Preface

Audience .......................................................................................................................................................... v
Downloading Oracle Communications Documentation .............................................................................. v
Documentation Accessibility ................................................................................................................... v
Document Revision History .................................................................................................................... v

1 Connecting J2EE-Compliant Applications to BRM

About Connecting J2EE-Compliant Applications to the BRM System ........................................... 1-1
Overview of the BRM JCA Resource Adapter Architecture .......................................................... 1-1
About the BRM JCA Resource Adapter ............................................................................................ 1-2
  About Sending Data to the BRM JCA Resource Adapter ............................................................... 1-3
  About BRM Web Services and WSDL Files ............................................................................... 1-4
  About Validating Input and Output XML Data............................................................................. 1-6
  About Validating the XML Schema for BRM Opcodes .............................................................. 1-9
  About Validating the XML Schema for Storable Classes and Subclasses ................................ 1-9
  About Converting XML Data and Calling the BRM API .......................................................... 1-9
  About Converting Output flist Data into XML Format ............................................................... 1-10
  About Invoicing Output XML Data .............................................................................................. 1-10
About BRM JCA Resource Adapter Connection Management .................................................. 1-10
About BRM JCA Resource Adapter Transaction Management ............................................. 1-10
  About Creating Multiple Accounts in a Single Transaction ....................................................... 1-11
About BRM JCA Resource Adapter Security ................................................................................. 1-11
About BRM JCA Resource Adapter Logging .................................................................................. 1-11
About Deploying and Connecting the BRM JCA Resource Adapter ........................................ 1-12
About the BRMAAdapterServletClient Application .................................................................. 1-12

2 Installing the BRM JCA Resource Adapter

System Requirements .............................................................................................................................. 2-1
  Software Requirements .................................................................................................................. 2-1
  Information Requirements .............................................................................................................. 2-1
Installing the BRM JCA Resource Adapter .................................................................................. 2-1
What’s Next? ........................................................................................................................................ 2-3

3 Deploying and Configuring the BRM JCA Resource Adapter

Overview of the BRM JCA Resource Adapter Configuration Procedure ........................................ 3-1
Creating and Packaging Custom Classes and Fields .......................................................... 3-2
Generating the Schema Files for Your System ................................................................. 3-2
  Generating the Schema for Your Opcodes ........................................................................ 3-3
  Creating Custom Opcode Specification XML Files .......................................................... 3-4
  Specifying the XSL Rules to Create the Opcode Schema ................................................. 3-4
  Converting flist Specs into XSD Schema Files with pin_opspec_to_schema .................. 3-4
  Generating the Schema for Your Storable Classes and Subclasses ............................... 3-4
  Determining the Storable Classes to Convert Into Schema .............................................. 3-5
  Generating Storable Class Schema Files with pin_dd_to_schema ................................... 3-5
  Specifying the Location of the Storable Class Schema Files in the Opcode Schema Files .... 3-6
Specifying the XML Tags for Extended Fields ................................................................. 3-7
Generating the WSDL Files for Your System ..................................................................... 3-8
  Defining a Web Service ...................................................................................................... 3-9
  Defining Default Values for a Web Service ...................................................................... 3-9
  Defining Settings for Individual Opcodes ......................................................................... 3-10
  Sample Web Services Group Configuration File ............................................................ 3-11
  Generating WSDL Files for Web Services ...................................................................... 3-12
Configuring How to Represent Infinite Date Values ......................................................... 3-12
Deploying and Connecting the BRM JCA Adapter on Oracle Application Server ............ 3-13
  Deploying the BRM JCA Resource Adapter on Oracle Application Server .................... 3-13
  Connecting the Adapter to BRM in Oracle Application Server ...................................... 3-14
  Configuring the Transaction Mode in Oracle Application Server .................................... 3-17
  Configuring Java Logging on Oracle Application Server ................................................. 3-18
  Changing the Java Logging Level .................................................................................... 3-19
Deploying and Connecting the BRM JCA Adapter on Oracle WebLogic Server ............ 3-19
  Deploying the BRM JCA Resource Adapter on Oracle WebLogic Server ....................... 3-19
  Connecting the Adapter to BRM in Oracle WebLogic Server ........................................ 3-20
  Configuring Java Logging in Oracle WebLogic Server .................................................. 3-22
  Creating a Startup Class .................................................................................................. 3-22
  Changing the Java Logging Level in Oracle WebLogic Server ...................................... 3-23
Deploying and Connecting the BRM JCA Adapter on Oracle GlassFish Server .............. 3-25
  Configuring the GlassFish Server .................................................................................. 3-25
  Deploying and Configuring JCA Resource Adapter on the GlassFish Server .................. 3-25
Setting Up JCA Resource Adapter to Support Custom Opcodes ...................................... 3-26
Using the BRMAdapterServletClient Application ......................................................... 3-27

4 JCA Resource Adapter Utilities
  pin_dd_to_schema ............................................................................................................. 4-2
  pin_opspec_to_schema .................................................................................................. 4-4
  pin_wsdll_generator ...................................................................................................... 4-5
Preface

This book describes how to use the BRM JCA Resource Adapter with Oracle Communications Billing and Revenue Management (BRM).

Audience

This document is intended for developers and system administrators.

Downloading Oracle Communications Documentation

Product documentation is located on Oracle Technology Network:

http://docs.oracle.com

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

http://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>E25091-01</td>
<td>March 2013</td>
<td>Initial release.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Removed incorrect information about XA transaction support.</td>
</tr>
<tr>
<td>Version</td>
<td>Date</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Updated the &quot;About Sending Data to the BRM JCA Resource Adapter&quot; section.</td>
</tr>
</tbody>
</table>
Connecting J2EE-Compliant Applications to BRM

This chapter describes how a J2EE-compliant application can connect to the Oracle Communications Billing and Revenue Management (BRM) system using the BRM JCA Resource Adapter and how BRM operations are exposed through the adapter.

Before using the adapter, you should be familiar with the following:

- BRM concepts and architecture. See "Introducing BRM" in BRM Concepts.
- J2EE and J2EE Connector Architecture (JCA) 1.5 Specification.
- Web Service Invocation Framework (WSIF) and Web Services Description Language (WSDL) with JCA bindings.

About Connecting J2EE-Compliant Applications to the BRM System

You connect J2EE-compliant external applications to your BRM system by using the BRM JCA Resource Adapter. The adapter is deployed on a J2EE-compliant application server, such as Oracle Application Server or Oracle WebLogic Server. External applications send requests for information to the adapter, which then calls BRM opcodes and returns data to the external application. This enables any J2EE-compliant application to integrate with the BRM software.

For example, a customer relationship management (CRM) application, such as Oracle’s Siebel CRM, that is connected to an application server can send billing and rating requests to the BRM software for processing. For information about using the adapter as part of Oracle Application Integration Architecture, see "Using BRM with Oracle Application Integration Architecture" in BRM Concepts.

Overview of the BRM JCA Resource Adapter Architecture

When a J2EE-compliant application sends a request to the application server, the Business Process Execution Language (BPEL) process on the application server routes the request to the adapter on the application framework (AF). Figure 1–1 illustrates the BRM JCA Resource Adapter architecture.
When the adapter receives the request, it performs the following:

1. Retrieves the opcode to call from the interaction specification.
2. Optionally validates the input XML against the schema for the target opcode.
3. Converts the input XML into an input flist.
4. Calls the BRM opcode through the BRM context, using the opcode name, opcode flags, and other attributes passed in the interaction specification.

The opcode processes the request and returns the result in an output flist to the adapter. The adapter then performs the following:

1. Converts the opcode’s output flist into output XML and stores it in the payload of the output XML record.
2. Optionally validates the output XML against the schema for the opcode.
3. Returns the output XML to the calling application.
4. In case of an error, returns a resource exception.

The adapter connects to BRM by using internal connection pooling, which can be configured as part of the adapter’s deployment when creating connection factories.

**About the BRM JCA Resource Adapter**

The adapter is a system-level software driver that is used by a J2EE-compliant application to connect to the BRM system. The adapter complies with J2EE Connector Architecture 1.5 and follows the guidelines for a standalone packaged adapter. It supports synchronous outbound communication, which means that the request message originates from the J2EE-compliant application and is returned by the BRM system.

The adapter exposes the BRM API through the following:

- JCA common client interface (CCI) as an Interaction.
- WSIF provider and WSDL with JCA bindings as a WSIF API.

The adapter performs the following main functions:
About the BRM JCA Resource Adapter

- Receives data sent by the J2EE-compliant application. See "About Sending Data to the BRM JCA Resource Adapter".
- Validates the input and output XML data. See "About Validating Input and Output XML Data".
- Converts XML data into BRM flist format and calls the appropriate opcode. See "About Converting XML Data and Calling the BRM API".
- Converts output flists into XML format and returns them to the calling application. See "About Converting Output flist Data into XML Format".

About Sending Data to the BRM JCA Resource Adapter

The adapter implements CCI as part of the client API implementation. Any J2EE-compliant application that uses CCI can send data to the adapter, but the XML data it sends must conform to the BRM opcode XML schema.

To send data to the adapter, J2EE-compliant applications send the following data in the interaction specification:

- **The BRM opcode to call.** For a list of opcodes that can be called, see the BRM WSDL files. By default, the WSDL files are installed in the `BRM_Home/apps/brm_integrations/wsdls` directory. For information about WSDL files, see "About BRM Web Services and WSDL Files".

  `BRM_Home/apps/brm_integrations/wsdls` contains sub-directories to support BPEL flows on Oracle Application Integration Architecture (AIA) for Communications. For Oracle AIA 11.x, use the WSDL files in `BRM_Home/apps/brm_integrations/wsdls/11g`.

- **(Optional) Opcode schema file name.** This specifies the location of the input and output schema files. By default, the schema files are installed in the `BRM_Home/apps/brm_integrations/schemas` directory.

  **Note:** The schema file name is mandatory if validation is required.

- **(Optional) Opcode flags.** This defaults to zero.

- **(Optional) Whether it is a base opcode.** This specifies whether the opcode is a base opcode. See "Base Opcodes" in BRM Developer’s Reference for more information. By default, an opcode is not a base opcode.

- **(Optional) Validation requirements.** This specifies whether the adapter validates the input XML and output XML against the opcode schema. By default, validation is turned off.

J2EE-compliant applications send the input data for the opcode in the XML record payload. This includes all data that is required by the opcode’s input flist.

For more information about the interaction specification parameters, see the BRM WSDL files. For more information about the input data, see the appropriate opcode flist specification.

The adapter returns a resource exception in the following situations:

- If any of the mandatory fields are not passed in the interaction specification.
- If the input XML is not in an XML record or is not an instance of the XML record.
- If there is any error calling the opcode.
- If there is any error in the opcode’s execution.
- If there is any error in the input or output validation.

**About BRM Web Services and WSDL Files**

The WSDL files included with the adapter define the opcodes that can be called as well as the attributes required to call a specific opcode. The opcodes are grouped by functional area into a Web service. For example, one Web service defines the billing opcodes and another Web service defines the payment opcodes. The adapter includes one WSDL file for each Web service.

Table 1–1 describes the WSDL files installed, by default, in the `BRM_Home/apps/brm_integrations/wsdls` directory.

<table>
<thead>
<tr>
<th>WSDL File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRMARServices.wsdl</td>
<td>Defines the accounts receivable Web service, which currently includes these opcodes:</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_AR_ACCOUNT_ADJUSTMENT in</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_AR_BILL_ADJUSTMENT</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_AR_EVENT_ADJUSTMENT</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_AR_GET_ACCT_ACTION_ITEMS</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_AR_GET_ACCT_BAL_SUMMARY</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_AR_GET_ACCT_BILLS</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_AR_GET_ACTION_ITEMS</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_AR_GET_BAL_SUMMARY</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_AR_GET_BILLS</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_AR_GET_BILL_ITEMS</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_AR_ITEM_ADJUSTMENT</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_ARRESOURCE_AGGREGATION</td>
</tr>
<tr>
<td></td>
<td>See “Accounts Receivable FM Standard Opcodes” in <em>BRM Developer’s Reference</em> for more information.</td>
</tr>
<tr>
<td>BRMBalServices.wsdl</td>
<td>Defines the balances Web service, which currently includes these opcodes:</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_BAL_GET_ACCT_BAL_GRP_AND_SVC</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_BAL_GET_ACCT_BILLINFO</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_BAL_GET_BALANCES</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_BAL_GET_BAL_GRP_AND_SVC</td>
</tr>
<tr>
<td></td>
<td>See “Balance FM Standard Opcodes” in <em>BRM Developer’s Reference</em> for more information.</td>
</tr>
<tr>
<td>BRMBaseServices.wsdl</td>
<td>Defines the base Web service, which currently includes these opcodes:</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_READ_FLDS</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_READ_OBJ</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_SEARCH</td>
</tr>
<tr>
<td></td>
<td>See “Base Opcodes” in <em>BRM Developer’s Reference</em> for more information.</td>
</tr>
</tbody>
</table>
### Table 1–1 (Cont.) WSDL Default Files

<table>
<thead>
<tr>
<th>WSDL File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRMBillServices.wsdl</td>
<td>Defines the billing Web service, which currently includes these opcodes:</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_BILL_DEBIT</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_BILL_GET_ITEM_EVENT_CHARGE_DISCOUNT</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_BILL_GROUP_GET_PARENT</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_BILL_GROUP_MOVE_MEMBER</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_BILL_MAKE_BILL_NOW</td>
</tr>
<tr>
<td>BRMCollectionsServices.wsdl</td>
<td>Defines the collections Web service, which currently includes this opcode:</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_COLLECTIONS_SET_ACTION_STATUS</td>
</tr>
<tr>
<td></td>
<td>See “Collections Manager FM Standard Opcodes” in BRM Developer’s Reference for more information.</td>
</tr>
<tr>
<td>BRMCustServices.wsdl</td>
<td>Defines the customer Web service, which currently includes these opcodes:   - PCM_OP_CUST_COMMIT_CUSTOMER</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_CUST_CREATE_PROFILE</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_CUST_DELETE_PAYINFO</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_CUST_DELETE_PROFILE</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_CUST_MODIFY_CUSTOMER</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_CUST_MODIFY_PROFILE</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_CUST_SET_STATUS</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_CUST_UPDATE_CUSTOMER</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_CUST_UPDATE_SERVICES</td>
</tr>
<tr>
<td></td>
<td>See “Customer FM Standard Opcodes” in BRM Developer’s Reference for more information.</td>
</tr>
<tr>
<td>BRMInvServices.wsdl</td>
<td>Defines the invoicing Web service, which currently includes this opcode:   - PCM_OP_INV_VIEW_INVOICE</td>
</tr>
<tr>
<td></td>
<td>Important: You must configure your client application to convert the invoice data received from the PCM_OP_INV_VIEW_INVOICE opcode into the appropriate format. See “About Invoicing Output XML Data”.</td>
</tr>
<tr>
<td></td>
<td>See “Invoicing FM Standard Opcodes” in BRM Developer’s Reference for more information.</td>
</tr>
<tr>
<td>BRMPymtServices.wsdl</td>
<td>Defines the payment Web service, which currently includes this opcode:   - PCM_OP_PYMT_COLLECT</td>
</tr>
<tr>
<td></td>
<td>See “Payment FM Standard Opcodes” in BRM Developer’s Reference for more information.</td>
</tr>
</tbody>
</table>
You can create a new WSDL file or add an opcode description to an existing WSDL file. For more information, see "Generating the WSDL Files for Your System".

### About Validating Input and Output XML Data

The adapter optionally validates the input and output XML by comparing the XML fields and values against the opcode XML schema.

During the validation process, the adapter verifies that all date values passed in the input and output XML meet the ISO 8601 standard.

The adapter supports the following ISO-8601 date formats:

#### Short Formats (without Seconds)

- yyyy-mm-ddT
- yyyy-mm-ddTh
- yyyy-mm-ddThhZ
- yyyy-mm-ddT+hh
- yyyy-mm-ddT-hh
- yyyy-mm-ddThh
- yyyy-mm-ddThh+hh
- yyyy-mm-ddThh-hh
- yyyy-mm-ddThhZ
- yyyy-mm-ddThhZ+hh
- yyyy-mm-ddThhZ-hh
- yyyy-mm-ddThh:mm
- yyyy-mm-ddThh:mmZ
- yyyy-mm-ddThh:mm

---

**Table 1–1 (Cont.) WSDL Default Files**

<table>
<thead>
<tr>
<th>WSDL File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRMSubscriptionService</td>
<td>Defines the subscription Web service, which currently includes these opcodes:</td>
</tr>
<tr>
<td>BRMSWebService.wsdl</td>
<td>- PCM OP_SUBSCRIPTION_CANCEL_DISCOUNT</td>
</tr>
<tr>
<td></td>
<td>- PCM OP_SUBSCRIPTION_CANCEL_PRODUCT</td>
</tr>
<tr>
<td></td>
<td>- PCM OP_SUBSCRIPTION_CANCEL_SUBSCRIPTION</td>
</tr>
<tr>
<td></td>
<td>- PCM OP_SUBSCRIPTION_CHANGE_DEAL</td>
</tr>
<tr>
<td></td>
<td>- PCM OP_SUBSCRIPTION_GET_PURCHASED_OFFERINGS</td>
</tr>
<tr>
<td></td>
<td>- PCM OP_SUBSCRIPTION_PURCHASE_DEAL</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_SUBSCRIPTION_SET_BUNDLE</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_SUBSCRIPTION_SET_DISCOUNTINFO</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_SUBSCRIPTION_SET_DISCOUNT_STATUS</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_SUBSCRIPTION_SET_PRODINFO</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_SUBSCRIPTION_SET_PRODUCT_STATUS</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_SUBSCRIPTION_TRANSFER_SUBSCRIPTION</td>
</tr>
</tbody>
</table>

See “Subscription Management FM Standard Opcodes” in BRM Developer’s Reference for more information.
- yyyy-mm-ddThh:mmZ
- yyyy-mm-ddT+hh:mm
- yyyy-mm-ddT-hh:mm
- yyyy-mm-ddThh:mm+hh
- yyyy-mm-ddThh:mm-hh
- yyyy-mm-ddThh:mmZ+hh
- yyyy-mm-ddThh:mmZ-hh
- yyyy-mm-ddThhmm+hh
- yyyy-mm-ddThhmm-hh
- yyyy-mm-ddThhmm+hh
- yyyy-mm-ddThhmmZ+hh
- yyyy-mm-ddThhmmZ-hh
- yyyy-mm-ddThh:mm:ss

Long Formats (with Seconds)
- yyyy-mm-ddThh:mm:ss
You can use a negative or positive offset from -12:00 to +14:00; for example, \(-05:00\) or \(+10:00\). Optionally, you can use a colon to separate the offset hours and offset minute values.

The adapter includes XML schema for the BRM opcodes specified in the WSDL files. If support for other opcodes, other storable classes, or subclasses is needed, you must convert them into XSD schema before you can use them with the adapter.

- For information about opcode schemas, see "About Validating the XML Schema for BRM Opcodes".
- For information about storable class schemas, see "About Validating the XML Schema for Storable Classes and Subclasses".
About Validating the XML Schema for BRM Opcodes
The adapter validates the XML data passed in by the J2EE-compliant application by using the opcode XSD schema. For example, if the input XML includes data for the PCM_OP_CUST_COMMIT_CUSTOMER opcode, the adapter validates the input XML data against the PCM.OP.CUST.COMMIT_CUSTOMER input schema. Likewise, when returning the request, the adapter validates the output XML data against the opcode’s output schema.

You must generate the schema for any opcode that the adapter needs to validate by using the `pin_opspec_to_schema` utility. For more information, see "Generating the Schema for Your Opcodes".

About Validating the XML Schema for Storable Classes and Subclasses
Opcode schemas define the schema for each opcode only and contain an extension section that references storable classes. To validate flist fields that reference storable class fields, the adapter includes the schema for storable classes in the schema for the opcodes.

The opcode schemas packaged with the adapter have `include` sections for the following storable class schema files:

- `/account`
- `/billinfo`
- `/discount`
- `/event`
- `/event/billing`
- `/payinfo`
- `/product`
- `/profile`
- `/purchased_discount`
- `/purchased_product`
- `/service`

You must generate schema files for these storable classes and their subclasses. If you added any opcode schemas to the adapter, you must also generate schema files for any storable classes they need to access.

You generate the XSD schema for your storable classes and subclasses by using the `pin_dd_to_schema` utility. The utility generates the schema for all storable classes and their subclasses. For example, if you specify to generate schema for the `/service/telco/gsm` class, the utility generates the schema for `/service/telco/gsm`, `/service/telco/gsm/telephony`, `/service/telco/gsm/data`, and all other subclasses of the `/service/telco/gsm` class in your system.

To create the schema for storable classes, see "Generating the Schema for Your Storable Classes and Subclasses".

About Converting XML Data and Calling the BRM API
The adapter converts the input XML into an opcode input flist according to the rules in the XSL style sheet. The adapter then passes the flist to the appropriate opcode through the Java Portal Communications Module (PCM). In case of an error calling the
opcode or an error in the opcode functionality, the adapter returns an error to the calling application.

### About Converting Output flist Data into XML Format

The adapter converts output flists into XML format by using the BRM short tag name format. That is, the adapter creates XML tags by removing the “PIN_FLD_” prefix from the flist field and array name. For example, the PIN_FLD_LOGIN flist field is mapped to the LOGIN XML tag, and the PIN_FLD_SERVICES flist array is mapped to the SERVICES XML tag.

The XML tags for the PIN_FLD_INHERITED_INFO and PIN_FLD_EXTENDED_INFO substructs, however, are constructed using the following syntax:

```
Class Extension
```

The value of Class is defined in the `brm_extensions.xml` file, which specifies the flist field from which to build the tag. The file is preconfigured for the opcode and storable class schemas shipped with the adapter. If your system contains custom opcodes that reference storable class fields, you must modify this file. For information, see "Specifying the XML Tags for Extended Fields".

### About Invoicing Output XML Data

When you use the PCM_OP_INV_VIEW_INVOICE opcode to retrieve and display invoices, the adapter returns an unformatted invoice to the calling application. The adapter returns invoicing data in the BUFFER XML field in hexBinary format and the type of invoice format in the TYPE_STR XML field, as described in the opcode schema. To view a formatted invoice, you must configure your client application to convert the invoicing data into the appropriate format.

### About BRM JCA Resource Adapter Connection Management

Connection management allows J2EE-compliant applications to connect to the BRM system through the adapter. The adapter manages all connections and establishes physical connections through the Java PCM.

When a J2EE-compliant application sends an XML record, the adapter performs the following:

- Instantiates a CCI Connection object.
- Establishes a connection to BRM through the Java PCM.
- Stores a reference of the BRM context in the CCI Connection object.
- Returns the CCI Connection object to the application server.

### About BRM JCA Resource Adapter Transaction Management

The adapter provides the following levels of transaction support as defined in the JCA 1.5 specification. You can deploy JCA Resource Adapter in either mode:

- **No transaction**: In this mode, the calling application controls the transaction.
- **Local transaction**: In this mode, the JCA Resource Adapter does not participate in a global transaction; however, the transaction management initiated and managed by the transaction manager is required between the Business Process Execution Language (BPEL) process manager and the JCA Resource Adapter.
You can deploy the JCA Resource Adapter in different transaction modes by configuring the `ra.xml` file, which is packaged with the JCA Resource Adapter archive (OracleBRMJCA15Adapter.rar) file.

### About Creating Multiple Accounts in a Single Transaction

You can create multiple BRM accounts in one transaction with the JCA Resource Adapter. This is useful, for example, when a single order in Oracle Communications Order and Service Management (OSM) produces multiple new customers. It enables Application Integration Architecture (AIA) to maintain the cross-references between the OSM order and its associated BRM accounts.

To create multiple accounts in one transaction, make all calls to the `PCM_OP_CUST_COMMIT_CUSTOMER` opcode any time within that transaction. The order of calling the `PCM_OP_CUST_COMMIT_CUSTOMER` opcode is not relevant.

For more information on the `PCM_OP_CUST_COMMIT_CUSTOMER` opcode, see BRM Developer’s Reference.

### About BRM JCA Resource Adapter Security

The adapter logs a J2EE-compliant application into the BRM software using basic security:

- **Client authentication.** The adapter authenticates a valid J2EE-compliant application with a user name and password. You can implement more robust authentication by using the adapter policies.

- **Client authorization.** By default, the adapter does not perform authorization, but you can implement it by using the Oracle Identity Manager. You can also use Oracle Identity Manager to implement roles and authorization based on roles.

- **Secure communication.** A secure communication channel between the adapter and BRM is dependent on the implementation.

**Note:** The adapter does not support reauthentication or reauthorization.

If the client passes erroneous security information, the adapter returns a security exception to indicate the error.

You set security by using entries in the adapter deployment descriptor (`ra.xml`) file. For more information, see "Connecting the Adapter to BRM in Oracle WebLogic Server".

### About BRM JCA Resource Adapter Logging

The JCA Resource Adapter supports Java Unified Logging (JUL), which allows the adapter to use the application server’s logging library.

You specify how the adapter generates log files by editing the OC4J logging configuration file (`j2ee-logging.xml`). For information, see "Configuring Java Logging on Oracle Application Server”.

You view the adapter’s log files by using the Oracle Enterprise Manager Application Server Control Console. See Oracle Containers for J2EE Configuration and Administration Guide for instructions on viewing log files generated by an OC4J instance.
About Deploying and Connecting the BRM JCA Resource Adapter

After installing the adapter package, you need to deploy it on your application server. The connection parameters to connect the adapter to BRM are defined in the deployment descriptor packaged along with the adapter archive (RAR) file. The deployment descriptor includes the following files:

- Adapter deployment descriptor file (ra.xml): Includes parameters for connecting the adapter to BRM and the application server. The format of the file is defined in the J2EE Connector Architecture 1.5 Specification.
- Oracle Application Server deployment descriptor file (oc4j-ra.xml): Defines operation parameters that are unique to Oracle Application Server.
- Oracle WebLogic Server deployment descriptor file (weblogic-ra.xml): Defines operation parameters that are unique to Oracle WebLogic Server.

For more information, see "Deploying and Connecting the BRM JCA Adapter on Oracle Application Server" and "Deploying and Connecting the BRM JCA Adapter on Oracle WebLogic Server".

About the BRMApapterServletClient Application

You can use the BRMApapterServletClient sample application to test your custom opcodes and develop customized applications that communicate with the JCA Resource Adapter.

For more information, see "Using the BRMApapterServletClient Application".
Installing the BRM JCA Resource Adapter

This chapter explains how to install the Oracle Communications Billing and Revenue Management (BRM) JCA Resource Adapter on your BRM system.

System Requirements
The BRM JCA Resource Adapter is available for the HP-UX IA64, Linux, Solaris, and AIX operating systems.

Software Requirements
Before installing the BRM JCA Resource Adapter, you must install:

- Third-Party software, which includes the Perl libraries and JRE required for installing BRM components. See "Installing the Third-Party software" in BRM Installation Guide.
- BRM Release 7.4.

The adapter must be deployed on a J2EE 1.4-compliant application server that has implemented the JCA 1.5 specification. The adapter runs only in a managed environment. Currently, the BRM JCA Resource Adapter is deployable on the Oracle Application Server only.

Information Requirements
You need the following information about your existing system during the BRM JCA Resource Adapter installation:

- The location of your BRM_Home directory and log directories.
- The Connection Manager (CM) port number.

Installing the BRM JCA Resource Adapter
BRM JCA Resource Adapter is bundled with Synchronization Queue Data Manager and Web Services Manager in the Web Services Manager downloadable package. You must download the Web Services Manager package and extract these three components before you can install JCA Resource Adapter. For more information about Synchronization Queue Manager and Web Services Manager, see "Understanding the Synchronization Queue Data Manager" in BRM Synchronization Queue Manager and "Installing and Configuring Web Services Manager" in BRM Web Services Manager.

To install the BRM JCA Resource Adapter:
1. Download the Web Services Manager package (7.4_WebServicesMgr_platform_32_opt.tar.Z) to a temporary directory (temp_dir). For information on downloading the software, see "Downloading the BRM Applications Media Pack" in BRM Web Services Manager.

Important:
- If you download to a Windows workstation, use FTP to copy the package to a temporary directory on your UNIX server.
- You must increase the heap size used by the Java Virtual Machine (JVM) before running the installation program to avoid “Out of Memory” error messages in the log file. For information, see "Increasing Heap Size to Avoid “Out of Memory” Error Messages" in BRM Installation Guide.

2. Go to temp_dir and uncompress and extract the .tar.Z file. The following files are extracted:
   - 7.4_BRM_JCA_Adapter_platform_32_opt.bin: JCA Resource Adapter
   - 7.4_DM_AQ_platform_32_opt.bin: Synchronization Queue Data Manager
   - 7.4_WebServicesMgr_platform_32_opt.bin: Web Services Manager

3. Go to the directory where you installed the Third-Party package and source the source.me file.

   Note: You must source the source.me file to proceed with installation, otherwise “suitable JVM not found” and other error messages appear.

   Bash shell:
   ```sh```
   source source.me.sh
   ```
   C shell:
   ```csh```
   source source.me.csh
   ```
   For more information, see "Installing the Third-Party Software" in BRM Installation Guide.

4. Go to temp_dir and enter the appropriate command for your operating system.
   - For Solaris, AIX, HP-UX IA64, and 64-bit Linux systems, enter this command:
     ```bash```
     7.4_BRM_JCA_Adapter_platform_32_opt.bin
     ```
   - For 32-bit Linux systems, enter this command, where OracleAS_home is the directory where you installed the Oracle Application Server (AS) software:
     ```bash```
     7.4_BRM_JCA_Adapter_linux_32_opt.bin -W
     ThirdPartyCheckSequence.active="false" -is:javahome OracleAS_home/jdk/jre/bin/java
     ```
Follow the instructions displayed during installation. The default installation directory for the BRM JCA Resource Adapter is `BRM_Home/apps/brm_integrations`.

**Note:** The `pin_setup` script starts all required BRM processes.

Go to the `BRM_Home/setup` directory and run the `pin_setup` script.

Your BRM JCA Resource Adapter installation is now complete.

**What's Next?**

Deploy and configure the BRM JCA Resource Adapter on your application server and generate the schema for any custom BRM opcodes and storable classes in your system. See "Deploying and Configuring the BRM JCA Resource Adapter".
Deploying and Configuring the BRM JCA Resource Adapter

This chapter describes how to deploy and configure the Oracle Communications Billing and Revenue Management (BRM) JCA Resource Adapter.

Before configuring the adapter, you should be familiar with the following:

- BRM concepts and architecture. See "Putting Together Your BRM System" in BRM Installation Guide.
- BRM opcodes and flists. See "Understanding Flists and Storable Classes" in BRM Developer’s Guide.
- J2EE Connector Architecture (JCA) 1.5 Specification.
- XML, XML Schema Definition (XSD), and XML StyleSheet Language (XSL).
- Web Services Description Language (WSDL).

Overview of the BRM JCA Resource Adapter Configuration Procedure

The procedure for setting up the BRM JCA Resource Adapter includes the following tasks:

1. Installing the adapter on your BRM system, if you have not already done so. See "Installing the BRM JCA Resource Adapter".
2. Compiling your custom Java Portal Connection Module (PCM) classes and adding them to the adapter. See "Creating and Packaging Custom Classes and Fields".
3. Generating the schema files for the adapter. See "Generating the Schema Files for Your System".
4. Specifying how to construct XML tags. See "Specifying the XML Tags for Extended Fields".
5. Generating the WSDL files for the adapter. See "Generating the WSDL Files for Your System".
6. Deploying and connecting the adapter on your application server:
   - For Oracle Application Server, see "Deploying and Connecting the BRM JCA Adapter on Oracle Application Server".
   - For Oracle WebLogic Server, see "Deploying and Connecting the BRM JCA Adapter on Oracle WebLogic Server".
   - For Oracle GlassFish Server, see "Deploying and Connecting the BRM JCA Adapter on Oracle GlassFish Server".
Creating and Packaging Custom Classes and Fields

The JCA Resource Adapter uses Java PCM APIs to connect to BRM. If the adapter will be calling opcodes that reference custom storable class fields, you must make the custom storable class fields available to the adapter by:

- Compiling your custom classes and adding them to the adapter.
- Adding the `Infranet.properties` file as a resource to the adapter. The `Infranet.properties` file defines which package contains the custom field definitions.

**Note:** The application server class loader does not load the `Infranet.properties` file, so you must make the file available as a resource.

To make your custom and modified storable classes available to the JCA Resource Adapter:

1. Use Storable Class Editor to create or modify your storable classes and fields. For information, see Storable Class Editor Help.
   
   For information about editing storable classes, see "Creating, Editing, and Deleting Fields and Storable Classes" in BRM Developer's Guide.

2. Choose **File - Generate Custom Fields Source** to create source files for your custom fields. See Storable Class Editor Help for detailed instructions.

   Storable Class Editor creates a C header file called `cust_flds.h`, a Java properties file called `InfranetPropertiesAdditions.properties`, and a Java source file for each custom field.

3. For each Java application that will use these fields, copy the contents of the `InfranetPropertiesAdditions.properties` file and paste it into each application’s `Infranet.properties` file.

4. In the directory where Storable Class Editor created the Java source files, compile the source files:

   ```
   javac -d . *.java
   ```

5. Package the class files created in step 4 into a JAR file:

   ```
   jar cvf filename.jar *.class
   ```

6. In the CLASSPATH, add the location of the JAR file.

7. Create a package that includes both the JAR file and the updated BRM `Infranet.properties` file.

8. Place the package in the adapter’s working directory (`Oracle_home/j2ee/Instance/connectors/AdapterDeploymentName/AdapterDeploymentName`).

9. Restart your application server.

Generating the Schema Files for Your System

The JCA Resource Adapter uses schema files to validate data it sends to or receives from the BRM software.

To generate the schema files for your system, perform these tasks:
1. If you modified any opcodes, generate schemas for the opcodes in your BRM system. See "Generating the Schema for Your Opcodes".

2. Generate schemas for the storable classes and subclasses in your BRM system. See "Generating the Schema for Your Storable Classes and Subclasses".

3. In your opcode schema files, specify the location of your storable class schema files. See "Specifying the Location of the Storable Class Schema Files in the Opcode Schema Files".

---

**Note:** After generating the opcode and storable class schema files, copy the schema files to a location that is accessible to the adapter. Make sure that this location is the same as the one specified in the `include` section of the opcode schema files and in the opcode schema `InteractionSpec` attribute in the WSDL files. See "Specifying the Location of the Storable Class Schema Files in the Opcode Schema Files" and "Generating the WSDL Files for Your System".

---

**Generating the Schema for Your Opcodes**

The adapter package includes all of the opcode schemas and flist specifications you need for a default integration.

If you customized any of the opcodes that are supported by the adapter or if you added support for new opcodes, you must generate XSD schema files for the opcodes. The steps you need to take depend on the type of changes you made, as shown in Table 3–1:

**Table 3–1  Procedure for Modifying Opcodes**

<table>
<thead>
<tr>
<th>Opcode Modification</th>
<th>Procedure</th>
</tr>
</thead>
</table>
| Customized an opcode that the JCA Resource Adapter already supports | 1. Modify the opcode’s XML specification file. By default, the opcode specification XML files are installed in the `BRM_Home/apps/brm_integrations/opspecs` directory. See "Creating Custom Opcode Specification XML Files".  
2. Run the `pin_opspec_to_schema` utility. See "Converting flist Specs into XSD Schema Files with pin_opspec_to_schema". |
| Added a custom data type to an opcode that the JCA Resource Adapter already supports | 1. Modify the opcode’s XML specification file. By default, the opcode specifications are installed in the `BRM_Home/apps/brm_integrations/opspecs` directory. See "Creating Custom Opcode Specification XML Files".  
2. Modify the `pin_opspec_to_schema` style sheet to handle custom data types or custom data structures. See "Specifying the XSL Rules to Create the Opcode Schema".  
3. Run the `pin_opspec_to_schema` utility. See "Converting flist Specs into XSD Schema Files with pin_opspec_to_schema". |
Creating Custom Opcode Specification XML Files

You must create XML specification files for any opcodes that you customize or add to the adapter. Create the XML specification files by following the `BRM_Home/apps/brm_integrations/stylesheets/opspec.xsd` file. You can then convert the opcode specifications into XSD schema by using the `pin_opspec_to_schema` utility.

Specifying the XSL Rules to Create the Opcode Schema

The `pin_opspec_to_schema` utility uses the `BRM_Home/brm_integrations/stylesheets/pin_opspec_to_schema.xsl` style sheet to generate the schema for BRM opcodes. If your opcode references custom data types or custom data structures, you must customize the `pin_opspec_to_schema.xsl` style sheet to handle your customizations.

For a list of the supported BRM data types, see "Understanding the BRM Data Types" in `BRM Developer’s Guide`.

Converting flist Specs into XSD Schema Files with pin_opspec_to_schema

You convert opcode flist specifications into XSD schema by using the `pin_opspec_to_schema` utility.

To generate XSD schema, go to the `BRM_Home/apps/brm_integrations` directory and run the following command:

```
pin_opspec_to_schema -i InputFile [-o OutputFile]
```

where:

- `-i InputFile` specifies the name and location of the opcode’s XML flist specification. By default, the utility looks for the file in the current directory.
- `-o OutputFile` creates the XSD schema output file using the name you specify. By default, the utility creates a file named `opcodename.xsd` in the directory from which you run the utility.

Generating the Schema for Your Storable Classes and Subclasses

The adapter uses storable class schema files to validate storable class extensions passed in the input XML. See “Connecting J2EE-Compliant Applications to BRM”.

To generate storable class schema files for your system:

1. Determine which storable class files you need to convert into XSD schema. See "Determining the Storable Classes to Convert Into Schema".

---

### Table 3–1 (Cont.) Procedure for Modifying Opcodes

<table>
<thead>
<tr>
<th>Opcode Modification</th>
<th>Procedure</th>
</tr>
</thead>
</table>
| Added a custom opcode to the JCA Resource Adapter | 1. Configure BRM to recognize the custom opcode and manually create the opcode’s XML specification file. See “Writing a custom Facilities Module” in `BRM Developer’s Guide`.  
2. If your opcode supports custom data types or data structures, modify the `pin_opspec_to_schema` style sheet. See “Specifying the XSL Rules to Create the Opcode Schema”.  
3. Run the `pin_opspec_to_schema` utility. See “Converting flist Specs into XSD Schema Files with pin_opspec_to_schema”.  
Note: Make sure you add your custom opcode to a WSDL file. See "Generating the WSDL Files for Your System". |
2. Convert your storable classes and subclasses by running the **pin_dd_to_schema** utility. See "Generating Storable Class Schema Files with pin_dd_to_schema".

**Determining the Storable Classes to Convert Into Schema**

The adapter package includes all of the storable class schema files required for a default integration. By default, they are installed in the `BRM_Home/apps/brm_integrationsschemas` directory.

You will need to generate additional schema files if your system contains the following:

- Extensions to the `/account`, `/billinfo`, `/discount`, `/event`, `/event/billing`, `/payinfo`, `/product`, `/profile`, `/purchased_discount`, `/purchased_product`, or `/service` storable classes. If you extended any of these storable classes, you must create a new schema file for the base class. For example, if you added a subclass to the `/profile` storable class, you must regenerate the `profile.xsd` schema file.

- Storable classes required by opcodes you added to the adapter. If you added an opcode schema to the adapter, you can determine whether it requires a storable class by checking for an `include schemaLocation` entry at the beginning of the schema file. If the entry exists, it lists the required storable class schema files. For example, the `PCM_OP_CUST_COMMIT_CUSTOMER.xsd` schema file includes these entries, which indicate that the opcode needs schema files for the `/service` and `/profile` classes:

```xml
<?xml version = '1.0' encoding = 'UTF-8'?>
<xsd:schema
    targetNamespace="http://www.xmlns.oracle.com/BRM/schemas/BusinessOpcodes"
    elementFormDefault="qualified"
    xmlns="http://www.xmlns.oracle.com/BRM/schemas/BusinessOpcodes"
    xmlns:op="http://www.portal.com/schemas/BusinessOpcodes"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <xsd:include schemaLocation="service.xsd"/>
    <xsd:include schemaLocation="profile.xsd"/>
    ...
</xsd:schema>
```

**Generating Storable Class Schema Files with pin_dd_to_schema**

You use the **pin_dd_to_schema** utility to generate the schema for storable classes and their subclasses from the data dictionary. The adapter uses the schema to validate the object fields that are passed in the input XML.

---

**Note:** The utility requires a configuration file to access the BRM database. For more information, see "Using Configuration Files to Connect and Configure Components" in *BRM System Administrator’s Guide*.

---

Before you run the **pin_dd_to_schema** utility, make sure you have added the storable class changes to the BRM data dictionary.

To generate the schema for the storable classes and subclasses in your system:

1. Using a text editor, create a configuration file that lists all storable classes you would like to convert. For example, create a `sample.txt` file that includes the following entries:
2. Save and close the file.

3. On Linux systems only, set the LC_ALL environment variable to C:

   ```
   setenv LC_ALL C
   ```

4. Go to the `BRM_Home/apps/brm_integrations` directory and enter the following command:

   ```
   pin_dd_to_schema -e ConfigFile | -r StorableClass
   ```

   where:
   - `-e ConfigFile` generates the schema for all storable classes listed in the configuration file as well as their subclasses.
   - `-r StorableClass` generates the schema for the specified storable class and all of its subclasses.

   The utility generates the schema files in the directory from which you run the utility.

5. Copy the schema files to a location that is accessible to the adapter during the XML validation process. For example, the `BRM_Home/apps/brm_integrations/schemas` directory.

### Specifying the Location of the Storable Class Schema Files in the Opcode Schema Files

Opcode schema files contain an `include` section that lists each storable class schema file that they need. You must manually update the opcode schema files to point to the storable class schema file's correct location.

Not all opcodes need storable classes, so only a subset of the opcode schema files have the `include` section. The following list shows the opcode schemas that are packaged with the adapter and have an `include` section:

- `PCM_OP_CUST_COMMIT_CUSTOMER`
- `PCM_OP_CUST_CREATE_PROFILE`
- `PCM_OP_CUST_MODIFY_CUSTOMER`
- `PCM_OP_CUST_MODIFY_PROFILE`
- `PCM_OP_CUST_UPDATE_CUSTOMER`
- `PCM_OP_CUST_UPDATE_SERVICES`
- `PCM_OP_PYMT_COLLECT`
- `PCM_OP_READ_FLDS`
- `PCM_OP_READ_OBJ`
- `PCM_OP_SUBSCRIPTION_CANCEL_DISCOUNT`
- `PCM_OP_SUBSCRIPTION_CANCEL_PRODUCT`
- `PCM_OP_SUBSCRIPTION_CANCEL_SUBSCRIPTION`
- `PCM_OP_SUBSCRIPTION_CHANGE DEALINGS`
Specifying the XML Tags for Extended Fields

The **brm_extensions.xml** file defines how to construct XML tags for PIN_FLD_INHERITED_INFO and PIN_FLD_EXTENDED_INFO substructs when converting output flists into XML. The default file is preconfigured for the opcode and storable class schemas shipped with the adapter. You must modify this file if you added to the adapter any opcodes that reference storable class fields.

The adapter creates XML tags for PIN_FLD_INHERITED_INFO and PIN_FLD_EXTENDED_INFO substructs by using the following syntax:

```
ClassExtension
```

where `Class` is the subclass name, using an underscore (_) notation. For example, the XML tag for `/service/telco/gsm` is written as `service_telco_gsmExtension`. You configure how the adapter determines the subclass name by using the **brm_extensions.xml** file. The file specifies the following for each opcode:

```
Note: The adapter does not support accessing schemas through URLs.
```

3. Save and close the file.
The flist array that contains the PIN_FLD_INHERITED_INFO or PIN_FLD_EXTENDED_INFO substruct.

The flist field in which to look up the subclass name.

To specify how to create XML tags for custom extensions, perform the following tasks:

1. Open the `brm_extensions.xml` file in an XML editor.

   ```xml
   <brm_extensions>
   <opcode name="PCM_OP_CUST_COMMIT_CUSTOMER">
     <container name="PIN_FLD_SERVICES">PIN_FLD_SERVICE_OBJ</container>
   </opcode>
   ...
   </brm_extensions>
   ```

   In this example, if the PIN_FLD_SERVICE_OBJ output flist field is 0.0.0.1 /service/ip 12345 0, Class is `service_ip` and the corresponding XML extension tag is `service_ipExtension`.

2. Enter the opcode that requires the subclass fields in the `opcode name` entry:

   ```xml
   <opcode name="Opcodename"/>
   ```

3. Enter the flist field used to construct the XML tag in the `container name` entry:

   ```xml
   <container name="Arrayname">Fieldname</container>
   ```

   where:

   - `Arrayname` specifies the array that contains the PIN_FLD_INHERITED_INFO or PIN_FLD_EXTENDED_INFO substruct.
   - `Fieldname` specifies the flist field in which to look up the subclass name.

   For example, the following entry for the PCM_OP_CUST_COMMIT_CUSTOMER opcode specifies that when the PIN_FLD_SERVICES output flist array contains PIN_FLD_INHERITED_INFO or PIN_FLD_EXTENDED_INFO, the adapter looks up the subclass name from the PIN_FLD_SERVICE_OBJ flist field.

   ```xml
   <brm_extensions>
   <opcode name="PCM_OP_CUST_COMMIT_CUSTOMER">
     <container name="PIN_FLD_SERVICES">PIN_FLD_SERVICE_OBJ</container>
   </opcode>
   ...
   </brm_extensions>
   ```

   In this example, if the PIN_FLD_SERVICE_OBJ output flist field is 0.0.0.1 /service/ip 12345 0, Class is `service_ip` and the corresponding XML extension tag is `service_ipExtension`.

4. Save and close the file.

5. Restart the adapter.

Generating the WSDL Files for Your System

The adapter package includes all of the WSDL files you need for a default implementation. If you customized the adapter to support additional opcodes, you must create or modify a WSDL file.

To generate a WSDL file:

- Create an XML configuration file that specifies the opcodes associated with a Web service. See "Defining a Web Service".
Generate the WSDL file for the Web service by running the `pin_wsdl_generator` utility. See "Generating WSDL Files for Web Services".

#### Defining a Web Service

To define a Web service, you specify the following in a configuration file:

- The Web service name.
- The opcodes associated with the Web service.
- The location of each opcode’s schema files.

You create the configuration file manually by using an XML editor. The adapter provides a sample XML configuration file (`BRM_Home/apps/brm_integrations/config/pin_wsdl_generator.xml`) that you can use as a guide.

The configuration file includes four main sections: namespces, groups, binding-info, and service-info. You configure the groups section only; the namespaces, binding-info, and service-info sections are preconfigured for Oracle Application Server.

#### Defining Default Values for a Web Service

You specify the default values for a Web service in the `group-info name` entry. The default values apply to all opcodes in the Web service, but they are overridden by any opcode-specific values defined in the `opcode name` entry. For information, see "Defining Settings for Individual Opcodes".

```xml
<group-info name="BASE" base="True" includePath="..\schemas/"
            wsdlFileInFix="Base" validation="ValidationNotRequired"
            validationAttr="ValidationNotRequired"/>
```

where:

- `group-info name` specifies the name of the Web service.
- `base` specifies whether the opcodes in this Web service are base opcodes. See "Base Opcodes" in BRM Developer’s Reference. The default is `False`.
- `includePath` specifies the relative path to the schema files. The utility adds this value to the WSDL file’s `include` entries.
- `wsdlFileInFix` specifies the value to add to the WSDL file name, using this syntax: `BRMValueServices.wsdl`. For example, if `wsdlFileInFix` is set to `Bill`, the WSDL file name is `BRMBillServices.wsdl`. If this attribute is not present, the utility uses the `group-info name` value in the file name’s syntax.
- `validation` specifies whether the adapter validates the input and output XML for the opcodes in this group. The valid values are:
  - `ValidationRequired`, which specifies to validate the input and output XML. You can limit validation to just the input XML or just the output XML by using the `validationAttr` entry.
  - `ValidationNotRequired`, which specifies to skip the validation process.
  - `InternalSchemaValidation`, which specifies that the utility determines whether to validate the input and output XML by reading the validation requirement from the interaction specification.

The default is `ValidationRequired`. 

Generating the WSDL Files for Your System

- **validationAttr** specifies whether to validate only the input XML or only the output XML. This entry is valid only when the **validation** entry is set to **ValidationRequired**. The valid values are:
  - **InputValidationNotRequired**, which specifies to validate only the output XML.
  - **OutputValidationNotRequired**, which specifies to validate only the input XML.

**Defining Settings for Individual Opcodes**

You use the **opcode name** lines to list the opcodes that are in the Web service. Each opcode in the Web service must have its own **opcode name** entry.

---

**Note:** The opcode-specific settings in this section override the default settings in the **group-info name** entry. If you set both default values and values specific to an individual opcode, the utility uses the opcode-specific settings first.

```xml
<opcode name="PCM_OP_SEARCH"
  base="True"
  custom="no"
  opcodeNumber="89"
  includePath="../schemas/"
  wsdlFileInFix="Base"
  validation="ValidationRequired"
  validationAttr="InputValidationNotRequired">
  <opcodeFlag>1</opcodeFlag>
</opcode>
```

where:

- **opcode name** specifies the name of an opcode in the Web service.
- **base** specifies whether the opcode is a base opcode. See "Base Opcodes" in BRM Developer’s Reference. The default is **False**.
- **custom** specifies whether the opcode is a custom opcode. The valid values are **yes** and **no**. The default is **no**.
- **opcodeNumber** specifies the opcode number. This field is mandatory if the **custom** field is set to **yes**.
- **includePath** specifies the relative path to the schema files. The utility adds this value to the WSDL file’s **include** entries.
- **wsdlFileInFix** specifies the value to add to the WSDL file name, using this syntax: **BRMValueServices.wsdl**. For example, if **wsdlFileInFix** is set to **Bill**, the WSDL file name is **BRMBillServices.wsdl**. If this attribute is not present, the utility uses the **group-info name** value in the file name’s syntax.
- **validation** specifies whether the adapter validates the input and output XML. The valid values are:
  - **ValidationRequired**, which specifies to validate the input and output XML.
    You can limit validation to just the input XML or just the output XML by using the **validationAttr** entry.
  - **ValidationNotRequired**, which specifies to skip the validation process.
- **InternalSchemaValidation**, which specifies that the utility determines whether to validate the input and output XML by reading the validation requirements from the interaction specification.

The default is **ValidationRequired**.

- **validationAttr** specifies whether to validate only the input XML or only the output XML. This entry is valid only when the **validation** entry is set to **ValidationRequired**. The valid values are:
  - **InputValidationNotRequired**, which specifies to validate only the output XML.
  - **OutputValidationNotRequired**, which specifies to validate only the input XML.

- **opcodeFlag** is the flag to pass to the opcode when it is called by the adapter. The default is 0.

**Sample Web Services Group Configuration File**

The following shows a sample XML configuration file:

```xml
<?xml version='1.0'?>
<wsdl-generate>
  <namespaces>
    <partnerlink-namespace
      prefix="plt">http://schemas.xmlsoap.org/ws/2003/05/partner-link</partnerlink-namespace>
    <jca-namespace
      prefix="jca">http://xmlns.oracle.com/pcbpel/wsdl/jca</jca-namespace>
    <soap-namespace
      prefix="soap">http://schemas.xmlsoap.org/wsdl/soap</soap-namespace>
    <default-namespace>http://schemas.xmlsoap.org/wsdl/</default-namespace>
    <brm-nameSpace
      prefix="brm">http://xmlns.oracle.com/BRM/schemas/BusinessOpcodes</brm-nameSpace>
  </namespaces>
  <groups>
    <group-info name="BAL" includePath="../schemas/" wsdlFileInFix="Bal">
      validation="ValidationRequired" validationAttr="InputValidationNotRequired"/
      <opcode name="PCM_OP_BAL_GET_BAL_GRP_AND_SVC">
        <opcodeFlag>1</opcodeFlag>
      </opcode>
    </group-info>
  </groups>
  <binding-info>
    <!-- binding information specific to J2CA and SOAP. The general binding information is filled by the tool. -->
    <JCA>
      <InteractionSpec>oracle.tip.adapter.brm.BRMInteractionSpec</InteractionSpec>
    </JCA>
    <SOAP>
      <Action>http://localhost/operationName</Action>
      <encodingStyle>http://schemas.xmlsoap.org/soap/encoding/</encodingStyle>
    </SOAP>
  </binding-info>
  <service-info>
    <!-- Service information specific to J2CA and SOAP -->
    <JCA>
      <!-- JNDI name of the BRM JCA Adapter -->
    </JCA>
  </service-info>
</wsdl-generate>
```
Generating WSDL Files for Web Services

After you define a Web service in the XML configuration file, you generate the WSDL file by running the `pin_wsdl_generator` utility.

To generate WSDL files, go to the `BRM_Home/apps/brm_integrations/config` directory and run the following command:

```
pin_wsdl_generator [-c ConfigFile] [-j | -s]
```

where:

- `-c ConfigFile` specifies the name and location of the XML configuration file that describes the grouping of opcodes. By default, the utility uses the `BRM_Home/apps/brm_integrations/config/pin_wsdl_generator.xml` file.
- `-j` specifies to generate WSDL files with JCA bindings. This is the default.
- `-s` specifies to generate WSDL files with SOAP bindings.

The utility generates the WSDL files in the current directory.

Configuring How to Represent Infinite Date Values

In some external applications, the infinite date value is represented as a NULL (empty XML element) value and in other external applications as the epoch time (01-01-1970 1200 AM UTC).

By default, when the JCA Resource Adapter sends data to your external application, the infinite date value is the start of the epoch time.

You can define how the JCA Resource Adapter sets infinite date values by using the `ZeroEpochAsNull` connection factory field.

**Note:** Before configuring the new connection factory field, you must redeploy the JCA Resource Adapter on the server running Oracle Application Server or Oracle WebLogic Server.

To configure how to represent infinite date values:

1. Start the application server or the J2EE instance.
2. From the Oracle Application Server ASC page or the WebLogic Server Administration Console, go to the resource adapter home page and create connection factories.
3. Set the `ZeroEpochAsNull` connection factory field to either of the following values:
   - `true` sends the date field value as NULL to represent an infinite start or end date.
   - `false` sends the date field value as the start of the epoch time. This is the default.
Deploying and Connecting the BRM JCA Adapter on Oracle Application Server

To deploy and configure the adapter on Oracle Application Server, perform these tasks:

- Deploying the BRM JCA Resource Adapter on Oracle Application Server
- Connecting the Adapter to BRM in Oracle Application Server
- Configuring Java Logging on Oracle Application Server

Deploying the BRM JCA Resource Adapter on Oracle Application Server

The adapter is dependent on JAR files to deploy properly. Table 3–2 lists the JAR files that the adapter requires from each application in your system.

<table>
<thead>
<tr>
<th>Application</th>
<th>JAR files</th>
</tr>
</thead>
<tbody>
<tr>
<td>J2EE application server</td>
<td>classes12.jar, connector15.jar, and jta.jar</td>
</tr>
<tr>
<td>Oracle BPEL process</td>
<td>bpm-infra.jar, orabpel-thirdparty.jar, orabpel.jar, and xmlparserv2.jar</td>
</tr>
<tr>
<td>Apache</td>
<td>xercesImpl.jar</td>
</tr>
</tbody>
</table>

If you are deploying the adapter in a standalone OC4J instance, make sure these JAR files are available to the class loader that is loading the adapter.

If you are deploying the adapter by using Oracle SOA Suite, these JAR files are available as part of the oracle.bpel.common code source. You import these libraries as follows:

1. Open the Oracle_home/j2ee/Instance/config/applications.xml configuration file for the J2EE instance.
2. Add the oracle.bpel.common entry (shown in bold below) to the imported-shared-libraries section of the file:
   ```xml
   <imported-shared-libraries>
     <import-shared-library name="adf.oracle.domain"/>
     <import-shared-library name="oracle.bpel.common"/>
   </imported-shared-libraries>
   ```
3. Save and close the file.
4. Restart the application server or the J2EE instance.

After you make the JAR files available, deploy the adapter on Oracle Application Server by using either Oracle Application Server Application Server Control (ASC) or the Oracle admintool.jar file. Copy the adapter archive file (BRM_Home/apps/brm_integrations/jca_adapter/OracleBRMJCA15Adapter.rar) from the installation directory to a location that is accessible to the adapter deployment tool. You can then open and deploy the archive file on your application server. When deploying the archive file, make sure you:

- Name the adapter BRMJCAAdapter.

Note: Connection factory field values must be lowercase.
Choose the default deployment plan.

Choose the private connection pool.

After successful deployment, return the applications.xml file to its original settings and add the oracle.bpel.common codesource to the BRM Adapter oc4j-ra.xml file:

1. Open the Oracle_Home/j2ee/Instance/config/applications.xml configuration file for the J2EE instance.

2. Remove the following oracle.bpel.common entry (shown in bold below):

   ```xml
   <imported-shared-libraries>
     <import-shared-library name="adf.oracle.domain"/>
     <import-shared-library name="oracle.bpel.common"/>
   </imported-shared-libraries>
   ``

3. Save and close the file.

4. Open the JCA Resource Adapter oc4j-ra.xml file from the Oracle_Home/j2ee/Instance/application-deployments/default/BRMJCAAdapter directory.

5. Add the oracle.bpel.common entry (shown in bold below) to the oc4j-connector-factories section of the file:

   ```xml
   <oc4j-connector-factories...>
     <imported-shared-libraries>
       <import-shared-library name="oracle.bpel.common"/>
     </imported-shared-libraries>
   </oc4j-connector-factories>
   ``

6. Save and close the file.

7. Restart the application server or the J2EE instance.

For more information about deploying the adapter, see your application server’s documentation.

Connecting the Adapter to BRM in Oracle Application Server

You connect the adapter to the BRM software by creating connection pools and connection factories. As part of the adapter deployment, the application server creates the oc4j-ra.xml file from the packaged ra.xml file. The ra.xml file is located in the Oracle_Home/j2ee/Instance/connectors/AdapterDeploymentName/AdapterDeploymentName/META-INF directory. For example, Oracle_Home/j2ee/home/connectors/BRMAAdapter/BRMAAdapter/META-INF/ra.xml.

Create your connection pool by following the performance and tuning guidelines in Configuring Connection Pooling in OC4J in Oracle Containers for J2EE Resource Adapter Administrator’s Guide found at:

http://www.oracle.com/technetwork/documentation/index.html

To configure the JCA Resource Adapter connection factory and connection pool:

1. Restart the application server or the J2EE instance.

2. Use the resource adapter home page from the Oracle Application Server ASC page to create connection pools and connection factories. Create the adapter’s Connection Factory with the connection pooling scheme set to Use PrivateConnection Pool.
For each connection factory, specify the following along with the connection factory field values listed in Table 3–3:

- The JNDI location for the connection factory.
- The connection pool to use.
- How to connect to BRM by using these entries:

Note: Connection factory field values must be lower case.

Table 3–3 Connection Factory Field Values

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AverageOpcodeCount</td>
<td>Specifies the average number of opcode calls per thread.</td>
</tr>
<tr>
<td>BRMConnectionPoolMaxIdleTime</td>
<td>Specifies the maximum amount of time in milliseconds that a free connection in the Connection Pool can be idle. If a free connection is idle for a time greater than or equal to BRMConnectionPoolMaxIdleTime, the connection is removed from the pool. Note: Only the connections created after the limit set by BRMConnectionPoolMinSize is reached are verified for the idle time; if the condition is met, the connection is removed from the pool.</td>
</tr>
<tr>
<td>BRMConnectionPoolMaxSize</td>
<td>Specifies the maximum number of connections the Connection Pool can create.</td>
</tr>
<tr>
<td>BRMConnectionPoolMinSize</td>
<td>Specifies the minimum number of connections created by the Connection Pool when the pool is initialized.</td>
</tr>
<tr>
<td>BRMConnectionPoolTimeout</td>
<td>Specifies the maximum amount of time in milliseconds that a connection request is queued.</td>
</tr>
<tr>
<td>ConnectionString</td>
<td>Specify the protocol, host name, and port number for connecting to the BRM software. For example: ip server1 12006. Note: The protocol must be set to ip.</td>
</tr>
<tr>
<td>InputValidation</td>
<td>Specifies whether to validate the input XMLRecord:</td>
</tr>
<tr>
<td></td>
<td>true: The adapter validates the input XMLRecord against the opcode schema.</td>
</tr>
<tr>
<td></td>
<td>false: The adapter does not validate the input XMLRecord.</td>
</tr>
<tr>
<td></td>
<td>The default is false.</td>
</tr>
<tr>
<td></td>
<td>This overrides any other validation parameter specified in the WSDL file. For information about setting the validation parameters in the WSDL file, see &quot;Defining a Web Service&quot;.</td>
</tr>
<tr>
<td>InteractionTimeZone</td>
<td>Specifies the time zone of the date fields in which the JCA Resource Adapter sends the date fields to the AIA components. The valid values are UTC and LOCAL. The default is UTC.</td>
</tr>
<tr>
<td></td>
<td>For AIA 2.5 and later systems, do not change the default value. For AIA 2.4 and earlier systems, set the value to LOCAL.</td>
</tr>
</tbody>
</table>
3. Open the JCA Resource Adapter oc4j-ra.xml file from the Oracle_home/j2ee/Instance/application-deployments/default/BRMJCAAdapter directory.

4. Make sure the following entries are set as shown below:

```
<connection-pooling use="private">
  <property name="waitTimeout" value="300" />
```

### Table 3–3 (Cont.) Connection Factory Field Values

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LoginType</td>
<td>Specifies the authentication method:</td>
</tr>
<tr>
<td></td>
<td>■ 1: The adapter logs in to BRM by using the specified login name and password.</td>
</tr>
<tr>
<td></td>
<td>■ 0: The adapter logs in to BRM by using the specified service type and POID ID.</td>
</tr>
<tr>
<td></td>
<td>The default is 1.</td>
</tr>
<tr>
<td>MaxRequestListSize</td>
<td>Specifies the maximum number of connection requests the Connection Pool can queue before returning a NAP_CONNECT_FAILED error.</td>
</tr>
<tr>
<td></td>
<td>Typically, the requests are queued when all the connections in the Connection Pool are occupied.</td>
</tr>
<tr>
<td>MultiDB</td>
<td>Specifies whether to enable connections to multiple databases:</td>
</tr>
<tr>
<td></td>
<td>■ true: Enables connections to multiple databases.</td>
</tr>
<tr>
<td></td>
<td>■ false: Disables connections to multiple databases.</td>
</tr>
<tr>
<td>OutputValidation</td>
<td>Specifies whether to validate the output XMLRecord:</td>
</tr>
<tr>
<td></td>
<td>■ true: The adapter validates the output XMLRecord against the opcode schema.</td>
</tr>
<tr>
<td></td>
<td>■ false: The adapter does not validate the output XMLRecord.</td>
</tr>
<tr>
<td></td>
<td>The default is false.</td>
</tr>
<tr>
<td></td>
<td>This overrides any other validation parameter specified in the WSDL file.</td>
</tr>
<tr>
<td></td>
<td>For information about setting the validation parameters in the WSDL file, see “Defining a Web Service”.</td>
</tr>
<tr>
<td>Password</td>
<td>Specify the password the adapter uses for logging in to the BRM software.</td>
</tr>
<tr>
<td></td>
<td>This entry is required only if LoginType is set to 1.</td>
</tr>
<tr>
<td></td>
<td>Note: The default password is encrypted on the screen.</td>
</tr>
<tr>
<td></td>
<td>However, when you type a new password, it is not encrypted.</td>
</tr>
<tr>
<td>PoidID</td>
<td>Specifies the POID. This entry should be set to 1.</td>
</tr>
<tr>
<td></td>
<td>If the JCA Resource Adapter is deployed under a brand, set PoidID to the account POID of the brand login.</td>
</tr>
<tr>
<td>ServiceType</td>
<td>Specifies the service the adapter uses to log in to the BRM software.</td>
</tr>
<tr>
<td></td>
<td>The default is /service/pcm_client.</td>
</tr>
<tr>
<td>UserName</td>
<td>Specifies the login name the adapter uses for logging in to the BRM software.</td>
</tr>
<tr>
<td></td>
<td>Note: This entry is required only if LoginType is set to 1.</td>
</tr>
</tbody>
</table>
<property name="scheme" value="fixed_wait" />
<property name="maxConnections" value='300' />
<property name="inactivity-timeout-check" value="periodic" />
<property name="minConnections" value='100' />
<property name="initial-capacity" value='25' />
<property name="inactivity-timeout" value='300' />
</connection-pooling>

5. Save and close the file.
You have successfully configured the adapter to connect to BRM.

Configuring the Transaction Mode in Oracle Application Server

If you are deploying JCA Resource Adapter on Oracle Application Server, set the JCA Resource Adapter transaction mode as follows:

1. Unzip the `BRM_Home/apps/brm_integrations/jca_adapter/OracleBRMJCA15Adapter.rar` file by using the following command:
   
   ```
   unzip OracleBRMJCA15Adapter.rar -d OracleBRMJCA15Adapter
   ```

2. In the META-INF directory, open the `ra.xml` file.

   Note: If you are deploying the JCA Resource Adapter on a WebLogic server, configure the transaction mode in the `weblogic-ra.xml` file instead of the `ra.xml` file.

3. In the `config-property` section, replace the value in the `<config-property-value>` element with the appropriate transaction mode:

   ```
   <config-property>
   <config-property-name>transactionMode</config-property-name>
   <config-property-type>java.lang.String</config-property-type>
   <config-property-value>TransactionMode</config-property-value>
   </config-property>
   ```

   where `TransactionMode` is:

   - NO_TRANSACTION for no transaction mode
   - LOCAL_TRANSACTION for local transaction mode

4. In the `<transaction-support>` element, replace the value with the appropriate transaction supported value:

   ```
   <transaction-support>TransactionSupported</transaction-support>
   ```

   where `TransactionSupported` is:

   - NoTransaction for no transaction
   - LocalTransaction for local transaction

   Note: `TransactionMode` in the `<config-property-value>` element and `TransactionSupported` in the `<transaction-support>` element must match. If they do not match, the JCA Resource Adapter generates a run-time error.
5. Save and close the file.

6. Repackage the META-INF directory in the OracleBRMJCA15Adapter.rar file by using the following command:

   ```
jar -cvf ../AdapterName.rar .
   
   where AdapterName is the name of the JCA Resource Adapter.
   

Configuring Java Logging on Oracle Application Server

The JCA Resource Adapter supports Java Unified Logging (JUL), which allows the adapter to use the Oracle Application Server logging library. You specify how the adapter generates log files by editing the OC4J logging configuration file (j2ee-logging.xml).

To specify how the adapter generates log files:

1. Create the following new directory: Oracle_home/j2ee/Instance/application-deployments/default/BRMAdapter/logs.

2. Open the Oracle_home/j2ee/Instance/config/j2ee-logging.xml file in an XML editor.

3. Add following lines under the <log_handlers> element:

   ```xml
   <log_handlers>
   <log_handler name="brm-handler" class="oracle.core.ojdl.logging.ODLHandlerFactory">
   <property name="path" value="Oracle_home/j2ee/oc4j-soa/application-deployments/default/BRMAdapter/logs/BRMAdapterLog.log"/>
   <property name="maxFileSize" value="10485760"/>
   <property name="maxLogSize" value="104857600"/>
   <property name="encoding" value="UTF-8"/>
   <property name="format" value="ODL-TEXT"/>
   </log_handler>
   </log_handlers>
   
   where:
   - Path specifies the directory in which the adapter creates the log file.
   - maxFileSize specifies the maximum size, in bytes, of the log file. When a log file reaches this limit, a new log file is generated.
   - maxLogSize specifies the maximum size, in bytes, of the logging directory.
   - encoding must be set to UTF-8.
   - format must be set to ODL-TEXT. This enables you to view the log files directly in the Oracle Application Server control console.

4. Add the following lines under the <loggers> element:

   ```xml
   <loggers>
   <logger name="oracle.tip.adapter.brm" level="LogLevel" useParentHandlers="false">
   <handler name="oc4j-handler"/>
   <handler name="brm-handler"/>
   </logger>
   
   where LogLevel is set to one of the following:
   - SEVERE to log problems that require attention from the system administrator.
WARNING to log actions that should be reviewed and may require action before an error occurs.

INFO to log normal actions or events. This includes user operations, such as user log ins, and automatic operations, such as log file rotations.

CONFIG to log configuration-related messages.

FINE to log trace or debug messages for performance monitoring.

FINER to log highly detailed trace or debug message.

FINEST to log the most detailed trace or debug messages.

5. Save and close the file.
6. Restart the application server or the J2EE instance.

Changing the Java Logging Level
To change the logging level by using the Oracle Application Server ASC:

1. On the OC4J home page, click Administration.
2. From the administration tasks, select Logger Configuration.
   The Logger Configuration page appears.
3. Click Expand All to view the entire list of loggers currently loaded for the OC4J instance.
4. Select NULL for oracle.tip.adapter.brm.
5. Click Apply.

Deploying and Connecting the BRM JCA Adapter on Oracle WebLogic Server
To deploy and configure the adapter on Oracle WebLogic Server, perform these tasks:

- Deploying the BRM JCA Resource Adapter on Oracle WebLogic Server
- Connecting the Adapter to BRM in Oracle WebLogic Server
- Configuring Java Logging in Oracle WebLogic Server

Deploying the BRM JCA Resource Adapter on Oracle WebLogic Server
The adapter is dependent on JAR files to deploy properly. Table 3–4 lists the JAR files that the adapter requires from each application in your system.

<table>
<thead>
<tr>
<th>Application</th>
<th>JAR Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>J2EE application server</td>
<td>classes12.jar, connector15.jar, and jta.jar</td>
</tr>
<tr>
<td>Oracle BPEL process</td>
<td>bpm-infra.jar, orabpel-thirdparty.jar, orabpel.jar, and xmlparsev2.jar</td>
</tr>
<tr>
<td>Apache</td>
<td>xercesImpl.jar</td>
</tr>
</tbody>
</table>

If you are deploying the adapter in a standalone WebLogic Server instance, make sure these JAR files are available to the class loader that is loading the adapter.

To deploy the adapter on WebLogic Server:
1. Start the WebLogic Server domain, if it is not already started.
2. Log in to the Oracle WebLogic Server Administration Console. The default is: 
   http://localhost:8001/console
3. Click Lock and Edit.
4. In the Domain Structure tree, click Deployments.
   The Summary of Deployments pane appears.
5. Click Install.
   The Install Application Assistant pane appears.
6. Browse to the deploy directory of the ResourceAdapterDeployment project, select
   the OracleBRMJCAAdapter.rar file, and click Next.
7. Select Install this deployment as an application and click Next.
8. In the server list, select the required server, and click Next.
9. Accept all other defaults on the remaining screens.
10. Click Finish.
11. Click Activate Changes, which deploys the resource adapter.
Check the state of the OracleBRMJCAAdapter application in the WebLogic Server
Administration Console. Start the application if the state is set to Prepared.

Connecting the Adapter to BRM in Oracle WebLogic Server

You connect the adapter to the BRM software by creating connection pools and
connection factories. As part of the adapter deployment, the application server creates
a weblogic-ra.xml file from the packaged ra.xml file.

To configure the JCA Resource Adapter connection factory and connection pools:
1. Use the resource adapter home page from the WebLogic Server Administration
Console to create connection pools and connection factories.
   For each connection factory, specify the following along with the connection
   factory field values listed in Table 3–5:
   ■ The JNDI location for the connection factory.
   ■ The connection pool to use.
   ■ How to connect to BRM by using these entries:

   Note: Connection factory field values must be lower case.

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectionString</td>
<td>Specify the protocol, host name, and port number for connecting to the BRM software. For example: ip server1 12006.</td>
</tr>
<tr>
<td></td>
<td>Note: The protocol must be set to ip.</td>
</tr>
</tbody>
</table>

Table 3–5 Connection Factory Field Values
Table 3–5 (Cont.) Connection Factory Field Values

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
</table>
| InputValidation      | Specifies whether to validate the input XMLRecord:  
  ■ true: The adapter validates the input XMLRecord against the opcode schema.  
  ■ false: The adapter does not validate the input XMLRecord.  
  The default is false.  
  This overrides any other validation parameter specified in the WSDL file.  
  For information about setting the validation parameters in the WSDL file, see “Defining a Web Service”. |
| OutputValidation     | Specifies whether to validate the output XMLRecord:  
  ■ true: The adapter validates the output XMLRecord against the opcode schema.  
  ■ false: The adapter does not validate the output XMLRecord.  
  The default is false.  
  This overrides any other validation parameter specified in the WSDL file.  
  For information about setting the validation parameters in the WSDL file, see “Defining a Web Service”. |
| LoginType            | Specifies the authentication method:  
  ■ 1: The adapter logs in to BRM by using the specified login name and password.  
  ■ 0: The adapter logs in to BRM by using the specified service type and POID ID.  
  The default is 1. |
| UserName             | Specifies the login name the adapter uses for logging in to the BRM software.  
  Note: This entry is required only if LoginType is set to 1. |
| Password             | Specify the password the adapter uses for logging in to the BRM software.  
  Note: This entry is required only if LoginType is set to 1. |
| PoidID               | Specifies the POID. This entry should be set to 1.  
  If the JCA Resource Adapter is deployed under a brand, set PoidID to the account POID of the brand log in. |
| ServiceType          | Specifies the service the adapter uses to log in to the BRM software.  
  The default is /service/pcm_client. |
| BRMConnection PoolMaxSize | Specifies the maximum number of connections the Connection Pool can create. |
| BRMConnection PoolMinSize | Specifies the minimum number of connections created by the Connection Pool when the pool is initialized. |
| BRMConnection PoolTimeout | Specifies the maximum amount of time in milliseconds that a connection request is queued. |
| BRMConnection PoolMaxIdleTime | Specifies the maximum amount of time in milliseconds that a free connection in the Connection Pool can be idle.  
  If a free connection is idle for a time greater than or equal to BRMConnectionPoolMaxIdleTime,  
  the connection is removed from the pool.  
  Note: Only the connections created after the limit set by BRMConnectionPoolMinSize are verified for the idle time; if the condition is met, the connection is removed from the pool. |
Save and close the file.

You have successfully configured the adapter to connect to BRM.

**Configuring Java Logging in Oracle WebLogic Server**

The JCA Resource Adapter supports Java Unified Logging (JUL), which allows the adapter to use the Java Unified Logging library.

**Creating a Startup Class**

To create a Startup class:

1. Copy the `weblogic_start.jar` file provided with the installation package to `Domain_Home/lib`.

2. Log in to the WebLogic Server Administration Console. The default is:
   
   `http://localhost:8001/console`

3. Click **Lock and Edit**.

4. In the **Domain Structure** tree, expand **Environment** and then click **Startup and Shutdown classes**.

   The Startup and Shutdown Classes pane appears.

5. Click **New**.

   The Configure a New Startup or Shutdown Class: Class Type pane appears.

6. Select **Startup Class** and click **Next**.

   The Configure a New Startup or Shutdown Class: Startup Class Properties pane appears.

7. In the **Name** field, enter `BRMStartupClass`.

8. In the **Class Name** field, enter `oracle.tip.adapter.brm.BRMLoggerStartUP`.

---

**Table 3–5 (Cont.) Connection Factory Field Values**

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InteractionTimeZone</td>
<td>Specifies the time zone of the date fields in which the JCA Resource Adapter sends the date fields to the AIA components. The valid values are UTC and LOCAL. The default is UTC. For AIA 2.5 and later systems, do not change the default value. For AIA 2.4 and earlier systems, set the value to LOCAL.</td>
</tr>
<tr>
<td>MaxRequestListSize</td>
<td>Specifies the maximum number of connection requests the Connection Pool can queue before returning a NAP_CONNECT_FAILED error. Typically, the requests are queued when all the connections in the Connection Pool are occupied.</td>
</tr>
<tr>
<td>MultiDB</td>
<td>Specifies whether to enable connections to multiple databases: ■ true: Enables connections to multiple databases. ■ false: Disables connections to multiple databases. When the MultiDB connection factory entry is set to true, you do not need to supply the target database number; the adapter automatically opens transactions on the correct database and generates an error if a transaction attempts to manipulate data across databases</td>
</tr>
<tr>
<td>AverageOpcodeCount</td>
<td>Specifies the average number of opcode calls per thread.</td>
</tr>
</tbody>
</table>
9. Click Next.
   The Configure a New Startup or Shutdown Class: Select Targets pane appears.
10. From the Servers list, select the server on which to deploy the class and click Finish.
   The Startup and Shutdown Classes pane appears.
11. Click BRMStartupClass.
   The Settings for BRMStartupClass pane appears.
12. Select Run Before Application Deployments and Run Before Application Activations and click Save.
13. Click Activate Changes.
14. Restart WebLogic Server to apply the changes.
   By default, log files will be created in the Domain_Home/logs/BRMAdapterLogs directory.

Changing the Java Logging Level in Oracle WebLogic Server
You change the logging level by using JConsole.

   1. Go to the WebLogic_home/jdk160_05/bin directory and enter the following command:
      
      jconsole

      The New Connection dialog box appears as shown in Figure 3–1.
2. Select **Remote Process**, enter the WebLogic Server host name and port number, enter your user name and password, and then click **Connect**.

**Note:** When WebLogic Server is running on the same system, you can use **Local Process** without authentication. The **Local Process** list shows the WebLogic Server process name and PID.

The Java Monitoring and Management Console pane appears.

3. Click the **MBeans** tab.

4. In the MBean tree, expand **java.util.logging**, then expand **Attributes**, and then select **LoggerNames**.

5. Copy the **oracle.tip.adapter.brn.BRMConnectionFactory** line.

6. In the MBean tree, expand **java.util.logging** and then select **Operations**.

   The Operation invocation pane appears.

7. In the **void setLoggerLevel p0** and **p1** fields, paste the logger name and log level that you copied in step 4.

8. Click **setLoggerLevel**, which updates the log level.

Deploying and Connecting the BRM JCA Adapter on Oracle GlassFish Server

To deploy and configure the JCA adapter on Oracle GlassFish Server, perform these tasks:

- Configuring the GlassFish Server
- Deploying and Configuring JCA Resource Adapter on the GlassFish Server

Configuring the GlassFish Server

To configure the GlassFish Server:

1. Set the PATH environment variable to Glassfish_Home/bin, where Glassfish_Home is the directory in which the GlassFish software is installed.

2. Copy the orabpel.jar file located in the BRM_Home/apps/brm_integrations/jca_adapter/dependentjars directory to the Glassfish_Home/glassfish/domains/Domain_Name/lib directory, where Domain_Name is the name of the home directory for the domain you are configuring.

3. Copy the xmlparserv2.jar file located in the BRM_Home/apps/brm_integrations/jca_adapter/dependentjars directory to the Glassfish_Home/glassfish/domains/Domain_Name/lib/applibs directory, where Domain_Name is the name of the home directory for the domain you are configuring.

Deploying and Configuring JCA Resource Adapter on the GlassFish Server

To deploy and configure JCA Resource Adapter on the GlassFish Server:

1. Start the GlassFish Server domain, if it is not already started.

2. Run the following command:

   asadmin deploy --libraries xmlparserv2.jar BRM_Adapter_Directory/OracleBRMJCA15Adapter.rar

   where BRM_Adapter_Directory is the path to the directory where the OracleBRMJCA15Adapter.rar file is located.

3. Start the GlassFish Server Administration Console.

4. Create a connector connection pool for JCA Resource Adapter. For more information on creating a connector connection pool, see the Oracle GlassFish Server Help.

5. Create a Java Naming and Directory Interface (JNDI) connector resource for JCA Resource Adapter. For more information on creating a JNDI connector resource, see the Oracle GlassFish Server Help.

6. Set the visibility of the JCA Resource Adapter classes to global by running the following command:

   asadmin set server.connector-service.class-loading-policy="global"

For more information about JConsole, access the Java documentation at: http://download.oracle.com/javase/6/docs/technotes/guides/management/jconsole.html.
7. Set the log level for JCA Resource Adapter to **FINEST** by running the following command:

   asadmin set-log-levels oracle.tip.adapter.brm=FINEST

---

**Note:** JCA Resource Adapter logs are stored in the Glassfish/Home/glassfish/domains/Domain_Name/logs/server.log file.

---

### Setting Up JCA Resource Adapter to Support Custom Opcodes

You can enable JCA Resource Adapter to support custom opcodes.

To enable JCA Resource Adapter to support custom opcodes:

1. Run the `parse_custom_ops_fields` Perl script, which creates the `CustomOp.java` class in the Java default package.
   
   For more information about the script, see the BRM documentation.

2. Verify that the `CustomOp.java` file contains the following:
   
   - The opcode-name-to-opcode-number mapping for all the custom opcodes in the class
   
   **Important:** Verify that the mapping includes the full name of each opcode. If any opcode name is truncated, replace the truncated name with the full name.

   - The `opToString` method, which converts opcode numbers to opcode names
   - The `stringToOp` method, which converts opcode names to opcode numbers

   For example:

   ```java
   public class CustomOp {

   public static final int Custom_Opcode_Name = Custom_Opcode_Number;

   public static String opToString( int op ) {
       -------
   }
   .
   public static int stringToOp( String op ) {
       -------
   }
   .
   .
   }
   ```

3. Create a JAR file that includes the `CustomOp.java` class.

4. Make the JAR file available to JCA Resource Adapter at run time by doing one of the following:
   
   - If you have not deployed the adapter, put the JAR file in `BRM/Home/apps/brm_integrations/jca_adapter/OracleBRMJCA15Adapter.rar`

   The JCA Resource Adapter package will then contain the custom opcode JAR file created in step 3 along with other JAR files such as `pcm.jar`, `pcmext.jar` and `OracleBRMJCA15Adapter.jar`. 
If you have deployed the adapter, put the JAR file in the adapter’s working directory on the application server.

For Oracle application servers, the working directory is Oracle_Home/j2ee/Instance/connectors/AdapterDeploymentName/AdapterDeployment Name.

**Using the BRMAAdapterServletClient Application**

Use the BRMAAdapterServletClient application to test the JCA Resource Adapter’s configuration and connectivity to BRM. The application allows you to provide input to an opcode and view its output.

To set up the BRMAAdapterServletClient application, you must deploy and configure it on your Oracle Application Server system.

---

**Note:** Make sure the BRM JCA Resource Adapter is deployed first, because BRMAAdapterServletClient uses classes from the adapter.

---

Deploy the BRMAAdapterServletClient application on Oracle Application Server by using either Oracle Application Server ASC or the Oracle admintool.jar file. When deploying the application, make sure you perform the following:

- Copy the BRMAAdapterServletClient.war file from the BRM_Home/apps/brm_integrations/jca_adapter/BRMAAdapterServletClient directory to a location that is accessible to the adapter deployment tool.
- Specify to deploy BRMAAdapterServletClient in the same instance as that of the BRM JCA Resource Adapter.
- Upload the BRMAAdapterServletClient.war archive file.
- Specify that BRMAAdapterServletClient is dependent on the oracle.bpel.common shared library.

After the application is deployed, you can begin using it to run and test the adapter. You can access the application’s default page at the following URL:

http://HostName:PortNumber/BRMAAdapterServletClient

where HostName is the host name of the machine where the application is deployed and PortNumber is the port where the application is deployed.
This chapter provides reference information for Oracle Communications Billing and Revenue Management (BRM) system administration utilities.
**pin_dd_to_schema**

Use the **pin_dd_to_schema** utility to generate the XSD schema for your storable classes and subclasses.

For more information, see "Generating the Schema for Your Storable Classes and Subclasses".

---

**Note:** To connect to the Oracle Communications Billing and Revenue Management (BRM) database, the **pin_dd_to_schema** 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_dd_to_schema  -e ConfigFile | -r StorableClass | -h
```

**Parameters**

- **-e ConfigFile**
  Generates the schema for all storable classes listed in the configuration file as well as their subclasses. For example, if the configuration file lists the `/service/telco/gsm` class, the utility generates the schema for `/service/telco/gsm`, `/service/telco/gsm/telephony`, `/service/telco/gsm/sms`, and any other `/service/telco/gsm` subclasses in your system.

  **Note:** You must specify both the name and path of the configuration file.

- **-r StorableClass**
  Generates the schema for the specified storable class and all of its subclasses. For example:

  ```
  pin_dd_to_schema -r /service/telco/gsm
  -h
  ```

  Displays the syntax and parameters for this utility.

**Results**

The **pin_dd_to_schema** utility generates the output files in the directory from which it was run.

The utility notifies you only if it encounters errors. For errors, look in the `default.pinlog` file, which is created in the directory from which the utility was run.

**Note:** If you receive the following error when running **pin_dd_to_schema** on Linux systems, set the LC_ALL environment variable to `C` (```
(setenv LC_ALL C)
``` and then rerun the utility.
Malformed UTF-8 character (unexpected continuation byte 0xac, with no preceding start byte) in bitwise and (&) at BRM_Home/lib/5.8.0/Switch.pm line 251. Malformed UTF-8 character (unexpected continuation byte 0xab, with no preceding start byte) in bitwise and (&).
pin_opspec_to_schema

Use the `pin_opspec_to_schema` utility to generate XSD schema files for opcodes. For more information, see "Generating the Schema for Your Opcodes".

**Location**

`BRM_Home/bin`

**Syntax**

```
pin_opspec_to_schema -i InputFile [-o OutputFile] [-h]
```

**Parameters**

- `-i InputFile`
  Specifies the name and location of the opcode XML flist specification file to convert into XSD schema. If you don’t specify the absolute path to the file, the utility looks in the current directory.

- `-o OutputFile`
  Creates the XSD schema output file using the name and location you specify. By default, the utility generates an output file named `opcodename.xsd` in the directory from which you run the utility.

- `-h`
  Displays the syntax and parameters for this utility.

**Results**

The `pin_opspec_to_schema` utility notifies you when it successfully generates schema files. Any errors are displayed on the console.
pin_wsdl_generator

Use the pin_wsdl_generator utility to generate WSDL files for Web services.
For more information, see "Generating WSDL Files for Web Services".

Location

BRM_Home/bin

Syntax


Parameters

-c ConfigFile
Specifies the name and location of the XML configuration file that describes how to
group opcodes into Web services. By default, the utility uses the BRM_Home/apps/brm_integrations/config/pin_wsdl_generator.xml file. See "Defining a Web Service".

-v SchemaFile
Validates the pin_wsdl_generator.xml configuration file against the specified schema file.

-j | -s
Specifies whether to create WSDL files with JCA (-j) or SOAP (-s) bindings.
The default is to generate WSDL files with JCA bindings.

-d
Runs in debug mode and displays more detailed messages.

-h
Displays the syntax and parameters for this utility.

Results

The pin_wsdl_generator utility generates the output WSDL files in the directory from
which it was run.
The utility notifies you when it successfully generates WSDL files. Any errors are
displayed on the console.