is called nonpaying.</td>
<td>10008</td>
</tr>
<tr>
<td>Beta</td>
<td>10009</td>
</tr>
<tr>
<td>Internal</td>
<td>10010</td>
</tr>
<tr>
<td>Guest</td>
<td>10011</td>
</tr>
<tr>
<td>Cash</td>
<td>10012</td>
</tr>
<tr>
<td>Check</td>
<td>10013</td>
</tr>
<tr>
<td>Wire transfer</td>
<td>10014</td>
</tr>
<tr>
<td>Payorder</td>
<td>10015</td>
</tr>
<tr>
<td>Postal order</td>
<td></td>
</tr>
</tbody>
</table>
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.

---

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

---

To export invoices to files:

1. Open the invoicing configuration file (`BRM_Home/apps/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 you want 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
   ```


5. Run the `pin_inv_export` utility by entering at the command line:
   ```
   pin_inv_export
   ```

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

---

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

---

To generate trial invoices, run the following command:

---

### Table 1–4 (Cont.) Payment Method IDs

<table>
<thead>
<tr>
<th>Payment Method</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voucher</td>
<td>10016</td>
</tr>
</tbody>
</table>

---
pin_inv_export -verbose -trial

To generate detail 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.

---

**Important:** When exporting invoices for parent A/R accounts, and the invoices for the subordinate bills were not consolidated into the A/R account’s invoice, a separate invoice file for each subordinate bill unit and the A/R bill unit is exported.

The file-naming convention is accountPOID_billPOID_date.ext.

If a subordinate invoice or parent invoice fails during generation, subordinate invoices are not exported.

---

**Note:** 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".

### Setting Invoicing Defaults

You can specify the following system defaults for invoices:

- Payment due date. See "Setting the Default Payment Due Date".
- Invoice type. See "Setting the Default Invoice Type".
- Hierarchical group settings. See "Setting Defaults for Hierarchical Group Invoices".
- Maximum invoice size viewable in Customer Center. See "Setting the Maximum Size of Invoices Viewable in Customer Center".

### Setting the Default Payment Due Date

The default payment due date for invoice payments is 30 days from the billing date. To change the default, customize the PCM_OP_CUST_POL_PREP_PAYINFO policy opcode.

### Setting the Default Invoice Type

The PIN_FLD_INV_DETAIL field in the /payinfo object contains the invoice value: detailed (0) or summary (1). You can set this value by passing the value in the PIN_FLD_INV_TYPE field in the input flist of:

- PCM_OP_CUST_COMMIT_CUSTOMER when creating a customer account.
- PCM_OP_CUST_SET_PAYINFO when changing a customer’s payment information.
Setting Defaults for Hierarchical Group Invoices

BRM automatically generates invoices for the parent bill unit and each subordinate bill unit in an account hierarchy. For example, if an account hierarchy contains two subordinate bill units, BRM generates three invoices: one for the parent bill unit and one for each subordinate bill unit.

You can configure whether invoices for parent bill units also include information from subordinate bill units by modifying fields in the invoicing instance of the /config/business_params object. Specifically, you can configure the following:

- Whether subordinate A/R items are displayed on a parent A/R account’s invoice.
- The maximum number of subordinate bill units that can be included in a parent A/R account’s summary invoice.
  
  When the number of subordinate bill units is less than the maximum, all subordinate bill units are included on the parent’s summary invoice. When the number of subordinate bill units exceeds the maximum, none of the subordinate bill units are included on the parent’s invoice.
- The maximum number of subordinate bill units that can be included in a parent A/R account’s detailed invoice.
  
  When the number of subordinate bill units is less than the maximum, the parent’s detailed invoice lists the items and events for all subordinate bill units. When the number of subordinate bill units exceeds the maximum, the parent’s detailed invoice does not include subordinate bill unit data.

To specify whether subordinate details are displayed on parent invoices:

1. Use the following command to create an editable XML file for the Invoicing parameter class:

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

   This command creates the XML file named `bus_params_Invoicing.xml.out` in your working directory. If you do not want this file in your working directory, specify the full path as part of the file name.

2. (Optional) Specify whether to display A/R items for each subordinate bill unit on parent invoices:

   Set the `SubArItemsIncluded` entry to `enabled` to display subordinate A/R items on parent invoices and to `disabled` to display only parent A/R items on parent invoices.

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

3. (Optional) Set the threshold for summary invoices:

   Search the XML file for the following line and change 0 to the number of subordinate bill units allowed in a parent’s summary invoice:

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

   If the threshold value is exceeded, the parent A/R account summary invoice will not contain the subordinate bill unit data.

---

**Note:** A value of 0 means that BRM does not check the size of the account hierarchy and the parent’s A/R account summary invoice automatically includes all subordinate bill unit data.
4. (Optional) Set the threshold for detailed invoices:

Search the XML file for the following line and change 0 to the number of subordinate bill units allowed in a parent’s detailed invoice:

```xml
<TresholdSubordsDetail>0</TresholdSubordsDetail>
```

If the number of subordinate bill units exceeds the threshold value, the parent’s detailed invoice includes only the parent bill unit’s details.

**Note:** A value of 0 means that BRM does not check the size of the account hierarchy and the parent’s A/R account summary invoice automatically includes all subordinate bill unit data.

5. Save the file and rename it to `bus_params_Invoicing.xml`.

6. Use the following command to load the change into the `/config/business_params` object:

```bash
pin_bus_params bus_params_Invoicing.xml
```

Run this command from the `BRM_Home/sys/data/config` directory, which includes support files used by the utility. To run it from a different directory, see “pin_bus_params” in *BRM Developer’s Guide*.

7. Read the object with the `testnap` utility or the Object Browser to verify that all fields are correct. See “Reading an Object and Fields” in *BRM Developer’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 Account Groups".

**Setting the Maximum Size of Invoices Viewable in Customer Center**

To limit the size of invoices that can be displayed in Customer Center, you set the `view_invoice_max_size` entry in the `Customized.properties` file.

**Important:** Do not modify the default Customer Center properties file (`CustomerCenter.properties`). The parameters and values specified in the `Customized.properties` file take precedence over values for identical parameters in the default properties file.

1. Open the `Customized.properties` file with a text editor. This file is located in the `CCSDK_home/CustomerCareSDK/CustCntr/custom` directory.

2. Add the following entry to the file:

```properties
view_invoice_max_size=kilobytes
```

where `kilobytes` is the maximum invoice size.

This sets the maximum size of an invoice in kilobytes that can be displayed in Customer Center. This value is passed in the input flist of `PCM_OP_INV_VIEW_INVOICE`. See "Displaying Invoices".

3. Save and close the file.
Adding Siebel CRM Promotion Names to Invoices

Promotions are marketing bundles that contain multiple products and services, often at reduced prices. When customers order promotions, their invoices, by default, display the 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 products and discounts 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.

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 products and discounts 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 products and discounts (/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 "Managing Promotions with Siebel CRM".

Use an invoice template that displays promotion names and details. See "Using Invoice Templates that Display Promotion Names".

---

**Important:** If you configure BRM to exclude plan, deal, and product pricing details from invoices, the invoice will not display Siebel CRM promotion names. See "Improving Performance by Removing Invoice Details You Do Not Need".

---

### Specifying Whether BRM Displays Promotion Details on Invoices

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

You can enable and disable this search for promotion details by modifying a field in the invoicing instance of the /config/business_params object.

- 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.

You modify the /config/business_params object by using the pin_bus_params utility. For information on this utility, see "pin_bus_params" in BRM Developer’s Guide.

To specify whether BRM invoices display Siebel CRM promotion names:

1. Open the BRM_Home/sys/data/config/bus_params_Invoicing.xml file in an XML editor.
2. Edit the <PromotionDetailDisplay> element in the XML file:
   - Use disabled to disable the search for promotion details.
   - Use enabled to enable the search for promotion details.
   
   ```xml
   <PromotionDetailDisplay>enabled</PromotionDetailDisplay>
   ```

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

3. Save and close the file.
4. Use the following command to load this change into the /config/business_params object:

   ```shell
   pin_bus_params bus_params_Invoicing.xml
   ```

   Execute this command from the BRM_Home/sys/data/config directory, which includes support files used by the utility. To execute it from a different directory, see "pin_bus_params" in BRM Developer’s Guide.
5. Read the object with the `testnap` utility or Object Browser to verify that all fields are correct. See "Reading an Object and Fields" in BRM Developer’s Guide.

**Note:** See "Using testnap" in BRM Developer’s Guide for general instructions on using `testnap`. See “Reading Objects by Using Object Browser” in BRM Developer’s Guide for information on how to use Object Browser.

６. Stop and restart the CM. For more information, see "Starting and Stopping the BRM System" in BRM System Administrator’s Guide.

7. For multiple databases, run the `pin_multidb` script with the `-R CONFIG` parameter. For more information on this script, see "pin_multidb" in BRM System Administrator’s Guide.

**Managing Promotions with Siebel CRM**

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:

- Promotion name
- Promotion description
- Products and discounts bundled with the promotion
- Promotion creation date
- Promotion validity dates
- Promotion status: active, inactive, or canceled

In an AIA system, you pass this information from Siebel CRM to BRM through the J2EE Connector Architecture (JCA) Resource Adapter, which exposes the BRM API through a JCA common client interface (CCI) as an Interaction. Data required for sending data to PCM_OP_SUBSCRIPTION_SET_BUNDLE is detailed in the Subscription Web service. For more information, see "Connecting J2EE-compliant Applications to BRM" in BRM JCA Resource Adapter.

To implement the ability to display Siebel CRM promotions on invoices, add the following functionality to Siebel CRM by using the PCM_OP_SUBSCRIPTION_SET_BUNDLE opcode:

- Create `/purchased_bundle` objects. See "Adding a promotion to an account".
- Modify `/purchased_bundle` objects. See "Modifying an account’s promotion".
- Inactivate `/purchased_bundle` objects. See "Inactivating an account’s promotion".
- Cancel `/purchased_bundle` objects. See "Canceling an account’s promotion".

**Adding a promotion to an account**

To add a promotion to a customer’s account, pass in details about the promotion to PCM_OP_SUBSCRIPTION_SET_BUNDLE. You must also set the following fields in the opcode’s input flist:

- In the PIN_FLD_BUNDLE_INFO array:
  - Set the PIN_FLD_POID field to a type-only POID for `/purchased_bundle`.
  - Set the PIN_FLD_STATUS field to 1 to set the promotion’s status to Active.
Adding Siebel CRM Promotion Names to Invoices

In the PIN_FLD_OFFERINGS array, specify the products and discounts associated with the promotion. You must create a separate array for each product and discount in the promotion. In the PIN_FLD_OFFERINGS array:

- Set the PIN_FLD_POID field to the POID of the /purchased_product or /purchased_discount object.
- Set the PIN_FLD_BUNDLE_OBJ field to NULL.
- Set the PIN_FLD_BUNDLE_INFO array to the rec_id of the PIN_FLD_BUNDLE_INFO array at the 0th level of the flist. This specifies to associate the newly created /purchased_bundle object with the specified product or discount during account creation.

Modifying an account’s promotion

To modify an account’s existing promotion, pass in the promotion details that changed to the PCM_OP_SUBSCRIPTION_SET_BUNDLE opcode:

- To change a promotion’s attributes: In the PIN_FLD_BUNDLE_INFO array, set the PIN_FLD_POID field to the existing /purchased_bundle POID and then pass in only the fields that have changed.
- To transfer a promotion from one account to another: In the PIN_FLD_BUNDLE_INFO array, set the PIN_FLD_POID field to the existing /purchased_bundle POID and set the PIN_FLD_ACCOUNT_OBJ field to the new /account POID.
- To add products and discounts to a promotion: In the PIN_FLD_OFFERINGS array, set the PIN_FLD_POID field to the complete POID of the product or discount that is being added to the promotion and set the PIN_FLD_BUNDLE_OBJ field to the complete POID of the /purchased_bundle object.
- To remove an existing product or discount from a promotion: In the PIN_FLD_OFFERINGS array, set the PIN_FLD_POID field to the complete POID of the product or discount that is being removed from the promotion and set the PIN_FLD_BUNDLE_OBJ field to NULL.

When a promotion is modified, the opcode:

- Changes the specified /purchased_bundle object.
- Changes the PIN_FLD_PLAN_OBJ field of the /purchased_product or /purchased_discount object, if a product or discount changed.
- Generates an /event/billing/bundle/modify object for auditing purposes, if a promotion’s attributes changed.

Inactivating an account’s promotion

When you inactivate an account’s promotion, the PCM_OP_SUBSCRIPTION_SET_BUNDLE opcode sets the promotion’s status to Inactive. It does not update the associated product or discount object’s PIN_FLD_PLAN_OBJ field. Also, unlike the canceled status, a promotion’s details can be modified after it has been inactivated.

To inactivate a promotion, pass in details about the promotion to PCM_OP_SUBSCRIPTION_SET_BUNDLE and set the PIN_FLD_STATUS field in the PIN_FLD_BUNDLE_INFO array to 2.

Note: If PIN_FLD_STATUS is not passed in, the promotion’s status is set to active by default.
Canceling an account’s promotion

You can no longer make changes to an account’s promotion after it has been canceled. When you cancel an account’s promotion, the PCM_OP_SUBSCRIPTION_SET_BUNDLE opcode:

- Sets the promotion’s validity end date to the current date.
- Sets the promotion’s status to cancelled.

In addition, the opcode can optionally disassociate the products and discounts from the promotion by changing the PIN_FLD_PLAN_OBJ field of the /purchased_product object and the /purchased_discount object to NULL.

To cancel an account’s promotion, call PCM_OP_SUBSCRIPTION_SET_BUNDLE and set the PIN_FLD_STATUS field in the PIN_FLD_BUNDLE_INFO array to 3. To configure the opcode to also disassociate the products and discounts from the promotion, perform one of the following tasks:

- Set the PIN_FLD_FLAGS field to 1. When set, the opcode searches for and disassociates the account’s products and discounts from the promotion. This option decreases processing performance.
- In the PIN_FLD_OFFERINGS array, set the PIN_FLD_POID field to the complete POID of the product or discount that is being canceled and set the PIN_FLD_BUNDLE_OBJ field to NULL. When set, the opcode disassociates the products and discounts you specify from the promotion.

---

**Important:** To improve processing performance, pass in all products and discounts that are associated with the promotion you are canceling.

---

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 "Using XSLT Invoice Templates".
- Customize the pin_invoice_template.html template by using the PCM_OP_INV_POL_FORMAT_INVOICE_HTML policy opcode. See "Customizing the Format for HTML Invoices".

Setting Up Invoicing on Multiple Databases

To run invoicing on multiple BRM databases, you can:

- Run invoicing on one database at a time by using one instance of the invoicing utilities. See "Running Invoicing on Multiple Databases One at a Time".
Run invoicing on multiple databases simultaneously by using multiple instances of the invoicing utilities. See "Running Invoicing on Multiple Databases Simultaneously".

Running Invoicing on Multiple Databases One at a Time

Running invoicing utilities on multiple databases one at a time requires that you edit the invoicing configuration file each time you run the invoicing utilities. Perform the following steps before you run invoicing:

1. Open the invoicing configuration file (BRM_Home/apps/pin_inv/pin.conf).
2. Change the value of the login_name entry to a database account in the database against which you want to run invoicing.

   For example, to run invoicing using the account root.0.0.0.2, change the login_name entry as follows:
   
   `- nap login_name root.0.0.0.2`

3. Save and close the file.
4. Run the invoicing utilities.

Running Invoicing on Multiple Databases Simultaneously

Running invoicing on multiple database simultaneously requires that you create parallel instances of the invoicing configuration files, each of which is configured for a particular database. Then, you run all instances of your invoicing utilities.

1. For each database you want to run invoicing on, create a subdirectory under BRM_Home/apps/pin_inv.

   For example, BRM_Home/apps/pin_inv/db1 for database 1, BRM_Home/apps/pin_inv/db2 for database 2, and so forth.

2. Copy the BRM_Home/apps/pin_inv/pin.conf file into each new subdirectory.
3. In each invoicing subdirectory, do the following:
   
   a. Open the pin.conf file.
   b. Change the database number in the login_name entry to a database account that resides in the database against which you want to run invoicing.

      For example, to run invoicing using the account root.0.0.0.2, change the login_name entry as follows:

      `- nap login_name root.0.0.0.2`

   c. Save and close the file.

4. Run the invoicing utilities from the new subdirectories.

Using an /invoice Subclass

If you add a subclass to the /invoice object and want to use the subclass for invoicing, you must specify the subclass type. There are two ways you can do this:

- Specify the invoice object database number and type in the PIN_FLD_INVOICE_OBJ field in the /payinfo class. Then, using testnap, change the invoice_obj_type in the payinfo_t table to the subclass type.
Edit the CM configuration file (BRM_Home/sys/cm/pin.conf) by doing the following:

a. In the Invoice_db entry, add the subclass extension name to /invoice.

**Important:** This entry is read during account creation time. Changing it will have no affect on existing accounts. If you want the /invoice subclass to apply to existing accounts, you must use the previous method instead.

b. Change this 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.
Designing and Generating Invoices in Oracle Business Intelligence Publisher 11g

This chapter describes how to design, generate, and customize Oracle Communications Billing and Revenue Management (BRM) invoice documents using Oracle Business Intelligence (BI) Publisher 11g.

Before reading this document, you should be familiar with the following:

- Basic database administration concepts. See your database documentation.
- BI Publisher 11g functions. See your BI Publisher documentation.

Note: For information on integrating BRM with BI Publisher 10g, see "Designing and Generating Invoices in Oracle Business Intelligence Publisher 10g”.

About BRM Integration with BI Publisher for Invoicing

BRM comes with a pre-integrated invoicing framework. This framework uses BI Publisher as the publishing application, allowing users to create more stylized and detailed invoices in PDF format. BRM also includes sample RTF templates for individual customers and for corporate customers. In these templates, some of the core business functions are pre-built and can be customized. You can configure and extend these templates for your specific business needs.

With BI Publisher 11g, 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. With increased throughput, you reduce the number of times you have to extract data to create invoices. For example, in a non-bursting environment, 50,000 invoices would require BI Publisher to connect to BRM 50,000 times, which can significantly reduce system performance. In a bursting environment, you can tune BI Publisher by setting a chunk size parameter. In the example of creating 50,000 invoices, BI Publisher only has to connect to BRM 5 times if you set the chunk size to 10,000.

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.
The pre-integrated invoicing framework eliminates the need for any proprietary invoice design tools, resulting in lowered costs. BI Publisher leverages familiar desktop software, so using the software requires minimal learning.

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

---

**Note:** BRM-BI Publisher 11g invoicing integration is certified only on WebLogic Server, which is bundled with the BI Publisher product. The Oracle BRM-BI Publisher invoicing integration is certified on BI Publisher 11.1.1.7.

---

**Overview of BRM and BI Publisher Invoice Generation**

BI Publisher is an enterprise reporting solution to author, manage, and deliver all types of formatted documents. You can customize the layout of the generated invoice document using the template builder capabilities of BI Publisher. A typical invoice document generated using BRM-BI Publisher integration is in "Understanding Invoice Layout".

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, and stores it in the BI Publisher database.

Figure 2–1 shows the BRM and BI Publisher invoice generation process:
About the Invoicing Process

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

1. **BRM associates bill units with a BI Publisher invoice and report.** When accounts are created, BRM associates each bill unit with a BI Publisher invoice report and layout template. See "About Associating Bill Units with a BI Publisher Invoice and Report".

2. **Integrating BRM and BI Publisher for generating invoice documents.** To generate invoice documents using BRM-BI Publisher integration framework, you must enable the integration between BRM and BI Publisher. See "Enabling the BRM-BI Publisher Integration".

3. **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.

4. **BRM generates the invoice and stores it in the BRM database.** When the `pin_inv_accts` utility is executed, 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".

5. **The pin_inv_doc_gen utility:**
   - **Executes BI Publisher reports.** This utility, based on the input parameters, invokes and executes appropriate BI Publisher reports for invoice documents. It executes a BI Publisher invoicing report, with the SQL bursting data query, through a scheduled job. See "Generating BI Publisher Invoice Documents in BI Publisher 11g”.
   - **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”.

**About Associating Bill Units with a BI Publisher Invoice and Report**

When the BRM-BI Publisher integration is enabled, BRM automatically associates bill units with a BI Publisher report name and invoice template name during the account creation process. When an account is created, BRM performs the following for each bill unit (`/billinfo` object) in the account:
Determines the BI Publisher invoice report and template that the bill unit qualifies for by reading the `/config/business_profile` object. This object defines your invoice types, the criteria a bill unit must meet for the invoice type, and the BI Publisher invoice report and template names associated with the invoice type. For example, a `/config/business_profile` might define the following in Table 2–1 for a regular bill:

### Table 2–1 Business Profile Example

<table>
<thead>
<tr>
<th>Invoice Type</th>
<th>Criteria</th>
<th>BI Publisher Invoice Report and Template Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monthly billing</td>
<td>The <code>/billinfo</code> object’s PIN_FLD_WHEN field is set to 1.</td>
<td><code>monthly_invoice_report</code> and <code>monthly_invoice</code></td>
</tr>
<tr>
<td>Quarterly billing</td>
<td>The <code>/billinfo</code> object’s PIN_FLD_WHEN fields is set to 3.</td>
<td><code>quarter_invoice_report</code> and <code>quarter_invoice</code></td>
</tr>
</tbody>
</table>

Creates an `/associated_bus_profile` object for the bill unit. This object stores the following information associated with invoicing for the bill unit as an element in a template array called PIN_FLD_TEMPLATE_ARRAY.

- PIN_FLD_TEMPLATE_NAME. The Template name configured in the pair key Template_Name of the corresponding invoicing `/config/business_profile` object.
- PIN_FLD_REPORT_NAME. The Report name configured in the pair key Report_Name of the corresponding invoicing `/config/business_profile` object.
- PIN_INV_TYPE. The type of invoice stored as PIN_INV_TYPE_REGULAR or PIN_INV_TYPE_CORRECTIVE.

Associates the bill unit with its `/associated_bus_profile` object by populating the `/billinfo` object’s PIN_FLD_ASSOC_BUS_PROFILE_OBJ_LIST field.

BRM creates an `/associated_bus_profile` object for each bill unit in an account by calling the PCM_OP_CUST_CREATE_ASSOCIATED_BUS_PROFILE opcode. This opcode is called internally by the PCM_OP_CUST_CREATE_BILLINFO opcode during the account creation process.

Whenever invoice business profiles are modified in the `/config/business_profile` object, use the PCM_OP_CUST_SET_ASSOCIATED_BUS_PROFILE opcode to update all related `/associated_bus_profile` objects.

### About BRM-Business Intelligence Publisher Invoicing Integration Package Components

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

You can use the following capabilities of BI Publisher to design 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".

About Invoice Layout Templates for Invoice Document Generation

The BRM-BI Publisher invoicing integration package contains two 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 (/billinfo objects) of the account are associated with an /associated_bus_profile object. The /associated_bus_profile object contains the report and template names that are used for generating invoice documents for this account on the BI Publisher server. See "Specifying BI Publisher Invoice Report and Template Names in BRM".

Default Templates for Regular Invoices

BRM provides the following default templates for regular invoices:
- BRM_Corporate_Invoice.rtf
- BRM_Consumer_Invoice.rtf

Report File for Default Invoice Templates

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 and HTML. The default output format specified in the reports is PDF.

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

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.
- In BRM, specify the BI Publisher invoice report and layout templates names. See "Specifying BI Publisher Invoice Report and Template Names in BRM".
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 Taxation Details in Invoices".

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".

Configure data source and data model.

Configure for multischema. Store and modify copies of the report in each schema.

**Specifying BI Publisher Invoice Report and Template Names in BRM**

When the BRM-BI Publisher integration is enabled, BRM automatically performs the following during the bill unit (/billinfo object) creation process:

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

1. 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. See "Creating Bill Unit Validation Templates for Invoices".
3. Specify the validation template, invoice report, and layout template to be associated with the bill unit. See "Specifying the BI Publisher Report and Template".
4. Load the business profile information into the BRM database by using the `load_pin_business_profile` utility. See "Loading Invoicing Business Profiles into the Database".
5. If your system includes accounts that were created before you integrated BRM invoicing with 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 Publisher Integration is disabled in BRM. You can enable this feature by modifying a field in the `invoicing` instance of the `/config/business_params` object.

You modify the `/config/business_params` object by using the `pin_bus_params` utility. For information on this utility, see "pin_bus_params" in BRM Developer’s Guide.

To enable the BRM-BI Publisher integration for invoices:
1. Go to the `BRM_Home/sys/data/config` directory, where `BRM_Home` is the directory in which you installed BRM components.

2. Run the following command, which creates an editable XML file from the `invoicing` instance of the `/config/business_params` object:

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

   This command creates the XML file named `bus_params_Invoicing.xml.out` in your working directory. To place this file in a different directory, specify the path as part of the file name.

3. Open the `bus_params_Invoicing.xml.out` file.

4. Search for the following line:

   ```xml
   <EnableInvoicingIntegration>disabled</EnableInvoicingIntegration>
   ```

5. Change `disabled` to `enabled`.

6. Save the file as `bus_params_Invoicing.xml`.

7. Go to the `BRM_Home/sys/data/config` directory, which includes support files used by the `pin_bus_params` utility.

8. Run the following command, which loads this change into the appropriate `/config/business_params` object:

   ```bash
   pin_bus_params PathToWorkingDirectory/bus_params_Invoicing.xml
   ```

   where `PathToWorkingDirectory` is the directory in which `bus_params_Invoicing.xml` resides.

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

   **Note:** To run this command from a different directory, see "pin_bus_params" in BRM Developer’s Guide.

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


10. Stop and restart the Connection Manager (CM). For more information, see "Starting and Stopping the BRM System" in BRM System Administrator’s Guide.

11. (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.

---

**Creating Bill Unit Validation Templates for Invoices**

You specify the criteria a bill unit must meet to be associated with an invoicing configuration business profile by creating a bill unit validation template.
Specifying BI Publisher Invoice Report and Template Names in BRM

(/config/template/billinfo). For more information about validation templates, see "About Validation Templates" in BRM Managing Customers.

To create a bill unit validation template for invoices, edit the BRM_Home/sys/data/config/pin_business_profile.xml file. The file specifies the following:

- The valid business profiles for each invoice type
- The criteria a bill unit must meet to qualify for each business profile
- The templateID name.

Note: Each type of invoice can have multiple business profiles, but they must all point to the same TemplateID name.

The default pin_business_profile.xml file includes entries for the consumer and corporate invoice documents. You can edit these entries or add entries for any other type of invoice document you want to generate.

For each type of invoice document you want to generate, edit the following entries:

- Set the TemplateId name entry to InvoiceIntegration and the type entry to /billinfo, as shown below:

  <TemplateId name="InvoiceIntegration" type="/billinfo"/>

- Specify the criteria a bill unit must meet to be associated with an invoicing business profile in the Iscript section. Enter your conditions in the if and else if blocks and the name of the business profile in PIN_FLD_NAME.

  Important: The order in which criteria is written in the if and else if blocks defines the order in which a bill unit is evaluated against the criteria.

If a particular condition is satisfied in the if and else if blocks, the business profile name associated with the condition is set with the OUT label as one of the Iscript return values.

For example:

  <Iscript>
  Long parentFlags = flistGetLong(IN,PIN_FLD_PARENT_FLAGS);

  if(parentFlags == 0)
  {
   flistSetEnum(OUT, PIN_FLD_RESULT, 1);
   flistSetStr(OUT, PIN_FLD_NAME, "ConsumerInvoice");
  }
  else if(parentFlags > 0)
  {
   flistSetEnum(OUT, PIN_FLD_RESULT, 1);
   flistSetStr(OUT, PIN_FLD_NAME, "CorporateInvoice");
  }
  </Iscript>
Specifying the BI Publisher Report and Template

You specify the BI Publisher invoice reports and layout template names by editing the business profile configuration file (pin_business_profile.xml). The default XML file contains entries for generating a consumer invoice document. You can modify these entries and add entries for any other type of invoice document that you want to generate. For example, you can add entries to generate an invoice document for Internet customers or wireless telephony customers.

To specify the BI Publisher invoice report and layout template to use for generating an invoice document, edit the following entries in the BRM_Home/sys/data/config/pin_business_profile.xml file. Add a set of entries for each type of invoice document you want to generate:

- Specify the name of the business profile in the BusinessProfile name entry. The name must match the PIN_FLD_NAME entry from the Iscript element. For example:

  `<BusinessProfile name="ConsumerInvoice" type="Invoice">`

- Make sure TemplateId name is set to InvoiceIntegration and type is set to /billinfo. All invoice business profiles must use these values.

  `<TemplateId name="InvoiceIntegration" type="/billinfo"/>`

- Specify the name of the BI Publisher XDO report to use for generating invoice documents in the Report_Name value entry. For example:

  `<NameValue key="Report_Name" value="Bursting Invoice Report"/>`

- Specify the name of the BI Publisher XDO report to use and the name of the layout template to use for each type of invoice, the regular invoice, the Replacement Invoice and the Invoice Correction Letter.

The following is a sample of the BusinessProfile tag entries for the default Corporate Invoice type in the pin_business_profile.xml file:

```
<BusinessProfile name="CorporateInvoice" type="Invoice">
  <Desc> Description of the business profile </Desc>
  <!-- List templates -->
  <TemplateId name="InvoiceIntegration" type="/billinfo"/>
  <!-- List key values -->
  <NameValue key="Report_Name" value="Bursting Invoice Report"/>
  <NameValue key="Template_Name" value="BRM_Corporate_Invoice"/>
</BusinessProfile>
```

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 that were created prior to your BRM-BI Publisher integration will not be linked with 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 7.5 from a prior release.

To generate BI Publisher invoice documents for /billinfo objects created prior to 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 with its /associated_bus_profile object by populating the /billinfo object’s PIN_FLD_ASSOC_BUS_PROFILE_OBJ_LIST field.

To associate pre-existing bill units with business profiles:

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

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

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

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

3. Go to the BRM_Home/apps/pin_billd directory.

   **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.

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.

   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.
about the /associated_bus_profile object

if you are using brm-bi publisher integration framework to generate invoice documents, at the time of customer account creation, the /associated_bus_profile object is created. see "enabling the brm-bi publisher integration". the /associated_bus_profile object stores the invoicing business profile information for a /billinfo object. the /associated_bus_profile object contains the layout template name in the PIN_FLD_TEMPLATE_NAME field and the report name in the PIN_FLD_REPORT_NAME field.

creating /associated_bus_profile objects

if the brm-bi publisher invoicing integration is enabled, during customer account creation, internally the PCM_OP_CUST_CREATE_BILLINFO opcode calls the PCM_OP_CUST_CREATE_ASSOCIATED_BUS_PROFILE opcode to create one /associated_bus_profile object for each bill unit in the account.

viewing invoices individually on the bi publisher console

in the brm_invoices reports directory (BIP_Home/user_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.

creating /associated_bus_profile objects for accounts in a branded environment

To create /config/business_profile objects for accounts in a branded environment:

1. In the BRM_Home/sys/data/config directory, create a new directory for the brand. For example, BrandA.
2. Copy pin.conf, business_configuration.xsd, and pin_business_profile.xml files from BRM_Home/sys/data/config to BRM_Home/sys/data/config/BrandA.
3. Open pin.conf and add the following entries:

   ```
   nap login_name brandloginname
   nap login_pw brandpassword
   ```

   where brandloginname is the login name for the brand and brandpassword is the password for the brand.
4. Save and close the file.
5. Open pin_business_profile.xml and replace the default values with the new values listed in Table 2–2:
Creating /associated_bus_profile Objects for Accounts in a Branded Environment

6. Save and close the file.
7. Go to the BRM_Home/sys/data/config/BrandA directory and enter the following command:

   load_pin_business_profile pin_business_profile.xml

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

   See "Using testnap" in BRM Developer’s Guide for general instructions on using the testnap utility. See "Reading Objects by Using Object Browser" in BRM Developer’s Guide for information on how to use Object Browser.

Creating Reports for Accounts in a Branded Environment in BI Publisher
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.

2. Log in to BI Publisher.
3. Create a new RTF template to associate it with the layout template specific for the accounts in a brand. See the discussion of creating an RTF template using the Template Builder for Word in the BI Publisher documentation.

   Note: The layout template name must be same as the template name specified in the updated pin_business_profile.xml file. For example, if you specify the layout template as BRM_Consumer_Invoice_BrandA, you must name the RTF template file as BRM_Consumer_Invoice_BrandA.rtf.

4. In Shared Folders, click BRM_Invoices.
5. Click Create a new report link.

6. Enter the report name as BRM_Bursting_Invoice_Report_BrandA.

   Note: The report name must be same as the report name specified in the updated pin_business_profile.xml file.

7. For the BRM_Bursting_Invoice_Report_BrandA report, click Edit. To specify report properties, see the discussion of creating the report entry and specify general properties in the BI Publisher documentation.

   Note:
   - The configuration of the BRM_Bursting_Invoice_Report_BrandA report must be same as the sample BRM_Bursting_Invoice_Report report in the BRM-BI Publisher invoicing integration package. For example, the BRM_Bursting_Invoice_Report_BrandA report must have the same WSDL URL as given in the BRM_Bursting_Invoice_Report report.
   - The layout template name must be BRM_Consumer_Invoice_BrandA.

8. Click Save.

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”.
- Install Java Runtime Environment (JRE) 1.5, and set the PORTAL_JRE environment variable to the path of the JRE 1.5 installation. See "Setting the PORTAL_JRE Environment Variable”.
- Manually remove previously installed versions of the BRM-BI Publisher 11g invoicing integration layout template files. See "Removing Older BRM-BI Publisher 11g Invoicing Integration Layout Template Files”.
- Run the BRM-BI Publisher invoicing integration package. See "Enabling the BRM-BI Publisher Invoicing Integration Package Application”.

Pre-Installation Requirements

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

- BI Publisher 11g is installed.
- BI Publisher home environment variable has been successfully set.
- BRM 7.4 Patch Set 18 is installed. The patch set contains the pin_inv_doc_gen utility updates that are required for BI Publisher 11g integration.
Third-Party package is installed, for example; 7.4PS18_ThirdParty_linux_32_opt.bin. The Third-Party package includes the PERL libraries and JRE required for installing BRM components. For more information, see “Installing the Third-Party Software” in BRM Installation Guide.

For more information on these requirements, see BRM Reports.

Setting the PORTAL_JRE Environment Variable

The BRM-BI Publisher invoicing integration package application requires JRE 1.5, and the PORTAL_JRE environment variable is set to the path of the JRE 1.5 installation.

To set the PORTAL_JRE environment variable:

- If JRE 1.5 is installed on your BI Publisher server:
  1. Open the .cshrc file.
  2. Set the PORTAL_JRE environment variable to the path where JRE 1.5 is installed:

     ```
     setenv PORTAL_JRE java_path
     ```

     where java_path is the directory in which JRE 1.5 is installed.
  3. Save and close the file.
  4. Source the .cshrc file:

     ```
     source .cshrc
     ```

- If JRE 1.5 is not installed on your BI Publisher server:
  1. Go to the directory where you installed the Third-Party package.
  2. Source the source.me file.

     For Bash shell:

     ```
     source source.me.sh
     ```

     For C shell:

     ```
     source source.me.csh
     ```

     The PORTAL_JRE environment variable is set to the path of the JRE 1.5 installation.

Removing Older BRM-BI Publisher 11g Invoicing Integration Layout Template Files

When you install the BRM-BI Publisher invoicing integration package, the older versions of BRM-BI Publisher 11g 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 11g invoicing integration layout template files.

To remove older BRM-BI Publisher 11g invoicing integration layout template files:

1. 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:
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_Invoice_en_US.rtf*
   - BRM_Corporate_Invoice_en_US.rtf*

Enabling the BRM-BI Publisher Invoicing Integration Package Application

To enable the BRM-BI Publisher invoicing integration package application:

1. Download the BRM-BI Publisher invoicing integration package 7.4PS18_BRM_BIP_Invoicing_Integration_platform_32_opt.bin to a temporary directory (temp_dir).

   Where platform is the platform on which BI Publisher is installed. For example, linux.

2. Go to the temp_dir and enter the following command:

   7.4PS18_BRM_BIP_Invoicing_Integration_platform_32_opt.bin -console

3. Follow the instructions displayed during installation.

   Note: The search and verify step of the installation is optional. If the installation hangs, or appears to be taking too long during the search and verify step, you may interrupt the installer without impacting your installation.

4. If you are using a custom BI Publisher server setup, do the following:
   a. Copy the BIP_Home/user_projects/domains/bifoundation_domain/servers/AdminServer/tmp/WebLogic_user/bipublisher_11.1.1.1/6uc731/lib/BIPExtension.jar file to your domain lib directory.
   b. Stop and restart the Oracle WebLogic Server, which adds the BIPExtension.jar file to the Oracle WebLogic server classpath and enables the BI Publisher extension library.

5. If you are using custom paths, copy the BRM-BI Publisher template files from BIP_Home/user_projects/domains/bifoundation_domain/config/bipublisher/repository/Reports to your custom directories.

Installed Files

The BRM-BI Publisher invoicing integration program copies the files listed in Table 2-3 to your computer during installation:
To use BI Publisher 10g templates in BI Publisher 11g, make the following BI Publisher 11g modifications listed in Table 2–4:
Configuring the pin_inv_doc_gen Utility

Configuring pin_inv_doc_gen involves:

- Install JDK version 1.6.0_37 or above.
- Setting the JAVA_HOME Environment Variable
- Configuring the Infranet.properties File for pin_inv_doc_gen with BI Publisher 11g
- Configuring the docgen.sh File
- 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

**Table 2–4  BI Publisher 10g Template Modifications for BI Publisher 11g**

<table>
<thead>
<tr>
<th>Modification</th>
<th>BI Publisher 10g</th>
<th>BI Publisher 11g</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML Structure</td>
<td>[data source = web service]</td>
<td>[data source = sql query for bursting]</td>
</tr>
<tr>
<td></td>
<td>&lt;invoice&gt;</td>
<td>&lt;ROWSET&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;POID&gt;0.0.0.1 /bill151665 3&lt;/POID&gt;</td>
<td>&lt;INVOICE_POID&gt;159696&lt;/INVOICE_POID&gt;</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>&lt;ROW&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;OBJ_ID&gt;159696&lt;/OBJ_ID&gt;</td>
<td>&lt;OBJ_ID&gt;0&lt;/OBJ_ID&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;REC_ID&gt;0&lt;/REC_ID&gt;</td>
<td>&lt;REC_ID&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;BUFFER_BUF&gt;</td>
<td>&lt;BUFFER_BUF&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;invoice&gt;</td>
<td>&lt;row&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;POID&gt;0.0.0.1 /bill151665 3&lt;/POID&gt;</td>
<td>&lt;POID&gt;0.0.0.1 /bill151665 3&lt;/POID&gt;</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>&lt;/invoice&gt;</td>
<td>&lt;/invoice&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/BUFFER_BUF&gt;</td>
<td>&lt;/BUFFER_BUF&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/ROW&gt;</td>
<td>&lt;/ROW&gt;</td>
</tr>
<tr>
<td>Referencing fields from the root</td>
<td>&lt;?xdoxslt:set_variable($_XDOCTX,'currency',/invoice/BILLINFO/CURRENCY)?&gt;</td>
<td>&lt;?xdoxslt:set_variable($_XDOCTX,'currency',/ROWSET/ROW/BUFFER_BUF/invoice/BILLINFO/CURRENCY)?&gt;</td>
</tr>
<tr>
<td>'Contains' function</td>
<td>&lt;?if:contains(EVENT_OBJ,'adjustment')='true'?&gt;</td>
<td>&lt;?if:contains(EVENT_OBJ,'adjustment')?&gt;</td>
</tr>
<tr>
<td>'Doesn’t contain' function</td>
<td>contains(EVENT_OBJ,'writeoff_reversal')!='true'</td>
<td>not(contains(EVENT_OBJ,'writeoff_reversal'))</td>
</tr>
</tbody>
</table>

**Configuring the JAVA_HOME Environment Variable**

The pin_inv_doc_gen utility requires JDK 1.6.0_37 or above and that the JAVA_HOME environment variable be set to the absolute path of the JDK 1.6 software installation.
To set the JAVA_HOME environment variable:

1. Go to BRM_Home/apps/pin_inv_doc_gen.

2. Open the `docgen.sh` file in a text editor.

   **Note:** The `docgen.sh` file is a batch script for setting environment variables and launching the `pin_inv_doc_gen` utility.

3. Search for the following line:

   ```
   $JAVA_HOME/bin/java invoicedocgen.pin_inv_doc_gen -status pending
   ```

4. Replace with the following:

   ```
   BRM_HOME/jdk1.6.0_37/bin/java invoicedocgen.pin_inv_doc_gen -status pending
   ```

5. Save and close the file.

### Configuring the Infranet.properties File for `pin_inv_doc_gen` with BI Publisher 11g

To configure the `Infranet.properties` file for `pin_inv_doc_gen`:

1. Go to BRM_Home/apps/pin_inv_doc_gen.

2. Save the `Infranet.properties.sample` file as `Infranet.properties`.

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

   **Note:** The user must have read-write permissions to access the `/invoice` object. The BRM system administrator creates this user and grants the required permissions.

3. Open the `Infranet.properties` file in a text editor and provide the required values for configuring `pin_inv_doc_gen`.

   Table 2–5 contains a list of `Infranet.properties.sample` configuration entries for the `pin_inv_doc_gen` utility.

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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.chunksiz e | 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. |
Configuring the **docgen.sh** File

To configure the **docgen.sh** file:

1. Go to `BRM_Home/apps/pin_inv_doc_gen`.
2. Open the **docgen.sh** file in a text editor.
3. Edit the default values available in the file to the values specific to your system environment.
4. Save and close the file.

### Table 2–5 (Cont.) Infranet.properties.sample File Entries

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>infranet.burst.threadpool.size</td>
<td>Specifies the number of core threads in the pool.</td>
</tr>
<tr>
<td>infranet.connection</td>
<td>Specifies the URL to the BRM service.</td>
</tr>
<tr>
<td>infranet.INITIAL_CONTEXT_FACTORY</td>
<td>Environment property that specifies the initial context factory to use.</td>
</tr>
<tr>
<td>infranet.INITIAL_CONTEXT_FACTORY</td>
<td>Note: Do not modify the default value, “weblogic.jndi.WLInitialContextFactory”.</td>
</tr>
<tr>
<td>infranet.log.file</td>
<td>Specifies the log file name for the <strong>pin_inv_doc_gen</strong> utility.</td>
</tr>
<tr>
<td>infranet.log.level</td>
<td>Specifies the level of log for the <strong>pin_inv_doc_gen</strong> utility.</td>
</tr>
<tr>
<td>infranet.login.type</td>
<td>Specifies the type of login. A type 1 login requires the application to provide a user name and password. A type 0 login in a trusted login that comes through a Connection Manager (CM) proxy. It doesn’t require a user name or password. Use 1 or 0.</td>
</tr>
<tr>
<td>infranet.dmsearchargs.size</td>
<td>Specifies the number of account and bill unit pairs the utility should process at one time. The maximum value is 14.</td>
</tr>
<tr>
<td>infranet.dupinvdir.path</td>
<td>Specifies the directory path for storing duplicate invoice documents.</td>
</tr>
<tr>
<td>infranet.polling.sleepTime</td>
<td>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).</td>
</tr>
<tr>
<td>infranet.PROVIDER_URL</td>
<td>Environment property that specifies the BI Publisher provider URL in the following format: t3s://BIP_Hostname:Port where BIP_Hostname is where BI Publisher is installed and Port is the port number. Note: Enable HTTPS in BI Publisher by using the t3s (secure) protocol.</td>
</tr>
<tr>
<td>infranet.SECURITY_CREDENTIALS</td>
<td>Environment property that specifies the credentials of the principal for authenticating the caller to the service. Note: The password can be encrypted with Advanced Encryption Standard (AES) for security.</td>
</tr>
<tr>
<td>infranet.SECURITY_PRINCIPAL</td>
<td>Environment property that specifies the identity of the principal for authenticating the caller to the service.</td>
</tr>
<tr>
<td>infranet.threadpool.fetchsize</td>
<td>Specifies the number of records that can be fetched by the thread. The default value is 5.</td>
</tr>
<tr>
<td>infranet.threadpool.maxsize</td>
<td>Specifies the maximum number of threads. The default value is 5.</td>
</tr>
<tr>
<td>infranet.threadpool.size</td>
<td>Specifies the number of threads. The default value is 3.</td>
</tr>
</tbody>
</table>

4. Save and close the file.
Changing Command-Line Parameters for pin_inv_doc_gen

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

1. Go to BRM_Home/apps/pin_inv_doc_gen.
2. Open the docgen.sh file 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. Go to the BRM_Home/apps/pin_inv_doc_gen directory
7. Enter the following command which runs the docgen.sh file:
   
   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 the pin_inv_doc_gen utility.

To configure HTTPS for secure communication:

1. Enable HTTPS in BI Publisher. See the BI Publisher documentation for details.
2. Open the Infranet.properties file in a text editor.
3. Set and edit infranet.bip.url to use HTTPS and the SSL port. For example:
   
   infranet.bip.url=https://BIP_Hostname:SSL_port/xmlpserver
4. Modify infranet.PROVIDER_URL to use T3S and the SSL port. For example:
   
   infranet.PROVIDER_URL=t3s://BIP_Hostname:SSL_port
5. Launch pin_inv_doc_gen using docgen.sh.
6. 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 job ID and the Invoice_POID for which an invoice document is generated.

To create the XMLP_INVOICE_JOB table in the Scheduler database:

1. Go to `BRM_Home/apps/pin_inv_doc_gen`.
2. Open the `Create_Xmlp_Invoice_Job.sql` file.
3. Search for `<tablespacename>`.
4. Replace `<tablespacename>` with the tablespace name where all the BI Publisher Scheduler tables in the BI Publisher Scheduler database are created.
5. Save and close the file.
7. At the prompt, enter the following command which opens SQL*Plus:

   ```
   sqlplus user_name/password@Database_Name
   ```

   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.
8. At the SQL prompt, enter the following command:

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

      ```
      Name=invoiceXREF
      JNDI name= jdbc/OracleBRM/invoiceXREF
      ```
   g. Click Next.
   h. Enter the Scheduler database details that you used when you installed the XMLP_INVOICE_JOB table.
   i. Click Next.
   j. Select the server on which the BI Publisher software is installed.
   k. Click Finish.
   l. 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:

1. Go to `BRM_Home/apps/pin_inv_doc_gen`.
2. Enter the following commands:
   ```
   sqlplus user/password@ORACLE_SID
   >&invoice_bursting.plb;
   quit;
   ```

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 11g.
   - 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
     ```
Designing and Generating Invoices in Oracle Business Intelligence Publisher 11g

Generating BI Publisher Invoice Documents in BI Publisher 11g

You use the `pin_inv_doc_gen` 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 `pin_inv_doc_gen`, the utility fetches the corresponding invoice POID, BI Publisher report name, and layout template name of an /invoice 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 11g".

To run the `pin_inv_doc_gen` utility:

1. Configure the `docgen.sh` file to execute 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.

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.
3. Enter the following command which runs the `docgen.sh` file:

    `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 subordinate accounts, the `/invoice` objects of the subordinate account are not selected for generating the invoice document.

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

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

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

   **Note:** You can use the `BRM_Home/apps/pin_inv_doc_gen/InvoiceList.xml` file as an example.

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

   `invoicedocgen.pin_inv_doc_gen -status pending`

6. Replace with:

   `invoicedocgen.pin_inv_doc_gen -accts_list InvoiceList`

   where `InvoiceList` specifies the name and location of the file that includes a list of accounts and bill units for which to create invoice documents. This is the file you created in step 1.

7. Save and close the file.
8. Go to the BRM_Home/apps/pin_inv_doc_gen directory.
9. Enter the following command which runs the docgen.sh file:
   
   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_Home/apps/pin_inv_doc_gen directory.
2. Open the docgen.sh file in a text editor.
3. Search for the following line:
   
   invoicedocgen.pin_inv_doc_gen -status pending

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

5. Save and close the file.
6. Go to the BRM_Home/apps/pin_inv_doc_gen directory.
7. Enter the following command which runs the docgen.sh file:
   
   docgen.sh

Generating Duplicate Invoice Documents with BI Publisher 11g

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_Home/apps/pin_inv_doc_gen directory.
2. Make sure the infranet.dupinvdir.path entry in the pin_inv_doc_gen utility 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 11g".
3. Open the docgen.sh file in a text editor.
4. Search for the following line:
   
   invoicedocgen.pin_inv_doc_gen -status pending

5. Replace with:
   
   invoicedocgen.pin_inv_doc_gen -status generated

6. Save and close the file.
7. Go to the BRM_Home/apps/pin_inv_doc_gen directory.
8. Enter the following command which runs the docgen.sh file:
   
   docgen.sh
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 Oracle Fusion Middleware Report Designer’s Guide for Oracle Business Intelligence 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

Important: 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 customize the invoice documents by:

- Customizing Invoice Layout Templates
- Enriching Invoice Data

Customizing Invoice Layout Templates

You can customize the invoice layout template by:
Updating the existing layout template file

Adding a new layout template file

Updating the Invoice Layout Template File

You can update 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 of creating an RTF template in the BI Publisher documentation.

Adding a New Invoice Layout Template

To add a new layout template:

1. Create a new layout template in BI Publisher. For information on how to create a layout template, see the discussion of creating an RTF template in the BI Publisher documentation.

2. Create a new invoicing \config\business_profile object in BRM to reflect the addition of the new layout template. See “Specifying BI Publisher Invoice Report and Template Names in BRM” and "About Associating Bill Units with a BI Publisher Invoice and Report”.

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.

To enrich BRM invoice data:

- The /invoice objects containing data in XML format should be present in the BRM database.
- The additional data source must be configured in 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.
Specifying the Default Format in Which to Store Invoices in BRM

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

You modify the /config/business_params object by using the pin_bus_params utility. For information on this utility, see "pin_bus_params" in BRM Developer’s Guide.

To specify the default format in which to store invoices:

1. Go to the BRM_Home/sys/data/config directory.
2. Run the following command, which creates an editable XML file for the invoicing parameter class:

   `pin_bus_params -r BusParamsInvoicing bus_params_Invoicing.xml`

   This command creates the XML file named bus_params_Invoicing.xml.out in your working directory. To place this file in a different directory, specify the path as part of the file name.
3. Open the bus_params_Invoicing.xml.out file.
4. Search for following line:

   `<InvoiceStorageType>1</InvoiceStorageType>`

5. Do one of the following:
   - To specify flist as the output format, set InvoiceStorageType to 0.
   - To specify XML as the output format, set InvoiceStorageType to 1.
6. Save this file as bus_params_Invoicing.xml.
7. Go to the BRM_Home/sys/data/config directory, which includes support files used by the pin_bus_params utility.
8. Run the following command, which loads this change into the appropriate /config/business_params object:

   `pin_bus_params PathToWorkingDirectory/bus_params_Invoicing.xml`

   where PathToWorkingDirectory is the directory in which bus_params_Invoicing.xml resides.

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

   **Note:** To run this command from a different directory, see "pin_bus_params" in BRM Developer’s Guide.

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

   See "Using testnap" in BRM Developer’s Guide for general instructions on using the testnap utility. See "Reading Objects by Using Object Browser" in BRM Developer’s Guide for information on how to use Object Browser.
10. Stop and restart the CM. For more information, see “Starting and Stopping the BRM System” in BRM System Administrator’s Guide.

11. (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.
Specifying the Default Format in Which to Store Invoices in BRM
This chapter describes how to design, generate, and customize Oracle Communications Billing and Revenue Management (BRM) invoice documents using Oracle Business Intelligence (BI) Publisher 10g.

Before reading this document, you should be familiar with the following:

- Basic database administration concepts. See your database documentation.
- BI Publisher 10g functions. See your BI Publisher documentation.

**About BRM Integration with BI Publisher for Invoicing**

BRM comes with a pre-integrated invoicing framework. This framework uses BI Publisher as the publishing application, allowing users to create more stylized and detailed invoices in PDF format. BRM also includes sample RTF templates for individual customers and for corporate customers. In these templates, some of the core business functions are pre-built and can be customized. You can configure and extend these templates for your specific business needs.

The integration methodology is based on the J2EE framework and Web Services-based communication, enabling a complete open-ended, seamless and scalable solution. BI Publisher allows you to choose from a variety of desktop tools, such as Microsoft Word and Adobe Acrobat, to create document templates. It can also access multiple data sources, such as database and HTTP, and it provides multiple document output options, such as email and FTP.

The pre-integrated invoicing framework eliminates the need for any proprietary invoice design tools, resulting in lowered costs. BI Publisher leverages familiar desktop software, so using the software requires minimal learning.

To use this feature, you must install BI Publisher, BI Publisher Desktop, and a web services component.

**Overview of BRM and BI Publisher Invoice Generation**

BI Publisher is an enterprise reporting solution to author, manage, and deliver all types of formatted documents. You can customize the layout of the generated invoice document using the template builder capabilities of BI Publisher. A typical invoice document generated using BRM-BI Publisher integration is in "Understanding Invoice Layout".

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 and stores it in the BI Publisher database.

Figure 3–1 shows the BRM and BI Publisher invoice generation process:

Figure 3–1  BRM and BI Publisher Invoice Generation Process

About the Invoicing Process

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

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

1. **BRM associates bill units with a BI Publisher invoice and report.** When accounts are created, BRM associates each bill unit with a BI Publisher invoice report and layout template. See "About Associating Bill Units with a BI Publisher Invoice and Report".

2. **Integrating BRM and BI Publisher for generating invoice documents.** To generate invoice documents using BRM-BI Publisher integration framework, you must enable the integration between BRM and BI Publisher. See "Enabling the BRM-BI Publisher Integration".

3. **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*.

4. **BRM generates the invoice and stores it in the BRM database.** When the `pin_inv_accts` utility is executed, 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". If the invoice data is stored in pin_flist format, you must perform additional steps when deploying the custom BRM Web service. See "Deploying the BRM Web Service".

---

5. **The `pin_inv_doc_gen` utility:**

   - **Executes BI Publisher reports.** This utility, based on the input parameters, invokes and executes appropriate BI Publisher reports for invoice documents. It executes a BI Publisher invoicing report, with the BRM Web service as the data source, through a scheduled job. See "Generating BI Publisher Invoice Documents".

   - **Calls `PublicBRMfetchInvoiceXMLService` to pull the XML data from the BRM database.** The BRM Web service, `PublicBRMfetchInvoiceXMLService`, 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.

   - **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".

**About Associating Bill Units with a BI Publisher Invoice and Report**

When the BRM-BI Publisher integration is enabled, BRM automatically associates bill units with a BI Publisher report name and invoice template name during the account creation process. When an account is created, BRM performs the following for each bill unit ( `/billinfo` object) in the account:

- Determines the BI Publisher invoice report and template that the bill unit qualifies for by reading the `/config/business_profile` object. This object defines your invoice types, the criteria a bill unit must meet for the invoice type, and the BI Publisher invoice report and template names associated with the invoice type. For example, a `/config/business_profile` might define the following as shown in Table 3–1, for a regular bill:

**Table 3–1  Business Profile Example**

<table>
<thead>
<tr>
<th>Invoice Type</th>
<th>Criteria</th>
<th>BI Publisher Invoice Report and Template Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monthly billing</td>
<td>The <code>/billinfo</code> object’s PIN_FLD_WHEN field is set to 1.</td>
<td><code>monthly_invoice_report</code> and <code>monthly_invoice</code></td>
</tr>
<tr>
<td>Quarterly billing</td>
<td>The <code>/billinfo</code> object’s PIN_FLD_WHEN fields is set to 3.</td>
<td><code>quarter_invoice_report</code> and <code>quarter_invoice</code></td>
</tr>
</tbody>
</table>
Designing Invoices

- Creates an `/associated_bus_profile` object for the bill unit. This object stores the invoice template name in the PIN_FLD_TEMPLATE_NAME field and the invoice report name in the PIN_FLD_REPORT_NAME field.

  This object stores the following information associated with invoicing for the bill unit as an element in a template array called PIN_FLD_TEMPLATE_ARRAY.

- Associates the bill unit with its `/associated_bus_profile` object by populating the `/billinfo` object’s PIN_FLD_ASSOC_BUS_PROFILE_OBJ_LIST field.

BRM creates an `/associated_bus_profile` object for each bill unit in an account by calling the PCM_OP_CUST_CREATE_ASSOCIATED_BUSPROFILE opcode. This opcode is called internally by the PCM_OP_CUST_CREATE_BILLINFO opcode during the account creation process.

Whenever invoice business profiles are modified in the `/config/business_profile` object, use the PCM_OP_CUST_SET_ASSOCIATED_BUSPROFILE opcode to update all related `/associated_bus_profile` objects.

About BRM-Business Intelligence Publisher Invoicing Integration Package Components

The BRM-BI Publisher invoicing integration package contains:

- The Java integration utility, `pin_inv_doc_gen`. See "Generating BI Publisher Invoice Documents".
- Custom BRM Web service.
- Six BI Publisher reports for generating invoice documents for consumer and corporate accounts. See "About BI Publisher Reports for Invoice Document Generation".
- Six sample layout templates in rich text format (RTF) for consumer and corporate accounts. See "About Invoice Layout Templates for Invoice Document Generation".

Designing Invoices

You can use the following capabilities of BI Publisher to design 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".

About Invoice Layout Templates for Invoice Document Generation

The BRM-BI Publisher invoicing integration package contains two layout templates in RTF for the consumer and corporate account. The layout templates have the XSL formatting code that is understood by the BI Publisher reporting engine that guides the look and feel of the invoice document.

Layout templates that are part of the BRM-BI Publisher invoicing integration package are based on the invoice document and whether the invoice is intended for a corporate or consumer customer. If the BRM-BI Publisher integration framework is enabled, when you create an account, the bill units (`/billinfo` objects) of the account are associated with an `/associated_bus_profile` object. The `/associated_bus_profile` object contains the report and template names that are used for generating invoice documents for this account on the BI Publisher server. See "Specifying BI Publisher
Invoice Report and Template Names in BRM”.

**Default Templates for Regular Invoices**

BRM provides the following default templates for regular invoices:

- BRM_Corporate_Invoice.rtf
- BRM_Consumer_Invoice.rtf

**Report Files for Default Invoice Templates**

The report files that BI Publisher associates with each the default templates for invoices are:

- BRM_Corporate_Invoice_Report.xdo
- BRM_Consumer_Invoice_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 PublicBRMfetchInvoiceXMLService Web service.
- 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, HTML, and XML (data). The default output format specified in the reports is PDF.

---

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

Reports that are part of the BRM-BI Publisher invoicing integration package:

- BRM_Consumer_Invoice_Report.xdo
- BRM_Corporate_Invoice_Report.xdo

**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.
- In BRM, specify the BI Publisher invoice report and layout templates names. See "Specifying BI Publisher Invoice Report and Template Names in BRM”.
- 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 Taxation Details in Invoices”.
- Install the BRM-BI Publisher invoicing integration package on the BI Publisher server. See "Installing BRM-BI Publisher Invoicing Integration Package".
Specifying BI Publisher Invoice Report and Template Names in BRM

When the BRM-BI Publisher integration is enabled, BRM automatically performs the following during the bill unit (/billinfo object) creation process:

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

1. 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. See "Creating Bill Unit Validation Templates for Invoices".
3. Specify the validation template, invoice report, and layout template to be associated with the bill unit. See "Specifying the BI Publisher Report and Template".
4. Load the business profile information into the BRM database by using the load_pin_business_profile utility. See "Loading Invoicing Business Profiles into the Database".
5. If your system includes accounts that were created before you integrated BRM invoicing with 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 Publisher Integration is disabled in BRM. You can enable this feature by modifying a field in the invoicing instance of the /config/business_params object.

You modify the /config/business_params object by using the pin_bus_params utility. For information on this utility, see "pin_bus_params" in BRM Developer’s Guide.

To enable the BRM-BI Publisher integration for invoices:

1. Go to the BRM_Home/sys/data/config directory, where BRM_Home is the directory in which you installed BRM components.
2. Run the following command, which creates an editable XML file from the invoicing instance of the /config/business_params object:
Specifying BI Publisher Invoice Report and Template Names in BRM

Designing and Generating Invoices in Oracle Business Intelligence Publisher 10g

3. Open the bus_params_Invoicing.xml.out file.
4. Search for the following line:
   <EnableInvoicingIntegration>disabled</EnableInvoicingIntegration>
5. Change disabled to enabled.
6. Save the file as bus_params_Invoicing.xml.
7. Go to the $HOME/sys/data/config directory, which includes support files used by the pin_bus_params utility.
8. Run the following command, which loads this change into the /config/business_params object:
   
   pin_bus_params PathToWorkingDirectory/bus_params_Invoicing.xml
   
   where PathToWorkingDirectory is the directory in which bus_params_Invoicing.xml resides.

---

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

---

**Note:** To run this command from a different directory, see “pin_bus_params” in BRM Developer’s Guide.

---

9. Read the object with the testnap utility or Object Browser to verify that all fields are correct.
   
   See "Using testnap" in BRM Developer’s Guide for general instructions on using the testnap utility. See "Reading Objects by Using Object Browser" in BRM Developer’s Guide for information on how to use Object Browser.

10. Stop and restart the Connection Manager (CM). For more information, see "Starting and Stopping the BRM System" in BRM System Administrator’s Guide.

11. (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.

---

Creating Bill Unit Validation Templates for Invoices

You specify the criteria a bill unit must meet to be associated with an invoicing configuration business profile by creating a bill unit validation template (/config/template/billinfo). For more information about validation templates, see "About Validation Templates" in BRM Managing Customers.

To create a bill unit validation template for invoices, edit the $HOME/sys/data/config/pin_business_profile.xml file. The file specifies the following:
Specifying BI Publisher Invoice Report and Template Names in BRM

- The valid business profiles for each invoice type
- The criteria a bill unit must meet to qualify for each business profile
- The templateID name.

**Note:** Each type of invoice can have multiple business profiles, but they must all point to the same TemplateID name.

The default `pin_business_profile.xml` file includes entries for the consumer and corporate invoice documents. You can edit these entries or add entries for any other type of invoice document you want to generate.

For each type of invoice document you want to generate, edit the following entries:

- Set the **TemplateId name** entry to **InvoiceIntegration** and the **type** entry to `/billinfo`, as shown below:

  ```xml
  <TemplateId name="InvoiceIntegration" type="/billinfo"/>
  ```

- Specify the criteria a bill unit must meet to be associated with an invoicing business profile in the **Iscript** section. Enter your conditions in the **if** and **else if** blocks and the name of the business profile in PIN_FLD_NAME.

  **Important:** The order in which criteria is written in the **if** and **else if** blocks defines the order in which a bill unit is evaluated against the criteria.

If a particular condition is satisfied in the **if** and **else if** blocks, the business profile name associated with the condition is set with the **OUT** label as one of the Iscript return values.

For example:

```xml
<Iscript>
Long parentFlags = flistGetLong(IN, PIN_FLD_PARENT_FLAGS);

if(parentFlags == 0)
{
    flistSetEnum(OUT, PIN_FLD_RESULT, 1);
    flistSetStr(OUT, PIN_FLD_NAME, "ConsumerInvoice");
}
else if(parentFlags > 0)
{
    flistSetEnum(OUT, PIN_FLD_RESULT, 1);
    flistSetStr(OUT, PIN_FLD_NAME, "CorporateInvoice");
}
</Iscript>
```

**Specifying the BI Publisher Report and Template**

You specify the BI Publisher invoice reports and layout template names by editing the business profile configuration file (**pin_business_profile.xml**). The default XML file contains entries for generating a consumer invoice document. You can modify these entries and add entries for any other type of invoice document that you want to generate. For example, you can add entries to generate an invoice document for Internet customers or wireless telephony customers.
To specify the BI Publisher invoice report and layout template to use for generating an invoice document, edit the following entries in the `BRM_Home/sys/data/config/pin_business_profile.xml` file. Add a set of entries for each type of invoice document you want to generate:

- Specify the name of the business profile in the `BusinessProfile name` entry. The name must match the PIN_FLD_NAME entry from the Iscript element.

  **Important:** The `type` attribute entry for the BusinessProfile tags must be set to `Invoice`.

For example:

```xml
<BusinessProfile name="ConsumerInvoice" type="Invoice">
  <Desc> Description of the business profile </Desc>
  <!-- List templates -->
  <TemplateId name="InvoiceIntegration" type="/billinfo"/>
  <!-- List key values -->
  <NameValue key="Report_Name" value="BRM_Consumer_Invoice_Report"/>
  <NameValue key="Template_Name" value="BRM_Consumer_Invoice"/>
</BusinessProfile>
```

- Make sure `TemplateId name` is set to `InvoiceIntegration` and the `type` is set to `/billinfo`. All invoice business profiles must use these values.

```xml
<TemplateId name="InvoiceIntegration" type="/billinfo"/>
```

- Specify the name of the BI Publisher XDO report to use for generating invoice documents in the `Report_Name` value entry. For example:

```xml
{NameValue key="Report_Name" value="BRM_Consumer_Invoice_Report"}
```

- Specify the name of the BI Publisher XDO report to use and the name of the layout template to use for each type of invoice, the regular invoice, the Replacement Invoice and the Invoice Correction Letter.

The following is a sample of the BusinessProfile tag entries for the default Corporate Invoice type in the `pin_business_profile.xml` file:

```xml
<BusinessProfile name="CorporateInvoice" type="Invoice">
  <Desc> Description of the business profile </Desc>
  <!-- List templates -->
  <TemplateId name="InvoiceIntegration" type="/billinfo"/>
  <!-- List key values -->
  <NameValue key="Report_Name" value="BRM_Corporate_Invoice_Report"/>
  <NameValue key="Template_Name" value="BRM_Corporate_Invoice"/>
</BusinessProfile>
```

**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 that were created prior to your BRM-BI Publisher integration will not be linked with 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 7.5 from a prior release.

To generate BI Publisher invoice documents for `/billinfo` objects created prior to 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 with its /associated_bus_profile object by populating the /billinfo object’s PIN_FLD_ASSOC_BUS_PROFILE_OBJ_LIST field.

To associate pre-existing bill units with business profiles:

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

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

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

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

3. Go to the BRM_Home/apps/pin_billd directory.

   **Note:** The utility 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.

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.

   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.

**About the /associated_bus_profile Object**

If you are using BRM-BI Publisher integration framework to generate invoice documents, at the time of customer account creation, the /associated_bus_profile object is created. See "Enabling the BRM-BI Publisher Integration". The /associated_bus_profile object stores the invoicing business profile information for a /billinfo object. The /associated_bus_profile object contains the layout template name in the PIN_FLD_TEMPLATE_NAME field and the report name in the PIN_FLD_REPORT_NAME field.
Creating /associated_bus_profile Objects
If the BRM-BI Publisher invoicing integration is enabled, during customer account creation, internally the PCM_OP_CUST_CREATE_BILLINFO opcode calls the PCM_OP_CUST_CREATE_ASSOCIATED_BUS_PROFILE opcode to create one /associated_bus_profile object for each bill unit in the account.

Creating /associated_bus_profile Objects for Accounts in a Branded Environment

To create /config/business_profile objects for accounts in a branded environment:

1. In BRM_Home/sys/data/config directory, create a new directory for the brand.
   For example, BrandA.
2. Copy pin.conf, business_configuration.xsd, and pin_business_profile.xml files from BRM_Home/sys/data/config to BRM_Home/sys/data/config/BrandA.
3. Open pin.conf and add the following entries:
   
   ```
   nap login_name brandloginname
   nap login_pw brandpassword
   ```

   where brandloginname is the login name for the brand and brandpassword is the password for the brand.
4. Save and close the file.
5. Open pin_business_profile.xml and replace the default values with the new values listed in Table 3–2:

   Table 3–2 Default Values in pin_business_profile.xml File
<table>
<thead>
<tr>
<th>Entry</th>
<th>Default Value</th>
<th>New Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BusinessProfile name</td>
<td>ConsumerInvoice</td>
<td>ConsumerInvoice_BrandA</td>
</tr>
<tr>
<td>TemplateId name</td>
<td>InvoiceIntegration</td>
<td>InvoiceIntegration_BrandA</td>
</tr>
<tr>
<td>NameValue value</td>
<td>BRM_Consumer_Invoice_Report</td>
<td>BRM_Consumer_Invoice_Report_BrandA</td>
</tr>
<tr>
<td>NameValue value</td>
<td>BRM_Consumer_Invoice</td>
<td>BRM_Consumer_Invoice_BrandA</td>
</tr>
<tr>
<td>BusinessProfile name</td>
<td>CorporateInvoice</td>
<td>CorporateInvoice_BrandA</td>
</tr>
<tr>
<td>TemplateId name</td>
<td>InvoiceIntegration</td>
<td>InvoiceIntegration_BrandA</td>
</tr>
<tr>
<td>NameValue value</td>
<td>BRM_Corporate_Invoice_Report</td>
<td>BRM_Corporate_Invoice_Report_BrandA</td>
</tr>
<tr>
<td>NameValue value</td>
<td>BRM_Corporate_Invoice</td>
<td>BRM_Corporate_Invoice_BrandA</td>
</tr>
<tr>
<td>Template name</td>
<td>InvoiceIntegration</td>
<td>InvoiceIntegration_BrandA</td>
</tr>
<tr>
<td>flistSetStr PIN_FLD_NAME</td>
<td>ConsumerInvoice</td>
<td>ConsumerInvoice_BrandA</td>
</tr>
<tr>
<td>flistSetStr PIN_FLD_NAME</td>
<td>CorporateInvoice</td>
<td>CorporateInvoice_BrandA</td>
</tr>
</tbody>
</table>
6. Save and close the file.
7. Go to the BRM_Home/sys/data/config/BrandA directory and enter the following command:
   ```
   load_pin_business_profile pin_business_profile.xml
   ```
8. Read the object with the testnap utility or Object Browser to verify that all fields are correct.
Creating /associated_bus_profile Objects for Accounts in a Branded Environment


Creating Reports for Accounts in a Branded Environment in BI Publisher

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.

2. Log in to BI Publisher.

3. Create a new RTF template to associate it with the layout template specific for the accounts in a brand. See the discussion of creating an RTF template using the Template Builder for Word in the BI Publisher documentation.

4. In Shared Folders, click BRM_Invoices.

5. Click Create a new report link.

6. Enter the report name as BRM_Bursting_Invoice_Report_BrandA.

   Note: The report name must be same as the report name specified in the updated pin_business_profile.xml file.

7. For the BRM_Bursting_Invoice_Report_BrandA report, click Edit. To specify report properties, see the discussion of creating the report entry and specify general properties in the BI Publisher documentation.

   Note: The configuration of the BRM_Bursting_Invoice_Report_BrandA report must be same as the sample BRM_Bursting_Invoice_Report report in the BRM-BI Publisher invoicing integration package. For example, the BRM_Bursting_Invoice_Report_BrandA report must have the same WSDL URL as given in the BRM_Bursting_Invoice_Report report.

   ■ The layout template name must be BRM_Consumer_Invoice_BrandA.

8. Click Save.
Installing 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.

Windows Installation
To install the BRM-BI Publisher invoicing integration package:

- Follow the pre-installation requirements. See "Pre-Installation Requirements".
- Set the BI Publisher environment variables. See "Setting the BI Publisher System Environment Variable".
- Run the BRM-BI Publisher invoicing integration package. See "Installing the BRM-BI Publisher Invoicing Integration Package Application".
- Set the LD_LIBRARY_PATH environment variable. See "Setting the LD_LIBRARY_PATH Environment Variable".

Pre-Installation Requirements
Before running the BRM-BI Publisher invoicing integration package application, do the following:

- Install Java 1.5 (or higher version).
- Set the BIP_HOME and JAVA_HOME environment variables. See "Setting the BI Publisher System Environment Variable".

Note: After you install BI Publisher on your machine, the BIP_HOME environment variable is created. If the BIP_HOME environment variable is not set, the BRM-BIP invoicing integration package application will quit.

Setting the BI Publisher System Environment Variable
To set the BI Publisher home environment variable:

1. On the desktop, right-click **My Computer**, and select **Properties** from the shortcut menu.
2. Click the **Advanced** tab.
3. Click **Environment Variable**.
4. In **System variables**, click **New**.
5. In the **Variable name** field, enter **BIP_HOME**.
6. In the **Variable value** field, enter the path where the BI Publisher server is installed.
7. Click **OK**.
8. In **Environment Variables** dialog box, click **OK**.
9. In **System Properties** dialog box, click **OK**.

Installing the BRM-BI Publisher Invoicing Integration Package Application
To install the BRM-BI Publisher invoicing integration package application:
Installing BRM-BI Publisher Invoicing Integration Package

1. Go to the directory containing the BRM-BI Publisher invoicing integration package application, 7.4_BRM_BIP_Invoicing_Integration_nt_32_opt.exe.

2. Run 7.4_BRM_BIP_Invoicing_Integration_nt_32_opt.exe.
   The installation wizard starts.

3. Read the instructions and enter the required details in the installation wizard screens.

4. After the installation completes, click Finish.

Setting the LD_LIBRARY_PATH Environment Variable
To set the LD_LIBRARY_PATH environment variable:
1. On the server on which the BRM-BI Publisher integration is deployed, set the LD_LIBRARY_PATH environment variable to portal.dll.
2. Go to BIP_Home\oc4j_bi\j2ee\home\applications\pin_inv_doc_gen.
3. Run the following command:
   BipEnv.bat

Linux Installation
To install the BRM-BI Publisher invoicing integration package:

- Ensure that the pre-installation requirements are satisfied. See "Pre-Installation Requirements".
- Run the BRM-BI Publisher invoicing integration package. See "Enabling the BRM-BI Publisher Invoicing Integration Package Application".
- Set the LD_LIBRARY_PATH environment variable. See "Setting the LD_LIBRARY_PATH Environment Variable".

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

- BI Publisher is installed on Linux.
- BI Publisher home environment variable has been successfully set.

For more information on these requirements, see BRM Reports.

Enabling the BRM-BI Publisher Invoicing Integration Package Application
To enable the BRM-BI Publisher invoicing integration:

1. Download the 7.4_BRM_BIP_Invoicing_Integration_linux_32_opt.bin file from your BI Publisher media pack to a temporary directory (temp_dir).
2. Go to the directory where you installed the Third-Party package and source the source.me file.

Caution: You must source the source.me file to proceed with installation.
If you fail to do so, “suitable JVM not found” and other error messages appear.
For Bash shell:

```bash
source source.me.sh
```

For C shell:

```bash
source source.me.csh
```

3. Go to `temp_dir` and enter the following command:

```bash
7.4_BRM_BIP_Invoicing_Integration_linux_32_opt.bin -console
```

### Setting the LD_LIBRARY_PATH Environment Variable

To set the LD_LIBRARY_PATH environment variable:

1. On the server on which the BRM-BI Publisher integration is deployed, set the LD_LIBRARY_PATH environment variable in your shell startup script (for example, in `.cshrc`) to the directory that has the `libportal.so` shared library.

2. Go to `BIP_Home/oc4j_bi/j2ee/home/applications/pin_inv_doc_gen`.

3. Run the following command:

```bash
./BipEnv.sh
```

### Installed Files

The BRM-BI Publisher invoicing integration program copies the following files listed in Table 3–3 to your computer during installation:

<table>
<thead>
<tr>
<th>Filename</th>
<th>Install Path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BipEnv.bat</td>
<td><code>BIP_home/oc4j_bi/j2ee/home/application\s\pin_inv_doc_gen</code></td>
<td>Batch script for setting environment variables. Before running the BipEnv.bat file, ensure that JAVA_HOME environment variable is set.</td>
</tr>
<tr>
<td>BIPExtension.jar</td>
<td><code>BIP_home/oc4j_bi/j2ee/home/application\s/xmlpserver\xmlpserver\WEB-INF\lib</code></td>
<td>JAR file used by RTF file for date conversions.</td>
</tr>
<tr>
<td>BRM_Consumer_Invoice.rtf</td>
<td><code>BIP_home/xmlp/XMLLP\Report\s\BRM_Invoices\BRM_Consumer_Invoice_Report</code></td>
<td>RTF file associated with layout template for consumer invoice document.</td>
</tr>
<tr>
<td>BRM_Consumer_Invoice_Report.xdo</td>
<td><code>BIP_home/xmlp/XMLLP\Report\s\BRM_Invoices\BRM_Consumer_Invoice_Report</code></td>
<td>XDO file for consumer invoice document.</td>
</tr>
<tr>
<td>BRM_Corporate_Invoice.rtf</td>
<td><code>BIP_home/xmlp/XMLLP\Report\s\BRM_Invoices\BRM_Corporate_Invoice_Report</code></td>
<td>RTF file associated with layout template for corporate invoice.</td>
</tr>
<tr>
<td>BRM_Corporate_Invoice_Report.xdo</td>
<td><code>BIP_home/xmlp/XMLLP\Report\s\BRM_Invoices\BRM_Corporate_Invoice_Report</code></td>
<td>XDO file for corporate invoice document.</td>
</tr>
<tr>
<td>Create_Xmlp_Invoice_Job.sql</td>
<td><code>BIP_home/oc4j_bi/j2ee/home/application\s\pin_inv_doc_gen</code></td>
<td>Script for creating XMLP_INVOICE_JOB table in BI Publisher Scheduler database.</td>
</tr>
</tbody>
</table>
Configuring the pin_inv_doc_gen Utility

Configuring the pin_inv_doc_gen Utility involves:

- Configuring the Infranet.properties File for pin_inv_doc_gen
- Configuring the BipEnv.bat File
- Changing Command-Line Parameters for pin_inv_doc_gen
- Setting Scheduler Configuration
- Configuring the Email Server
- Creating the XMLP_INVOICE_JOB Table in the Scheduler Database

Configuring the Infranet.properties File for pin_inv_doc_gen

To configure the Infranet.properties file for pin_inv_doc_gen:

1. Go to BIP_Home/oc4j_bi/j2ee/home/applications/pin_inv_doc_gen.
2. Open the Infranet.properties file in a text editor and provide the required values for configuring pin_inv_doc_gen.

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

Note: The user must have read-write permissions to access the /invoice object. The BRM system administrator creates this user and grants the required permissions.

Table 3–4 shows the entries for this utility in the Infranet.properties file.

### Table 3–3 (Cont.) BI Publisher-Related Files Placed on Your Computer

<table>
<thead>
<tr>
<th>Filename</th>
<th>Install Path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infranet.properties</td>
<td>BIP_home\oc4j\bi\j2ee\home\applications\pin_inv_doc_gen</td>
<td>Infranet.properties file for pin_inv_doc_gen. Contains the configuration entries for the pin_inv_doc_gen utility.</td>
</tr>
<tr>
<td>InvoiceList.xml</td>
<td>BIP_home\oc4j\bi\j2ee\home\applications\pin_inv_doc_gen</td>
<td>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.</td>
</tr>
<tr>
<td>InvoiceList.xsd</td>
<td>BIP_home\oc4j\bi\j2ee\home\applications\pin_inv_doc_gen</td>
<td>XSD file that validates the InvoiceList.xml.</td>
</tr>
<tr>
<td>pin_inv_doc_gen.jar</td>
<td>BIP_home\oc4j\bi\j2ee\home\applications\pin_inv_doc_gen</td>
<td>JAR file for pin_inv_doc_gen application.</td>
</tr>
<tr>
<td>PublicBRMfetchInvoiceXMLService.war</td>
<td>BIP_home\oc4j\bi\j2ee\home\applications\pin_inv_doc_gen\WS-fetchInvoiceXML</td>
<td>Application WAR file for the custom BRM Web service to be deployed on to the OC4J application server.</td>
</tr>
</tbody>
</table>
### Table 3–4 Infranet.properties File Entries

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>infranet.bip.deliveryoption1</td>
<td>Specifies the first delivery option for the invoice document. See Setting Up Delivery Options in your BI Publisher documentation.</td>
</tr>
<tr>
<td>infranet.bip.deliveryoption2</td>
<td>Specifies the second delivery option for the invoice document.</td>
</tr>
<tr>
<td>infranet.bip.emailfromaddress</td>
<td>Specifies the email address of the sender who sends the Scheduler delivery request.</td>
</tr>
<tr>
<td>infranet.bip.ftpremotedirectory</td>
<td>Specifies the ftp remote directory for ftp delivery option.</td>
</tr>
</tbody>
</table>
| infranet.bip.ftpsecureftp | Valid entries are:  
  - false. (Default)  
  - true]                                                                                                                                          |
| infranet.bip.ftpservername | Specifies the ftp server name for ftp delivery option.                                                                                           |
| infranet.bip.ftpusername   | Specifies the ftp user name for ftp delivery option.                                                                                             |
| infranet.bip.ftpuserpassword | Specifies the ftp user password for ftp delivery option.                                                                                         |
| infranet.bip.outputdocformat | Specifies the output format of the invoice document: PDF or HTML. See “Specifying the Utility Output Format”.                                  |
| infranet.bip.password      | Specifies the password.  
  **Note:** BI Publisher does not encrypt or decrypt the password. Ensure that the password to connect to the BI Publisher server is valid. |
| infranet.bip.port          | Specifies the BI Publisher port number.                                                                                                            |
| infranet.bip.server        | Specifies the name of the BI Publisher server.                                                                                                    |
| infranet.bip.toemailaddress | Specifies the receiver’s email address who receives the Scheduler delivery request. It also specifies the receiver’s email address who receives the invoice document as an email attachment. |
| infranet.bip.username      | Specifies the user name.                                                                                                                         |
| infranet.log.file          | Specifies the log file name for the PublicBRMfetchInvoiceXMLService Web service.                                                                |
| infranet.log.level         | Specifies the level of log for the PublicBRMfetchInvoiceXMLService Web service.                                                                 |
| 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.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.                                                                                         |

3. Save and close the file.

**Specifying the Utility Output Format**

You can generate the invoice document in PDF, HTML, and XML (data) formats.

To specify the output format for the invoice document:
Configuring the pin_inv_doc_gen Utility

1. Open the `BIP_Home\oc4j_bi\j2ee\home\applications\pin_inv_doc_gen\Infranet.properties` file in a text editor.
2. Set the `infranet.bip.outputdocformat` parameter to: PDF or HTML.
   
infranet.bip.outputdocformat=PDF
3. Save and close the file.

After generating an invoice document, if you want to change the output format, you need to update the `Infranet.properties` file for `pin_inv_doc_gen` with the desired format and regenerate the invoice document.

Configuring the BipEnv.bat File

To configure the `BipEnv.bat` file:

1. Go to `BIP_Home\oc4j_bi\j2ee\home\applications\pin_inv_doc_gen`.
2. Open the `BipEnv.bat` file.
3. Edit the default values available in the file to the values specific to your system environment.
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 `BipEnv.bat` file:

1. Go to the `BIP_Home\oc4j_bi\j2ee\home\applications\pin_inv_doc_gen`.
2. Open the `BipEnv.bat` file 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 `BIP_Home\oc4j_bi\j2ee\home\applications\pin_inv_doc_gen` directory.
7. Enter the following command which runs the `BipEnv.bat` file:
   
   BipEnv.bat

   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.
Creating the XMLP_INVOICE_JOB Table in the Scheduler Database

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

To create XMLP_INVOICE_JOB table in the Scheduler database:

1. Go to BIP_Home\oc4j\bi\j2ee\home\applications\pin_inv_doc_gen.
2. Open Create_Xmlp_Invoice_Job.sql file.
3. Search for <tablespacename>.
4. Replace <tablespacename> with the tablespace name where all the BI Publisher Scheduler tables in the BI Publisher Scheduler database are created.
5. Save and close the file.
6. At the DOS command prompt, go to: BIP_Home\oc4j\bi\j2ee\home\applications\pin_inv_doc_gen.
7. At the prompt, enter the following command to open SQL*Plus:

   sqlplus user_name/password@Database_Name

   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.
8. At the SQL prompt of the Scheduler database connection, enter the following command:

   SQL> @Create_Xmlp_Invoice_Job.sql

Deploying the BRM Web Service

---

**Important:** If the invoice data is stored in pin_flist format instead of XML format, before deploying the BRM Web service, you must create a shared library in the OC4J server to overcome the issue of xmlparserv2.jar of OC4J server. See "Creating Shared Library in OC4J Server".

---

To deploy the BRM Web service, PublicBRMfetchInvoiceXMLService:

1. In a Web browser, enter the following URL to log in to the Oracle Enterprise Manager Application Server Control console:

   http://localhost:port/em

   ■ localhost is the host on which Application Sever Control is installed.
   ■ port is the console port. The default port is 9704.
2. Click the Applications link.
3. Click Deploy.
4. In the Deploy: Select Archive page, click Browse to select the PublicBRMfetchInvoiceXMLService.war file.
Deploying the BRM Web Service

5. Click Next.

6. In the Deploy: Application Attributes page, in the Application Name field, enter PublicBRMfetchInvoiceXMLService.

Note: You must name the Web service as PublicBRMfetchInvoiceXMLService.

7. Click Next.

Important: If the invoice data is stored in pin_flist format, you must perform the additional steps from 8 through 12. If the invoice data is stored in XML format, skip the additional steps from 8 through 12 and start from step 13.

8. In the Deploy: Deployment Settings page, click the Go to Task icon for Configure Class Loading.

The shared libraries currently defined in the server display.

9. Select the Import check box for the shared library that you created. See "Creating Shared Library in OC4J Server".

10. Click Next 10.

11. Clear the Import check box for oracle.xml shared library.

Note: The oracle.xml shared library may appear on another page. If you do not see it, keep clicking the Next 10 link until you find it.

12. Click OK at the top of the page.

13. Click Deploy.

The confirmation of successful deployment of the application appears.

14. Click Return.

15. Edit the Infranet.properties file to define your configuration entries.

Note: After deploying the PublicBRMfetchInvoiceXMLService.war file, the Infranet.properties file is located at BIP_Home\oc4j\bi\j2ee\home\applications\PublicBRMfetchInvoiceXMLService\PublicBRMfetchInvoiceXMLService\WEB-INF\classes directory.

16. Restart your BI Publisher server.

After deploying the Web service, you should test that the Web service is fetching the data correctly. See "Testing the Newly Deployed Web Service".
Creating Shared Library in OC4J Server

To create a shared library:

1. In a Web browser, enter the following URL to log in to the Oracle Enterprise Manager Application Server Control console:
   
   \[\text{http://localhost:port/em}\]

   \[\text{localhost}\] is the host on which Application Server Control is installed.

   \[\text{port}\] is the console port. The default port is \[\text{9704}\].

2. Click the Administration link.

3. Click the Go to Task icon for Shared Libraries.

4. In the Shared Libraries page, click the Create button.

5. In the Create Shared Library: Attributes page, do the following:
   
   a. In the Shared Library Name field, enter \[\text{apachexerces.xml}\].

   b. In the Shared Library Version field, enter \[\text{1.0}\].

   c. Click Next.

6. In the Create Shared Library: Add Archives page, click the Add button.

7. Select the first option and browse for the \[\text{xercesImpl.jar}\] file.

   This file is located at: \[BIP\_Home\oc4j\_bi\assert\lib\xercesImpl.jar\].

8. Click Continue.

9. Click Next.

10. Click Finish.

   The Shared Libraries page displays the newly added shared library.

Testing the Newly Deployed Web Service

1. In the Oracle Enterprise Manager Application Server Control console, click the Application link to access the newly deployed Web service.

2. Click the PublicBRMfetchInvoiceXMLService link.

3. Click the Web Services link and click fetchInvoiceXMLPort.

4. Click the Test Service link.

5. In the Discovered Websites page, click Test Web Service.

6. In the Invoice_POID field, enter an invoice POID for which you want to fetch the invoice data from the BRM database.

---

**Note:** The sample XDO reports that are part of the BRM-BI Publisher invoicing integration package have the default BRM Web service URL defined as:

\[\text{http://localhost:9704/PublicBRMfetchInvoiceXMLService/fetchInvoiceXMLPort?WSDL}\]

Replace the default URL with the URL that you get after deploying the Web service.
7. Click Invoke.
   If the Web service is working correctly, you should be able to view the fetched
   invoice data in XML format.

Configuring How PublicBRMfetchInvoiceXMLService Connects to the
BRM Database

The **Infranet.properties** 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.*

After you deploy the PublicBRMfetchInvoiceXMLService Web service, you can
configure how it connects to the BRM database by using the Web service’s
**Infranet.properties** file.

You can access the **Infranet.properties** file from the
*BIP_Home\oc4j_ bi\j2ee\home\applications\PublicBRMfetchInvoiceXMLService\PublicBRMfetch
InvoiceXMLService\WEB-INF\classes* directory. See "Deploying the BRM Web
Service".

Table 3–4 lists the entries in the **Infranet.properties** file. Open the **Infranet.properties**
file and enter the following required entries:

- **infranet.db.username**
- **infranet.db.sid**
- **infranet.db.log.file**
- **infranet.db.log.level**

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 connection pool and the data source for BRM schemas. See
  "Configuring the Connection Pool and the Data Source" for more information.

- Configure the connection pool and the data source for the BI Publisher Scheduler
database. See "Configuring the Connection Pool and the Data Source" for more
information.

- Add Java Naming and Directory Interface (JNDI) entries for your BI Publisher
administration server. See "Adding JNDI Entries in the Infranet.properties File of
pin_inv_doc_gen" for more information.

Configuring the Connection Pool and the Data Source

To configure the connection pool and the data source for the BRM schema or the BI
Publisher Scheduler database:

1. In a Web browser, enter the following URL to log in to the Oracle Enterprise
   Manager Application Server Control console:

   http://localhost:port/em

   - *localhost* is the host on which Application Server Control is installed.
Configuring the BRM-BI Publisher Invoicing Integration to Support Multischema Systems

1. *port* is the console port. The default port is 9704.
2. Click the **Administration** link.
3. Click the **Go to Task** icon for **JDBC Resources**.
   The **JDBC Resources** page appears.
4. In the **Connection Pools** section, click the **Create** button.
   The Create Connection Pool - Application page opens.
5. Do the following:
   a. From the **Application** list, select **default**.
   b. From the **Connection Pool Type** section, select the **New Connection Pool** option.
   c. Click **Continue**.
      The Create Connection Pool page opens.
6. Do the following:
   a. In the **Name** field, enter the name of the connection pool.
      - For the BRM schema, enter the BRM schema name. For example, enter 0.0.0.1.
      - For the BI Publisher Scheduler database, enter **scheduler**.
   b. In the **Connection Factory Class** field, enter `oracle.jdbc.pool.OracleDataSource`.
   c. In the **JDBC URL** field, enter the connection string.
      - For the BRM schema, enter the connection string to the BRM schema.
      - For the BI Publisher Scheduler database, enter the connection string to the BI Publisher Scheduler database.
   d. (Optional) To verify the connectivity to the BRM schema or to the BI Publisher Scheduler database, click **Test Connection**.
   e. Click **Finish**.
      In the **JDBC Resources** page, the **Connection Pools** section displays the newly added connection pool.
7. (Optional for BRM schemas) In a multischema BRM system, to create connection pools for different BRM schemas, such as 0.0.0.2, 0.0.0.3, repeat step 4 through step 6.
8. In the **Data Sources** section, click the **Create** button.
   The **Create Data Source - Application & Type** page appears.
9. Do the following:
   a. From the **Application** list, select **default**.
   b. From the **Data Source Type** section, select the **Managed Data Source** option.
   c. Click **Continue**.
      The Create Data Source - Managed Data Source page opens.
10. Do the following:
a. In the Name field, enter the name of the BRM schema or the BI Publisher Scheduler database for which you configured the connection pool.
   For example:
   - For a BRM schema, enter 0.0.0.1.
   - For a BI Publisher Scheduler database, enter scheduler.

b. In the JNDI Location field, enter one of the following:
   - For a BRM schema, enter jdbc/OracleBRM/BRMSchema, where
     BRMSchema is the BRM schema for which the data source is being created.
     For example, if the BRM schema is 0.0.0.1, enter jdbc/OracleBRM/0.0.0.1.
   - For the BI Publisher Scheduler database, enter jdbc/OracleBIP/scheduler.

c. Click Finish.
   In the JDBC Resources page, the Data Sources section displays the newly added data source.

11. (Optional for BRM schemas) In a multischema BRM system, to create data sources for connection pools belonging to different BRM schemas, such as 0.0.0.2, 0.0.0.3, repeat step 8 through step 10.

Adding JNDI Entries in the Infranet.properties File of pin_inv_doc_gen

To add JNDI entries for fetching data from the configured data sources:

1. Open the BIP_Home/oc4j_bi/j2ee/home/applications/pin_inv_doc_gen/Infranet.properties file in a text editor.
2. Add the entries shown in Table 3–5 to the end of the file:

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>infranet.PROVIDER_URL</td>
<td>The environment property that specifies the BI Publisher provider URL in the</td>
</tr>
<tr>
<td></td>
<td>following format: ormi://localhost:port/xmlpserver</td>
</tr>
<tr>
<td></td>
<td>where localhost is the host on which the BI Publisher server is installed</td>
</tr>
<tr>
<td></td>
<td>and port is the Oracle Remote Method Invocation port. The default port is</td>
</tr>
<tr>
<td></td>
<td>25791.</td>
</tr>
<tr>
<td>infranet.SECURITY_</td>
<td>The environment property that specifies the identity of the principal for</td>
</tr>
<tr>
<td>PRINCIPAL</td>
<td>authenticating the caller to the service.</td>
</tr>
<tr>
<td>infranet.SECURITY_</td>
<td>The environment property that specifies the credentials of the principal for</td>
</tr>
<tr>
<td>CREDENTIALS</td>
<td>authenticating the caller to the service.</td>
</tr>
<tr>
<td>infranet.INITIAL_</td>
<td>The environment property that specifies the initial context factory to use.</td>
</tr>
<tr>
<td>CONTEXT_FACTORY</td>
<td>The default is com.evermind.server.ApplicationClientInitialContextFactory.</td>
</tr>
<tr>
<td></td>
<td>Do not modify this value.</td>
</tr>
</tbody>
</table>

3. Save and close the file.

Generating BI Publisher Invoice Documents

You use the pin_inv_doc_gen 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 `pin_inv_doc_gen`, the utility fetches the corresponding invoice POID, BI
Publisher report name, and layout template name of an `/invoice` object.

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

To run the `pin_inv_doc_gen` utility:

1. Configure the `BipEnv.bat` file to execute the appropriate `pin_inv_doc_gen`
   command. To view the utility’s syntax and parameters, see "`pin_inv_doc_gen`".
2. In the command prompt, navigate to the `BIP_Home\oc4j\bi\j2ee\home\applications\pin_inv_doc_gen` directory.
3. Enter the following command which runs the `BipEnv.bat` file:
   ```bash
   BipEnv.bat
   ```

**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 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 subordinate accounts, the `/invoice` objects of the
subordinate account are not selected for generating the invoice document.

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

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

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

   **Note:** You can use the `BIP_Home\oc4j\bi\j2ee\home\applications\pin_inv_doc_gen\InvoiceList.xml` file
   as an example.
2. Go to the `BIP_Home\oc4j_bi\j2ee\home\applications\pin_inv_doc_gen` directory.

3. Open the `BipEnv.bat` file in a text editor.

4. Search for the following line:
   
   ```
   invoicedocgen.pin_inv_doc_gen -status pending
   ```

5. Replace with:
   
   ```
   invoicedocgen.pin_inv_doc_gen -accts_list InvoiceList
   ```
   
   where `InvoiceList` specifies the name and location of the file that includes a list of accounts and bill units for which to create invoice documents. This is the file you created in step 1.

6. Save and close the file.

7. In the command prompt, navigate to the `BIP_Home\oc4j_bi\j2ee\home\applications\pin_inv_doc_gen` directory.

8. Enter the following command to execute the `BipEnv.bat` file:
   
   ```
   BipEnv.bat
   ```

---

**Generating BI Publisher Invoice Documents Based on the Account Type**

To generate invoices based on the account type:

1. Go to the `BIP_Home\oc4j_bi\j2ee\home\applications\pin_inv_doc_gen` directory.

2. Open the `BipEnv.bat` file in a text editor.

3. Search for the following line:
   
   ```
   invoicedocgen.pin_inv_doc_gen -status pending
   ```

4. Do one of the following:

   - To generate invoice document for consumer account types, replace with:
     
     ```
     invoicedocgen.pin_inv_doc_gen -type consumer
     ```

   - To generate invoice document for corporate account types, replace with:
     
     ```
     invoicedocgen.pin_inv_doc_gen -type corporate
     ```

5. Save and close the file.

6. In the command prompt, navigate to the `BIP_Home\oc4j_bi\j2ee\home\applications\pin_inv_doc_gen` directory.

7. Enter the following command to execute the `BipEnv.bat` file:
   
   ```
   BipEnv.bat
   ```

---

**Generating Duplicate BI Publisher Invoice Documents**

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 $BIP_HOME/oc4j_bi/j2ee/home/applications/pin_inv_doc_gen$ directory.

2. Make sure the $infranet.dupinvdir.path$ entry in the utility’s $Infranet.properties$ file specifies the directory in which to write the file. See "Configuring the $Infranet.properties$ File for $pin_inv_doc_gen$".

3. Open the $BipEnv.bat$ file in a text editor.

4. Search for the following line:
   
   invoicedocgen.pin_inv_doc_gen -status pending
   
5. Replace with:
   
   invoicedocgen.pin_inv_doc_gen -status generated
   
6. Save and close the file.

7. In the command prompt, navigate to the $BIP_HOME/oc4j_bi/j2ee/home/applications/pin_inv_doc_gen$ directory.

8. Enter the following command to execute the $BipEnv.bat$ file:

   BipEnv.bat

---

**Important:** 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.

---

### Regenerating BI Publisher Invoice Documents

If an /invoice object changes after you created a BI Publisher invoice document, you can generate a new BI Publisher invoice document by using the $pin_inv_doc_gen$ utility $-status regen$ parameter.

To regenerate BI Publisher invoice documents:

1. Go to the $BIP_HOME/oc4j_bi/j2ee/home/applications/pin_inv_doc_gen$ directory.

2. Open the $BipEnv.bat$ file in a text editor.

3. Search for the following line:
   
   invoicedocgen.pin_inv_doc_gen -status pending
   
4. Replace with:
   
   invoicedocgen.pin_inv_doc_gen -status regen
   
5. Save and close the file.

6. In the command prompt, navigate to the $BIP_HOME/oc4j_bi/j2ee/home/applications/pin_inv_doc_gen$ directory.

7. Enter the following command to execute the $BipEnv.bat$ file:

   BipEnv.bat

The utility generates a BI Publisher invoice document for all /invoice objects in the BRM database that have a status of PIN_INV_STATUS_GENERATED.
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 in BI Publisher. See "Configuring the Email Server".
- Configure the invoice document delivery option in the Infranet.properties file of pin_inv_doc_gen utility. Email is configured as the default delivery option. See "Configuring the Infranet.properties File for pin_inv_doc_gen".

Customizing Invoices

---

Important: 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 customize the invoice documents by:

- Customizing Invoice Layout Templates
- Enriching Invoice Data

Customizing Invoice Layout Templates

You can customize the invoice layout template by:

- Updating the existing layout template file
- Adding a new layout template file

Updating the Invoice Layout Template File

You can update 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 of creating an RTF template in the BI Publisher documentation.

Adding a New Invoice Layout Template

To add a new layout template:

1. Create a new layout template in BI Publisher. For information on how to create a layout template, see the discussion of creating an RTF template in the BI Publisher documentation.

2. Create a new invoicing /config/business_profile object in BRM to reflect the addition of the new layout template. See "Specifying BI Publisher Invoice Report and Template Names in BRM" and "About Associating Bill Units with a BI
Publisher Invoice and Report”.

**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.

---

**Important:** The sample layout templates that are available with BRM-BI Publisher invoicing integration package display data fetched from only BRM database. If you want to enrich your invoice document with additional data sources, you need to customize the sample layout templates. See "Customizing Invoice Layout Templates".

---

To enrich BRM invoice data:

- The `/invoice` objects containing data in XML format should be present in the BRM database.
- The additional data source must be configured in BI Publisher to enrich the invoice data. See the discussion of setting up data sources in the BI Publisher documentation.
- BI Publisher reports should be configured with at least two data sources: Custom BRM Web service, PublicBRMfetchInvoiceXMLService, for retrieving invoice XML data from the BRM database, and the enriching data source. For information on data sources that BI Publisher supports, see the BI Publisher documentation.

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.

**Example: Enriching Invoice Data**

You can enrich the BRM invoice data with additional data from various data sources. In this example, you configure the consumer invoice report to display data from a file data source in addition to the data from BRM Web service. You use the XML demo file provided along with BI Publisher (`BIP_Home/xmlp/XMLP/DemoFiles` directory).

You must customize the `BRM_Consumer_Invoice.rtf` file to display data from additional source. See "Customizing Invoice Layout Templates".

To configure file data source in `BRM_Consumer_Invoice_Report.xdo` report:

1. In a Web browser, enter the following URL:
   
   `http://localhost:port/xmlpserver/`

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

2. Enter your credentials to log in to BI Publisher.

3. In **Shared Folders**, click **BRM_Invoices**.

4. For `BRM_Consumer_Invoice_Report`, click the **Edit** link.
5. In the Report pane, in Data Model, click BRM_Consumer_Invoice_Report data set. The Data Set pane displays data set name as BRM_Consumer_Invoice_Report and type as Web Service.

6. In Layouts, click BRM_Consumer_Invoice_Report layout. The Layout pane displays template as BRM_Consumer_Invoice.rtf and template type as RTF Template.

7. Click Data Model.

8. Click New. The Data Model displays the newly added data set, New DataSet2.

9. In the Data Set pane, perform the following:
   a. In the Name field, enter a name for the new data set.
   b. From the Type list, select File option.
   c. From the Data Source list, select demo files.
   d. In the File Name field, enter the file name that you want to use to enrich the invoice data.

10. Click Layouts to associate the updated layout template file with the report.

11. Click Browse to browse the customized BRM_Consumer_Invoice.rtf file.

12. Click Upload.

13. Click Save.

14. Click View to open the report in View mode.

15. Select the BRM_Consumer_Invoice template from the Template list and select the output format as pdf.

16. Click View to view the report.

---

**Specifying the Default Format in Which to Store Invoices in BRM**

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

You modify the /config/business_params object by using the pin_bus_params utility. For information on this utility, see "pin_bus_params" in BRM Developer’s Guide.

To specify the default format in which to store invoices:

1. Go to the BRM_Home/sys/data/config directory.

2. Run the following command, which creates an editable XML file for the invoicing parameter class:
   ```
   pin_bus_params -r BusParamsInvoicing bus_params_Invoicing.xml
   ```
This command creates the XML file named `bus_params_Invoicing.xml.out` in your working directory. To place this file in a different directory, specify the path as part of the file name.

3. Open the `bus_params_Invoicing.xml.out` file.

4. Search for the following line:
   ```xml
   <InvoiceStorageType>1</InvoiceStorageType>
   ```

5. Do one of the following:
   - To specify flist as the output format, set `InvoiceStorageType` to 0.
   - To specify XML as the output format, set `InvoiceStorageType` to 1.

6. Save this file as `bus_params_Invoicing.xml`.

7. Go to the `BRM_Home/sys/data/config` directory, which includes support files used by the `pin_bus_params` utility.

8. Run the following command, which loads this change into the appropriate `/config/business_params` object:
   ```bash
   pin_bus_params PathToWorkingDirectory/bus_params_Invoicing.xml
   ```
   where `PathToWorkingDirectory` is the directory in which `bus_params_Invoicing.xml` resides.

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

   **Note:** To run this command from a different directory, see “pin_bus_params” in BRM Developer’s Guide.

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


10. Stop and restart the CM. For more information, see "Starting and Stopping the BRM System" in BRM System Administrator’s Guide.

11. (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.
This chapter provides reference information for Oracle Communications Billing and Revenue Management (BRM) invoice utilities.
load_pin_invoice_data_map

Use this utility to load invoice data templates into the /config/invoice_data_map object in the BRM database. You define the invoice data template in the BRM_Home/data/config/pin_invoice_data_map file.

For more information, see "Specifying Invoice Data from Pipeline Manager and Custom Applications".

**Note:** You cannot load separate /config/invoice_data_map objects for each brand. All brands use the same object.

**Caution:** When you run the load_pin_invoice_data_map utility, 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 load_pin_invoice_data_map utility.

**Important:** To connect to the BRM database, the load_pin_invoice_data_map utility needs a configuration file in the directory from which you run the utility. See "Creating Configuration Files for BRM Utilities" in BRM System Administrator’s Guide.

**Location**

BRM_Home/bin

**Syntax**

load_pin_invoice_data_map [-d] [-v] [-h] invoice_data_map_file

**Parameters**

- **-d**
  Creates a log file for debugging purposes. Use this parameter when the utility appears to have run with no errors, but the invoice data templates have not been loaded into the database.

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

To redirect the output to a log file, use the following command. Replace filename.log with the name of the log file:

load_pin_invoice_data_map -v > filename.log

**Note:** If a file with the same name exists, it is overwritten.

- **-h**
  Displays the syntax and parameters for this utility.
invoice_data.map_file

The name and location of the file that defines invoice data templates. The default `pin_invoice_data_map` file is in the `BRM_Home/sys/data/config` directory.

If you copy the `pin_invoice_data_map` file to the same directory from which you run the `load_pin_invoice_data_map` utility, you do not need to specify the path or the file name.

If you use the default `pin_invoice_data_map` file name, you do not need to specify it on the command line when it is located in the same directory as this utility.

Results

The `load_pin_invoice_data_map` utility notifies you only if it encounters errors.

---

**Important:** 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 invoices. You run this utility as part of your daily billing to create an invoice for each account that is billed on that day.

---

**Note:** 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 invoices, see "Designing and Generating Invoices".

For information on `pin_bill_day`, see "About Running the Billing Scripts" in *BRM Configuring and Running Billing*.

---

**Important:** To connect to the Oracle Communications Billing and Revenue Management (BRM) database, the `pin_inv_accts` utility needs a configuration file in the directory from which you run the utility. See "Creating Configuration Files for BRM Utilities" in *BRM System Administrator’s Guide*.

---

**Location**

`BRM_Home/bin`

**Syntax**

```
pin_inv_accts  [-test] [-start date] [-end date] [-regen]
```

**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 date**
- **-end date**
  Generates invoices for accounts whose billing cycle ended during the time period defined by the start and end date, including the end date. If you do not specify an end date, `pin_inv_accts` uses the current date. If you do not specify a start date, `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. A programmer specifies the format in the `PCM_OP_INV_POL_FORMAT_INVOICE` policy opcode. If the
opcode specifies a format that the invoices are already stored in, pin_inv_accts regenerates them in that format, overwriting the original invoices and updating the PIN_FLD_INVOICE_OBJ field in the /bill object.

To regenerate invoices within a certain 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 subordinate (nonpaying) bills in a billing hierarchy, run the pin_inv_accts utility with the -regen and -hierarchy parameters. By default, invoices are regenerated only for parent bills in a billing hierarchy and for paying non-hierarchical bills.

**Note:** You can regenerate invoices only if you still have all the original /event and /bill objects for the months you want to regenerate. If the balances are maintained correctly in the /bill objects and are not updated, you can regenerate invoices.

**-summary | -detail**
Generates summary invoices if the -summary parameter is set; generates detailed invoices if the -detail parameter is set. If you do not specify either parameter, pin_inv_accts uses -detail parameter.

**-pay_type payment_method**
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 the list in Table 4–1:

<table>
<thead>
<tr>
<th>Payment Method</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>undefined</td>
<td>0</td>
</tr>
<tr>
<td>prepaid</td>
<td>10000</td>
</tr>
<tr>
<td>invoice</td>
<td>10001</td>
</tr>
<tr>
<td>debit</td>
<td>10002</td>
</tr>
<tr>
<td>credit card</td>
<td>10003</td>
</tr>
<tr>
<td>direct debit</td>
<td>10005</td>
</tr>
<tr>
<td>smart card</td>
<td>10006</td>
</tr>
<tr>
<td>subordinate</td>
<td>10007</td>
</tr>
<tr>
<td>In Customer Center, this is called nonpaying.</td>
<td>10007</td>
</tr>
<tr>
<td>beta</td>
<td>10008</td>
</tr>
<tr>
<td>internal</td>
<td>10009</td>
</tr>
<tr>
<td>guest</td>
<td>10010</td>
</tr>
<tr>
<td>cash</td>
<td>10011</td>
</tr>
<tr>
<td>check</td>
<td>10012</td>
</tr>
<tr>
<td>wire transfer</td>
<td>10013</td>
</tr>
<tr>
<td>payorder</td>
<td>10014</td>
</tr>
</tbody>
</table>
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 A/R accounts in which the bill’s PIN_FLD_AR_HIERARCHY_SIZE value exceeds the invoicing threshold and generates a separate invoice for each subordinate bill unit.

For more information, see "About Invoicing for Hierarchical Account Groups".

**-file filename**

Generates an invoice for each bill in the specified file. For example:

```
pin_inv_accts -file bills_020207
```

<table>
<thead>
<tr>
<th>Payment Method</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>postal order</td>
<td>10015</td>
</tr>
<tr>
<td>voucher</td>
<td>10016</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0</th>
<th>PIN_FLD_RESULTS</th>
<th>ARRAY [0]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PIN_FLD_POID</td>
<td>POID [0] 0.0.0.1 /bill 26011 0</td>
</tr>
<tr>
<td>0</td>
<td>PIN_FLD_RESULTS</td>
<td>ARRAY [1]</td>
</tr>
<tr>
<td>1</td>
<td>PIN_FLD_POID</td>
<td>POID [0] 0.0.0.1 /bill 26091 0</td>
</tr>
</tbody>
</table>

For more information on the **pin_flist** format, see "About Formats for Storing Invoices".

**-skip_blank_billnos**

Excludes invoicing for subordinate bill units that do not contain bill numbers.

Subordinate bill units, which use the bill number of their parent account, are generated even when billing for the parent account is suppressed. In such cases, they will not contain bill numbers, which invoices require. Therefore, when bill suppression is enabled on a parent (A/R) account, and the subordinate account threshold is exceeded (multiple threads are used to process the invoices), invoicing fails because the parent bill number is not available. This parameter enables **pin_inv_accts** to proceed without processing these bill units.

(Subordinate bill units only contain bill numbers when their parent accounts are billed.)

**-verbose**

Displays information about successful or failed processing as the utility runs.
Note: This parameter is always used in conjunction with other parameters and commands. It is not position dependent. For example, you can enter `-verbose` at the beginning or end of a command to initiate the verbose parameter. To redirect the output to a log file, use the following syntax with the verbose parameter. Replace `filename.log` with the name of the log file:

```
pin_inv_accts any_other_parameter -verbose > filename.log
```

-help
Displays the syntax and parameters for this utility.
Use this utility to create, deliver, and store invoice documents in the Oracle Business Intelligence Publisher (BI Publisher) Scheduler database.

**Important:** The **pin_inv_doc_gen** utility requires Java Development Kit (JDK) 1.6.0_37 or above and that the JAVA_HOME environment variable be set to the absolute path of the JDK 1.6.0_37 software installation. For more information, see "Setting the JAVA_HOME Environment Variable".

After `/invoice` 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.

**For BI Publisher 11g:**
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 11g".

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 11g, see "Designing and Generating Invoices in Oracle Business Intelligence Publisher 11g".

**Important:**
- Before running the **pin_inv_doc_gen** utility, save the `BRM_Home/apps/pin_inv_doc_gen/Infranet.properties.sample` configuration file as `Infranet.properties`.
- To connect to the BRM database, run **pin_inv_doc_gen** from the directory that contains the `Infranet.properties` configuration file.
- BI Publisher 11g supports multischema systems.

For more information, see "Configuring the Infranet.properties File for pin_inv_doc_gen with BI Publisher 11g".

**For BI Publisher 10g:**
Start the **pin_inv_doc_gen** utility by running the `BipEnv.bat` file on the server running OC4J. For information on the `BipEnv.bat` file, see "Generating BI Publisher Invoice Documents".

Set the `infranet.dmsearchargs.size` parameter in the `Infranet.properties` configuration file to the number of account and bill unit pairs the utility processes at one time. The maximum value for `infranet.dmsearchargs.size` is 14.
For example, if the `InvoiceList.xml` file lists 20 account and bill unit pairs, `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 10g, see "Designing and Generating Invoices in Oracle Business Intelligence Publisher 10g".

---

Important:
- To connect to the BRM database, run `pin_inv_doc_gen` from the directory that contains the `Infranet.properties` configuration file. The `Infranet.properties` configuration file is located in `BIP_Home/oc4j_bi/j2ee/home/applications/pin_inv_doc_gen` directory.
- BI Publisher 10g supports multischema systems.

For more information, see "Configuring the Infranet.properties File for `pin_inv_doc_gen`".

---

Location

For BI Publisher 11g: `BRM_Home/apps/pin_inv_doc_gen`

For BI Publisher 10g: `BIP_Home/oc4j_bi/j2ee/home/applications/pin_inv_doc_gen`

Syntax

```
pin_inv_doc_gen [-accts_list InvoiceList.xml] [-type [-consumer | -corporate]] [-schema SchemaNumber] [-status pending | generated | regen] [-start startDate] [-end endDate] [-help]
```

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 11g.

**-accts_list InvoiceList.xml**

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

**-type**

Specifies the type of account for which the invoice document is to be generated:
- `-type consumer` generates a consumer account.
- `-type corporate` generates a corporate account. For a corporate account, all the `/invoice` objects of its subordinate accounts (if they have a separate `/invoice` POID) are selected for generating the invoice document. See "About Invoicing for Hierarchical Account Groups". However, the `/invoice` object of the paying child account 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 are the paying accounts in the account hierarchy, the PIN_FLD_PARENT_FLAGS field value is greater than zero.
-schema SchemaNumber
(Valid only for BI Publisher 11g) 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 /invoice object for which the invoice is to be generated:

- **-status pending** selects the /invoice objects for a document that has not been generated. It selects /invoice objects with a status of 0 and generates documents for them.

- **-status generated** generates duplicate documents for invoices for a document that has been generated. The duplicate document displays a Duplicate watermark on the document. For BI Publisher 11g, see "Generating Duplicate Invoice Documents with BI Publisher 11g". For BI Publisher 10g, see "Generating Duplicate BI Publisher Invoice Documents".

- **-status regen** (for BI Publisher 10g) generates the invoice document with a new /invoice object. See "Regenerating BI Publisher Invoice Documents" for more information.

-start startDate
-end endDate
Generates invoice documents for accounts or bill units whose /invoice objects were created during the time period defined by startDate and endDate.

If you do not specify a start date, **pin_inv_doc_gen** fetches all /invoice objects with a status of PIN_INV_STATUS_PENDING.

If you do not specify an end date, **pin_inv_doc_gen** uses the current date. The end date cannot be greater than the current date. The end date specified is inclusive, which means all /invoice objects created until 23:59:59 of the end date are selected by the search query.

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

-help
Displays the syntax and parameters for this utility.
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 subordinate bill invoices that are not consolidated into the parent invoice, pin_inv_export creates a separate invoice file for each subordinate bill. The file naming convention is:

```
inv_BillPOID_BillDate_SubordinateAccountPOID_ARaccountPOID.ext
```

For subordinate 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_SubordinateAccountPOID.ext
```

---

**Important:** To connect to the BRM database, the pin_inv_export utility needs a configuration file in the directory from which you run the utility. See "Creating Configuration Files for BRM Utilities" in BRM System Administrator’s Guide.

---

**Location**

`BRM_Home/bin`

**Syntax**

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

**Parameters**

- **-start date**
- **-end date**

Retrieves invoices for accounts whose billing cycle ended during the time period defined by the start and end date, including the end date.

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).

---

**Note:** If a custom /invoice subclass exists, then 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:

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

For more information on generating a pin_flist file of bills, see "pin_inv_accts".

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

---

**Note:** This parameter is always used in conjunction with other parameters and commands. It is not position dependent. For example, you can enter -verbose at the beginning or end of a command to initiate the verbose parameter. To redirect the output to a log file, use the following syntax with the verbose parameter.

```
pin_inv_export any_other_parameter -verbose > filename.log
```

where `filename` is the name of the log file.

---

-help
Displays the syntax and parameters for this utility.
**pin_inv_send**

Use this utility to email or print invoices, depending on the delivery method specified in Customer Center:

- If the delivery method is set to **Email**, `pin_inv_send` emails the invoice.
- If the delivery method is set to **Postal**, `pin_inv_send` prints the invoice.

This setting is in Payment Details panel of Customer Center.

You can configure the following behavior by editing the `pin_inv_send` utility’s configuration file (`BRM_Home/apps/pin_inv/pin.conf`).

- To set the maximum size in kilobytes of invoices that can be sent by email, set the `inv_send_size` entry. This entry is used by PCM_OP_INV_VIEW_INVOICE to restrict sending large invoices to the Email Data Manager (`dm_email`).
- To send individual subordinate 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".

---

**Important:** To connect to the BRM database, the `pin_inv_send` utility needs a configuration file in the directory from which you run the utility. See "Creating Configuration Files for BRM Utilities" in *BRM System Administrator’s Guide*.

---

**Location**

`BRM_Home/bin`

**Syntax**

```
pin_inv_send [-start date] [-end date] [-pay_type id] [-format_id id] [-v] [-h]
```

**Parameters**

- **-start date**
- **-end date**

Emails or prints invoices for accounts with invoices whose billing cycle ended during the time period defined by the start and end date, including the end date. 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 4–2**:

**Table 4–2  Payment Method IDs**

<table>
<thead>
<tr>
<th>Payment Method</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Undefined</td>
<td>0</td>
</tr>
<tr>
<td>Prepaid</td>
<td>10000</td>
</tr>
<tr>
<td>Invoice</td>
<td>10001</td>
</tr>
<tr>
<td>Debit</td>
<td>10002</td>
</tr>
<tr>
<td>Credit card</td>
<td>10003</td>
</tr>
<tr>
<td>Direct debit</td>
<td>10005</td>
</tr>
<tr>
<td>Smart card</td>
<td>10006</td>
</tr>
<tr>
<td>Subordinate</td>
<td>10007</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>In Customer Center, this is called <strong>nonpaying</strong>.</td>
<td>10007</td>
</tr>
<tr>
<td>Beta</td>
<td>10008</td>
</tr>
<tr>
<td>Internal</td>
<td>10009</td>
</tr>
<tr>
<td>Guest</td>
<td>10010</td>
</tr>
<tr>
<td>Cash</td>
<td>10011</td>
</tr>
<tr>
<td>Check</td>
<td>10012</td>
</tr>
<tr>
<td>Wire transfer</td>
<td>10013</td>
</tr>
<tr>
<td>Payorder</td>
<td>10014</td>
</tr>
<tr>
<td>Postal order</td>
<td>10015</td>
</tr>
<tr>
<td>Voucher</td>
<td>10016</td>
</tr>
</tbody>
</table>

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 4–3**:

**Table 4–3  Format ID Values**

<table>
<thead>
<tr>
<th>Format</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTML</td>
<td>1</td>
</tr>
<tr>
<td>DOC1</td>
<td>3</td>
</tr>
</tbody>
</table>
-v
Displays information about successful or failed processing as the utility runs.

---

**Note:** This parameter is always used in conjunction with other parameters and commands. It is not position dependent. For example, you can enter -v at the beginning or end of a command to initiate the verbose parameter. To redirect the output to a log file, use the following syntax with the verbose parameter. Replace *filename.log* with the name of the log file:

```
pin_inv_send any_other_parameter -v > filename.log
```

---

-h
Displays the syntax and parameters for this utility.
Use this utility to load brand-specific /config/invoice_events objects into the BRM database. This utility extends branding to invoices by associating specified event objects with a specific brand. When invoices are generated, BRM displays the brand’s events in the invoice. If you do not load /config/invoice_events objects, BRM uses the /config/invoice_events object that belongs to the root account.

For information about invoices for brands, see "About Brand-Specific Invoices".

**Note:** You cannot load separate /config/invoice_events objects for each brand. All brands use the same object.

**Important:** To connect to the BRM database, the pin_load_invoice_events utility needs a configuration file in the directory from which you run the utility. See "Creating Configuration Files for BRM Utilities" in BRM System Administrator’s Guide.

**Location**

BRM_Home/bin

**Syntax**

```bash
pin_load_invoice_events -brand "brand_poid" -eventfile file
[-logfile file] [-reload] [-debug] [-help]
```

**Parameters**

- **-brand "brand_poid"**
  Specifies the brand POID, for example, "0.0.0.1 /account 104 0".

- **-eventfile file**
  Specifies the pathname 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 file**
  Specifies the full pathname of the log file.
-reload
Reloads an existing /config object for a brand.

-debug
Logs the flist and detailed messages in the log file.

-help
Displays the syntax and parameters for this utility.

Results
This 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 brand specific objects into the BRM database. These objects contain the style sheet templates that you apply to invoices. The templates let you specify the look of invoices that belong to specific brands.

There are two usages for this utility:

- You can load /config/invoice_templates brand specific objects into the BRM database. If you do not load /config/invoice_templates objects, BRM uses the /config/invoice_templates object that belongs to the root account.
- You can 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 Invoice Templates".

---

**Note:** You cannot load separate /config/invoice_templates objects for each brand. All brands use the same object.

---

**Important:** To connect to the BRM database, the pin_load_invoice_template utility needs a configuration file in the directory from which you run the utility. See "Creating Configuration Files for BRM Utilities" in BRM System Administrator’s Guide.

---

**Location**

BRM_Home/bin

**Syntax**

pin_load_invoice_template -brand "brand_poid" -type template_type
-locale locale_string
-template template_file [-usexsl]
[-debug] [-logfile logfile] [-help]

**Parameters**

- **-brand "brand_poid"**
  Specifies the brand account POID for which you are loading the template. For example, "0.0.0.1 /account 104 0"
  
  If your system is not brand enabled, you must use the POID of the root account: "0.0.0.1 /account 1".

  **Important:** 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".
-locale locale_string
Specifies the locale. For example, en_US.

-template template_file
Specifies the full pathname of the template file.

-usexsl
Uses the XSL invoice formatting opcode. This option sets the PIN_FLD_FLAG field in the /config/invoice_templates object to XSL style sheets.

-debug
Logs the flist and detailed messages in the log file.

-logfile logfile
Specifies the full pathname of the log file.

-help
Displays the syntax and parameters for this utility.

Changing the Template Flag

To change the flag that specifies whether to use an HTML template or an XSL style sheet, set the PIN_FLD_FLAGS field in the /config/invoice_templates object by using this syntax:

- For an HTML template:
  pin_load_invoice_template -brand brand_poid

- For an XSL style sheet:
  pin_load_invoice_template -brand brand_poid [-usexsl]

  When the -usexsl option is not specified, an HTML template will be used.

Results

This utility should notify you that it was successful. If errors are returned, look in the utility log file (default.pinlog) for detailed error messages. The log file is located either in the directory from which the utility was started, or in a directory specified in the configuration file.
pin_upd_assoc_bus_profile

Use this utility to associate bill units with an invoicing business profile. It searches the BRM database for /billinfo objects with an empty PIN_FLD_ASSOC_BUS_PROFILE_OBJ_LIST field and populates the field with an /associated_bus_profile object. See "Associating Pre-Existing Bill Units with Business Profiles".

Before running pin_upd_assoc_bus_profile, you must:

- Load invoicing business profiles into the database. See "Loading Invoicing Business Profiles into the Database".
- Enable the BRM-BI Publisher integration framework. See "Enabling the BRM-BI Publisher Integration".

**Note:** This utility fails if you attempt to run it when the BRM-BI Publisher integration framework is disabled.

This utility is supported in branding environments. See "About Branding" in BRM Managing Customers.

**Important:** The pin_upd_assoc_bus_profile utility needs the configuration file in BRM_Home/apps/pin_billd. This configuration file must be present in the directory from which you run the utility.

Because pin_upd_assoc_bus_profile is based on the BRM multithreaded application (MTA) framework, the configuration file also requires some performance-related configuration entries. For more information, see "Configuring Your Multithreaded Application" in BRM Developer’s Guide.

**Location**

BRM_Home/bin

**Syntax**

pin_upd_assoc_bus_profile [-file filename] [-verbose] [-help]

**Parameters**

- **-file filename**
  Specifies the name and location of the input file that lists the /billinfo objects to update. This is an optional parameter.

  The input file should be in the following format:

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