Oracle® Communications
Billing and Revenue Management
Web Services Manager
Release 7.4
E25107-02

November 2014
## Contents

**Preface**........................................................................................................................................................................ v

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

### 1 Installing and Configuring Web Services Manager

- **System Requirements** ........................................................................................................................................ 1-1
- **Installing Web Services Manager** ........................................................................................................ 1-1
- **Connecting Web Services Manager to the BRM System** ........................................................................ 1-3
- **About Deploying Web Services Manager** ........................................................................................... 1-5
  - Deploying the Web Service ......................................................................................................................... 1-5
  - Setting Up Web Services Manager to Support Custom Opcodes ....................................................... 1-5
  - Creating a Custom Web Service ............................................................................................................... 1-7
    - Creating Opcode Specification Schema Files ..................................................................................... 1-9
    - Specifying the XSL Rules to Create the Opcode Schema .................................................................. 1-10
  - Using Web Services Programs .................................................................................................................. 1-10
- **Uninstalling Web Services Manager** ........................................................................................................ 1-10
- **About Validating Input and Output XML Data** ....................................................................................... 1-10
- **About BRM Web Services and WSDL Files** ........................................................................................... 1-11
- **Configuring Java Logging in Oracle WebLogic Server** ......................................................................... 1-14
  - Specifying the JUL Mechanism .................................................................................................................. 1-14
  - Creating a Startup Class ............................................................................................................................. 1-14
  - Changing the Java Logging Level in Oracle WebLogic Server ............................................................ 1-15
This book contains guidelines for installing and setting up Oracle Communications Billing and Revenue Management (BRM) Web Services Manager. The sample procedures use WebLogic Server, but you can apply the concepts to any standards-compliant, vendor-supported application server that supports Web services.

Before reading this book, you should be familiar with implementing Web services using WebLogic Server. See your WebLogic Server documentation for more information.

Audience

This book is intended for systems integrators, system administrators, database administrators, and other individuals who are responsible for installing and configuring Web services for BRM.

Downloading Oracle Communications Documentation

Product documentation is located on Oracle Technology Network:

http://docs.oracle.com

Additional Oracle Communications documentation is available from Oracle software delivery Web site:

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 guide:
<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E25107-01</td>
<td>March 2013</td>
<td>Initial release.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Added documentation about customizing Web services:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Connecting Web Services Manager to the BRM System</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Creating a Custom Web Service</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Updated the &quot;Using Web Services Programs&quot; section.</td>
</tr>
</tbody>
</table>
Installing and Configuring Web Services Manager

This chapter contains information on the requirements for installing and setting up Oracle Communications Billing and Revenue Management (BRM) Web Services Manager.

Before reading this chapter, you should be familiar with implementing Web services using Oracle WebLogic Server. See your Oracle WebLogic Server documentation for more information.

System Requirements

Web Services Manager is available for the HP-UX IA64, Oracle Solaris, Oracle Linux, and AIX operating systems.

Before you install and configure Web Services Manager, you must install the third-party software, which includes the Perl libraries and Java Runtime Environment (JRE) required for installing BRM components. See "Installing the Third-Party Software" in BRM Installation Guide.

Installing Web Services Manager

**Note:** If you have already installed the product, features that are already installed cannot be reinstalled without uninstalling them first. To reinstall a feature, uninstall it and then install it again.

The Web Services Manager package includes Web Services Manager, Synchronization Queue Data Manager, and JCA Resource Adapter. You must download the Web Services Manager package and extract these three components before you can install Web Services Manager. For more information about Synchronization Queue Data Manager and JCA Resource Adapter, see:

- "Understanding the Synchronization Queue Data Manager" in BRM Synchronization Queue Manager.
- "Connecting J2EE-Compliant Applications to BRM" in BRM JCA Resource Adapter.

To install Web Services Manager:

1. Go to the Oracle software delivery Web site:

   https://edelivery.oracle.com
and download the appropriate software pack to a temporary directory (temp_dir):

- **7.4_WebServicesMgr_platform_32_opt.tar.Z**
  where *platform* is the operating system name.

---

**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

where *platform* is the operating system name.

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

---

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

---

Bash shell:

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

C shell:

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

4. Go to temp_dir and enter the following command:

```
7.4_WebServicesMgr_platform_32_opt.bin
```

where *platform* is the operating system name.

---

**Note:** You can use the -console parameter to run the installation in command-line mode. To enable a graphical user interface (GUI) installation, install a GUI application such as X Windows and set the DISPLAY environment variable before you install the software.

---

5. Follow the instructions displayed during installation. The default installation directory for Web Services Manager is BRM_Home.
Note: The installation program does not prompt you for the installation directory if BRM or Web Services Manager is already installed on the machine and automatically installs the package in BRM_Home.

Your Web Services Manager installation is now complete.

Connecting Web Services Manager to the BRM System

After you install Web Services Manager, configure connection settings for Web Services Manager to connect to the BRM system.

To connect Web Services Manager to the BRM system:

1. Do one of the following:
   - If you customized Web services:
     - Extract the infranetwebsvc.war file to a local directory (local_dir) on the machine on which you installed your application server.
     - Copy the BRM_Home/deploy/web_services/Infranet.properties file to the local_dir/WEB-INF/classes directory, where local_dir is a directory on the machine on which you installed your application server.
   - If you did not customize Web services, copy the BRM_Home/deploy/web_services/Infranet.properties file to either of the following directories on the machine on which you installed your application server:

     Note: If you want to copy the Infranet.properties file to the local_dir/WEB-INF/classes directory, extract the infranetwebsvc.war file to a local directory (local_dir) on the machine on which you installed your application server.

     - local_dir/WEB-INF/classes directory
     - home directory on the machine on which you installed your application server

2. Open the Infranet.properties file that you copied to the machine on which you installed your application server.

3. Locate the following lines and update the connection parameters:

   infranet.connection=pcp://root.0.0.0.1:password@ipAddress:port/0.0.0.1/service/admin_client 1
   infranet.login.type=1

   where:
   - password is the BRM server password
   - ipAddress is the IP address of the application server on which BRM is installed
   - port is the port number of the application server on which BRM is installed

4. Locate the following lines and update the parameters:

   - (Optional) Minimum number of connections:
Connecting Web Services Manager to the BRM System

```ini
infranet.connectionpool.minsize=minsize
```

where `minsize` is the minimum number of connections allowed in the pool. The default number is 1.

- (Optional) Maximum number of connections:
  ```ini
  infranet.connectionpool.maxsize=maxsize
  ```

  where `maxsize` is the maximum number of connections allowed in the pool.

- (Optional) Connection timeout:
  ```ini
  infranet.connectionpool.timeout=timeout
  ```

  where `timeout` is the connection pool timeout in milliseconds.

- (Optional) Logging for Web Services Manager:
  ```ini
  webservices.log.enabled=value
  ```

  where `value` is:
  - `true` to enable logging. This option saves and displays the log files as standard output in the WebLogic Server console.
  - `false` to disable logging. This option saves the log files in the `/domain/logs/BRMWebSvcMgr.log` file. Configure the `BRM_Home/deploy/web_services/lib/weblogic_ws_startup.jar` file to use this option. For more information, see "Configuring Java Logging in Oracle WebLogic Server".

5. If you added custom opcodes or custom fields for Web services, add the enum values of the custom fields.

For example, if you created the `custom_fld_usage_id` custom field and the enum value for the `custom_fld_usage_id` field is 10001, add the following entry:

```ini
infranet.custom.field.10001=custom_fld_usage_id
```

The following examples show other custom fields with enum values that you can add to the `Infranet.properties` file:

- Package name:
  ```ini
  infranet.custom.field.package=packagename
  ```

  where `packagename` is name of the custom package that you created.

- Custom field usage type ID:
  ```ini
  infranet.custom.field.10000=custom_fld_usage_type_id
  ```

  where `custom_fld_usage_type_id` is the usage type ID of the custom field.

- Custom field user ID:
  ```ini
  infranet.custom.field.10001=custom_fld_user_id
  ```

  where `custom_fld_user_id` is the user ID of the custom field.

6. Save and close the file.
About Deploying Web Services Manager

You can deploy Web Services Manager on any standards-compliant, vendor-supported application server that supports Java servlets. The package is shipped in a WAR (Web archive) file.

Web Services Manager:
- Includes a servlet-based implementation that hosts the deployed Web service.
- Uses the Apache AXIS library to support SOAP, WSDL, and other low-level Web service protocols.

Deploying the Web Service

You can deploy the Web service on WebLogic Server through the WebLogic Server Administration Console.

To deploy the Web service:
1. Extract the BRM_Home/deploy/web_services/Infranetwebsvc.war file to a local directory (local_dir), where BRM_Home is the directory in which you installed the BRM software.
2. Copy any CustomFields.jar files to the local_dir/WEB-INF/lib directory.
   - Note: Make sure the JRE version that was used to generate CustomFields.jar is the same or lower than the version of the WebLogic Server JRE.
3. Copy the Infranet.properties file to the local_dir/WEB-INF/classes directory.
4. Deploy the application as a directory on WebLogic Server or convert the directory into a WAR file and deploy it.

Setting Up Web Services Manager to Support Custom Opcodes

To expose custom opcodes as a Web service, enable Web Services Manager to support custom opcodes. For more information on custom opcodes, see BRM Developer’s Guide.

To enable Web Services Manager to support custom opcodes:
1. Do one of the following:
   - Create the CustomOp.java file by running the following command:
     ```bash
     parse_custom_ops_fields -L pcmjava -I input -O output -P java_package
     ```
     where:
     - `input` is the header file you create for your custom opcodes and fields.
About Deploying Web Services Manager

1. **output** is the memory-mapped file or directory for the output of the script. **output** must be a directory having some correspondence with the Java package. For example, if `java_package` is in `com.portal.classFiles`, **output** must be `f:/mysource/com/portal/classFiles`.

   **java_package** is the Java package in which to put the generated classes.

   For more information, see the discussion about the **parse_custom_ops_fields** utility in BRM Developer’s Guide.


3. Verify that the **CustomOp.java** file contains the following:
   - The opcode-name-to-opcode-number mapping for all the custom opcodes in the file
   - **opToString** method, which converts opcode numbers to opcode names
   - **stringToOp** method, which converts opcode names to opcode numbers

   The following is a sample **CustomOp.java** file:

   ```java
   public class CustomOp {
       public static final int MY_CUSTOM_OPCODE = 100001;
       public static String opToString( int op ) {
           ...
       }
       public static int stringToOp( String op ) {
           ...
       }
   }
   ```

4. Compile the **CustomOp.java** file into the **CustomOp.class** file by running the following command:

   ```sh
   javac -d . path/CustomOp.java
   ```

   where `path` is the path to the **CustomOp.java** file.

   For example:

   ```sh
   javac -d . com/portal/classFiles/CustomOp.java
   ```

5. Package the **CustomOp.class** file into the **CustomFields.jar** file by running the following command:

   ```sh
   Note: Make sure the JRE version that was used to generate the CustomFields.jar file is the same or lower than the version of the application server JRE.
   ```

   ```sh
   jar cvf CustomFields.jar path.CustomOp.class
   ```
where path is the path to the CustomOp.class file.

For example:

```
jar cvf CustomFields.jar com.portal.classFiles.CustomOp.class
```

5. Make the CustomFields.jar file available to Web Services Manager by doing one of the following:

- If you have not deployed Web Services Manager, do the following:
  
  a. Extract the BRM_Home/deploy/web_services/Infranetwebsvc.war file to a local directory (local_dir).

  b. Copy the path/CustomFields.jar file to the local_dir/WEB-INF/lib directory, where path is the path to the CustomFields.jar file (for example, com/portal/classFiles).

  c. Open the BRM_Home/deploy/web_services/Infranet.properties file in a text editor.

  d. Add or modify the following entry:

  ```
infranet.custom.field.package = path
  ```

  where path is the path to the CustomOp.java file; for example, com.portal.classFiles.

  e. Save and close the file.

  f. Copy the BRM_Home/deploy/web_services/Infranet.properties file to the local_dir/WEB-INF/classes directory.

  g. Deploy the application as a directory on the application server or convert the directory into a WAR file and deploy it.

- If you have deployed Web Services Manager, do the following:

  a. Copy the path/CustomFields.jar file to the Webservices_deployment_dir/WEB-INF/lib directory.

  where path is the path to the CustomFields.jar file (for example, com/portal/classFiles) and Webservices_deployment_dir is the directory in which you have deployed the Web services on the application server.

  b. Open the Webservices_deployment_dir/WEB-INF/classes/Infranet.properties file in a text editor.

  c. Add or modify the following entry:

  ```
infranet.custom.field.package = path
  ```

  where path is the path to the CustomOp.java file; for example, com.portal.classFiles.

  d. Save and close the file.

  e. Deploy the application as a directory on the application server or convert the directory into a WAR file and deploy it.

### Creating a Custom Web Service

You can extend Web Services Manager to support custom Web services. Before you create a custom Web service or customize an existing Web service in Web Services
Manager, implement your custom opcodes in the BRM system. For more information, See "Creating Custom Fields and Storable Classes" in BRM Developer’s Guide.

To create a custom Web service:

1. If you created an opcode with custom fields for your custom Web service, configure BRM to recognize the custom fields. See "Creating Custom Fields and Storable Classes" in BRM Developer’s Guide.

2. Create a WSDL file for the Web service. See "Generating WSDL Files for Web Services" in BRM JCA Resource Adapter.

   To create a WSDL file manually, see the deploy/web_services/wsd1 sample file and create the WSDL file.

3. Create the XML specifications for your custom opcodes. See "Creating Opcode Specification Schema Files".

4. Create a/service class for your custom Web service. See the BRM_Home/deploy/web_services/sample/SampleWebService.java.template sample.

   Note: When you create the /service class for your custom Web service, do the following:
   - Ensure that the names of the opcodes in the Java class match the names of the opcodes that are defined in the WSDL file.
   - Use a common name for the WebService class and the WSDL file. For example, SampleWebService.wsdl and SampleWebService.class.

5. Create the customField.jar file and include all the class files of the custom fields and the CustomOp.class file. See "Setting Up Web Services Manager to Support Custom Opcodes" for more information.

6. Add all the custom field enum constants to the Infranet.properties file. See "Connecting Web Services Manager to the BRM System" for more information.

7. Extract the infranetwebsvc.war file to local_dir, where local_dir is the directory in which you deployed Web Services Manager on your application server.

8. Extract the local_dir/WEB-INF/services/InfranetWebservices.aar file.

9. Copy the SampleWebService.class file you created in step 1 to local_dir/WEB-INF/services/com/portal/webservices/.

10. Open local_dir/WEB-INF/services/META-INF/services.xml file in a text editor.

11. Add your custom service.

   The following example shows the services.xml configuration file for Web services that take payload as an XML string:

   ```xml
   <serviceGroup>
     <service name="CustomWebServices"
      targetNamespace="http://xmlns.oracle.com/BRM/schemas/BusinessOpcodes">
       <description>CustomWebServices</description>
       <parameter name="ServiceClass">
         com.portal.webservices.SampleWebService
       </parameter>
       <schema>
         schemaNamespace="http://xmlns.oracle.com/BRM/schemas/BusinessOpcodes"/
         <messageReceivers>
   ```
<messageReceiver
  mep="http://www.w3.org/2004/08/wsd1/in-out"
  class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
</messageReceivers>

<operation name="customOpOperationName"
  mep="http://www.w3.org/2004/08/wsd1/in-out">
  <actionMapping>PcmOpSearch</actionMapping>
</operation>
</service>
</serviceGroup>

Note:
- Ensure that the value in the <parameter name="ServiceClass"> element includes the full path to the custom Web Service class file.
- Ensure that the operation name matches the name that is defined in the WSDL file and in the custom Web Service class file.

12. Go to local_dir/WEB-INF/services/ and package the InfranetWebservices.aar file by doing the following:
   a. Delete the existing InfranetWebservices.aar file.
   b. Create a new InfranetWebservices.aar file by running the following command:
      - jar -cvf InfranetWebservices.aar *

Verify that the InfranetWebservices.aar file has the following directory structure:

META-INF/          Custom WSDLs
    Custom XSDs
    wsdls.list
    services.xml
    com/custom/webservices/SampleWebService.class

13. Go to local_dir and package the infranetwebsvc.war file by doing the following:
   a. Delete the existing infranetwebsvc.war file.
   b. Create a new infranetwebsvc.war file by running the following command:
      - jar -cvf infranetwebsvc.war *

Creating Opcode Specification Schema Files
You must create opcode flist specification files for opcodes that you customize or add to the Web Services Manager. Create the specification XML files by following the BRM_Home/apps/brm_integrations/stylesheets/opspec.xsd file.

You then convert the opcode flist specification XML files into XSD schema by using the pin_opspec_to_schema utility.

To convert opcode flist specification XML files into XSD schema, go to the BRM_Home/apps/brm_integrations directory and do the following:
- For Web services that take payload as XML string, run the following command:
  pin_opspec_to_schema -i InputFile [-o OutputFile]
Uninstalling Web Services Manager

To uninstall Web Services Manager:

1. Go to the `BRM_Home/uninstaller/WebServicesMgr` directory.
2. Enter the following command:
   
   ```sh
   uninstaller.bin
   ```

About Validating Input and Output XML Data

Web Services Manager validates the input and output XML by comparing the XML fields and values against the opcode XML schema.

The opcode specifications, schemas, and WSDL files are packaged along with Web Services Manager. The package includes the `opspec.xsd` file and the `pin_opspec_to_schema` utility. Use the `opspec.xsd` file to write opcode specifications for custom opcodes that need to be exposed as a Web service. Use the `pin_opspec_to_schema` utility to generate the schema files from the opcode specification files.

To configure Web Services Manager to validate the input and output XML against the target opcode XML schema:

1. Open the `local_dir/WEB-INF/classes/Infranet.properties` directory.
2. Add the following entries to the file:

---

Where:

- **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.

- **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.

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 fields, you must customize the `pin_opspec_to_schema.xsl` style sheet to handle your custom fields.

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

Using Web Services Programs

Web Services Manager includes sample programs that demonstrate how to write code for various tasks when customizing Web services. To test your Web services implementation, write a client application that communicates with the Web service using the SOAP protocol or test Web services using SoapUI.

The following sample shows an input payload for the `pcmOpBalGetBalances` Web service:

```xml
<![CDATA[<PCM_OP_BAL_GET_BALANCES_inputFlist
xmlns:brm="http://xmlns.oracle.com/BRM/schemas/BusinessOpcodes"
xmlns="http://xmlns.oracle.com/BRM/schemas/BusinessOpcodes"><POID>0.0.0.1 /account 1 0</POID></PCM_OP_BAL_GET_BALANCES_inputFlist>]]>
```
About BRM Web Services and WSDL Files

Web Services Manager exposes the opcodes as operations through the different WSDLs in JCA Resource Adapter.

The WSDL files included with Web Services Manager define the opcodes that can be called and 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. Web Services Manager includes one WSDL file for each Web service.

Table 1–1 describes the WSDL files.

<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 includes the following opcodes:</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_AR_ACCOUNT_ADJUSTMENT</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_AR_BILL_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_ITEM_ADJUSTMENT</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_AR_EVENT_ADJUSTMENT</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_AR_GET_ACTION_ITEMS</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_AR_GET_BILLS</td>
</tr>
<tr>
<td></td>
<td>- PCM_OP_ARRESOURCE_AGGREGATION</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>BRMBalServices.wsdl</td>
<td>Defines the balances Web service, which includes the following opcodes:&lt;br&gt;  ■ PCM_OP_BAL_GET_BALANCES&lt;br&gt;  ■ PCM_OP_BAL_GET_BAL_GRP_AND_SVC&lt;br&gt;  ■ PCM_OP_BAL_GET_ACCT_BAL_GRP_AND_SVC&lt;br&gt;  ■ PCM_OP_BAL_GET_ACCT_BILLINFO&lt;br&gt;See “Balance FM Standard Opcodes” in BRM Developer’s Reference for more information.</td>
</tr>
<tr>
<td>BRMBillServices.wsdl</td>
<td>Defines the billing Web service, which includes the following opcodes:&lt;br&gt;  ■ PCM_OP_BILL_GET_ITEM_EVENT_CHARGE_DISCOUNT&lt;br&gt;  ■ PCM_OP_BILL_GROUP_MOVE_MEMBER&lt;br&gt;  ■ PCM_OP_BILL_MAKE_BILL_NOW&lt;br&gt;  ■ PCM_OP_BILL_DEBIT&lt;br&gt;  ■ PCM_OP_BILL_GROUP_GET_PARENT&lt;br&gt;See “Billing FM Standard Opcodes” in BRM Developer’s Reference for more information.</td>
</tr>
<tr>
<td>BRMCollectionsServices.wsdl</td>
<td>Defines the collections Web service, which includes the following opcode:&lt;br&gt;  ■ PCM_OP_COLLECTIONS_SET_ACTION_STATUS&lt;br&gt;See “Collections Manager FM Standard Opcodes” in BRM Developer’s Reference for more information.</td>
</tr>
<tr>
<td>BRMCustcareServices.wsdl</td>
<td>Defines the customer care Web service, which includes the following opcode:&lt;br&gt;  ■ PCM_OP_CUSTCARE_MOVE_ACCT</td>
</tr>
<tr>
<td>BRMCustServices.wsdl</td>
<td>Defines the customer Web service, which includes the following opcodes:&lt;br&gt;  ■ PCM_OP_CUST_COMMIT_CUSTOMER&lt;br&gt;  ■ PCM_OP_CUST_MODIFY_CUSTOMER&lt;br&gt;  ■ PCM_OP_CUST_UPDATE_CUSTOMER&lt;br&gt;  ■ PCM_OP_CUST_UPDATE_SERVICES&lt;br&gt;  ■ PCM_OP_CUST_DELETE_ACCT&lt;br&gt;  ■ PCM_OP_CUST_DELETE_PAYINFO&lt;br&gt;  ■ PCM_OP_CUST_CREATE_PROFILE&lt;br&gt;  ■ PCM_OP_CUST_MODIFY_PROFILE&lt;br&gt;  ■ PCM_OP_CUST_DELETE_PROFILE&lt;br&gt;See “Customer FM Standard Opcodes” in BRM Developer’s Reference 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>BRMInvServices.wsdl</td>
<td>Defines the invoicing Web service, which includes the following opcode:</td>
</tr>
<tr>
<td></td>
<td>• PCM_OP_INV_VIEW_INVOICE</td>
</tr>
<tr>
<td></td>
<td><strong>Important:</strong> 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” in BRM JCA Resource Adapter. See “Invoicing FM Standard Opcodes” in BRM Developer’s Reference for more information.</td>
</tr>
<tr>
<td>BRMPricesServices.wsdl</td>
<td>Defines the prices Web service, which includes the following opcodes:</td>
</tr>
<tr>
<td></td>
<td>• PCM_OP_PRICE_COMMIT_PRODUCT</td>
</tr>
<tr>
<td></td>
<td>• PCM_OP_PRICE_COMMIT_DISCOUNT</td>
</tr>
<tr>
<td></td>
<td>• PCM_OP_PRICE_SET_PRICE_LIST</td>
</tr>
<tr>
<td></td>
<td>• PCM_OP_PRICE_GET_PRICE_LIST</td>
</tr>
<tr>
<td></td>
<td>See “Price List 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 includes the following opcode:</td>
</tr>
<tr>
<td></td>
<td>• 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>
<tr>
<td>BRMReadServices.wsdl</td>
<td>Defines the read Web service, which includes the following 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 “LDAP Base Opcodes” in BRM Developer’s Reference for more information.</td>
</tr>
<tr>
<td>BRMSubscriptionServices.wsdl</td>
<td>Defines the subscription Web service, which includes the following opcodes:</td>
</tr>
<tr>
<td></td>
<td>• PCM_OP_SUBSCRIPTION_CANCEL_PRODUCT</td>
</tr>
<tr>
<td></td>
<td>• PCM_OP_SUBSCRIPTION_CANCEL_DISCOUNT</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_PURCHASE_DEAL</td>
</tr>
<tr>
<td></td>
<td>• PCM_OP_SUBSCRIPTION_SET_BUNDLE</td>
</tr>
<tr>
<td></td>
<td>• PCM_OP_SUBSCRIPTION_SET_DISCOUNT_STATUS</td>
</tr>
<tr>
<td></td>
<td>• PCM_OP_SUBSCRIPTION_SET_DISCOUNTINFO</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>
<tr>
<td></td>
<td>• PCM_OP_SUBSCRIPTION_GET_PURCHASED_OFFERINGS</td>
</tr>
<tr>
<td></td>
<td>See “Subscription Management 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" in BRM JCA Resource Adapter.

Configuring Java Logging in Oracle WebLogic Server

To configure Java logging in Oracle WebLogic server, do the following:

1. Specify the Java Unified Logging (JUL) mechanism. See "Specifying the JUL Mechanism".
2. Create a startup class. See "Creating a Startup Class".
3. Change the logging level by using JConsole. See "Changing the Java Logging Level in Oracle WebLogic Server".

Specifying the JUL Mechanism

Specifying the Java Unified Logging (JUL) mechanism allows Web Services Manager to use JUL in addition to the WebLogic Server Administration console logging.

To specify the JUL mechanism:

1. Open the BRM_Home/deploy/web_services/Infranet.properties file in a text editor.
2. Uncomment the following entry:
   
   `# webservices.log.enabled = true`

3. Change the value to `false`:
   
   `webservices.log.enabled = false`

4. Save and close the file.

Creating a Startup Class

You define a startup class to enable JUL and create log files for the following Web service classes:

- `com.portal.webservices.BRMFlistToXML`
- `com.portal.webservices.BRMXMLToFlist`
- `com.portal.webservices.OpcodeCaller`
- `com.portal.webservices.WebServicesUtilities`

To create a startup class:

1. Copy the `BRM_Home/deploy/web_services/weblogic_ws_startup.jar` file to the `local_dir/WEB-INF/lib` directory.
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 BRMWSLoggerStartUpClass.

8. In the Class Name field, enter com.portal.webservices.BRMWSLoggerStartUp.

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 BRMWSLoggerStartUpClass.
The Settings for BRMWSLoggerStartUpClass pane appears.

12. Select Run Before Application Deployments and Run Before Application Activations and click Save.

13. Click Activate Changes.

14. Restart the WebLogic Server, which applies the changes.

By default, log files are created in the WebLogic_Home/user_projects/domains/Domain_Name/logs/BRMWebServicesMgrLogs/BRMWebServicesMgr.log directory.
where:

- WebLogic_Home is the directory in which you installed WebLogic Server.
- Domain_Name is the name of the domain you are configuring.

### Changing the Java Logging Level in Oracle WebLogic Server

To change the Java logging level in WebLogic Server:

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 1–1.
Figure 1–1 New Connection Dialog Box

3. In the Host Name field, enter the WebLogic Server host name.
4. In the Port Number field, enter the WebLogic Server port number.
5. In the User Name field, enter your user name.
6. In the Password field, enter your password.
7. 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.

8. Click the MBeans tab.
9. In the MBean tree, expand java.util.logging, then expand Attributes, and then select LoggerNames.

The LoggerNames pane appears.

10. Copy the com.portal.webservices.BRMFlistToXML logger name.
11. In the MBean tree, expand java.util.logging and then select Operations.

The Operation invocation pane appears.
12. In the `void setLoggerLevel p0` field, paste the logger name that you copied.

13. In the `void setLoggerLevel p1` field, change the logging level to one of the following:
   - To log problems that require attention from the system administrator, change the value to `SEVERE`.
   - To log actions that should be reviewed and may require action before an error occurs, change the value to `WARNING`.
   - To log normal actions or events, change the value to `INFO`. This includes user operations, such as user log ins, and automatic operations, such as log file rotations.
   - To log configuration-related messages, change the value to `CONFIG`.
   - To log trace or debug messages for performance monitoring, change the value to `FINE`.
   - To log highly detailed trace or debug message, change the value to `FINER`.
   - To log the most detailed trace or debug messages, change the value to `FINEST`.

14. Click `setLoggerLevel`, which updates the logging level.

15. In the MBean tree, select `LoggerNames`.
    The Logger Names pane appears.

16. Copy the `com.portal.webservices.BRMXMLToFlist` logger name.

17. Repeat steps 11 to 14 to change the logging level for the `com.portal.webservices.BRMXMLToFlist` logger.


19. Repeat steps 11 to 14 to change the logging level for the `com.portal.webservices.OpcodeCaller` logger.


21. Repeat steps 11 to 14 to change the logging level for the `com.portal.webservices.WebServicesUtilities` logger.

22. Close the Java Monitoring and Management Console pane.

For more information, see the discussion about using JConsole in *Java SE Monitoring and Management Guide*: [http://docs.oracle.com/javase/6/docs/technotes/guides/management/jconsole.html](http://docs.oracle.com/javase/6/docs/technotes/guides/management/jconsole.html).