Installation Guide for Oracle Billing Insight

Version 7.0, Rev. B

November 2016

ORACLE

Contents

Chapter 4: Configuring Oracle WebLogic

1 What's New in This Release

What's New in Installation Guide for Oracle Billing Insight, Version 7.0, Rev. B

Table 1 lists the changes in this version of the documentation to support this release of the software.

Table 1. What's New in Installation Guide for Oracle Billing Insight, Version 7.0, Rev. B

Topic	Description
"Process of Migrating Oracle Self-Service E-Billing Version 6.2 to Oracle Billing Insight Version 7.0" on page 117	Modified topics. Added a step to run the master key update for implementations that used encryption in version 6.2. Also added steps to migrate users.
"Process of Migrating Oracle Self-Service E-Billing Version 6.1 to Oracle Billing Insight Version 7.0" on page 122	Modified topics. Added a step to run the master key update for implementations that used encryption in version 6.1. Also added steps to migrate users.

What's New in Installation Guide for Oracle Billing Insight, Version 7.0, Rev. A

Table 2 lists the changes in this version of the documentation to support this release of the software.

Table 2. What's New in Installation Guide for Oracle Billing Insight, Version 7.0, Rev. A

Topic	Description
"Process of Migrating Oracle Self-Service E-Billing Version 6.2 to Oracle Billing Insight Version 7.0" on page 117	New topic. Added steps for migrating to Oracle Self-Service E-Billing Version 7.0 from version 6.2.
"Process of Migrating Oracle Self-Service E-Billing Version 6.1 to Oracle Billing Insight Version 7.0" on page 122	New topic. Added steps for migrating to Oracle Self-Service E-Billing version 7.0 from version 6.1.
"Process of Implementing TDE Column Encryption" on page 33	Modified topic. Corrected the name of the directory where the encrypt_olap.log file is located.

What's New in Installation Guide for Oracle Billing Insight, Version 7.0

Table 3 lists the changes in this version of the documentation to support this release of the software.

Table 3. What's New in Installation Guide for Oracle Billing Insight, Version 7.0

Topic	Description
"Modifying the csr.xma.xml File for the Assisted Service Application Properties" on page 84	Modified topic. With the use of the Single-Sign On feature for accessing impersonation, it is no longer necessary to set the URL for logging in and out of the Assisted Service application in the csr.xma.xml file. Oracle Self-Service E-Billing uses the values set in the globalConfig.xma.xml file.
"Process of Deploying Oracle Billing Insight Applications on Oracle WebLogic" on page 95	New topic. This topic lists the individual tasks needed to deploy Oracle Self-Service E-Billing applications.
"Deploying the Jersey 2.5.1 (JAX-RS 2.0 RI) Shared Library for Use By the Web Services Application" on page 96	New topic. Describes how to deploy the Jersey 2.5.1 (JAX-RS 2.0 RI) to register the library with Oracle WebLogic on the servers where you intend to deploy the Web Services application. You no longer need to download the third-party library.
"Configuring and Starting Scheduler on Oracle WebLogic" on page 100	Modified topics. Updated the steps for starting Scheduler to remove database information that is no longer required.
"Running the Sample Oracle Billing Insight Applications on Oracle WebLogic" on page 102	Modified topic. Updated the procedure for running sample applications.
"Sample Users" on page 103	New topic. Shows the enhanced list of sample users and related data available in the sample Self-Service and Assisted Service applications.
Chapter 5, "Configuring the ODI Data Load Processes"	New chapter. This chapter describes the steps for configuring Oracle Data Integrator with Oracle Self-Service E-Billing.

Additional Changes

The following changes have been made in this version of the guide:

This book also contains the following name changes in this version:

- The product name changed to Oracle Self-Service E-Billing.
- The Customer Service Representative application name changed to the Assisted Service application.
- The Billing and Payment application name changed to the Self-Service application.

- Many directory and file names have changed to reflect the new product and application names.

2 Installing Oracle Billing Insight

This chapter covers the tasks you must perform to prepare your platform and install Oracle Billing Insight. It includes the following topics:

Roadmap for Installing Oracle Billing Insight 7.0

This topic describes the tasks necessary to install a new implementation of Oracle Billing Insight.

CAUTION: If you are migrating to this version of Oracle Billing Insight, then do not use this roadmap. Follow the particular roadmap appropriate for migrating your current product and version. If you perform a full installation on an existing implementation instead of the proper migration, then the migration will fail and the database will become unstable from deployment of the target schema.

For information about migration, see Chapter 6, “Migrating to Oracle Billing Insight 7.0.”

To install a new implementation of Oracle Billing Insight, perform the following processes and tasks:

1. “Preparing Your Platform” on page 12
2. “Checking the Integrity of the Oracle Billing Insight Installer Package” on page 12
3. “Installing the Oracle Billing Insight Software Using InstallAnywhere” on page 13

   For distributed environments, it is recommended that you install and configure Oracle Billing Insight in the same top-level directory structure, first on the Oracle Billing Insight database server, then on the Oracle Billing Insight application server.
4. “Configuring Log File Paths for Log4j” on page 15
5. “Adding Foreign Language Fonts to Your Application Server” on page 17
6. “Roadmap for Configuring the Oracle Billing Insight Database” on page 23

7 "Roadmap for Configuring Oracle WebLogic for Oracle Billing Insight" on page 47

8 "Roadmap for Configuring the Data Load Processes for Oracle Billing Insight" on page 107

9 Enroll the bootstrap user ID in the Command Center application and create the system administrator users. For details, see *Administration Guide for Oracle Billing Insight*.

Also enroll the bootstrap user ID in the Assisted Service application, creating a new personal system administrator. For details about enrolling the bootstrap user in Assisted Service application, see *Assisted Service Application Guide for Oracle Billing Insight*.

Preparing Your Platform

Before installing Oracle Billing Insight, you must verify that your platform is ready.

This task is a step in "Roadmap for Installing Oracle Billing Insight 7.0" on page 11.

To verify that your platform is ready to install Oracle Billing Insight

1 Install and test required hardware and software.

For a list of system requirements, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

2 Define required user and group permissions for your Oracle Billing Insight database and application servers.

For details on configuring your application server, see "Preparing to Configure Oracle WebLogic" on page 48.

3 Start and test your Oracle Billing Insight application server.

For details, see your server documentation.

4 For distributed database environments, make sure you have any required Oracle database client software installed on your application server and any other client computers of your Oracle Billing Insight database server.

Checking the Integrity of the Oracle Billing Insight Installer Package

After downloading Oracle Billing Insight from Oracle Software Delivery Cloud, you must check the integrity of the installer package, using a checksum utility. Run the integrity check before installing Oracle Billing Insight.

The purpose of the integrity check is to validate that the full package has been delivered and that no data was corrupted during the download of the Oracle Billing Insight installation files. The checksum utility verifies that the checksum value of the installer package matches the value stored in the checksum.md5 file included with the download. If the values are not the same, then either the download was interrupted or data was corrupted, and you must start the download again.

This task is a step in "Roadmap for Installing Oracle Billing Insight 7.0" on page 11.

To check the integrity of the Oracle Billing Insight installer package

1. Verify that you have a checksum tool, or download a free checksum tool from the Web if necessary:
   - **UNIX.** The md5sum utility is installed by default. You can also download the file from the following GNU Web site:

     http://ftp.gnu.org/gnu/coreutils/
   - **Windows.** Download one of a variety of free md5 utilities available for verifying the checksum, such as the following tool, and extract the content of the ZIP file:

     http://www.winmd5.com
2. Verify that the binary installer file for Oracle Billing Insight, BillingInsight.bin or WinMD5.exe, and the checksum.md5 file are both in the same directory.
3. Run the appropriate command for your operating system:
   - **UNIX.** You can perform the value check automatically or manually. To run the automatic check, use the automatic check option, -c. The result is either valid (OK) or failed, for example:

     ```
     $ md5sum -c /temp/checksum.md5

     BillingInsight.bin: OK
     ```

     To perform the value check manually, run the following command, then manually compare the value generated with the value in the checksum.md5 file:

     ```
     $ md5sum BillingInsight.bin
     ```
   - **Windows.** Run the `WinMD5.exe` command, click Browse, then select the downloaded BillingInsight.exe file. Copy the checksum value from the downloaded checksum.md5 file, then paste it in the text input area. Click Verify.

Installing the Oracle Billing Insight Software Using InstallAnywhere

Oracle distributes Oracle Billing Insight as an InstallAnywhere package. You can change the default installation directory when prompted during the installation procedure.

This document refers to the directory where you install Oracle Billing Insight as the EDX_HOME directory which is, by default:

- **UNIX.** `/opt/Oracle/BillingInsight`
- **Windows.** `Oracle\BillingInsight`

Scripts for creating and configuring the Oracle Billing Insight database are located in the following directory:

- **UNIX.** *EDX_HOME*/db
- **Windows.** *EDX_HOME*\db

Web applications you must deploy to your Oracle Billing Insight application server are located in the following directory:

- **UNIX.** *EDX_HOME*/J2EEApps
- **Windows.** *EDX_HOME*\J2EEApps

This task is a step in "Roadmap for Installing Oracle Billing Insight 7.0" on page 11.

To install the Oracle Billing Insight software using InstallAnywhere

1. Start InstallAnywhere in UI mode:
   - **UNIX.** Log in using the user and group name of the Oracle Billing Insight application server owner, such as edxadmin:edxadmin. Make sure DISPLAY is set, then type the following command:

     ```
     ./BillingInsight.bin
     ```

     To start InstallAnywhere in Console Mode, type the following command then follow the on-screen instructions:

     ```
     ./BillingInsight.bin -i console
     ```

   - **Windows.** Double-click the BillingInsight.exe file, and follow the on-screen instructions.
2. On the Introduction screen, read the Oracle Billing Insight introductory information. Click Next to continue.
3. On the License Agreement screen, read the licensing agreement carefully, select the terms acceptance, then click Next.
4. On the Enter Serial Number screen, enter your product serial number, then click Next.
5. (UNIX Only) On the Owner of Web Application Server screen, enter the name of the application server owner (if you have installed other Oracle Self-Service products, then use the same owner at this screen that you used for those product installations). Then click Next.
6. (UNIX Only) On the Group of Web Application Server screen, enter the name of the group for the application server. If you have installed other Oracle Self-Service products, then use the same group at this screen that you used for those product installations. Then click Next.
7. On the Choose Install Folder screen, accept the default installation folder or click Choose and enter the directory where you want to install the Oracle Billing Insight files and directories.

   The directory where you install Oracle Billing Insight is referred to in this document as EDX_HOME. Click Next to continue.
8. Select the vertical application to install, either Telco or Utility.
9. On the Choose Product Features screen, select Option 1, Oracle Billing Insight, and click Next.
10. (Windows Only) On the Choose Shortcut Folder, choose the New Program Group, then click Next.

11 On the Preinstallation Summary screen, verify that the information is correct, then click Install. To correct any entries, click Previous.

The installer copies the Oracle Billing Insight software components to the designated installation folder. A status bar on the bottom of the screen shows each component being installed.

If the installation is successful, then a congratulatory message appears with the directory that contains the Oracle Billing Insight components. Click Next.

12 Click Done to exit the installer.

If the installation fails, then determine the cause of the problem, and run InstallAnywhere again to reinstall Oracle Billing Insight.

CAUTION: The installation and configuration examples in this guide use default Oracle Billing Insight paths, privileges, and permissions. If you choose not to accept the default values, then make sure your values are consistent in all servers for your installation, or Oracle Billing Insight will not function properly.

Configuring Log File Paths for Log4j

By default, Oracle Billing Insight writes log files to the following directory:

- **UNIX.** `WL_HOME/user_projects/domains/domain_name`
- **Windows.** `WL_HOME\user_projects\domains\domain_name`

It is recommended that you specify the log output paths to the following directory:

- **UNIX.** `/opt/Oracle/BillingInsight/logs`
- **Windows.** `D:\Oracle\BillingInsight\logs`

Oracle Billing Insight applications generate log files for the different functionality in each application. To change log output paths, update the XML log configuration files associated with each application.

This task is a step in "Roadmap for Installing Oracle Billing Insight 7.0" on page 11.

To change the output path of log files

- Edit the output path name in the File parameter for each log4j configuration files, located in the following directory:
  - **UNIX.** *EDX_HOME*`/config`

- **Windows.** *EDX_HOME*\config

The log4j configuration files are listed in the following table.

Oracle Billing Insight Application	Log4j Configuration File	Log Files
Self-Service	log4j.xml	■ eBilling-log ■ apache-log ■ root-log ■ reporting-log ■ hierarchy-log ■ umf-log
Command Center	log4j_cc.xml	■ FILE_Thirdparty
Assisted Service	log4j_csr.xml	■ cba-log ■ csr-log ■ root-log ■ reporting-log ■ hierarchy-log ■ umf-log

For example, the log4j.xml configuration file specifies the default output path for the eBilling-log file as eBilling.log (under the application server's domain_name or profile_name directory):

```
<appender name="ebilling-log" class="org.apache.log4j.RollingFileAppender">
<param name="File" value="eBilling.log"/>
...
</ appender>
```

To change the output path for the eBilling-log file, set the File value to /opt/Oracle/BillingInsight/logs/eBilling.log, as shown in the following example:

```
<appender name="ebilling-log" class="org.apache.log4j.RollingFileAppender">
<param name="File" value="/opt/Oracle/BillingInsight/logs/eBilling.log"/>
...
</ appender>
```

Adding Foreign Language Fonts to Your Application Server

If you plan to localize Oracle Billing Insight, then you must copy the foreign language fonts in the TrueType Font (TTF) files to your application server directories.

This task is a step in "Roadmap for Installing Oracle Billing Insight 7.0" on page 11.

To copy foreign language fonts to your application server

- The TTF files are located in the following directory:
  - **UNIX.** *EDX_HOME*/config/fonts
  - **Windows.** *EDX_HOME*\config\fonts

  Copy all of the TTF files to the following folders in Oracle WebLogic:
  - **UNIX:**
    - WL_HOME\jdk150_11\jre\lib\fonts\
    - WL_HOME\jrockit_150_11\jre\lib\fonts\
  - **Windows:**
    - WL_HOME/jdk150_11/jre/lib/fonts/
    - WL_HOME/jrockit_150_11/jre/lib/fonts/

Importing the Chase Paymentech Certificate

If you intend to use the Chase Paymentech Orbital Payment Gateway, then you must download the product certificate and import it into the JDK Cacerts file. See the readme.txt file provided with Chase Paymentech for additional information.

This task is a step in "Roadmap for Installing Oracle Billing Insight 7.0" on page 11.

To import the Chase Paymentech certificate

1. Copy the certificate files to the following directory:
   - **UNIX.** *EDX_HOME*/payment/paymentech/certupdate
   - **Windows.** *EDX_HOME*\payment\paymentech\certupdate
2. Set JAVA_HOME to directory where Java/JDK is installed
   - **UNIX.** `export JAVA_HOME={JAVA Home Directory}`
   - **Windows.** `set JAVA_HOME={JAVA Home Directory}`
3. Run the ImportCert command:

- **UNIX.** `./ImportCert.sh`
- **Windows.** `./ImportCert.bat`

4 Click Yes.

Installing Digital Accessible Information System (DAISY)-Related Software

If you want to offer end users the option to receive billing statements in Digital Accessible Information System (DAISY) audio file format, then follow these steps to install the required DAISY-related software and configure your implementation. Oracle Billing Insight provides DAISY audio files as attachments to statement-ready email notifications. This feature is available to users in the Consumer Edition of Oracle Billing Insight only.

To install DAISY-related software

1 Follow the instructions contained in the following document to install and set up the Odt2daisy toolkit on the same computer where you installed Oracle Billing Insight. Do not use spaces in the save path name.

http://odt2daisy.sourceforge.net/doc/Dev_Documentation.pdf

2 Click the following link to download the DAISY Pipeline Core binaries to the same computer where you installed Oracle Billing Insight:

http://sourceforge.net/projects/daisymfc/files/pipeline/pipeline-20111215/pipeline-20111215.zip/download

3 Install and configure the DAISY Pipeline LAME MP3 encoder on the same computer where you installed Oracle Billing Insight. Download the software:

- **Linux and Oracle Solaris.** Go to the following Web site:

  http://lame.sourceforge.net/download.php

  By default. LAME installs in the path `/usr/local/bin/lame` directory. You can optionally specify a different installation directory.

- **Windows.** Go to the following site and click the Softmedia Mirror link appropriate for your implementation:

  http://www.softpedia.com/progDownload/LAME-MP3-Encoder-Download-6952.html

  Create a subdirectory called `ext` under the `pipeline-20111215` folder. Copy the LAME.exe file, found in the downloaded LAME.zip file, to this folder.

4 Edit the pipeline.user.properties file, found in the `pipeline-20111215` directory. Add the absolute path of the LAME.exe file in the Path to LAME section, for example:

- **Linux and Oracle Solaris**:

```
<!-- Path to LAME -->
```

```
<entry key="pipeline.lame.path">/usr/local/bin/lame</entry>
```

- **Windows:**

```
<!-- Path to LAME -->

<entry key="pipeline.lame.path">d:\devTools\pipeline-20111215\ext\lame.exe</
entry>
```

5 (Linux only) Install and configure the Software Speech synthesizer (Text-To-Speech) plug-in, eSpeak:

a From the main Linux menu, select System, Application, and then Add/Remove Software.

b Search for espeak. From the resulting list, select version espeak-1.40.02-4.el6 (x86_64).

By default, eSpeak installs in the `/usr/bin/espeak` directory.

c Edit the pipeline.user.properties file, found in the `pipeline-20111215` folder. Add the absolute installed path of eSpeak to Path to SOX section:

```
<!-- Path to sox (Sound eXchange - used on Linux) -->

<entry key="pipeline.espeak.path">/usr/bin/espeak</entry>
```

6 Open the reporting configuration file, globalConfig.properties, located in the following directory:

- **Linux and Oracle Solaris**. *EDX_HOME*/config/rpt
- **Windows.** *EDX_HOME*\config\rpt

7 Specify the following parameters.

Property	Description
odt2daisy.path	The location of the odt2daisy jar file, for example: odt2daisy.path=C:\\NetBeansProjects\\odt2daisy\\dist\\odt2daisy.jar

Property	Description
pipeline.command	The pipeline classpath. On Windows, the library name in the classpath must be separated by a semicolon, for example: pipeline.command=java -classpath C:\\pipeline-20111215\\pipeline.jar;C:\\pipeline-20111215 org.daisy.pipeline.ui.CommandLineUI On Linux or UNIX, the library in the classpath must be separated by a colon, for example: java -classpath /user/pipeline-20111215/pipeline.jar:/user/pipeline-20111215 org.daisy.pipeline.ui.CommandLineUI
pipeline.script	The location of the Pipeline software, for example: pipeline.script=D:\\pipeline-20111215\\scripts\\create_distribute\\dtb\\DTBookToDaisy3TextOnlyDTB.taskScript

8 To enable the log file for the StatementSummaryDaisyProvider job, open the log4j_cc.xml file, located in the following directory:

- **Linux and Oracle Solaris**. *EDX_HOME*/config/
- **Windows.** *EDX_HOME*\config\

Add the following content before the first category element:

```
<category name="report.daisy.log" additivity="false">
                <priority value="DEBUG"/>
            <!-- <appender-ref ref="FILE_SCHEDULER"/> -->
            <!-- <appender-ref ref="CONSOLE"/>  -->
                <appender-ref ref="CONSOLE"/>
        </category>
```

9 Restart the Command Center server.

10 To enable a user to select the alternative bill-ready notification method for DAISY audio file format, you must set the ISACCESSIBLE column in the EDX_BSL_UMF_USER OLTP table for the individual user to a value of 1 (the default value is 0, or disabled).

Uninstalling Oracle Billing Insight

You can uninstall and remove Oracle Billing Insight components and deployed J2EE applications using the Oracle Billing Insight Uninstaller.

Uninstall Oracle Billing Insight from the database server first, then uninstall it from the application server.

Note that the Uninstaller does not delete any directories that contain files modified since installation. Instead, it lists these items, which you must then remove manually.

Oracle Billing Insight does not provide a log for uninstalling; steps to capture the uninstall debug log are included in the following procedure.

This task is a step in "Roadmap for Installing Oracle Billing Insight 7.0" on page 11.

To uninstall Oracle Billing Insight

1. Stop your application server.
2. Stop your database instance and your database server.
3. Navigate to the `Uninstall` folder of your Oracle Billing Insight home directory.
4. On UNIX, direct debug output to either the console or to a file.
   - **Console**. Enter the following commands:

     ```
     export LAX_DEBUG=true
     ```

     ```
     - or - setenv LAX_DEBUG true - or - LAX_DEBUG=true set LAX_DEBUG
     ```

   - **File**. Enter the following commands:

     ```
     setenv LAX_DEBUG /export/home/temp/uninstall.log
     ```

     ```
     set LAX_DEBUG=/export/home/temp/uninstall.log
     ```

5. Start the Oracle Billing Insight Uninstaller:
   - **UNIX.** Run the `Uninstall_BillingInsight` script, located in the *EDX_HOME*`/Uninstall` directory. In the path, *EDX_HOME* is the location where Oracle Billing Insight is installed.
   - **Windows.** Click the Uninstall BillingInsight.exe file icon; immediately press and hold the Ctrl key to direct the debug log output to the console.
6. When the Uninstall screen appears, click Uninstall.

   Oracle Billing Insight lists the components while it removes them from your computer. When the Uninstaller is finished, a screen appears listing any items that could not be removed.
7. Change the directory to your Oracle Billing Insight home directory, and manually remove any remaining files and directories as necessary.
8. Click Done to close the Uninstaller.

9 Repeat this procedure on your application server and for any other installations of Oracle Billing Insight.

3 Configuring the Oracle Database

This chapter describes how to configure the Oracle database for Oracle Billing Insight. It includes the following topics:

Roadmap for Configuring the Oracle Billing Insight Database

To configure the Oracle Billing Insight database, perform the following processes and tasks:

1. "Preparing to Configure the Oracle Billing Insight Database" on page 24
2. "Configuring Oracle Services" on page 24
3. To create the Oracle Billing Insight database, choose the database installation process appropriate for your implementation:
   - "Creating the Oracle Billing Insight Database Using Ant" on page 26
   - "Creating the Oracle Billing Insight Database Using the Automated Ant Target" on page 31
4. If you plan to use database encryption, then see "Choosing a Database Encryption Method" on page 32 and one of the following database encryption processes:
   - "Process of Implementing TDE Column Encryption" on page 33
   - "Process of Implementing TDE Tablespace Encryption" on page 38
5. (Optional) "Loading Sample Data" on page 40

6 (Optional) “Enabling Oracle Auditing” on page 41

7 “Roadmap for Configuring the Data Load Processes for Oracle Billing Insight” on page 107

8 “Creating a Bootstrap Administrator User for Oracle Billing Insight” on page 42

Related Topic

“Roadmap for Installing Oracle Billing Insight 7.0” on page 11

Preparing to Configure the Oracle Billing Insight Database

Before running the Oracle Billing Insight database installation Ant script, you must complete the following steps.

This task is a step in “Roadmap for Configuring the Oracle Billing Insight Database” on page 23.

To prepare to install the Oracle Billing Insight database

1 Make sure you have met the minimum system requirements.

   For details on system requirements, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

2 Confirm that you have sufficient space on your Oracle Billing Insight database server. For updated disk space requirements, see 1358365.1 (Article ID) on My Oracle Support.

   CAUTION: Insufficient disk space can cause the Oracle Billing Insight database configuration to fail.

3 Upgrade your Oracle database server software as necessary. For distributed environments, make sure you have the required database client software installed on your application server and any other client computers of your Oracle Billing Insight database server.

4 Plan passwords with your system administrator, and have a sys account password available.

Using Database Clustering

Note that your application server handles database clustering, not Oracle Billing Insight. For help with clustered installations, contact your Oracle sales representative to request assistance from Oracle's Professional Services.

Configuring Oracle Services

You must edit the following Oracle configuration files that control access to the Oracle Billing Insight production database:

- **listener.ora.** Includes a list of service names and address of all listeners on a computer, the instance names of the databases for which they listen, and listener control parameters. The address for a server in the listener.ora file requires the SID (SID_NAME) of a database server in the tnsnames.ora file.

  You modify the listener.ora file on the database servers.

- **tnsnames.ora**. Includes a list of service names of network databases that are mapped to connect descriptors. Clients and distributed database servers use this file to identify potential server destinations. The address of a given database server in the tnsnames.ora file matches the address of a listener for that server in the listener.ora file.

  You modify the tnsnames.ora file on the database clients.

By default, these files install in the network administration directory of your Oracle Billing Insight database server:

- **UNIX.** `$ORACLE_HOME/network/admin`
- **Windows.** `%ORACLE_HOME%\network\admin`:

Consult with your onsite DBA to configure database connectivity, to make sure you comply with client standards for the enterprise.

For help with database connectivity, create a service request (SR) on My Oracle Support. Alternatively, you can phone Oracle Global Customer Support directly to create a service request or get a status update on your current SR. Support phone numbers are listed on My Oracle Support.

This task is a step in "Roadmap for Configuring the Oracle Billing Insight Database" on page 23.

To configure Oracle services

1. Change the directory to the network administration directory of your Oracle Billing Insight database server, for example:
   - **UNIX.** `$ORACLE_HOME/network/admin`
   - **Windows.** `%ORACLE_HOME%\network\admin`
2. Open the listener.ora file, and edit the SID_LIST_LISTENER section to reflect your Oracle SID and Oracle Billing Insight database home directory, for example:

```
SID_LIST_LISTENER=

(SID_LIST=

(SID_DESC =

 (SID_NAME = EBILL)

 (ORACLE_HOME = /opt/oracle/product/12.1.0.1)

))
```

3. Save and close the listener.ora file.
4. Change directory to the network administration directory of your Oracle Billing Insight database client, for example, `%ORACLE_HOME%\network\admin`.

5 Open the tnsnames.ora file, and edit the database service that identifies your protocol, host, and port.

The following text is an example of a tnsnames.ora file that uses the service name EBILL, installed on the database server localhost. Your service name might be different:

```
EBILL =

 (DESCRIPTION =

 (ADDRESS_LIST =

 (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))

)

(CONNECT_DATA =

 (SERVICE_NAME = EBILL)

 )

 )
```

6 Save and close the tnsnames.ora file.

7 Repeat Step 5 for the tnsnames.ora file on your application server.

This file installs with your database client software. Single computer environments can skip this step.

8 Stop and restart the Oracle listener with the following listener control commands.

```
lsnrctl stop

lsnrctl start
```

9 After the Oracle listener is restarted, to see the service summary for the Oracle Billing Insight instance, run:

```
lsnrctl status


Services Summary...

EBILL has 1 service handler(s)
```

This service handler must match the name you entered for the Oracle SID during Oracle Billing Insight database configuration, in this example, EBILL.

Creating the Oracle Billing Insight Database Using Ant

Oracle Billing Insight provides an ant script for creating the Oracle Billing Insight database. The ant script uses Oracle Database Configuration Assistant (DBCA).

Before running the Ant script you must configure the BillingInsight.properties file with the same EBILL database SID and tnsnames, and OLTP and OLAP user names and passwords. This properties file contain configuration parameters specific to each installation and are used by the Ant script that installs the Oracle Billing Insight database.

This task is a step in "Roadmap for Configuring the Oracle Billing Insight Database" on page 23.

To create the Oracle Billing Insight database using Ant

1. Open the BillingInsight.properties configuration file in a text editor. This file is located in the following directory:
   - **UNIX.** *EDX_HOME*/db/oracle
   - **Windows.** *EDX_HOME*\db\oracle

   In the directory, *EDX_HOME* is the location where you installed Oracle Billing Insight.
2. In the BillingInsight.properties file, specify the property values for the current installation. For property details, see Table 4 on page 28.
3. Go to the directory with the Oracle Database installation files:
   - **UNIX.** *EDX_HOME*/db/oracle
   - **Windows.** *EDX_HOME*\db\oracle
4. If you have not configured the Apache Ant environment, then do so now:
   - **UNIX.** Run the following commands, where JDK170_45 is your JDK version:

     ```
     export ANT_HOME=/opt/apache-ant-1.8.3

     export JAVA_HOME= /opt/JDK170_45

     export PATH=$JAVA_HOME/bin:$ANT_HOME/bin:$PATH
     ```

   - **Windows.** Run the following commands, where JDK170_45 is your JDK version:

     ```
     set ANT_HOME=C:\apache-ant-1.8.3

     set JAVA_HOME= c:\JDK170_45

     set PATH=%PATH%;%ANT_HOME%\bin;%JAVA_HOME%\bin
     ```

5. Enter Ant to run the build script.

   By default, the Ant command runs the build.xml file in the current directory.
6. Select Option 1, Create a Database Instance Using DBCA.

   This step can take anywhere from 20 minutes to 2 hours to complete, depending on the speed of your platform. DBCA creates a new instance for the Oracle Billing Insight database, and defines the data dictionaries and all system database objects.

   Ant returns to the current menu when finished. Review all log files for possible errors.
7. Select Option 3, Create Database Schemas and Objects.

   This step creates a schema and objects for the Oracle Billing Insight database.

8 Select Option 1, Create Tablespaces

9 Select Option 2, Create Schemas.

If you have set properties for TDE encryption, then the Ant script prompts you to create an Oracle Wallet password when it is ready to create the Master Encryption Key. Specify a password for the Oracle Wallet, then enter it again.

10 You have the option to create database users and objects in one step or separately:

- To create both users and objects in one step, select Option 3, Run Steps 1 and 2. Review all log files for possible errors.
- To create users and objects separately, select Option 1, Create Database Users. Select Option 2, Create Database Objects. Review all log files for possible errors.

The script creates new database users, grants the necessary permission and privileges, and creates all the database objects for the Oracle Billing Insight OLTP and OLAP schemas. Ant returns to the current menu when finished.

11 Select Option 6, Return to the Main Menu.

Oracle Billing Insight Installation Properties

Table 4 describes the properties you must specify in the BillingInsight.properties file for your installation.

Table 4. Properties in the BillingInsight.properties File

Property	Value
ORACLE_HOME	The root installation path for any Oracle product installed through an installer that is based on Oracle Universal Installer.
OLAP_USER	The OLAP schema user name.
OLAP_PASSWD	The password for the OLAP schema.
OLTP_USER	The OLTP schema user name.
OLTP_PASSWD	The password for the OLTP schema.
EBILL_CDB	The Oracle SID of the Oracle Billing Insight database.
EBILL_PDB	The Oracle SID of the Oracle Billing Insight database if pluggable database is being used.
EBILL_TNS_NAME	The Oracle Billing Insight database TNS (Transparent Network Substrate) name.
SYS_PASSWD	The password of the SYS user.

Table 4. Properties in the BillingInsight.properties File

Property	Value
L_DB_EDX_DATA_OLAP_FILE_LOC	The folder location where the installation script creates the L_DB_EDX_DATA_TB_FILE_LOC schema tablespace. The installation script automatically adds the Oracle SID, schema name, and /data to the path, as in: /home/oracle/oradata/ORACLE_SID/SCHEMA_NAME/data.
L_DB_EDX_INDX_OLAP_FILE_LOC	The folder location where the installation script creates the L_DB_EDX_INDX_OLAP_FILE_LOC schema tablespace. The installation script automatically adds the Oracle SID, schema name, and /data to the path, as in: /home/oracle/oradata/ORACLE_SID/SCHEMA_NAME/data.
ENCRYPTION_WALLET_LOCN	(Optional) If you are implementing database encryption, then specify the location of the Oracle wallet folder location. Use the same location that you set in the sqlnet.ora file.
WALLET_AUTO_OPEN	(Optional) If you are implementing database encryption, then specify whether the Oracle Wallet opens automatically when the database instance is restarted, or whether you must open it manually using the Ant Encryption menu. The value can be Y or N. It is recommended to set the value to Y.
TDE_ENCRYPT_OLAPCOLUMN	(Optional) When you are using encryption, specify whether to use column-level encryption in the OLAP schema. The value can be Y or N.
TDE_ENCRYPT_OLAPTSPACE	(Optional) When you are using encryption, specify whether to use tablespace-level encryption in the OLAP schema. The value can be TRUE or FALSE.
EBILL_LISTEN_PORT	The Oracle Billing Insight database listening port for creating the bootstrap user.
EBILL_HOSTNAME	The Oracle Billing Insight database host name for creating the bootstrap user.
L_DB_APP_DATA_OLTP_FILE_LOC	The folder location where the installation script creates the L_DB_APP_DATA_OLTP_FILE_LOC schema tablespace. The installation script automatically adds the Oracle SID, schema name, and /data to the path, as in: /home/oracle/oradata/ORACLE_SID/SCHEMA_NAME/data.
L_DB_APP_INDX_OLTP_FILE_LOC	The folder location where the installation script creates the L_DB_APP_INDX_OLTP_FILE_LOC schema tablespace. The installation script automatically adds the Oracle SID, schema name, and /data to the path, as in: /home/oracle/oradata/ORACLE_SID/SCHEMA_NAME/data.

Table 4. Properties in the BillingInsight.properties File

Property	Value
L_DB_LOAD_DATA_OLTP_FILE_LOC	The folder location where the installation script creates the L_DB_LOAD_DATA_OLTP_FILE_LOC schema tablespace. The installation script automatically adds the Oracle SID, schema name, and /data to the path, as in: /home/oracle/oradata/ORACLE_SID/SCHEMA_NAME/data.
L_DB_LOAD_INDX_OLTP_FILE_LOC	The folder location where the installation script creates the L_DB_LOAD_INDX_OLTP_FILE_LOC schema tablespace. The installation script automatically adds the Oracle SID, schema name, and /data to the path, as in: /home/oracle/oradata/ORACLE_SID/SCHEMA_NAME/data.
L_DB_FS_DATA_OLTP_FILE_LOC	The folder location where the installation script creates the L_DB_FS_DATA_OLTP_FILE_LOC schema tablespace. The installation script automatically adds the Oracle SID, schema name, and /data to the path, as in: /home/oracle/oradata/ORACLE_SID/SCHEMA_NAME/data.
L_DB_FS_INDX_OLTP_FILE_LOC	The folder location where the installation script creates the L_DB_FS_INDX_OLTP_FILE_LOC schema tablespace. The installation script automatically adds the Oracle SID, schema name, and /data to the path, as in: /home/oracle/oradata/ORACLE_SID/SCHEMA_NAME/data.
L_DB_STG_DATA_OLTP_FILE_LOC	The folder location where the installation script creates the L_DB_STG_DATA_OLTP_FILE_LOC schema tablespace. The installation script automatically adds the Oracle SID, schema name, and /data to the path, as in: /home/oracle/oradata/ORACLE_SID/SCHEMA_NAME/data.
L_DB_STG_INDX_OLTP_FILE_LOC	The folder location where the installation script creates the L_DB_STG_INDX_OLTP_FILE_LOC schema tablespace. The installation script automatically adds the Oracle SID, schema name, and /data to the path, as in: /home/oracle/oradata/ORACLE_SID/SCHEMA_NAME/data.
LARGE_DB_EDX_DATA_OLTP _FILE_LOC	The folder location where the installation script creates the LARGE_DB_EDX_DATA_OLTP_FILE_LOC schema tablespace. The installation script automatically adds the Oracle SID, schema name, and /data to the path, as in: /home/oracle/oradata/ORACLE_SID/SCHEMA_NAME/data.
LARGE_DB_EDX_INDX_OLTP _FILE_LOC	The folder location where the installation script creates the LARGE_DB_EDX_INDX_OLTP_FILE_LOC schema tablespace. The installation script automatically adds the Oracle SID, schema name, and /data to the path, as in: /home/oracle/oradata/ORACLE_SID/SCHEMA_NAME/data.

Table 4. Properties in the BillingInsight.properties File

Property	Value
MEDIUM_DB_EDX_DATA_OLTP_FILE_LOC	The folder location where the installation script creates the MEDIUM_DB_EDX_DATA_OLTP_FILE_LOC schema tablespace. The installation script automatically adds the Oracle SID, schema name, and /data to the path, as in: /home/oracle/oradata/ORACLE_SID/SCHEMA_NAME/data.
MEDIUM_DB_EDX_INDX_OLTP_FILE_LOC	The folder location where the installation script creates the MEDIUM_DB_EDX_INDX_OLTP_FILE_LOC schema tablespace. The installation script automatically adds the Oracle SID, schema name, and /data to the path, as in: /home/oracle/oradata/ORACLE_SID/SCHEMA_NAME/data.
TDE_ENCRYPT_OLTPCOLUMN	(Optional) When you are using encryption, specify whether to use column-level encryption in the OLTP schema. The value can be Y or N.
TDE_ENCRYPT_OLTPTSPACE	(Optional) When you are using encryption, specify whether to use tablespace-level encryption in the OLTP schema. The value can be TRUE or FALSE.

Creating the Oracle Billing Insight Database Using the Automated Ant Target

Instead of manually performing each of the Oracle Billing Insight database setup steps, you can use the automated Ant target to install the Oracle Billing Insight database.

Before running the Ant script you must configure the BillingInsight.properties file with the same EBILL database SID and tnsnames, and OLTP and OLAP user names and passwords. This properties file contain configuration parameters specific to each installation and are used by the Ant script that installs the Oracle Billing Insight database.

This task is a step in "Roadmap for Configuring the Oracle Billing Insight Database" on page 23.

To create the Oracle Billing Insight database using the automated Ant target

1. Open the BillingInsight.properties configuration file in a text editor. This file is located in the following directory:
   - **UNIX.** *EDX_HOME*/db/oracle
   - **Windows.** *EDX_HOME*\db\oracle

   In the directory, *EDX_HOME* is the location where you installed Oracle Billing Insight.
2. In the BillingInsight.properties file, specify the property values for the current installation. For details, see Table 4 on page 28.

3 Go to the directory with the Oracle Database installation files:

- **UNIX.** *EDX_HOME*/db/oracle
- **Windows.** *EDX_HOME*\db\oracle

4 If the EBILL instance is already created and you want to create tablespaces and schemas on them, then use the install-existing target to create tablespaces, schemas, and schema objects (tables, indexes, packages, procedures, and so on) with the user names and passwords specified in the properties file. Run:

```
ant install-existing
```

These commands create the Oracle Billing Insight seed data for OLTP and OLAP.

If you want to install sample data for both OLAP and OLTP, then use the following commands:

- **Telco Sample Data.** `ant install-existing -DloadSampleDataTelco=true`
- **Utility Sample Data.** `ant install-existing -DloadSampleDataUtility=true`

5 If you have set properties for TDE encryption, then the Ant script prompts you to create an Oracle Wallet password when it is ready to create the Master Encryption Key. Specify a password for the Oracle Wallet, then enter it again.

6 Install the Oracle Data Integrator (ODI).

For instructions on installing and setting up the Oracle Data Integrator, see "Creating a Standalone Agent" on page 109.

7 Configure ODI for your production environment.

For instructions on configuring ODI, see "Configuring Oracle Billing Insight for Your ODI Configuration" on page 111.

Choosing a Database Encryption Method

Oracle Billing Insight supports Transparent Data Encryption (TDE), which ships with the Oracle database. Encryption is an optional feature. If you decide to use encryption, then you must choose your security strategy.

There are two types of Transparent Data Encryption:

- TDE Column Encryption
- TDE Tablespace Encryption

Both types of TDE provide secure storage and management of encryption keys in Oracle Wallet.

TDE Column Encryption enables encryption of sensitive data in select table columns. Use TDE Column Encryption if you only have small amount of data to encrypt.

TDE Tablespace Encryption is an alternative to TDE Column Encryption, and enables encryption of an entire tablespace. Encrypting an entire tablespace eliminates the need to identify which columns contain sensitive data.

NOTE: You must know how to administer the Oracle Billing Insight database before choosing and implementing any database encryption strategy. See *Oracle Database Advanced Security Administrator's Guide* on Oracle Technology Network for details.

This task is a step in "Roadmap for Configuring the Oracle Billing Insight Database" on page 23.

To choose a database encryption method

- Identify which data is sensitive, and decide which encryption solution is suitable for your organization:
  - **TDE Column Encryption.** Identify which columns contain sensitive data.

    For details on implementing TDE Column Encryption, see "Process of Implementing TDE Column Encryption" on page 33.
  - **TDE Tablespace Encryption.** Identify which tablespaces contain sensitive data.

    For details on implementing TDE Tablespace Encryption, see "Process of Implementing TDE Tablespace Encryption" on page 38. Consider the potential for a reduction in performance of about 9% when database-level encryption is enabled.

    NOTE: There are some restrictions when using TDE Column Encryption. See *Oracle Database Advanced Security Administrator's Guide* on Oracle Technology Network for details.

Process of Implementing TDE Column Encryption

TDE Column Encryption encrypts the columns listed in two CSV files during the creation of the Oracle Billing Insight database.

By default, columns in the Oracle Billing Insight OLAP and OLTP schemas already known to contain sensitive data are listed in the CSV files:

- **tde_olap_columns.csv.** Columns in the OLAP schema that contain sensitive data; 0 columns included by default.
- **tde_oltp_columns.csv.** Columns in the OLTP schema that contain sensitive data; 95 columns included by default.

You can identify additional columns that contain sensitive data and add them to these files.

This process is a step in "Roadmap for Configuring the Oracle Billing Insight Database" on page 23.

To implement TDE Column Encryption, perform the following tasks:

1. Follow the steps in "Specifying the Oracle Wallet Location" on page 36.
2. (Optional) To encrypt additional columns, follow these steps:

a. Open the tde_olap_columns.csv and tde_oltp_columns.csv files, located in the following directory, where *EDX_HOME* is the location where you installed Oracle Billing Insight:

- **UNIX.** *EDX_HOME*/db/oracle/encrypt
- **Windows.** *EDX_HOME*\db\oracle\encrypt

b. Add the additional columns, using the following format:

table_name1, column_name1

table_name1, column_name2

...

table_name2, column_name1

table_name2, column_name2

...

table_nameN, column_name1

table_nameN, column_name2

...

3. Follow the steps in "Creating the Oracle Billing Insight Database Using Ant" on page 26. Set the following encryption properties in when configuring the BillingInsight.property file in those steps.

Encryption Property	Description
ENCRYPTION_WALLET_LOCN	Specify the location of the Oracle Wallet folder. Use the same location that you set in the sqlnet.ora file.
WALLET_AUTO_OPEN	Specify whether to automatically open the Oracle Wallet when the database instance restarts. Valid values are Y or N. It is recommended to set the value to Y.
TDE_ENCRYPT_OLAPCOLUMN	Set the value to Y to set the Master Encryption Key and enable column-level encryption in the OLAP schema.
TDE_ENCRYPT_OLAPTSPACE	Set the value to FALSE to suppress tablespace-level encryption in the OLAP schema.
TDE_ENCRYPT_OLTPCOLUMN	Set the value to Y to enable column-level encryption in the OLTP schema.
TDE_ENCRYPT_OLTPTSPACE	Set the value to FALSE to suppress tablespace-level encryption in the OLTP schema.

NOTE: If you do not set the encryption properties while setting the other properties in the BilingInsight.properties file, then you will have to open these files again to set the encryption properties before performing column encryption. You will also have to use the Ant Encryption menu to set the Master Encryption Key and open the Oracle Wallet, which require you to shut down and restart the database again.

4 The Ant script prompts you to create an Oracle Wallet password when it is ready to create the Master Encryption Key. Specify a password for the Oracle Wallet, then enter it again.

The installation script sets the Master Encryption Key for TDE column encryption after creating the schema objects.

5 Go to the directory where the Oracle Billing Insight database installation files are located:

- **UNIX.** *EDX_HOME*/db/oracle
- **Windows.** *EDX_HOME*\db\oracle

6 Enter Ant.

By default, the Ant command runs the build.xml file in the current directory.

7 From the Main Menu, select Option 2, Setup Database Encryption (Optional).

8 Select Option 3 to run an encryption precheck.

A precheck reviews the columns listed in the CSV files and reports how many columns can be encrypted and how many cannot. A precheck generates two log files: precheck_olap.log and precheck_oltp.log. Review these log files for details. The log files are located in the following directory:

- **UNIX.** *EDX_HOME*/db/oracle
- **Windows.** *EDX_HOME*\db\oracle

9 If you did not set the TDE_ENCRYPT_OLAPCOLUMN and TDE_ENCRYPT_OLTPCOLUMN properties in the BillingInsight.properties file to Y before running the Ant installation script, then do the following:

a Open the BillingInsight.properties file, located in the following directory, and specify the values described in Step 3 on page 34 for the current installation:

- **UNIX.** *EDX_HOME*/db/oracle
- **Windows.** *EDX_HOME*\db\oracle

b Follow the steps in "Setting the Master Encryption Key Using the Ant Encryption Menu" on page 37.

10 If you did not set the WALLET_AUTO_OPEN property in the BillingInsight.properties file to Y (to automatically open the Oracle Wallet when starting a database instance, then follow the steps in "Opening the Oracle Wallet Using the Ant Encryption Menu" on page 37.

11 Select Option 4, Encrypt OLAP sensitive data. Review the encrypt_olap.log file, located in the following directory, for detailed information:

- **UNIX.** *EDX_HOME*/db/oracle
- **Windows.** *EDX_HOME*\db\oracle

12 Select Option 5, Encrypt OLTP sensitive data. Review the encrypt_oltp.log file, located in the following directory, for detailed information:

- **UNIX.** *EDX_HOME*/db/oracle
- **Windows.** *EDX_HOME*\db\oracle

13 Select Q, Quit.

Specifying the Oracle Wallet Location

To implement Transparent Data Encryption on either columns or tablespaces, you must specify an Oracle Wallet location in the sqlnet.ora configuration file before creating the database instance.

After creating the database instance, the database installation script creates an Oracle Wallet in the location specified in the sqlnet.ora file. An external security module adds a Master Encryption Key to the Wallet.

This task is a step in "Process of Implementing TDE Column Encryption" on page 33 and in "Process of Implementing TDE Tablespace Encryption" on page 38.

To specify the Oracle Wallet location in the sqlnet.ora files

1 Change to the network administration directory of your Oracle Billing Insight database server, for example:

- **UNIX.** `$ORACLE_HOME/network/admin`
- **Windows.** `%ORACLE_HOME%\network\admin`

2 Open the sqlnet.ora file, and use the ENCRYPTION_WALLET_LOCATION parameter to specify the Oracle Wallet Location.

The following example shows a sqlnet.ora file that uses the `/export/home/oracle12/wallet` directory as the Oracle Wallet Location:

```
ENCRYPTION_WALLET_LOCATION=
  (SOURCE=
          (METHOD=FILE)
          (METHOD_DATA=
            (DIRECTORY=/export/home/oracle12/wallet)
                    )
                )
```

CAUTION: The wallet location directory must have an absolute path and end with a right parenthesis. Verify that there are no invisible characters at the end of the directory path. These characters can cause Oracle Database not to recognize the directory.

3 Save and close the sqlnet.ora file.

Setting the Master Encryption Key Using the Ant Encryption Menu

When implementing TDE column encryption, you can optionally set the Master Encryption Key, using the Ant Encryption menu after creating the Oracle Billing Insight database, but before encrypting the columns. Setting the Master Encryption Key using the Ant Encryption menu is only necessary if you did not set the TDE_ENCRYPT_OLAPCOLUMN and TDE_ENCRYPT_OLTPCOLUMN property values to Y in the BillingInsight.properties file before running the Oracle Billing Insight database installation using Ant.

This task is a step in "Process of Implementing TDE Column Encryption" on page 33.

To set master encryption key using the Ant Encryption menu

1. Change to the directory of the Oracle Billing Insight database installation files. In the directory, *EDX_HOME* is the location where you installed Oracle Billing Insight.
   - **UNIX.** *EDX_HOME*/db/oracle
   - **Windows.** *EDX_HOME*\db\oracle
2. Enter Ant to run the build script.

   By default, the Ant command runs the build.xml file in the current directory.
3. Select Option 2, Setup Database Encryption.
4. Select Option 1, Set Master Encryption Key. Enter your password twice.

   Ant prompts you to create an Oracle Wallet password when it is ready to create the Master Encryption Key.
5. Specify a password for the Oracle Wallet, then enter it again.
6. Specify whether you want to open the Oracle Wallet automatically when the database instance restarts.

   It is recommended to enter Y to keep the Oracle Wallet open when the database restarts. If you enter N, then you will have to open the Wallet each time you restart the database.

Opening the Oracle Wallet Using the Ant Encryption Menu

The external security module stores the Master Encryption Key in an Oracle Wallet. The Oracle Billing Insight database must load the Master Encryption Key into memory before it can encrypt or decrypt columns. Opening the Wallet allows the database to access the Master Encryption Key.

When the Wallet is opened, it remains open until you shut down the database instance. When you restart the instance, you must open the Wallet again if it is set to manual open mode. The Oracle Billing Insight database is configured for manual open mode when the WALLET_AUTO_OPEN property in the BillingInsight.properties file is set to N. To configure the Wallet to open automatically upon restarting the instance, change the property value to Y.

This task is a step in "Process of Implementing TDE Column Encryption" on page 33 and in "Process of Implementing TDE Tablespace Encryption" on page 38.

To open the Oracle Wallet using the Ant Encryption menu

1. Change to the directory of the Oracle Billing Insight database installation files. In the directory, *EDX_HOME* is the location where you installed Oracle Billing Insight.
   - **UNIX.** *EDX_HOME*/db/oracle
   - **Windows.** *EDX_HOME*\db\oracle
2. Enter Ant.

   By default, the Ant command runs the build.xml file in the current directory.
3. Select Option 2, Setup Database Encryption.
4. Select Option 2, Open Oracle Wallet.

NOTE: If you shut down an encrypted database (a shutdown abort), then the following error message might occur when restarting the database without opening the Wallet. This error can occur because during recovery, background processes might require access to encrypted redo and undo logs.

ORA-28365: wallet is not open

ORA-28365: wallet is not open

ORA-00600: internal error code, arguments: [kcrp_init_1], [], [], [], [], [],

[], []

The Wallet must be open before opening the database:

SQL> startup mount;

SQL> alter system set wallet open identified by "*example*";

SQL> alter database open;

Process of Implementing TDE Tablespace Encryption

By default, TDE Tablespace Encryption is disabled in Oracle Billing Insight. To enable TDE Tablespace Encryption, you must specify the encryption properties in BillingInsight.properties file when setting the configuration values for use by Ant script that creates the Oracle Billing Insight database.

When you enable TDE Tablespace Encryption, the Oracle Billing Insight database installation script encrypts the following OLTP tablespaces by default:

- EDX_DATA
- EDX_FS_DATA

- EDX_PWC_DATA
- EDX_APP_DATA

You can optionally identify any additional tablespaces you want to encrypt in two SQL files. No OLAP tablespaces are encrypted by default.

This process is a step in "Roadmap for Configuring the Oracle Billing Insight Database" on page 23.

To set up TDE Tablespace Encryption, perform the following tasks:

1. Follow the steps in "Specifying the Oracle Wallet Location" on page 36.
2. For other tablespaces you want to encrypt, in the setuptablespaces.sql file for the OLTP and OLAP schemas, change the encrypted field value from FALSE to TRUE in the corresponding insert statements. The setuptablespaces.sql files are located in the following directories:
   - **OLTP Database**:
     - **UNIX.** *EDX_HOME*/db/oracle/oltp/setuptablespaces.sql
     - **Windows.** *EDX_HOME*\db\oracle\oltp\setuptablespaces.sql
   - **OLAP Database**:
     - **UNIX.** *EDX_HOME*/db/oracle/olap/setuptablespaces.sql
     - **Windows.** *EDX_HOME%*\db\oracle\olap\setuptablespaces.sql

   In the following example of an insert statement, EDX_REPORT_IDX is the tablespace name. In the last two field values, FALSE and 3DES168, FALSE means this tablespace will not be encrypted by default, and 3DES168 is the encryption algorithm used for the Tablespace Encryption. Change the encrypted field value from FALSE to TRUE to enable encryption for this tablespace.

   ```
   :insert into "ebill_tablespace" (name, location, tablespace_size, auto_extend,
   auto_allocate, uniform_size, encrypted, encrypt_algorithm)

   value ('EDX_REPORT_IDX', '$L_DB_EDX_INDX_TB_FILE_LOC/edx_report_idx_01.dbf',
   '28M', 'TRUE', 'FALSE', '1m', 'FALSE', '3DES168');
   ```

3. Follow the steps in "Creating the Oracle Billing Insight Database Using Ant" on page 26 to create the Oracle Billing Insight database. You must set the encryption properties as follows when configuring the BillingInsight.properties file in those steps.

Encryption Property	Description
ENCRYPTION_WALLET_LOCN	Specify the location of the Oracle Wallet folder. Use the same location that you set in the sqlnet.ora file.
WALLET_AUTO_OPEN	Specify whether the Oracle wallet opens automatically when the database instance is restarted, or whether you must open it manually. Valid values are Y or N. It is recommended to set the value to Y.
TDE_ENCRYPT_OLAPCOLUMN	Set the value to N to suppress column-level encryption in the OLAP schema.
TDE_ENCRYPT_OLAPTSPACE	Set the value to TRUE to set the Master Encryption Key and encrypt tablespaces in the OLAP schema.

Encryption Property	Description
TDE_ENCRYPT_OLTPCOLUMN	Set the value to N to suppress column-level encryption in the OLTP schema.
TDE_ENCRYPT_OLTPTSPACE	Set the value to TRUE to encrypt tablespaces in the OLTP schema.

4 The Ant script prompts you to create an Oracle Wallet password when it is ready to create the Master Encryption Key. Specify a password for the Oracle Wallet, then enter it again.

The installation script sets the Master Encryption Key for TDE tablespace encryption and creates the encrypted tablespaces as you specified.

Loading Sample Data

You can load sample provisioning data into the Oracle Billing Insight database.

CAUTION: Do not run the HierarchyCopy job on the sample data files.

This task is a step in "Roadmap for Configuring the Oracle Billing Insight Database" on page 23.

To load sample data into your Oracle Billing Insight database

1 Change to the directory of the Oracle Billing Insight database installation files. In the directory, *EDX_HOME* is the location where you installed Oracle Billing Insight.

- **UNIX.** *EDX_HOME*/db/oracle
- **Windows.** *EDX_HOME*\db\oracle

2 Enter Ant to run the build script.

By default, the Ant command runs the build.xml file in the current directory.

3 From the Main Menu, select Option 3, Create Database Schemas and Objects.

4 Select Option 3, Load Sample Data.

5 Choose the sample data for your line of business only:

- Option 1, Load the Telco Sample Data
- Option 2, Load the Utility Sample Data

6 Review all log files for possible errors.

7 Select Option 3, Return to Previous Menu, or select Option Q, Quit.

8 You must remove sample data before going live. For information about the process of purging sample data, see *Administration Guide for Oracle Billing Insight*.

CAUTION: You must remove sample data from your production environment to comply with the Payment Card Industry Data Security Standard.

Enabling Oracle Auditing

You can enable Oracle auditing. You can audit as an administrator viewing the audit trail while it is active, or run standard auditing.

With standard auditing you can choose which tables to audit. It is recommended to audit the following tables, which contain sensitive data:

- EDX_BSL_AUTH_SECPROFILE
- EDX_UMF_SEC_PWD_HISTORY
- CDA_ATTRIBUTES
- USR_PASSWORD_ENTRIES
- EDX_UMF_SEC_VALIDATIONCODE
- PAYMENT_ACCOUNTS
- EDX_PMT_CHK_ACCT_ONETIME

This task is a step in "Roadmap for Configuring the Oracle Billing Insight Database" on page 23.

Enabling Standard Auditing

Follow these steps to enable standard database auditing.

To enable standard auditing

1. Open the init.ora initialization file for the Oracle Billing Insight EBILL database, located in the following directory:
   - **UNIX.** `$ORACLE_HOME/dbs`
   - **Windows.** `%ORACLE_HOME%\database`

   NOTE: If you use SPFILE to store the initialization parameter, then use Database Configuration Assistant (DBCA) to change the value of the audit_trail parameter.
2. Set the Oracle static initialization parameter audit_trail to db, extended.
3. Restart the database instance.
4. Using SQL*Plus, connect to the EBILL database as the OLTP schema user.
5. For each table you want to audit, run the following command. In the command, *Tablename* is the database tablename.

   ```
   audit all on Tablename by access;
   ```
6. Query the system view DBA_AUDIT_TRAIL to view the audit records.

Enabling Auditing by a System Administrator

Follow these steps to enable auditing by a system administrator.

To enable auditing by a system administrator

1. Open the init.ora initialization file for the Oracle Billing Insight EBILL database, located in the following directory:
   - **UNIX.** `$ORACLE_HOME/dbs`
   - **Windows.** `%ORACLE_HOME%\database`

   NOTE: If you use SPFILE to store the initialization parameter, then use Database Configuration Assistant (DBCA) to change the value of the audit_trail parameter.
2. Set the Oracle static initialization parameter audit_sys_operations to true.
3. Restart the database instance.
4. View the audit trail, using the Windows Event Viewer or syslog on UNIX.

Creating a Bootstrap Administrator User for Oracle Billing Insight

You must create a default bootstrap administrator user for Oracle Billing Insight. The bootstrap user is the only user with initial access to the Command Center and Assisted Service applications.

When you run the Command Center application for the first time, the bootstrap user is prompted to enroll (enter the bootstrap user ID and password) and then create a personal system administrator account. You must then create all of the other system administrator users you need for the Command Center using the bootstrap user ID. For details, see *Administration Guide for Oracle Billing Insight*.

When you run the Assisted Service application for the first time, the bootstrap user is prompted to enroll (enter the bootstrap user ID and password), and then create the first system administrator. For details about enrolling the bootstrap user in Assisted Service application, see *Assisted Service Application Guide for Oracle Billing Insight*.

Once you have created all of the system administrators you require in the Command Center, and created the initial administrator in the Assisted Service application, deactivate the bootstrap user ID in these applications to comply with the Payment Card Industry Data Security Standard (PCI DSS). For details on deactivating the bootstrap user, see *Implementation Guide for Oracle Billing Insight*.

NOTE: Preserve the bootstrap administrator ID and password. You can never change them. A database administrator can deactivate and reactivate the default bootstrap user access account. For details, see *Implementation Guide for Oracle Billing Insight*.

To create a bootstrap administrator user for Oracle Billing Insight

1. Go to the directory with the Oracle Database installation files:
   - **UNIX.** *EDX_HOME*`/db/oracle`

- **Windows.** *EDX_HOME*\db\oracle

2 Enter Ant to run the build script.

By default, the Ant command runs the build.xml file in the current directory.

3 From the Main Menu, select Option 4, Create boot strap user.

4 Enter a user name for the bootstrap administrator.

5 Enter a password for the bootstrap administrator. Enter the password again when prompted to confirm.

Configuring the Oracle Billing Insight Database on Oracle RAC

Configuring Oracle Billing Insight on Oracle Real Application Clusters (RAC), or on Oracle Exadata using RAC, requires that you manually create a database instance with Database Configuration Assistant (DBCA).

If you are using Oracle Automatic Storage Management (ASM) as the volume manager and file system for database files, then you must also modify the Ant script before creating the schemas. You do not need to modify the Ant script if you are using a conventional volume manager and file system for database files.

If you are using Oracle RAC One Node and creating a single instance of a database on Oracle Exadata, then you do not have to follow these steps. Use the procedures for UNIX described "Creating the Oracle Billing Insight Database Using Ant" on page 26.

To configure the Oracle Billing Insight database on Oracle RAC

1 Configure the Single Client Access Name (SCAN) Virtual IP address and SCAN listener to connect to the RAC database. For details on how to use SCAN, see *Oracle Grid Infrastructure Installation Guide*.

2 Create the database instance manually, using DBCA. Create an instance for the Self-Service application on each node.

a Enter Ant to run the build script.

By default, the Ant command runs the build.xml file in the current directory.

b Select Option 1, Create a Database Instance Using DBCA.

This step can take anywhere from 20 minutes to 2 hours to complete, depending on the speed of your platform. The script creates a new instance for the Oracle Billing Insight database, and defines the data dictionaries and all system database objects.

NOTE: DBCA automatically creates a RAC Net Service Name for a new instance. You do not have to manually configure the RAC Net Service Name in the tnsnames.ora file.

3 Create unique service names for each RAC node. Follow the instructions for ensuring the availability of service names for Oracle RAC nodes in *Oracle Warehouse Builder Installation and Administration Guide*. The following table shows sample data for each node.

Host	SID	Service	Net Service Name for the Node in the tnsnames.ora File
Hostname1	ebill1	ebill1	ebill1
Hostname2	ebill2	ebill2	ebill2

4 Make sure each RAC Net Service Name exists in the tnsnames.ora file, adding any unique Net Service Names defined for a node.

5 Use SQL*Plus to log on the EBILL instance as sysdba. Run the following scripts individually as the EBILL sysdba user:

```
$ORACLE_HOME/rdbms/admin/userlock.sql

$ORACLE_HOME/rdbms/admin/dbmslock.sql
```

6 If you are using Oracle Automatic Storage Management (ASM), then make the following modifications to the buildtablespaces.xml file, found in the *EDX_HOME*/db/oracle directory. Otherwise, skip this step.

- Remove the following code from the CheckdirOLAP target:

```
<available file="${filepath}" type="dir" property="direxists"/>
<fail unless="direxists"
message="${line.separator}${line.separator}Directory '${filepath}' does not
exists. Create the folder and install again; or try a different path for the
Tablespace"/>
```

- Remove the following code from the CheckdirOLTP target:

```
<available file="${filepath}" type="dir" property="direxists"/>
<fail unless="direxists"
message="${line.separator}${line.separator}Directory '${filepath}' does not
exists. Create the folder and install again; or try a different path for the
Tablespace"/>
```

- In the target SetupTspaceMstKey -> replace -> replacefilter, remove /${OLAP_USER}/data from each place where it appears in the following code:

```
<replacefilter
token="$TDE_ENCRYPT_OLAPTSPACE"
value="${TDE_ENCRYPT_OLAPTSPACE}"/>
<replacefilter
token="$L_DB_EDX_DATA_OLAP_FILE_LOC"
value="${L_DB_EDX_DATA_OLAP_FILE_LOC}/${OLAP_USER}/data"/>
<replacefilter
token="$L_DB_EDX_INDX_OLAP_FILE_LOC"
value="${L_DB_EDX_INDX_OLAP_FILE_LOC}/${OLAP_USER}/data"/>
```

- Also in the target SetupTspaceMstKey -> replace -> replacefilter, remove /${OLTP_USER}/ data from each place where it appears in the following code:

```
<replacefilter
token="$TDE_ENCRYPT_OLTPTSPACE"
value="${TDE_ENCRYPT_OLTPTSPACE}"/>
<replacefilter
token="$L_DB_EDX_DATA_OLTP_FILE_LOC"
value="${L_DB_EDX_DATA_OLTP_FILE_LOC}/${EBILL_SID}/${OLTP_USER}/data"/>
<replacefilter
token="$L_DB_APP_DATA_OLTP_FILE_LOC"
value="${L_DB_APP_DATA_OLTP_FILE_LOC}/${EBILL_SID}/${OLTP_USER}/data"/>
<replacefilter
token="$L_DB_EDX_INDX_OLTP_FILE_LOC"
value="${L_DB_EDX_INDX_OLTP_FILE_LOC}/${EBILL_SID}/${OLTP_USER}/data"/>
<replacefilter
token="$L_DB_APP_INDX_OLTP_FILE_LOC"
value="${L_DB_APP_INDX_OLTP_FILE_LOC}/${EBILL_SID}/${OLTP_USER}/data"/>
<replacefilter
token="$L_DB_LOAD_DATA_OLTP_FILE_LOC"
value="${L_DB_LOAD_DATA_OLTP_FILE_LOC}/${EBILL_SID}/${OLTP_USER}/data"/>
<replacefilter
token="$L_DB_LOAD_INDX_OLTP_FILE_LOC"
value="${L_DB_LOAD_INDX_OLTP_FILE_LOC}/${EBILL_SID}/${OLTP_USER}/data"/>
<replacefilter
token="$L_DB_FS_DATA_OLTP_FILE_LOC"
value="${L_DB_FS_DATA_OLTP_FILE_LOC}/${EBILL_SID}/${OLTP_USER}/data"/>
<replacefilter
token="$L_DB_FS_INDX_OLTP_FILE_LOC"
value="${L_DB_FS_INDX_OLTP_FILE_LOC}/${EBILL_SID}/${OLTP_USER}/data"/>
<replacefilter
```

```
token="$L_DB_STG_DATA_OLTP_FILE_LOC"
value="${L_DB_STG_DATA_OLTP_FILE_LOC}/${EBILL_SID}/${OLTP_USER}/data"/>
<replacefilter
token="$L_DB_STG_INDX_OLTP_FILE_LOC"
value="${L_DB_STG_INDX_OLTP_FILE_LOC}/${EBILL_SID}/${OLTP_USER}/data"/>
<replacefilter
token="$LARGE_DB_EDX_DATA_OLTP_FILE_LOC"
value="${LARGE_DB_EDX_DATA_OLTP_FILE_LOC}/${EBILL_SID}/${OLTP_USER}/data"/>
<replacefilter
token="$LARGE_DB_EDX_INDX_OLTP_FILE_LOC"
value="${LARGE_DB_EDX_INDX_OLTP_FILE_LOC}/${EBILL_SID}/${OLTP_USER}/data"/>
<replacefilter
token="$MEDIUM_DB_EDX_DATA_OLTP_FILE_LOC"
value="${MEDIUM_DB_EDX_DATA_OLTP_FILE_LOC}/${EBILL_SID}/${OLTP_USER}/data"/>
<replacefilter
token="$MEDIUM_DB_EDX_INDX_OLTP_FILE_LOC"
value="${MEDIUM_DB_EDX_INDX_OLTP_FILE_LOC}/${EBILL_SID}/${OLTP_USER}/data"/>
```

7 Make the following modifications to the Ant script OLAP property file, BillingInsight.properties:

- Set the SID for your software database server in the EBILL_CDB property, such as EBILL_CDB=ebill1.
- Set all DB_FILE_LOC properties to point to a shared file system location.
- If you are using ASM, then set all DB_FILE_LOC properties to this format: DISK_GROUP/EBILL_TNS/datafile. For example: L_DB_EDX_DATA_OLAP_FILE_LOC=+DATA/ebill/olap/datafile. Specify a local directory value for the property TRACE_FILE_LOCN, for example: TRACE_FILE_LOCN=/export/oracle/ebill.

8 Run the modified Ant script, using the install-existing option. Use either of the following procedures:

- "Creating the Oracle Billing Insight Database Using Ant" on page 26
- "Creating the Oracle Billing Insight Database Using the Automated Ant Target" on page 31

9 Follow the steps in "Configuring the ODI Data Load Processes" on page 107.

4 Configuring Oracle WebLogic

This chapter describes how to configure Oracle WebLogic for Oracle Billing Insight. It includes the following topics:

Roadmap for Configuring Oracle WebLogic for Oracle Billing Insight

You must configure Oracle WebLogic to use each Oracle Billing Insight application.

This roadmap is part of "Roadmap for Installing Oracle Billing Insight 7.0" on page 11.

To configure Oracle WebLogic for Oracle Billing Insight, perform the following processes and tasks:

1. "Preparing to Configure Oracle WebLogic" on page 48
2. Follow the process of configuring Oracle WebLogic for the Self-Service application appropriate for your implementation:
   - **On a single computer.** "Process of Configuring Oracle WebLogic for the Self-Service Application" on page 49
   - **On multiple computers with Oracle WebLogic Clustering.** "Process of Configuring the Self-Service Application on an Oracle WebLogic Cluster" on page 61
3. "Process of Configuring Oracle WebLogic for the Command Center Application" on page 68
4. "Process of Configuring Oracle WebLogic for the Assisted Service Application" on page 77
5. Process of Configuring Oracle WebLogic for RESTful Web Services Server Application on page 84

6 "Process of Repackaging the GNU Lesser General Public License" on page 91

7 "Process of Deploying Oracle Billing Insight Applications on Oracle WebLogic" on page 95

8 If you are configuring the Self-Service application to run on an Oracle WebLogic cluster, then follow the steps in "Starting the Oracle WebLogic Cluster" on page 99.

9 "Configuring and Starting Scheduler on Oracle WebLogic" on page 100

10 "Running the Sample Oracle Billing Insight Applications on Oracle WebLogic" on page 102

Preparing to Configure Oracle WebLogic

Before configuring Oracle WebLogic, you must complete the steps described in this topic.

This task is a step in "Roadmap for Configuring Oracle WebLogic for Oracle Billing Insight" on page 47.

To prepare to configure Oracle WebLogic

1 Verify that the database server components are installed and configured for Oracle Billing Insight.

2 For distributed environments, verify that you have any required database client software installed on the Oracle WebLogic application server and any other client computers of your database server.

3 Start the Oracle WebLogic Administration Console.

   If you cannot start the Administration Console, then you will be unable to proceed with configuring your application server for Oracle Billing Insight.

The instructions to configure Oracle WebLogic assume in-depth understanding of and practical experience with application server administration. Consult the Oracle WebLogic documentation as necessary.

NOTE: The installation and configuration examples in this guide use default Oracle Billing Insight paths, privileges and permissions. If you choose not to accept the default values, then make sure your values are consistent in all servers for your installation of Oracle Billing Insight.

Process of Configuring Oracle WebLogic for the Self-Service Application

This topic describes the process of configuring Oracle WebLogic for the Self-Service application.

If you are configuring the Self-Service application on an Oracle WebLogic cluster, then do not follow this process. Instead, follow "Process of Configuring the Self-Service Application on an Oracle WebLogic Cluster" on page 61.

This process is a step in "Roadmap for Configuring Oracle WebLogic for Oracle Billing Insight" on page 47.

To configure Oracle WebLogic for the Self-Service application, perform the following tasks:

1. "Creating the Oracle WebLogic Domain for the Self-Service Application" on page 49
2. "Defining the Oracle WebLogic Environment for the Self-Service Application Domain" on page 50
3. "Enabling HTTPS on Your Server for the Self-Service Application" on page 52
4. "Configuring JDBC Resources for the Self-Service Application" on page 53
5. "Setting the Mail Server Properties for the Self-Service Application" on page 55
6. "Setting the Global Configuration Properties for the Self-Service Application" on page 57
7. "Setting up Dynamic CSS and Image File Load Functionality" on page 59
8. "Setting Up Prenote Functionality" on page 60

Creating the Oracle WebLogic Domain for the Self-Service Application

You must create an Oracle WebLogic domain for the Self-Service application EAR file.

This task is a step in "Process of Configuring Oracle WebLogic for the Self-Service Application" on page 49.

To create an Oracle WebLogic domain for the Self-Service application EAR file

1. Go to the following directory:
   - **UNIX.** `WL_HOME/common/bin`
   - **Windows.** `WL_HOME\common\bin`
2. Run the following command:
   - **UNIX.** config.sh
   - **Windows.** config.cmd
3. On the Oracle WebLogic Configuration Wizard, select Create a new domain, then browse or enter the Domain Location. Click Next.

4 Leave the option to Create Domain Using Product Templates, accept the default selections, and then click Next.

5 Enter the user name and password of the user to administer the Oracle WebLogic domain. Confirm the password, and click Next.

6 Select the SUNJDK to use for this domain, and click Next.

7 In Advanced Configurations, select Administration Server, and then click Next.

8 Enter the Listen Port, such as port 7001, and click Next.

9 Review the Configuration Summary, and click Create.

Defining the Oracle WebLogic Environment for the Self-Service Application Domain

You must set environment variables and other options in the Oracle WebLogic environment to correctly set up the Self-Service domain.

This task is a step in "Process of Configuring Oracle WebLogic for the Self-Service Application" on page 49.

To set environment variables for the Self-Service application domain

1 Open the setDomainEnv file in a text editor.

This file is located in the domain's home directory, for example:

- **UNIX.** `WL_HOME/user_projects/domains/BillingInsight_domain/bin/setDomainEnv.sh`
- **Windows.** `%WL_HOME\user_projects\domains\BillingInsight_domain\bin\setDomainEnv.cmd`

2 In the setDomainEnv.sh file, define the environment variable EDX_HOME as the directory in which Oracle Billing Insight is installed, for example:

- **Oracle Solaris.** `export EDX_HOME=/opt/Oracle/BillingInsight`
- **Linux.** `export EDX_HOME=/opt/Oracle/BillingInsight`
- **Windows.** `set EDX_HOME=C:\oracle\BillingInsight`

3 Add the following entries to the file:

- **Oracle Solaris.** `export CLASSPATH=$CLASSPATH:$EDX_HOME/config:$EDX_HOME/config/resourcebundle:$EDX_HOME/lib/xercesImpl-2.7.1.jar:$EDX_HOME/lib/xalan-2.7.1.jar:$EDX_HOME/lib/serializer-2.7.1.jar`
- **Linux.** `CLASSPATH=$CLASSPATH:$EDX_HOME/config:$EDX_HOME/config/resourcebundle:$EDX_HOME/lib/xercesImpl-2.7.1.jar:$EDX_HOME/lib/xalan-2.7.1.jar:$EDX_HOME/lib/serializer-2.7.1.jar`

- **Windows.** set `CLASSPATH=%CLASSPATH%;%EDX_HOME%\config;%EDX_HOME%\config\resourcebundle;%EDX_HOME%\lib\xercesImpl-2.7.1.jar;%EDX_HOME%\lib\xalan-2.7.1.jar;%EDX_HOME%\lib\serializer-2.7.1.jar`

4 If you are using Chase Paymentech Orbital Payment Gateway, then add the following entries to the setDomainEnv file. Define the environment variable PAYMENTECH_HOME as the directory where Chase Paymentech configuration files are stored:

- **Oracle Solaris.** `export PAYMENTECH_HOME=$EDX_HOME/payment/paymentech`
- **Linux.** `export PAYMENTECH_HOME=$EDX_HOME/payment/paymentech`
- **Windows.** `set PAYMENTECH_HOME=%EDX_HOME%\payment\paymentech`

Set the config directory under PAYMENTECH_HOME in CLASSPATH:

- **Oracle Solaris.** `export CLASSPATH=$CLASSPATH:$PAYMENTECH_HOME/config`
- **Linux.** `export CLASSPATH=$CLASSPATH:$PAYMENTECH_HOME/config`
- **Windows.** `set CLASSPATH=%CLASSPATH%;%PAYMENTECH_HOME%\config`

5 In the JAVA OPTIONS section, add the Dedx.home Java option section to the end of the definition:

- **Oracle Solaris**. Add the Java as shown:

```
JAVA_VM="${JAVA_VM} ${JAVA_DEBUG} ${JAVA_PROFILE} -Dedx.home=${EDX_HOME} -
Dlog4j.configuration=file:${EDX_HOME}/config/log4j.xml -
Djavax.xml.parsers.SAXParserFactory=org.apache.xerces.jaxp.SAXParserFactoryI
mpl -
Djavax.xml.parsers.DocumentBuilderFactory=org.apache.xerces.jaxp.DocumentBui
lderFactoryImpl -Dorg.owasp.esapi.resources=${EDX_HOME}/config -
Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.Transform
erFactoryImpl"

export JAVA_VM
```

- **Linux**. Add the Java as shown:

```
JAVA_VM="${JAVA_VM} ${JAVA_DEBUG} -Dedx.home=${EDX_HOME} -
Dlog4j.configuration=file:${EDX_HOME}/config/log4j.xml -
Djavax.xml.parsers.SAXParserFactory=org.apache.xerces.jaxp.SAXParserFactoryI
mpl -
Djavax.xml.parsers.DocumentBuilderFactory=org.apache.xerces.jaxp.DocumentBui
lderFactoryImpl -Dorg.owasp.esapi.resources=${EDX_HOME}/config -
Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.Transform
erFactoryImpl"

export JAVA_VM
```

- **Windows.** Add the Java as shown. (The backslashes (\) in the following statement are correct.)

```
set JAVA_VM=%JAVA_VM% %JAVA_DEBUG% %JAVA_PROFILE% -Dedx.home=%EDX_HOME% -
Dlog4j.configuration=file:\\\%EDX_HOME%\config\log4j.xml -
Djavax.xml.parsers.SAXParserFactory=org.apache.xerces.jaxp.SAXParserFactoryI
mpl -
Djavax.xml.parsers.DocumentBuilderFactory=org.apache.xerces.jaxp.DocumentBui
```

```
lderFactoryImpl -Dorg.owasp.esapi.resources=%EDX_HOME%\config -
Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.Transform
erFactoryImpl
```

6 Specify the memory settings as follows:

```
MEM_MAX_PERM_SIZE_64BIT="-XX:MaxPermSize=1024m
```

7 If you are using Chase Paymentech Orbital Payment Gateway, then in the JAVA OPTIONS section, add the following section to the end of the definition:

- **Oracle Solaris.** `export JAVA_VM=$JAVA_VM -DPAYMENTECH_HOME=$PAYMENTECH_HOME`
- **Linux.** `export JAVA_VM=$JAVA_VM -DPAYMENTECH_HOME=$PAYMENTECH_HOME`
- **Windows.** `set JAVA_VM=%JAVA_VM% -DPAYMENTECH_HOME=%PAYMENTECH_HOME%`

8 Save the changes, and close the file.

Enabling HTTPS on Your Server for the Self-Service Application

Follow these steps to enable HTTPS on your server for the Self-Service application, required for compliance with the Payment Card Industry Data Security Standard.

This task is a step in "Process of Configuring Oracle WebLogic for the Self-Service Application" on page 49.

To enable HTTPS on your server for the Self-Service application

1 Log in to the Self-Service domain console, for example:

`http://localhost:7001/console/`

where:

- *localhost* is the name of the server where you installed the Self-Service application.
- *7001* is the port number where you installed the Self-Service application.

2 Click your domain name, Environment, and Servers.

3 In the Servers table, click the server where you want to deploy your application.

4 Select Configuration, General tab, and then click SSL Listen Port Enabled. Enter an SSL port number, then click Save.

5 Click Activate Changes to save the changes.

Configuring JDBC Resources for the Self-Service Application

You must configure the following JDBC resources for applications deployed with Oracle WebLogic, using the Oracle WebLogic Administration Console:

- Data sources
- Connection pools

This task is a step in "Process of Configuring Oracle WebLogic for the Self-Service Application" on page 49.

Configuring the Data Sources for the Self-Service Application

You must create the following data sources for the Self-Service application:

- edxAdminDataSource
- edxXMADataSource
- reportDataSource

To create the data sources for the Self-Service application

1. Start the newly created Self-Service domain, and open the Oracle WebLogic Administration Console in a browser. The default URL is

   `http://`*`Server_Name`*`:`*`Server_Port`*`/console`

   where:

   - *Server_Name* is the name of the server with the Self-Service domain.
   - *Server_Port* is the port number of the Self-Service domain server.
2. Log on to the Oracle WebLogic Administration Console, using the user name and password defined when the domain was created.
3. Click the Data Sources link under JDBC in the services section.
4. Click New to create a new data source.
5. Enter the data source name and the JNDI name.

6 For each data source, select the Database Type as Oracle, and enter other settings as shown in the following table. Be sure to select the correct version of the driver for your database version, including patches. Click Next.

Data Source Name	JNDI Name	Database and Driver	Settings	Database Details
edxAdminDataSource	edx.databasePool	Oracle Driver (Thin)	Emulate Two-Phase Commit	OLTP/ Hostname: ListenPort:SID
edxXMADataSource	edx.xma.databasePool	Oracle Driver (Thin XA)	None	OLTP/ Hostname: ListenPort:SID
reportDataSource	edx.report.databasePool	Oracle Driver (Thin XA)	None	OLAP/ Hostname: ListenPort:SID

7 Leave the default transaction options unchanged, then click Next.

8 For Connection Properties, provide the correct values.

Connection Property	Value
Database Name	Your Oracle Database Alias Name
Host Name	Your Oracle Database server host name
Port	DB server listening port
Database User Name	Your Oracle Database User Name
Password	Your Oracle Database Password

9 Click Next, then click Test Configuration to test whether the database connection is configured correctly.

10 Click Next, then target the data source to the server where you want to deploy the application. The default is AdminServer. Click Finish.

11 Click Activate Changes to save the changes.

12 Repeat from Step 4 to create each remaining data source.

Configuring the Connection Pools for the Self-Service Application

You must configure a connection pool for each data source.

To configure the connection pools for the Self-Service application

1. Go to the Summary of JDBC Data Sources.
2. Click any data source name link.
3. On the Connection Pool tab, enter the values for the connection pool settings for each JDBC data source you created.

Property	Value
Initial Capacity	5
Maximum Capacity	20
Capacity Increment	5
Statement Cache Type	FIXED
Statement Cache Size	300
Test Connections on Reserve	Checked
Test Frequency	120
Test Table Name	DUAL
Shrink Frequency	15
Login Delay	1

Setting the Mail Server Properties for the Self-Service Application

You must configure the notification mail server properties file, globalConfig.xma.xml, with the mail host, message transport protocol, and mail account authentication properties for your organization.

This task is a step in "Process of Configuring Oracle WebLogic for the Self-Service Application" on page 49.

To set the notification mail server properties

1. Open the globalConfig.xma.xml file, located in the following directory:
   - **UNIX.** *EDX_HOME*/xma/config/modules
   - **Windows.** *EDX_HOME*\xma\config\modules
2. Find the mailProperties property. Modify the following:
   - **mail.host**. Specify a fully qualified IP address or name of a host running the SMTP which can be used to send email.

- **mail.transport.protocol**. Specify the default message transport protocol.

3 If your company mail server requires mail account authentication, then set the following properties. If not, then set the mail.smtp.auth property to false, or remove the three properties:

- **mail.smtp.auth**. If the value is true, then your mail server attempts to authenticate the user. Set a mail account registered in your mail server.
- **mail.user.** If mail.smtp.auth is true, then set the user name to use when connecting to the mail server.
- **mail.password.** If mail.smtp.auth is true, then set the user password to use when connecting to the mail server.

4 If your company mail server requires SSL connection, then set the mail.smtp.socketFactory.class property. If not, then remove the following property:

```
"mail.smtp.socketFactory.class: If set, specifies the name of a class that
implements the javax.net.SocketFactory interface. This class will be used to
create SMTP sockets. For SSL connection, please set to
javax.net.ssl.SSLSocketFactory"
```

Example of the mail properties:

```
<property name="mailProperties">
   <props>
    <!-- For non authentication mail server
    <prop key="mail.host">mail.example.com</prop>
    <prop key="mail.transport.protocol">SMTP</prop>
    -->
    <prop key="mail.host">stbeehive.example.com</prop>
    <prop key="mail.transport.protocol">SMTP</prop>
    <!-- For requiring authentication mail server -->
    <prop key="mail.smtp.auth">true</prop>
    <prop key="mail.user">eBillingAdmin_WW@example.com</prop>
    <prop key="mail.password">Password</prop>
    <!-- For SSL connection mail server-->
    <prop key="mail.smtp.socketFactory.class">javax.net.ssl.SSLSocketFactory
    </prop>
    <prop key="mail.smtp.socketFactory.port">465</prop>
    </props>
 </property>
<property name="messageFrom"><value>admin@example.com</value></property>
```

5 If your company mail server does not use the SMTP default port 25, then set the mail.smtp.socketFactory.port property, indicating which port to use with the specified socket factory.

Setting the Global Configuration Properties for the Self-Service Application

You must modify various properties in the global configuration file, globalConfig.xma.xml, including server IP addresses, names, and ports for the Self-Service and Assisted Service applications.

This task is a step in "Process of Configuring Oracle WebLogic for the Self-Service Application" on page 49.

To modify the global configuration properties for the Self-Service application

1 Open the globalConfig.xma.xml file, located in the following directory:

- **UNIX.** *EDX_HOME*/xma/config/modules
- **Windows.** *EDX_HOME*\xma\config\modules

2 Find the bean ID called globalConfig. Modify the following properties.

Property	Value
ebillingHostName	The correct server name or IP address where the Self-Service application is deployed.
csrHostName	The correct server name or IP address where the Assisted Service application is deployed.
ebillingApplicationName	The Self-Service application name.
csrApplicationName	The Assisted Service application name.
ebillingHttpPort	The Self-Service application HTTP port.
ebillingSSLPort	The Self-Service application SSL port.
csrhttpPort	The Assisted Service application HTTP port.
csrSSLPort	The Assisted Service application SSL port.
ebillingPortalName	The mobile portal application name, if any.
emailImgSrc	Email notifications use HTML templates. The property value is an HTML URL referencing an image located on the Self-Service application server, such as https://www.XXX.com/selfservice/_assets/swan/headerBg.jpg.
emailHtmlCSS	The location of the CSS file for formatting email HTML content.
unMaskedLength	The string length, which is not masked in the notification template.
maskSymbol	The symbol used to mask the string in the notification template.
ebillingHomePageURL	The correct URL of the Self-Service application home page.
csrHomePageURL	The correct URL of the Assisted Service application home page.

The content of the globalConfig bean is:

```
<beans>
 <bean id="globalConfig"
    class="com.edocs.common.configuration.core.GlobalConfig"
    scope="singleton">
 <property name="encryptAccountNumbers"><value>true</value></property>
    <!-- default value -->
 <property name="ebillingHostName"><value>localhost</value></property>
 <property name="csrHostName"><value>localhost</value></property>
 <property name="rsWebServiceHostName"><value>localhost</value></property>
 <property name="ebillingApplicationName"><value>ebilling</value></property>
 <property name="csrApplicationName"><value>ebillingcsr</value></property>
 <property name="rsApplicationName"><value>ebillingrs</value></property>
 <property name="ebillingHttpPort"><value>7001</value></property>
 <property name="ebillingSSLPort"><value>7002</value></property>
 <property name="csrhttpPort"><value>7005</value></property>
```

```
<property name="csrSSLPort"><value>7006</value></property>
<property name="rsWebServicePort"><value>7017</value></property>
<property name="rsWebServiceSSLPort"><value>7018</value></property>
<property name="rsSSLHostEnabled"><value>false</value></property>
<property name="ebillingPortalName"><value>ebillingPortal.portal</value>
</property>
<property name="emailImgSrc"><value>http://localhost:7001/selfservice/_assets
/swan/headerBg.jpg</value></property>
<property name="emailHtmlCSS"><value>notification/util/email.css</value>
</property>
<property name="unMaskedLength"><value>4</value></property>
<property name="maskSymbol"><value>x</value></property>
<!-- for sso setting -->
<property name="ebillingSingleSignOnEnabled"><value>false</value></property>
<property name="csrSingleSignOnEnabled"><value>false</value></property>
<property name="ebillingWebServiceSingleSignOnEnabled"><value>false</value>
</property>
<!--
<property name="singleSignOutUrl"><value>j_spring_cas_security_logout
</value></property>
-->
<!-- The link of Home in the page header
<property name="ebillingHomePageUrl"><value>http://localhost:7001/portlet
</value></property>
<property name="csrHomePageUrl"><value>http://localhost:7001/portlet</value>
</property>
-->
<property name="autoDetectExternalResourcePeriod"><value>5</value></property>
</bean>
</beans>
```

Setting up Dynamic CSS and Image File Load Functionality

Follow these steps to set up dynamic Cascading Spreadsheet (CSS) and image loading functionality. This feature lets you edit CSS and image files used in the Self-Service application without having to restart the application server.

This task is a step in "Process of Configuring Oracle WebLogic for the Self-Service Application" on page 49.

To set up the dynamic CSS and image file load feature

1. Rename the following directory:
   - **UNIX.** Change *EDX_HOME*/config/assets to *EDX_HOME*/config/_assets
   - **Windows.** Change *EDX_HOME*\config\assets to *EDX_HOME*\config_assets
2. Open the weblogic.xml file, located in the following directory:

- **UNIX.** *EDX_HOME*/J2EEApps/selfservice/weblogic/selfservice-weblogic-7.0.ear/selfservice-web-1.0-SNAPSHOT.war/WEB-INF
- **Windows.** *EDX_HOME*\J2EEApps\selfservice\weblogic\selfservice-weblogic-7.0.ear\selfservice-web-1.0-SNAPSHOT.war\WEB-INF

Add the following code under the XML node <weblogic-web-app>:

```
<virtual-directory-mapping>

    <local-path>[your $edx_home absolute path]/config</local-path>

    <url-pattern>/*</url-pattern>

</virtual-directory-mapping>
```

3 Restart the Self-Service application server.

You can edit CSS and image files in the following directory at any time, without restarting the server. The changes are implemented dynamically.

- **UNIX.** *EDX_HOME*/config/_assets
- **Windows.** *EDX_HOME*\config_assets

Setting Up Prenote Functionality

Use the following instructions to set up prenote functionality.

This task is a step in "Process of Configuring Oracle WebLogic for the Self-Service Application" on page 49.

To set up prenote functionality

1 Open the payment.xma.xml file, located in the following directory:

- **UNIX.** *EDX_HOME*\xma\config\modules\payment\file
- **Windows.** *EDX_HOME*/xma/config/modules/payment/file

2 Find the paymentConfigurationBean bean definition, and set the enablePreNote property value as follows:

```
<property name="enablePreNote">

            <value>Yes</value>

      </property>
```

Process of Configuring the Self-Service Application on an Oracle WebLogic Cluster

This topic describes the process of configuring the Oracle Billing Insight Self-Service application on an Oracle WebLogic cluster.

If you are configuring the Self-Service application on a single computer (nonclustered environment), then do not follow these instructions. Instead, see "Process of Configuring Oracle WebLogic for the Command Center Application" on page 68.

This process is a step in "Roadmap for Configuring Oracle WebLogic for Oracle Billing Insight" on page 47.

To configure the Self-Service application on an Oracle WebLogic cluster, perform the following tasks:

1. "Creating the Oracle WebLogic Domain and Managed Servers for the Self-Service Application on a Cluster" on page 61
2. "Defining the Oracle WebLogic Cluster Environment for the Self-Service Application" on page 64
3. "Enabling HTTPS on Your Server for the Self-Service Application" on page 52
4. "Configuring JDBC Resources for the Self-Service Application on an Oracle WebLogic Cluster" on page 65
5. "Setting the Mail Server Properties for the Self-Service Application" on page 55
6. "Setting the Global Configuration Properties for the Self-Service Application" on page 57
7. "Setting Up Prenote Functionality" on page 60

Creating the Oracle WebLogic Domain and Managed Servers for the Self-Service Application on a Cluster

To implement the Self-Service application on an Oracle WebLogic cluster, you must create a domain and admin server on the computer where you installed the Self-Service application. You must also create one or more managed servers on this computer and on the other computers in the cluster.

Table 5 shows an example of the entities created for the Self-Service application in a clustered environment.

Table 5. Example of a Cluster Environment for the Self-Service Application

Server	Associated Computer	Server Name	IP Address	Port
Admin Server	Computer 1	AdminServer	10.240.12.163	7001
Proxy Server	Computer 1	BillingInsightProxy	10.240.12.163	9007
Managed Server 1	Computer 1	BillingInsightMS1	10.240.12.163	9001

Table 5. Example of a Cluster Environment for the Self-Service Application

Server	Associated Computer	Server Name	IP Address	Port
Managed Server 2	Computer 2	BillingInsightMS2	10.240.8.244	9002
Managed Server 3	Computer 3	BillingInsightMS3	10.240.12.157	9003

This task is a step in "Process of Configuring the Self-Service Application on an Oracle WebLogic Cluster" on page 61.

To create a domain and servers for the Self-Service application on a cluster

1. On the computer where you installed the Self-Service application, create a domain for the Self-Service EAR file:
   a. Go to the following directory:
      - **UNIX.** `WL_HOME/common/bin`
      - **Windows.** `WL_HOME\common\bin`
   b. Run the following command:
      - **UNIX.** `config.sh`
      - **Windows.** `config.cmd`
   c. On the Oracle WebLogic Configuration Wizard, select Create a new domain, then browse or enter the Domain Location. Click Next.
   d. Leave the option to Create Domain Using Product Templates, accept the default selections, and then click Next.
   e. Enter the user name and password of the user to administer the Oracle WebLogic domain. Confirm the password, and click Next.
   f. Select the SUNJDK to use for this domain, and click Next.
   g. In Advanced Configurations, select Administration Server, and then click Next.
   h. Enter the Listen Port, such as port 7001, and click Next.
   i. Review the Configuration Summary, and click Create.
2. Start the domain, using the following command:

   `WL_HOME/common/bin/startWebLogic.sh`
3. Create a managed server:
   a. Log in to the Oracle WebLogic administration console, for example:

   `http://localhost:7001/console/`

   where:

- *localhost* is the host name of the server where you installed the Self-Service application. The host name can be the Domain Name System (DNS) name or the IP address, such as 10.240.12.163.
- *7001* is the port number where you installed the Self-Service application.

b Click Environment, Servers, and then New.

c Enter a name for the managed server, such as BillingInsightMS1.

d Specify the server listen address (the IP address of the managed server).

e Enter the server listen port, such as 9001.

f Select the Standalone Server option (not the Clustering option).

g Click Next, and click Finish.

If successful, then the following messages appear:

All changes have been activated. However, 1 items must be restarted for the changes to take effect.

Server created successfully.

4 Stop the domain, using the following command:

```
$WL_HOME/common/bin/stopWebLogic.sh
```

5 Configure the Node Manager for the computer:

a Make sure you start the Node Manager at least once. The configuration file, nodemanager.properties, is created automatically when you first start Node Manager.

b Edit the nodemanager.properties file, located in the `WL_HOME/common/nodemanager` directory. Set the following parameter values.

Parameter	Value	Purpose
AuthenticationEnabled	false	Creates a nonauthenticated connection to Node Manager.
SecureListener	false	Creates a plain connection to Node Manager, not a secure connection.

c Start the Node Manager, using the following command:

```
$WL_HOME/server/bin/startNodeManager.sh
```

6 Edit the required scripts:

a Add the following code to the commEnv.sh file, located in the `$WL_HOME/common/bin` directory to set EDX_HOME, the directory where you installed Oracle Billing Insight:

```
export EDX_HOME=/home/oracle/BillingInsight

export CLASSPATH=$CLASSPATH:$EDX_HOME/config:$EDX_HOME/config/
resourcebundle:$EDX_HOME/lib/xercesImpl-2.7.1.jar:$EDX_HOME/lib/xalan-
2.7.1.jar:$EDX_HOME/lib/serializer-2.7.1.jar
```

b Also in the commEnv.sh file, change the MaxPermSize value to 1024 for each platform.

c In the startManagedWebLogic.sh file, located in the `$WL_HOME/common/bin` directory, add the Dlog4j.configuration Java option to the end of the definition in the JAVA option section:

```
export JAVA_VM="${JAVA_VM}  -Dedx.home=${EDX_HOME} -
Dlog4j.configuration=file:${EDX_HOME}/config/log4j.xml -
Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.Transform
erFactoryImpl    -
Djavax.xml.parsers.SAXParserFactory=org.apache.xerces.jaxp.SAXParserFactoryI
mpl -
Djavax.xml.parsers.DocumentBuilderFactory=org.apache.xerces.jaxp.DocumentBui
lderFactoryImpl -Dorg.owasp.esapi.resources=${EDX_HOME}/config"
```

7 Repeat Step 3 through Step 6 as necessary to create and configure additional servers on this computer or other computers in the cluster.

Defining the Oracle WebLogic Cluster Environment for the Self-Service Application

You must configure the Oracle WebLogic cluster environment.

This task is a step in "Process of Configuring the Self-Service Application on an Oracle WebLogic Cluster" on page 61.

To configure the Oracle WebLogic cluster environment

1 Log in to the Oracle WebLogic administrative console on the computer where the admin server is located, using the following URL format:

`http://`*`IP_Address:7001`*`/console`

where:

- *IP_Address* is the name of the computer, such as 10.240.12.163.
- *7001* is the port number where you installed the Self-Service application.

2 Create the Oracle WebLogic cluster:

a Click Environment, Clusters, then New.

b Enter a name for the cluster, such as BillingInsightCluster.

c Select the messaging mode. For best performance, it is recommended to choose Multicast.

d Leave the default values for the other configuration parameters.

e Click OK.

3 Click the name of the cluster you just created. For each computer in the cluster, follow these steps:

a Click Environment, Machines, New, and Next.

b Select Plain as the protocol to use with the node manager on this computer.

c Enter the listen address (IP address) of the computer and the listen port number, such as 5556.

d Click Finish.

e Click the computer that contains the node manager. Click the Monitoring tab, and then click Clusters.

The Monitoring option continuously verifies that Node Manager is accessible.

f Click the name of the cluster you created. Click the Servers tab.

4 For each managed server in the cluster, follow these steps:

a Click Environment, Server, and then New.

b Enter a name for the managed server.

c Specify the server listen address (where the server will listen for incoming connections) and the port number.

d Choose the Stand-Alone cluster.

e Check the Listen Port Enabled option.

f Click Next, and then Finish.

5 Create the Proxy Server, which functions as a load balancer:

a Click Environment, Servers, and then New.

b Specify a name for the proxy server, such as BillingInsightProxy.

c Specify the listen address as the IP address of the computer where the admin server is located, such as 10.240.12.163. Specify the listen port, such as 9007.

d Choose the Standalone cluster.

6 Add each server to the cluster. For each server, perform the following steps:

a Click Clusters, click the name of the cluster, and then click the Servers tab.

b Select the servers to add to the cluster and click Add.

c Click Finish.

Configuring JDBC Resources for the Self-Service Application on an Oracle WebLogic Cluster

You must configure the following JDBC resources for applications deployed with Oracle WebLogic, using the Oracle WebLogic Administration Console:

- Data sources
- Connection pools

This task is a step in "Process of Configuring the Self-Service Application on an Oracle WebLogic Cluster" on page 61.

Configuring the Data Sources for the Self-Service Application on an Oracle WebLogic Cluster

You must create the following data sources for the Self-Service application:

- edxAdminDataSource
- edxXMADataSource
- reportDataSource

To create the data sources for the Self-Service application

1. Start the newly created Self-Service domain, and open the Oracle WebLogic Administration Console in a browser. The default URL is:

   `http://localhost:7001/console/`

   where:

   - *localhost* is the host name of the server where you installed the Self-Service application. The host name can be the Domain Name System (DNS) name or the IP address, such as 10.240.12.157.
   - *7001* is the port number where you installed the Self-Service application.

2. Log on to the Oracle WebLogic Administration Console, using the user name and password defined when the domain was created.
3. Click the Data Sources link under JDBC in the services section.
4. Click New to create a new data source.
5. Enter the data source name and the JNDI name.
6. For each data source, select the Database Type as Oracle, and enter other settings as shown in the following table. Be sure to select the correct version of the driver for your database version, including patches. Click Next.

Data Source Name	JNDI Name	Database and Driver	Settings	Database Details
edxAdminDataSource	edx.databasePool	Oracle Driver (Thin)	Emulate Two-Phase Commit	OLTP/ Hostname: ListenPort:SID
edxXMADataSource	edx.xma.databasePool	Oracle Driver (Thin XA)	None	OLTP/ Hostname: ListenPort:SID
reportDataSource	edx.report.databasePool	Oracle Driver (Thin XA)	None	OLAP/ Hostname: ListenPort:SID

7 Leave the default transaction options unchanged, then click Next.

8 For Connection Properties, provide the correct values.

Connection Property	Value
Database Name	Your Oracle Database Alias Name
Host Name	Your Oracle Database server host name
Port	DB server listening port
Database User Name	Your Oracle Database User Name
Password	Your Oracle Database Password

9 Click Next, then click Test Configuration to test whether the database connection is configured correctly.

10 Click Next. Click Finish.

11 Open each datasource and click Target. For each datasource, follow these steps:

- a Click Summary of JDBC DataSource
- b Select the datasource.
- c Select the Target tab.
- d Click the cluster name, and click Save.

12 Click Activate Changes to save the changes.

13 Repeat from Step 4 to create each remaining data source.

Configuring the Connection Pools for the Self-Service Application on an Oracle WebLogic Cluster

You must configure a connection pool for each data source.

To configure the connection pools for the Self-Service application

1 Go to the Summary of JDBC Data Sources.

2 Click any data source name link.

3 On the Connection Pool tab, enter the values for the connection pool settings for each JDBC data source you created.

Property	Value
Initial Capacity	5
Maximum Capacity	20
Capacity Increment	5

Property	Value
Statement Cache Type	FIXED
Statement Cache Size	300
Test Connections on Reserve	Checked
Test Frequency	120
Test Table Name	DUAL
Shrink Frequency	15
Login Delay	1

Process of Configuring Oracle WebLogic for the Command Center Application

This topic describes the process of configuring Oracle WebLogic for the Command Center application.

This process is a step in "Roadmap for Configuring Oracle WebLogic for Oracle Billing Insight" on page 47.

To configure Oracle WebLogic for the Command Center EAR file, perform the following tasks:

1. "Creating the Oracle WebLogic Domain for Command Center" on page 68
2. "Defining the Oracle WebLogic Environment for the Command Center Domain" on page 69
3. "Enabling HTTPS for the Command Center Server" on page 71
4. "Configuring JDBC Resources for the Command Center" on page 71
5. "Setting the Service Transaction Time (Linux Only)" on page 74
6. "Configuring JMS Resources for the Command Center on Oracle WebLogic" on page 75

Creating the Oracle WebLogic Domain for Command Center

You must create an Oracle WebLogic domain for the Oracle Billing Insight Command Center EAR file.

This task is a step in "Process of Configuring Oracle WebLogic for the Command Center Application" on page 68.

To create an Oracle WebLogic domain for the Command Center

1. Go to the following directory:
   - **UNIX.** `WL_HOME/common/bin`
   - **Windows.** `WL_HOME\common\bin`

2 Run the following command:

- **UNIX.** `config.sh`
- **Windows.** `config.cmd`

3 On the Oracle WebLogic Configuration Wizard, select Create a new domain, then browse or enter the Domain Location. Click Next.

4 Leave the option to Create Domain Using Product Templates, accept the default selections, and then click Next.

5 Enter the user name and password of the user to administer the Oracle WebLogic domain. Confirm the password, and click Next.

6 Select the SUNJDK to use for this domain, and click Next.

7 In Advanced Configurations, select Administration Server, and then click Next.

8 Enter the Listen Port, such as port 7003, and click Next.

9 Review the Configuration Summary, and click Create.

Defining the Oracle WebLogic Environment for the Command Center Domain

You must set environment variables and other options in Oracle WebLogic environment for Command Center.

This task is a step in "Process of Configuring Oracle WebLogic for the Command Center Application" on page 68.

To set environment variables for the Command Center application domain

1 Open the file setDomainEnv in a text editor. This file is located in the domain's home directory, for example:

- **UNIX.** `WL_HOME/user_projects/domains/cc_domain/bin/setDomainEnv.sh`
- **Windows.** `WL_HOME\user_projects\domains\cc_domain\bin\setDomainEnv.cmd`

2 In the file, define the environment variable EDX_HOME as the directory in which Oracle Billing Insight is installed, for example:

- **Oracle Solaris.** `export EDX_HOME=/opt/Oracle/BillingInsight`
- **Linux.** `export EDX_HOME=/opt/Oracle/BillingInsight`
- **Windows.** `set EDX_HOME=C:\oracle\BillingInsight`

3 Add the following entries to the file:

- **Oracle Solaris.** `export CLASSPATH=$CLASSPATH:$EDX_HOME/config:$EDX_HOME/config/resourcebundle:$EDX_HOME/lib/xercesImpl-2.7.1.jar:$EDX_HOME/lib/xalan-2.7.1.jar:$EDX_HOME/lib/serializer-2.7.1.jar`

- **Linux.** `export CLASSPATH=$CLASSPATH:$EDX_HOME/config:$EDX_HOME/config/resourcebundle:$EDX_HOME/lib/xercesImpl-2.7.1.jar:$EDX_HOME/lib/xalan-2.7.1.jar:$EDX_HOME/lib/serializer-2.7.1.jar`
- **Windows.** `set CLASSPATH=%CLASSPATH%;%EDX_HOME%\config;%EDX_HOME%\config\resourcebundle;%EDX_HOME%\lib\xercesImpl-2.7.1.jar;%EDX_HOME%\lib\xalan-2.7.1.jar;%EDX_HOME%\lib\serializer-2.7.1.jar`

4 If you are using Chase Paymentech Orbital Payment Gateway, then add the following entries to the setDomainEnv file. Define the environment variable PAYMENTECH_HOME as the directory where Paymentech configuration files are stored:

- **Oracle Solaris.** `export PAYMENTECH_HOME=$EDX_HOME/payment/paymentech`
- **Linux.** `export PAYMENTECH_HOME=$EDX_HOME/payment/paymentech`
- **Windows.** `set PAYMENTECH_HOME=%EDX_HOME%\payment\paymentech`

Set the config directory under PAYMENTECH_HOME in CLASSPATH:

- **Oracle Solaris.** `export CLASSPATH=$CLASSPATH:$PAYMENTECH_HOME/config`
- **Linux.** `export CLASSPATH=$CLASSPATH:$PAYMENTECH_HOME/config`
- **Windows.** `set CLASSPATH=%CLASSPATH%;%PAYMENTECH_HOME%\config`

5 In the JAVA OPTIONS section, add the `-Dedx.home` Java option section to the end of the JAVA_VM variable definition:

- **Oracle Solaris**. Add the Java as shown:

```
JAVA_VM="${JAVA_VM} ${JAVA_DEBUG} ${JAVA_PROFILE} -Dedx.home=${EDX_HOME} -
Djavax.xml.parsers.SAXParserFactory=org.apache.xerces.jaxp.SAXParserFactoryI
mpl -
Djavax.xml.parsers.DocumentBuilderFactory=org.apache.xerces.jaxp.DocumentBui
lderFactoryImpl

-
Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.Transform
erFactoryImpl"

export JAVA_VM
```

- **Linux**. Add the Java as shown:

```
JAVA_VM="${JAVA_VM} ${JAVA_DEBUG} -Dedx.home=${EDX_HOME} -
Djavax.xml.parsers.SAXParserFactory=org.apache.xerces.jaxp.SAXParserFactoryI
mpl -
Djavax.xml.parsers.DocumentBuilderFactory=org.apache.xerces.jaxp.DocumentBui
lderFactoryImpl -
Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.Transform
erFactoryImpl"

export JAVA_VM
```

- **Windows**. Add the Java as shown:

```
set JAVA_VM=%JAVA_VM% %JAVA_DEBUG% %JAVA_PROFILE% -Dedx.home=%EDX_HOME% -
Djavax.xml.parsers.SAXParserFactory=org.apache.xerces.jaxp.SAXParserFactoryI
mpl -
Djavax.xml.parsers.DocumentBuilderFactory=org.apache.xerces.jaxp.DocumentBui
lderFactoryImpl -
Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.Transform
erFactoryImpl
```

6 If you are using Chase Paymentech Orbital Payment Gateway, then in the JAVA OPTIONS section, add the following section to the end of the definition:

- **Oracle Solaris.** `export JAVA_VM=$JAVA_VM -DPAYMENTECH_HOME=$PAYMENTECH_HOME`
- **Linux.** `export JAVA_VM=$JAVA_VM -DPAYMENTECH_HOME=$PAYMENTECH_HOME`
- **Windows.** `set JAVA_VM=%JAVA_VM% -DPAYMENTECH_HOME=%PAYMENTECH_HOME%`

7 Save the changes, and close the file.

Enabling HTTPS for the Command Center Server

Follow these steps to enable HTTPS on your server for the Command Center application, required for compliance with the Payment Card Industry Data Security Standard.

This task is a step in "Process of Configuring Oracle WebLogic for the Command Center Application" on page 68.

To enable HTTPS on your server for the Command Center application

1 Log in to the Command Center domain console, for example:

`http://localhost:7003/console/`

where:

- *localhost* is the name of the server where you installed the Command Center application.
- *7003* is the port number where you installed the Command Center application.

2 Click your domain name, Environment, and then click Servers.

3 In the Servers table, click the server where you want to deploy your application.

4 Select Configuration, General tab, and then click SSL Listen Port Enabled. Enter an SSL port number.

Configuring JDBC Resources for the Command Center

You must configure JDBC resources for the Command Center application deployed with Oracle WebLogic, using the Oracle WebLogic Administration Console.

You must configure the following:

- Data sources
- Connection pools

This task is a step in "Process of Configuring Oracle WebLogic for the Command Center Application" on page 68.

Configuring Data Sources for the Command Center Domain

You must configure the following data sources for running Oracle Billing Insight jobs and the reporting application:

- edxAdminDataSource
- edxXMADataSource
- reportDataSource
- edxLoggerDataSource
- edxMessagingDataSource
- edxUserDataSource

To create data sources for the Command Center domain

1. Start the newly created Command Center domain, and open the Oracle WebLogic Administration Console in a browser. The default URL

   `http://`*`Server_Name`*`:`*`Server_Port`*`/console`

   where:

   - *Server_Name* is the name of the server with the Command Center domain.
   - *Server_Port* is the port number of the Command Center domain server.
2. Log on to the Oracle WebLogic Administration Console, using the user name and password defined when the domain was created.
3. Click the Data Sources link under JDBC in the services section.
4. Click New to create a new data source.
5. Enter the data source name and the JNDI name.

6 For each data source, select the database type as Oracle and enter other settings as shown in the following table. Be sure to select the correct version of the driver for your database version, including patches. Click Next.

Data Source Name	JNDI Name	Database and Driver	Settings	DB Details
edxAdminDataSource	edx.databasePool	Oracle Driver (Thin)	Emulate Two-Phase Commit	OLTP/ Hostname:Listen Port:SID
edxXMADataSource	edx.xma.database Pool	Oracle Driver (Thin XA)	Not applicable	OLTP/ Hostname:Listen Port:SID
reportDataSource	edx.report.database Pool	Oracle Driver (Thin XA)	Not applicable	OLAP/ Hostname:Listen Port:SID
edxLoggerDataSource	edx.logger.database Pool	Oracle Driver (Thin)	Emulate Two-Phase Commit	OLTP/ Hostname:Listen Port:SID
edxMessagingDataSource	edx.messaging. databasePool	Oracle Driver (Thin XA)	Not applicable	OLTP/ Hostname:Listen Port:SID
edxUserDataSource	edx.user.database Pool	Oracle Driver (Thin)	Emulate Two-Phase Commit	OLTP/ Hostname:Listen Port:SID

7 Select Supports Global Transactions for all Data Sources, select the corresponding transaction option, then click Next.

8 On the Connections Properties page, provide the correct values for the following properties.

Connection Property	Description
Database Name	Your Oracle Database Alias Name
Host Name	Your Oracle Database Server Host Name
Port	Database Server Listening Port
Database User Name	Your Oracle Database User Name
Password	Your Oracle Database Password

9 Click Test Configuration to test whether the database connection is configured correctly. Click Next.

10 Target the data source to the Command Center domain (default is AdminServer), then click Finish.

11 Click Activate Changes to save the changes.

Configuring Connection Pools for the Command Center Domain

You must configure connection pools for each JDBC data source you created.

To configure the connection pools for Command Center

1 Go to the Summary of JDBC Data Sources.

2 Click any data source name link.

3 On the Connection Pool tab, enter the values for the connection pool settings for each JDBC data source you created.

Property	Value
Initial Capacity	5
Maximum Capacity	20
Capacity Increment	5
Statement Cache Type	FIXED
Statement Cache Size	300
Text Connections on Reserve	Selected
Test Frequency	120
Test Table Name	DUAL
Shrink Frequency	15
Login Delay	1

Setting the Service Transaction Time (Linux Only)

Linux users must set the service transaction time.

This task is a step in "Process of Configuring Oracle WebLogic for the Command Center Application" on page 68.

To set the transaction time (Linux only)

1 Select Services, JTA.

2 Update the value of Transaction Timeout from 30 to 60.

3 Save the change.

4 Stop and restart the Command Center domain.

Configuring JMS Resources for the Command Center on Oracle WebLogic

You must configure the following JMS resources for the Command Center:

- JMS persistence store
- JMS servers
- JMS module and connection factory
- JMS queues

This task is a step in "Process of Configuring Oracle WebLogic for the Command Center Application" on page 68.

Configuring a JMS Persistence Store for the Command Center

You must configure a JMS persistence store for the Command Center.

To configure a JMS persistence store

1 Select Services, Persistence Stores.

2 Select New, Create File Store. Create a file store with the following values.

Name	Target	Directory
LoggerFileStore	Select Your Server (AdminServer)	Directory Location to Save Store (such as `WL_HOME/user_projects/domains/commandcenter_domain/config/FileStore`)
EventFileStore	Select Your Server (AdminServer)	Directory Location to Save Store

Configuring a JMS Server

You must set up a JMS server for the Command Center.

To configure a JMS server

- Select Services, Messaging, JMS Servers, and New. Create one JMS Server, using the following values.

Name	Persistent Store	Target
LoggerJMSServer	LoggerFileStore	Select Your Server (AdminServer)
EventJMSServer	EventFileStore	Select Your Server (AdminServer)

Creating the JMS Module and Connection Factory for the Command Center

You must create a JMS module and add a JMS connection factory to the module for the Command Center.

To create the JMS module and connection factory

1. Click Services, Messaging, and the JMS Modules link on the Administration Console page for the Command Center domain.
2. Click New to create a new JMS Module. Set the name of the JMS system Module, and provide the JMS Module properties. Click Next.

Property	Description	Example
Name	Your JMS Module Name	CC_JMSFactories
Descriptor File Name	Your Descriptor File Name	CC_JMSFactories
Location in Domain	Your Location in Domain	CC_JMSFactories
Targets	Select Your Server	AdminServer

3. Select the check boxes to target the system Module at the servers running the Command Center application, click Next, and then click Finish.
4. Click the JMS Modules link on the Administration Console page for the Command Center domain.
5. Click the link for the module you created in the previous procedure.
6. Click New, select Connection Factory, and then click Next.
7. Add one Connection Factory, using the following parameters. Click Finish.

Connection Factory Type	Name	JNDI Name	Target
Connection Factory	LoggerConnectionFactory	edx.lcf	Select Your Server (AdminServer)
Connection Factory	EventConnectionFactory	edx.qcf	Select Your Server (AdminServer)

8 Click the new connection factory, and select XA Transaction on the Transactions tab.

Configuring a JMS Queue for the Command Center

You must create two JMS queues for the Command Center application.

To create the JMS queues

1 Click the JMS Modules link on the Administration Console page for the Command Center domain.

2 Click the link for the module you created. Click New, select Queue, and then click Next.

3 Provide the Queue Name and the JNDI Name as shown in the following table. Click Next.

Name	JNDI Name	Target
LoggerQueue	edx.queue.logger	LoggerJMSServer
EventQueue	edx.queue.outbound	EventJMSServer

4 Click Create a New Subdeployment. Oracle WebLogic shows the newly created queue name as Subdeployment Name.

5 Accept the name, then click OK. If you plan to add more JMS resources to the subdeployment, then you can rename it. Select the new subdeployment and the corresponding target JMS Server.

6 Click Activate Changes to save changes after the JMS configuration is complete.

Process of Configuring Oracle WebLogic for the Assisted Service Application

This topic describes the process of configuring Oracle WebLogic for the Assisted Service application.

This process is a step in "Roadmap for Configuring Oracle WebLogic for Oracle Billing Insight" on page 47.

To configure Oracle WebLogic for the Assisted Service application, perform the following tasks:

1 "Creating the Oracle WebLogic Domain for the Assisted Service Application" on page 78

2 "Defining the Oracle WebLogic Environment for the Assisted Service Application Domain" on page 78

3 "Accessing the Oracle WebLogic Console" on page 80

4 "Enabling HTTPS on Your Server for the Assisted Service Application" on page 81

5 "Configuring JDBC Resources for the Assisted Service Application" on page 81

6 "Modifying the csr.xma.xml File for the Assisted Service Application Properties" on page 84

Creating the Oracle WebLogic Domain for the Assisted Service Application

You must create an Oracle WebLogic domain for the Assisted Service EAR file.

This task is a step in "Process of Configuring Oracle WebLogic for the Assisted Service Application" on page 77.

To create an Oracle WebLogic domain for the Assisted Service EAR file

1. Go to the following directory:
   - **UNIX.** `WL_HOME/common/bin`
   - **Windows.** `WL_HOME\common\bin`
2. Run the following command:
   - **UNIX.** config.sh
   - **Windows.** config.cmd
3. On the Oracle WebLogic Configuration Wizard, select Create a new domain, then browse or enter the Domain Location. Click Next.
4. Leave the option to Create Domain Using Product Templates, accept the default selections, and then click Next.
5. Enter the user name and password of the user to administer the Oracle WebLogic domain. Confirm the password, and click Next.
6. Select the SUNJDK to use for this domain, and click Next.
7. In Advanced Configurations, select Administration Server, and then click Next.
8. Enter the Listen Port, such as port 7005, and click Next.
9. Review the Configuration Summary, and click Create.

Defining the Oracle WebLogic Environment for the Assisted Service Application Domain

You must set environment variables and other options in the Oracle WebLogic environment to correctly set up the Assisted Service application domain.

This task is a step in "Process of Configuring Oracle WebLogic for the Assisted Service Application" on page 77.

To set environment variables for the Oracle WebLogic Assisted Service application domain

1. Go to the following directory:
   - **UNIX.** `WL_HOME/user_projects/domains/csr_domain/bin`

- **Windows.** `WL_HOME\user_projects\domains\csr_domain\bin`

2 Open the following file in a text editor:

- **UNIX.** setDomainEnv.sh
- **Windows.** setDomainEnv.cmd

3 In the file, define the environment variable EDX_HOME as the directory where the Self-Service application is installed, for example:

- **Oracle Solaris.** `export EDX_HOME=/opt/Oracle/BillingInsight`
- **Linux.** `export EDX_HOME=/opt/Oracle/BillingInsight`
- **Windows.** `set EDX_HOME=C:\oracle\BillingInsight`

4 Add the following entries to the file:

- **Oracle Solaris.** `CLASSPATH=$CLASSPATH:$EDX_HOME/config:$EDX_HOME/config/resourcebundle:$EDX_HOME/lib/xalan-2.7.1.jar:$EDX_HOME/lib/serializer-2.7.1.jar:$EDX_HOME/lib/xercesImpl-2.7.1.jar`
- **Linux.** `CLASSPATH=$CLASSPATH:$EDX_HOME/config:$EDX_HOME/config/resourcebundle:$EDX_HOME/lib/xalan-2.7.1.jar:$EDX_HOME/lib/serializer-2.7.1.jar:$EDX_HOME/lib/xercesImpl-2.7.1.jar`
- **Windows.** set `CLASSPATH=%CLASSPATH%;%EDX_HOME%\config;%EDX_HOME%\config\resourcebundle;%EDX_HOME%\lib\xalan-2.7.1.jar;%EDX_HOME%\lib\serializer-2.7.1.jar;%EDX_HOME%\lib\xercesImpl-2.7.1.jar`

5 If you are using Chase Paymentech Orbital Payment Gateway, then add the following entries to the setDomainEnv file. Define the environment variable PAYMENTECH_HOME as the directory where Paymentech configuration files are stored:

- **Oracle Solaris.** `export PAYMENTECH_HOME=$EDX_HOME/payment/paymentech`
- **Linux.** `export PAYMENTECH_HOME=$EDX_HOME/payment/paymentech`
- **Windows.** `set PAYMENTECH_HOME=%EDX_HOME%\payment\paymentech`

Set the config directory under PAYMENTECH_HOME in CLASSPATH:

- **Oracle Solaris.** `export CLASSPATH=$CLASSPATH:$PAYMENTECH_HOME/config`
- **Linux.** `export CLASSPATH=$CLASSPATH:$PAYMENTECH_HOME/config`
- **Windows.** `set CLASSPATH=%CLASSPATH%;%PAYMENTECH_HOME%\config`

6 In the JAVA OPTIONS section, add the Java option section to the end of the JAVA_VM variable definition:

- **Oracle Solaris.** Add the Java as shown:

```
JAVA_VM="${JAVA_VM} ${JAVA_DEBUG} ${JAVA_PROFILE} -Dedx.home=${EDX_HOME} -
Dlog4j.configuration=file:${EDX_HOME}/config/log4j_csr.xml -
Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.TransformerFactoryImpl -
Djavax.xml.parsers.SAXParserFactory=org.apache.xerces.jaxp.SAXParserFactoryImpl -
```

```
Dorg.owasp.esapi.resources=${EDX_HOME}/config -
Djavax.xml.parsers.DocumentBuilderFactory=org.apache.xerces.jaxp.DocumentBui
lderFactoryImpl"

export JAVA_VM
```

- **Linux.** Add the Java as shown:

```
JAVA_VM="${JAVA_VM} ${JAVA_DEBUG} -Dedx.home=${EDX_HOME} -
Dlog4j.configuration=file:${EDX_HOME}/config/log4j_csr.xml -
Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.Transform
erFactoryImpl -
Djavax.xml.parsers.SAXParserFactory=org.apache.xerces.jaxp.SAXParserFactoryI
mpl -
Dorg.owasp.esapi.resources=${EDX_HOME}/config -
Djavax.xml.parsers.DocumentBuilderFactory=org.apache.xerces.jaxp.DocumentBui
lderFactoryImpl"

export JAVA_VM
```

- **Windows.** Add the Java as shown. (The backslashes (\) in the following statement are correct.)

```
set JAVA_VM=%JAVA_VM% %JAVA_DEBUG% %JAVA_PROFILE% -Dedx.home=%EDX_HOME% -
Dlog4j.configuration=file:\\\%EDX_HOME%\config\log4j_csr.xml -
Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.Transform
erFactoryImpl

-
Djavax.xml.parsers.SAXParserFactory=org.apache.xerces.jaxp.SAXParserFactoryI
mpl -
Djavax.xml.parsers.DocumentBuilderFactory=org.apache.xerces.jaxp.DocumentBui
lderFactoryImpl
```

7. If you are using Chase Paymentech Orbital Payment Gateway, then in the JAVA OPTIONS section, add the following section to the end of the definition:
   - **Oracle Solaris.** `export JAVA_VM=$JAVA_VM -DPAYMENTECH_HOME=$PAYMENTECH_HOME`
   - **Linux.** `export JAVA_VM=$JAVA_VM -DPAYMENTECH_HOME=$PAYMENTECH_HOME`
   - **Windows.** `set JAVA_VM=%JAVA_VM% -DPAYMENTECH_HOME=%PAYMENTECH_HOME%`
8. Save the changes, and close the file.

Accessing the Oracle WebLogic Console

When the server is running, you can access the Oracle WebLogic console.

This task is a step in "Process of Configuring Oracle WebLogic for the Assisted Service Application" on page 77.

To access the Oracle WebLogic console after the server is running

- Use the following URL

`http://localhost:7005/console`

where:

- *localhost* is the name of the server where you installed the Assisted Service application.
- *7005* is the port number where you installed the Assisted Service application.

Enabling HTTPS on Your Server for the Assisted Service Application

Follow these steps to enable HTTPS on your server for the Assisted Service application, required for compliance with the Payment Card Industry Data Security Standard.

This task is a step in "Process of Configuring Oracle WebLogic for the Assisted Service Application" on page 77.

To enable HTTPS on your server for the Assisted Service application

1. Log in to the Assisted Service application domain console, for example:

   `http://localhost:7005/console/`

   - *localhost* is the name of the server where you installed the Assisted Service application.
   - *7005* is the port number where you installed the Assisted Service application.
2. Click your domain name, Environment, and Servers.
3. In the Servers table, click the server where you want to deploy your application.
4. Select Configuration, General tab, and then click SSL Listen Port Enabled. Enter an SSL port number.

Configuring JDBC Resources for the Assisted Service Application

You must configure the following JDBC resources for applications deployed with Oracle WebLogic, using the Oracle WebLogic Administration Console:

- Data sources
- Connection pools

This task is a step in "Process of Configuring Oracle WebLogic for the Assisted Service Application" on page 77.

Configuring the Data Sources for the Assisted Service Application

You must create the following data sources for the Assisted Service application:

- edxAdminDataSource
- edxXMADataSource
- reportDataSource

To create the data source for the Assisted Service application

1. Start the newly created Self-Service domain, and open the Oracle WebLogic Administration Console in a browser. The default URL is

   `http://`*`Server_Name`*`:`*`Server_Port`*`/console`

   where:

   - *Server_Name* is the name of the server with the Assisted Service application domain.
   - *Server_Port* is the port number of the Assisted Service application domain server.
2. Log on to the Oracle WebLogic Administration Console, using the user name and password defined when the domain was created.
3. Click the Data Sources link under JDBC in the services section.
4. Click New to create a new data source.
5. Enter the data source name and the JNDI name.
6. For each data source, select the Database Type as Oracle, and enter other settings as shown in the following table. Be sure to select the correct version of the driver for your database version, including patches. For the edxAdminDataSource, select Supports Global Transactions, select the corresponding transaction option, and click Next.

Data Source Name	JNDI Name	Database and Driver	Settings	Database Details
edxAdminDataSource	edx.databasePool	Oracle Driver (Thin)	Emulate Two-Phase Commit	OLTP/ Hostname: ListenPort:SID
edxXMADataSource	edx.xma.databasePool	Oracle Driver (Thin XA)	None	OLTP/ Hostname: ListenPort:SID
reportDataSource	edx.report.databasePool	Oracle Driver (Thin XA)	None	OLAP/ Hostname: ListenPort:SID

7 For Connection Properties, provide the correct values.

Connection Property	Value
Database Name	Your Oracle Database Alias Name
Host Name	Your Oracle Database Server Host Name
Port	Database Server Listening Port
Database User Name	Your Oracle Database User Name
Password	Your Oracle Database Password

8 Click Next, then click Test Configuration to test whether the database connection is configured correctly.

9 Click Next, then target the data source to the server where you want to deploy the application. The default is AdminServer. Click Finish.

10 Click Activate Changes to save the changes.

11 Repeat from Step 4 to create each remaining data source.

Configuring the Connection Pools for the Assisted Service Application

You must configure a connection pool for each data source.

To configure the connection pools for the Assisted Service application

1 Go to the Summary of JDBC Data Sources.

2 Click any data source name link.

3 On the Connection Pool tab, enter the values for the connection pool settings for each JDBC data source you created.

Property	Value
Initial Capacity	5
Maximum Capacity	20
Capacity Increment	5
Statement Cache Type	FIXED
Statement Cache Size	300
Test Connections on Reserve	Checked
Test Frequency	120
Test Table Name	DUAL

Property	Value
Shrink Frequency	15
Login Delay	1

Modifying the csr.xma.xml File for the Assisted Service Application Properties

You must specify the customer service phone number in the csr.xma.xml file. The default is xxx-xxx-xxxx.

This task is a step in "Process of Configuring Oracle WebLogic for the Assisted Service Application" on page 77.

To modify the csr.xma.xml file

1. Open the csr.xma.xml file, located in the following directory:
   - **UNIX.** *EDX_HOME*/xma/config/modules/application/csr
   - **Windows.** *EDX_HOME*\xma\config\modules\application\csr
2. Modify the customerServicePhone property with the customer service phone number for your organization.

Process of Configuring Oracle WebLogic for RESTful Web Services Server Application

RESTful Web Services run on a separate service domain that accesses the same database as the other Oracle Billing Insight applications. You must set up and configure a RESTful service domain for use with Oracle Billing Insight.

To configure Oracle WebLogic to use RESTful Web services with Oracle Billing Insight, perform the following tasks:

1. Configure the Web Services Server application:
   a. "Configuring the Oracle Billing Insight Web Services Server Domain" on page 85
   b. "Defining the Oracle WebLogic Environment for the Web Services Server Domain" on page 85
   c. "Enabling HTTPS on Your Server for the Web Services Server Application" on page 87
   d. "Configuring JDBC Resources for the Web Services Server Application" on page 88

Configuring the Oracle Billing Insight Web Services Server Domain

If you intend to use Web services with Oracle Billing Insight, then you must create an Oracle WebLogic domain for the RESTful Web Services Server application EAR file.

This task is a step in "Process of Configuring Oracle WebLogic for RESTful Web Services Server Application" on page 84.

To configure a server domain for the Oracle Billing Insight Web Services Server application

1. Go to the following directory:
   - **UNIX.** `WL_HOME/common/bin`
   - **Windows.** `WL_HOME\common\bin`
2. Run the following command:
   - **UNIX.** config.sh
   - **Windows.** config.cmd
3. On the Oracle WebLogic Configuration Wizard, select Create a new domain, then browse or enter the Domain Location. Click Next.
4. Leave the option to Create Domain Using Product Templates, accept the default selections, and then click Next.
5. Enter the user name and password of the user to administer the Oracle WebLogic domain. Confirm the password, and click Next.
6. Select the SUNJDK to use for this domain, and click Next.
7. In Advanced Configurations, select Administration Server, and then click Next.
8. Enter the Listen Port, such as port 7017, and click Next.
9. Review the Configuration Summary, and click Create.

Defining the Oracle WebLogic Environment for the Web Services Server Domain

You must set environment variables and other options in the Oracle WebLogic environment to correctly set up the server domain for use with Oracle Web services.

This task is a step in "Process of Configuring Oracle WebLogic for RESTful Web Services Server Application" on page 84.

To set environment variables for the Web Services Server domain

1. Open the setDomainEnv file in a text editor.

   This file is located in the domain's home directory, for example:

- **UNIX.** `WL_HOME/user_projects/domains/rs/bin/setDomainEnv.sh`
- **Windows.** `WL_HOME\user_projects\domains\rs\bin\setDomainEnv.cmd`

2 In the setDomainEnv.sh file, define the environment variable `EDX_HOME` as the directory in which Oracle Billing Insight is installed, for example:

- **Oracle Solaris.** `export EDX_HOME=/opt/Oracle/BillingInsight`
- **Linux.** `export EDX_HOME=/opt/Oracle/BillingInsight`
- **Windows.** `set EDX_HOME=C:\oracle\BillingInsight`

3 Add the following entries to the file:

- **Oracle Solaris.** `export CLASSPATH=$CLASSPATH:$EDX_HOME/config:$EDX_HOME/config/resourcebundle:$EDX_HOME/lib/xercesImpl-2.7.1.jar:$EDX_HOME/lib/xalan-2.7.1.jar:$EDX_HOME/lib/serializer-2.7.1.jar`
- **Linux.** `CLASSPATH=$CLASSPATH:$EDX_HOME/config:$EDX_HOME/config/resourcebundle:$EDX_HOME/lib/xercesImpl-2.7.1.jar:$EDX_HOME/lib/xalan-2.7.1.jar:$EDX_HOME/lib/serializer-2.7.1.jar`
- **Windows.** set `CLASSPATH=%CLASSPATH%;%EDX_HOME%\config;%EDX_HOME%\config\resourcebundle;%EDX_HOME%\lib\xercesImpl-2.7.1.jar;%EDX_HOME%\lib\xalan-2.7.1.jar;%EDX_HOME%\lib\serializer-2.7.1.jar`

4 If you are using Chase Paymentech Orbital Payment Gateway, then add the following entries to the setDomainEnv file. Define the environment variable PAYMENTECH_HOME as the directory where Paymentech configuration files are stored:

- **Oracle Solaris.** `export PAYMENTECH_HOME=$EDX_HOME/payment/paymentech`
- **Linux.** `export PAYMENTECH_HOME=$EDX_HOME/payment/paymentech`
- **Windows.** `set PAYMENTECH_HOME=%EDX_HOME%\payment\paymentech`

Set the config directory under PAYMENTECH_HOME in CLASSPATH:

- **Oracle Solaris.** `export CLASSPATH=$CLASSPATH:$PAYMENTECH_HOME/config`
- **Linux.** `export CLASSPATH=$CLASSPATH:$PAYMENTECH_HOME/config`
- **Windows.** `set CLASSPATH=%CLASSPATH%;%PAYMENTECH_HOME%\config`

5 In the JAVA OPTIONS section, add the `Dedx.home` Java option section to the end of the definition:

- **Oracle Solaris**. Add the Java as shown:

```
JAVA_VM="${JAVA_VM} ${JAVA_DEBUG} ${JAVA_PROFILE} -Dedx.home=${EDX_HOME} -
Dlog4j.configuration=file:${EDX_HOME}/config/log4j_rs.xml -
Djavax.xml.parsers.SAXParserFactory=org.apache.xerces.jaxp.SAXParserFactoryI
mpl -
Djavax.xml.parsers.DocumentBuilderFactory=org.apache.xerces.jaxp.DocumentBui
lderFactoryImpl -Dorg.owasp.esapi.resources=${EDX_HOME}/config -
Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.Transform
erFactoryImpl"

export JAVA_VM
```

- **Linux**. Add the Java as shown:

```
JAVA_VM="${JAVA_VM} ${JAVA_DEBUG} -Dedx.home=${EDX_HOME} -
Dlog4j.configuration=file:${EDX_HOME}/config/log4j_rs.xml -
Djavax.xml.parsers.SAXParserFactory=org.apache.xerces.jaxp.SAXParserFactoryI
mpl -
Djavax.xml.parsers.DocumentBuilderFactory=org.apache.xerces.jaxp.DocumentBui
lderFactoryImpl -Dorg.owasp.esapi.resources=${EDX_HOME}/config -
Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.Transform
erFactoryImpl"

export JAVA_VM
```

- **Windows.** Add the Java as shown. (The backslashes (\) in the following statement are correct.)

```
set JAVA_VM=%JAVA_VM% %JAVA_DEBUG% %JAVA_PROFILE% -Dedx.home=%EDX_HOME% -
Dlog4j.configuration=file:\\\%EDX_HOME%\config\log4j_rs.xml -
Djavax.xml.parsers.SAXParserFactory=org.apache.xerces.jaxp.SAXParserFactoryI
mpl -
Djavax.xml.parsers.DocumentBuilderFactory=org.apache.xerces.jaxp.DocumentBui
lderFactoryImpl -Dorg.owasp.esapi.resources=%EDX_HOME%\config -
Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.Transform
erFactoryImpl
```

6. If you are using Chase Paymentech Orbital Payment Gateway, then in the JAVA OPTIONS section, add the following section to the end of the definition:
   - **Oracle Solaris.** `export JAVA_VM=$JAVA_VM -DPAYMENTECH_HOME=$PAYMENTECH_HOME`
   - **Linux.** `export JAVA_VM=$JAVA_VM -DPAYMENTECH_HOME=$PAYMENTECH_HOME`
   - **Windows.** `set JAVA_VM=%JAVA_VM% -DPAYMENTECH_HOME=%PAYMENTECH_HOME%`
7. Specify the memory settings as follows:

```
MEM_MAX_PERM_SIZE_64BIT=”-XX:MaxPermSize=1024m
```

8. Save the changes, and close the file.

Enabling HTTPS on Your Server for the Web Services Server Application

Follow these steps to enable HTTPS on your server for the Web Services Server application, which is required for compliance with the Payment Card Industry Data Security Standard.

The following examples show Web Services Server URLs under an HTTPS configuration:

- URL for logging in or getting the token in your application:

  https://host1:ssl_port1/BillingInsight-rs/rsLogin.action

- URL for accessing RESTful Web services:

  https://host1:ssl_port1/BillingInsight-rs/rs/{E-Billing_RS_Resource}

This task is a step in "Process of Configuring Oracle WebLogic for RESTful Web Services Server Application" on page 84.

To enable HTTPS on your server for Web Services Server application

1. Log in to the Web Services Server domain console, for example:

   `http://localhost:7017/console/`

   where:

   - *localhost* is the name of the server where you installed the Web Services Server application EAR.
   - *7017* is the port number where you installed the Web Services Server application.
2. Click your domain name, Environment, and Servers.
3. In the Servers table, click the server where you want to deploy your application.
4. Select Configuration, General tab, and then click SSL Listen Port Enabled. Enter an SSL port number, then click Save.
5. Click Activate Changes to save the changes.

Configuring JDBC Resources for the Web Services Server Application

You must configure the following JDBC resources for the Web Services Server application deployed with Oracle WebLogic, using the Oracle WebLogic Administration Console:

- Data sources
- Connection pools

This task is a step in "Process of Configuring Oracle WebLogic for RESTful Web Services Server Application" on page 84.

Configuring the Data Sources for the Web Services Server Application

You must create the following data sources for the Web Services Server application:

- edxAdminDataSource
- edxXMADataSource
- reportDataSource

To create the data sources for the Web Services Server application

1. Start the newly created Web Services Server domain, and open the Oracle WebLogic Administration Console in a browser. The default URL is

   `http://`*`Server_Name`*`:`*`Server_Port`*`/console`

   where:

   - *Server_Name* is the name of the server with the Web Services Server domain.
   - *Server_Port* is the port number of the Web Services Server domain server.

2. Log on to the Oracle WebLogic Administration Console, using the user name and password defined when the domain was created.
3. Click the Data Sources link under JDBC in the services section.
4. Click New to create a new data source.
5. Enter the data source name and the JNDI name.
6. For each data source, select the Database Type as Oracle, and enter other settings as shown in the following table. Be sure to select the correct version of the driver for your database version, including patches. Click Next.

Data Source Name	JNDI Name	Database and Driver	Settings	Database Details
edxAdminDataSource	edx.databasePool	Oracle Driver (Thin)	Emulate Two-Phase Commit	OLTP/ Hostname: ListenPort:SID
edxXMADataSource	edx.xma.databasePool	Oracle Driver (Thin XA)	None	OLTP/ Hostname: ListenPort:SID
reportDataSource	edx.report.databasePool	Oracle Driver (Thin XA)	None	OLAP/ Hostname: ListenPort:SID

7. Leave the default transaction options unchanged, then click Next.
8. For Connection Properties, provide the correct values.

Connection Property	Value
Database Name	Your Oracle Database Alias Name
Host Name	Your Oracle Database server host name
Port	DB server listening port

Connection Property	Value
Database User Name	Your Oracle Database User Name
Password	Your Oracle Database Password

9 Click Next, then click Test Configuration to test whether the database connection is configured correctly.

10 Click Next, then target the data source to the server where you want to deploy the application. The default is AdminServer. Click Finish.

11 Click Activate Changes to save the changes.

12 Repeat from Step 4 to Step 11 to create each remaining data source.

Configuring the Connection Pools for the Web Services Server Application

You must configure a connection pool for each data source.

To configure the connection pools for the Web Services Server application

1 Go to the Summary of JDBC Data Sources.

2 Click any data source name link.

3 On the Connection Pool tab, enter the values for the connection pool settings for each JDBC data source you created.

Property	Value
Initial Capacity	5
Maximum Capacity	20
Capacity Increment	5
Statement Cache Type	FIXED
Statement Cache Size	300
Test Connections on Reserve	Checked
Test Frequency	120
Test Table Name	DUAL
Shrink Frequency	15
Login Delay	1

Process of Repackaging the GNU Lesser General Public License

You must repackage the GNU Lesser General Public License (LGPL) on each server where you have an Oracle Billing Insight application installed (all platforms), including the server where you have installed the Web Services Server application.

This process is a step in "Roadmap for Configuring Oracle WebLogic for Oracle Billing Insight" on page 47.

To complete repackaging the GNU LGPL, perform the following tasks on each server:

1. "Setting Up Maven" on page 91
2. "Setting Up a Proxy Maven Configuration" on page 91
3. "Repackaging LGPL" on page 92

Setting Up Maven

Before you can repackage the LGPL, you must set up Maven.

This task is a step in "Process of Repackaging the GNU Lesser General Public License" on page 91.

To set up Maven

1. Go to the following directory, and download Maven 2.0.7 (Oracle Billing Insight also supports versions 2.0.5 and 2.0.6):

   http://maven.apache.org

2. Unzip the downloaded archive to your local Maven home directory, such as `maven-2.0.7`.
3. Create the environment variable M2_HOME, where *Maven_Home_Dir* is the Maven home path. For example:
   - **UNIX.** `export M2_HOME=Maven_Home_Dir`
   - **Windows.** `set M2_HOME=Maven_Home_Dir`
4. Add the bin directory from `M2_HOME` to a PATH variable:
   - **UNIX.** `export PATH=$M2_HOME/bin:$PATH;`
   - **Windows.** `set PATH=%M2_HOME%\bin;%PATH%`

Setting Up a Proxy Maven Configuration

If your computer does not have direct access to the Internet, then set up a proxy in the Maven configuration.

This task is a step in "Process of Repackaging the GNU Lesser General Public License" on page 91

To set up a proxy Maven configuration

1. Edit the `settings.xml` file, found in the following file:
2. Go to the following directory:
   - **UNIX.** `%M2_HOME%/conf`
   - **Windows.** `%M2_HOME%\conf`
3. Add the following lines to the appropriate section of the file:

```
<settings>

  .

  .

  <proxies>

   <proxy>

      <active>true</active>

      <protocol>http</protocol>

      <host>www-YourCompanyproxy.com</host>

      <port>80</port>


<nonProxyHosts>localhost|*.YourCompanyX.com|YourCompanyY.com|YourCompanyZ.com</
nonProxyHosts>

    </proxy>

  </proxies>

  .

  .

</settings>
```

where:

- *YourCompanyproxy.com* is your company proxy site.
- *YourCompanyX.com*, *YourCompanyY.com*, and *YourCompanyZ.com* are your local host sites.

Repackaging LGPL

You must download and repackage LGPL libraries for Oracle Billing Insight.

This task is a step in "Process of Repackaging the GNU Lesser General Public License" on page 91.

To repackage LGPL

1. If you support Chase Paymentech Orbital Payment Gateway, then download the chasepaymentech_java_orbital.zip file from the following location:

   http://download.chasepaymentech.com/docs/orbital/chasepaymentech_java_orbital.zip

   Unzip the enclosed PaymentechSDK.jar file and save it to the following directory. The name of the JAR file can vary for different versions of Chase Paymentech.

   - **UNIX.** *EDX_HOME*/repackage
   - **Windows.** *EDX_HOME*\repackage

2. Run the Maven installation command. If you support Chase Paymentech, then use the following command, specifying the name of the JAR file for your version of Paymentech:

   ```
   mvn clean install -DpaymentTech.fileName=PaymentechSDK.jar
   ```

   Otherwise, use:

   ```
   mvn install
   ```

   The installation command places all EAR files in the packaged LGPL libraries in the following directory:

   - **UNIX.** *EDX_HOME*/J2EEApps

- **Windows.** *EDX_HOME*\J2EEApps

If you receive an error similar to the following examples, or any error indicating [INFO] Failed to resolve artifact or [INFO] Unable to find artifact, then the command was not able to access and download LGPL components from the Internet. Follow the instructions in "Downloading LGPL Jar Files Manually" on page 94 to manually download and install the components.

Example Error 1

```
[INFO] ------------------------------------------------------------------------
---
[INFO] Building eBilling application
[INFO]    task-segment: [install]
[INFO] ------------------------------------------------------------------------
---
[INFO] ------------------------------------------------------------------------
[ERROR] BUILD ERROR
[INFO] ------------------------------------------------------------------------
[INFO] Failed to resolve artifact.
```

Example Error 2

```
[ERROR] BUILD ERROR
[INFO] ------------------------
[INFO] Unable to find artifact.
```

Downloading LGPL Jar Files Manually

If you received an error while repackaging LGPL libraries that indicated JAR files were missing, then follow these steps to manually download and repackage the required components.

To manually download and repackage the required LGPL JAR files

1. Go to the following location, and download the swarmcache-1.0RC2.jar file

   http://mvnrepository.com/artifact/swarmcache/swarmcache/1.0RC2

2. Install the swarmcache-1.0RC2.jar file, using the following command:

```
mvn install:install-file -DgroupId=swarmcache -DartifactId=swarmcache \
          -Dversion=1.0RC2 -Dpackaging=jar -Dfile=/path/to/file
```

3. Go to the following location and download the hibernate-3.6.0.jar.Final file

   http://mvnrepository.com/artifact/org.hibernate/hibernate-core/3.6.0.Final

4. Install the hibernate-3.6.0.Final.jar file, using the following command:

```
mvn install:install-file -DgroupId=org.hibernate -DartifactId=hibernate \
          -Dversion=3.6.0.Final -Dpackaging=jar -Dfile=/path/to/file
```

5. Go to the following location, and download the jgroups-2.12.1.Final.jar file

   http://mvnrepository.com/artifact/org.jgroups/jgroups/2.12.1.Final

6. Install the jgroups-2.12.1.Final.jar file, using the following command:

```
mvn install:install-file -DgroupId=jgroups -DartifactId=jgroups \
          -Dversion=2.12.1.Final -Dpackaging=jar -Dfile=/path/to/file
```

7. If you support Chase Paymentech Orbital Payment Gateway, then download the chasepaymentech_java_orbital.zip file from the following location:

   http://download.chasepaymentech.com/docs/orbital/chasepaymentech_java_orbital.zip

   Unzip the enclosed PaymentechSDK.jar file and save it to the following directory. The name of the JAR file can vary for different versions of Chase Paymentech.

   - **UNIX.** *EDX_HOME*/repackage
   - **Windows.** *EDX_HOME*\repackage

8. Install the PaymentechSDK.jar file, using the following command:

```
mvn install:install-file -DgroupId=paymentech -DartifactId=paymentech -
Dversion=6.9.0 -Dpackaging=jar -Dfile=/path/to/file
```

9. If you support Chase Paymentech, run the following Maven installation command, specifying the name of the JAR file for your version of Paymentech:

```
mvn clean install -DpaymentTech.fileName=PaymentechSDK.jar
```

   Otherwise, use:

```
mvn install
```

   This command places all EAR files in the packaged LGPL libraries in the following directory:

   - **UNIX.** *EDX_HOME*/J2EEApps
   - **Windows.** *EDX_HOME*\J2EEApps

Process of Deploying Oracle Billing Insight Applications on Oracle WebLogic

This topic describes the process of deploying Oracle Billing Insight applications.

This process is a step in "Roadmap for Configuring Oracle WebLogic for Oracle Billing Insight" on page 47.

To configure Oracle WebLogic for the Self-Service application, perform the following tasks

1. For each of the following Oracle Billing Insight applications you have configured for your implementation, follow the steps in "Deploying the Oracle Billing Insight Applications on Oracle WebLogic" on page 97 to deploy the corresponding EAR files on your application server:
   - Self-Service application
   - Command Center application
   - Assisted Service application
2. If you have configured Oracle Billing Insight Web Services Server application for your implementation, perform the following tasks:
   a. Follow the steps in "Deploying the Jersey 2.5.1 (JAX-RS 2.0 RI) Shared Library for Use By the Web Services Application" on page 96 for each Oracle WebLogic Server instance where a Web Services application will be deployed.
   b. Follow the steps in "Deploying the Oracle Billing Insight Applications on Oracle WebLogic" on page 97 to deploy the Web Service applications.

Deploying the Jersey 2.5.1 (JAX-RS 2.0 RI) Shared Library for Use By the Web Services Application

If you intend to use the Web Services application, you must deploy the Jersey 2.5.1 (JAX-RS 2.0 RI) shared library to the Oracle WebLogic server instances where you deploy the Oracle Billing Insight Web Services application.

CAUTION: When you deploy a referencing application, such as Oracle Billing Insight Web Services, Oracle WebLogic Server merges the shared library files with the application. If you try to deploy the Web Services application to a server instance where a the Jersey library is not registered, deployment of the Web Services application will fail.

This task is a step in "Process of Configuring the Self-Service Application on an Oracle WebLogic Cluster" on page 61.

To register Jersey 2.5.1 (JAX-RS 2.0 RI) shared library with an Oracle WebLogic Server instance

1. Log into the Oracle WebLogic Server Administration Console. For example:

   http://localhost:7007/console
2. Select Deployments.
3. Click Install to install the shared library.
4. In the Path field, enter or navigate to the following directory:

   `WL_HOME\wlserver\common\deployable-libraries`

5 Select the jax-rs-2.0.war file, and click Next.

6 Select Install this Deployment as a Library, and click Next.

7 Click Finish.

8 Repeat Step 1 through Step 7 for each WebLogic Server instance where you deploy the Web Services application.

Deploying the Oracle Billing Insight Applications on Oracle WebLogic

You must deploy each of the Oracle Billing Insight applications that you have configured on Oracle WebLogic.

This task is a step in "Process of Deploying Oracle Billing Insight Applications on Oracle WebLogic" on page 95.

To deploy an Oracle Billing Insight application on Oracle WebLogic

1 If you are deploying the Self-Service application on an Oracle WebLogic cluster, then edit the weblogic.xml file, located in the edx_home\J2EEApps\selfservice\weblogic\selfservice-weblogic-7.0\selfservice-web-1.0-SNAPSHOT\WEB-INF\ directory. Add the following code to enable a session on one cluster computer member to be replicated to the other computer members:

```
<session-descriptor>
 <session-param>
    <param-name>PersistentStoreType</param-name>
    <param-value>replicated</param-value>
 </session-param>
</session-descriptor>
```

2 Open the Oracle WebLogic Administration Console.

3 Click the Deployments link to display the deployments page for the Oracle WebLogic domain for the particular application you are deploying, such as BillingInsight_domain.

4 Click Install to start the Install Application Assistant, which guides you through the steps required to deploy the Oracle Billing Insight applications.

5 Click the links to navigate to the location of the EAR file to deploy.

Oracle Billing Insight J2EE Application	File Name and Location
Self-Service	File Name: selfservice-weblogic-7.0.ear Location: ■ **UNIX.** *EDX_HOME*/J2EEApps/selfservice/weblogic ■ **Windows.** *EDX_HOME*\J2EEApps\selfservice\weblogic\
Command Center	File Name: command-center-weblogic-7.0.ear Location: ■ **UNIX.** *EDX_HOME*/J2EEApps/commandcenter/weblogic ■ **Windows.** *EDX_HOME*\J2EEApps\commandcenter\weblogic
Assisted Service	File Name: csr-app-7.0.ear Location: ■ **UNIX.** *EDX_HOME*/J2EEApps/csr/weblogic ■ **Windows.** *EDX_HOME*\J2EEApps\csr\weblogic
Web Services Server	File Name: rs-weblogic-7.0.ear Location: ■ **UNIX.** *EDX_HOME*/J2EEApps/rs/weblogic ■ **Windows.** *EDX_HOME*\J2EEApps\rs\weblogic

Select the deployed EAR file, and start all services located at the Start, Servicing all requests. If you are using the Self-Service application on a cluster, then deploy the Self-Service application to the cluster, not the Admin server.

6 Restart the server.

7 Repeat this procedure for each Oracle Billing Insight application that your implementation requires.

8 If you are using the Self-Service application on a cluster, then you must also deploy the Cluster Proxy application to the proxy server. Edit the web.xml file found in the BillingInsightProxy WAR file. In the following section, add the IP address and port number of each managed server in the cluster, for example:

```
<init-param>
 <param-name>WebLogicCluster</param-name>
    <param-value>10.240.12.163:9001|10.240.8.244:9002|10.240.12.157:9003
    </param-value>
</init-param>
```

9 If you are deploying the Self-Service application on a cluster, then start the cluster.

For details, see "Starting the Oracle WebLogic Cluster" on page 99.

Starting the Oracle WebLogic Cluster

If you are configuring the Self-Service application on an Oracle WebLogic cluster, then you must start the cluster after deployment.

This task is a step in "Process of Configuring the Self-Service Application on an Oracle WebLogic Cluster" on page 61.

To start the Oracle WebLogic cluster

1 On the computer where the Admin server is located, start the admin server:

```
$selfservice_domain/bin/startWeblogic.sh
```

2 On each computer, start the node manager:

```
$WL_HOME/server/bin/startNodeManager.sh
```

3 Start each of the managed servers that is on the same computer as the Admin server:

```
$WL_HOME/common/bin/startManagedWeblogic.sh BillingInsightMS1 http://
10.240.12.163:7001
```

where:

- *EBillingMS1* is the name you gave the managed server.
- *10.240.12.163* is the IP address of the computer where the Admin server is located.
- *7001* is the port number of the Admin server.

4 Start the proxy server. In this command, *EBillingProxy* is the name you gave the Proxy server:

```
$WL_HOME/common/bin/startManagedWebLogic.sh BillingInsightProxy http://
10.240.12.163:7001
```

where:

- *BillingInsightProxy* is the name you gave the Proxy server.
- *10.240.12.163* is the IP address of the computer where the Admin server is located.
- *7001* is the port number of the Admin server.

5 On each of the other computers in the cluster, follow these steps:

a Start the node manager:

```
$WL_HOME/server/bin/startNodeManager.sh
```

b Start each managed server, using the following command:

```
$weblogic_home /common/bin/startManagedWeblogic.sh BillingInsightMS2 http://
10.240.12.163:7001
```

where:

- *BillingInsightMS2* is the name you gave the managed server.
- *10.240.12.163* is the IP address of the computer where the Admin server is located.
- *7001*is the port number of the Admin server.

Stopping the Oracle WebLogic Cluster

Follow these instructions if you want to stop the Oracle WebLogic cluster where the Self-Service application is implemented.

To stop the Oracle WebLogic cluster

- Do one of the following:
  - Stop the process in the SSH window.
  - Go to the Admin console, using the following command, and then stop the managed server separately:

    http://*AdminServerIP:Port*/console

    where:

    - *AdminServerIP* is the IP address of the computer where the Admin server and the managed server are located.
    - *7001* is the port number of the managed server.

Configuring and Starting Scheduler on Oracle WebLogic

Scheduler is a program that administrators use to schedule Command Center jobs. Scheduler must be running for scheduled jobs to execute.

Follow the procedure appropriate for your operating system to configure Scheduler on Oracle WebLogic:

- “Configuring and Starting Scheduler on Oracle WebLogic and UNIX” on page 101
- “Configuring and Starting Scheduler on Oracle WebLogic on Windows” on page 101

NOTE: For information on using Scheduler, see *Administration Guide for Oracle Billing Insight*.

Configuring and Starting Scheduler on Oracle WebLogic and UNIX

Follow these steps to configure and start Scheduler on Oracle WebLogic and UNIX.

To configure and start Scheduler on Oracle WebLogic and UNIX

1. Go to the *EDX_HOME*/bin directory.
2. Run the following command:

   ```
   $./edx_config
   ```

3. Specify the appropriate details for the application server.

Field	Example Value
Application server	wl
Java root directory	/opt/bea12c/Middleware/jdk160_29
WebLogic Application Server root directory	/opt/bea12c/Middleware/wlserver

4. Start Scheduler, using the following command, located in the *EDX_HOME*/bin directory:

   ```
   ./wl_scheduler -start -url t3://localhost:7003 -verbose
   ```

   where:

   - *localhost* is the name of the Command Center application server.
   - *7003* is the port number of the Command Center application server.

   To stop Scheduler, replace the -start parameter with the -stop parameter:

   ```
   ./wl_scheduler -stop -url t3://localhost:7003 -verbose
   ```

Configuring and Starting Scheduler on Oracle WebLogic on Windows

Follow these steps to configure and start Scheduler on Oracle WebLogic and Windows.

To configure and start Scheduler on Oracle WebLogic and Windows

1. Open the edx_env.bat file in the *EDX_HOME*\config directory, and add the following lines. In the lines, *Your_Java_Home* is the directory where Java is installed.

   ```
   @set EDX_HOME=D:\oracle\BillingInsight

   @set APP_SERVER=wl

   @set WL_HOME=D:\oracle\wls12130\wlserver
   ```

```
@set JAVA_HOME=Your_Java_Home
```

2 Go to the *EDX_HOME*\bin directory, and run the following command:

```
wl_scheduler.bat -start -url t3://localhost:7003 -verbose
```

where:

- *localhost* is the name of the Command Center application server.
- *7003* is the port number of the Command Center application server.

To stop Scheduler, replace the -start parameter with the -stop parameter in the same command:

```
wl_scheduler.bat -stop -url t3://localhost:7003 -verbose
```

Running the Sample Oracle Billing Insight Applications on Oracle WebLogic

After successfully deploying the Oracle Billing Insight application EAR files, you can log in to the sample Oracle Billing Insight applications.

This task is a step in "Roadmap for Configuring Oracle WebLogic for Oracle Billing Insight" on page 47.

To run the sample Oracle Billing Insight applications

1 In your browser, point to the Oracle Billing Insight application name, shown in the following table, specifying the local host (server name) and port number where you deployed the Oracle Billing Insight application.

Sample Oracle Billing Insight Application	URL Example
Self-Service	http://localhost:7001/selfservice In a clustered environment, use the Proxy server host and port, such as: like http://10.240.12.163:9007/selfservice
Command Center	http://localhost:7003/eBilling
Assisted Service	http://localhost:7005/csr
Web Services Server	http://localhost:7017/rs

The sample log in page appears.

2 For the sample Self-Service application, log in as one of the enrolled administrative users. For a list of preconfigured sample user IDs, see "Sample Users" on page 103. For the preconfigured sample user passwords, see your system administrator.

The user you log in as determines whether you see the Business or Consumer Edition of the Self-Service application. For information about enrolling for the first time, see *Application Guide for Oracle Billing Insight (Business Edition)*.

Related Topic

"Sample Users" on page 103

Sample Users

Oracle Billing Insight provides pre-enrolled users with the sample Self-Service and Assisted Service applications. The following tables show the preconfigured sample data available:

- "Business Users in the Sample Self-Service Application" on page 103
- "Consumer Users in the Sample Self-Service Application" on page 104
- "Users in the Sample Assisted Service Application" on page 105

For the preconfigured sample user passwords, along with the secure question and answers, see your system administrator.

Business Users in the Sample Self-Service Application

Table 6 shows a list of the preconfigured (enrolled) business users created for the sample Self-Service application and the prepopulated data for each. All users are with the CUELLE company ID.

Table 6. Business Users in the Sample Self-Service Application

User ID	Role	Billing Data Types	Account Numbers	First Name / Last Name	Brand Code
ftown	Admin	Billed and unbilled data	AC-B2B-123361-01 AC-B2B-153561-05 AC-B2B-64521A AC-B2B-TF2104A	Frank Town	Default
dennismartin	Admin	Same as ftown	Same as ftown	Dennis Martin	Take2
joesmith	Admin	Same as ftown	Same as ftown	Joe Smith	Default
managersales	Manager	As assigned in the hierarchy	As assigned in the hierarchy	Sales Manager	Default
managerfin	PayerManager	As assigned in the hierarchy	As assigned in the hierarchy	Finance Manager	Take1

Table 6. Business Users in the Sample Self-Service Application

User ID	Role	Billing Data Types	Account Numbers	First Name / Last Name	Brand Code
managerpay	PayerManager	As assigned in the hierarchy	As assigned in the hierarchy	Payment Manager	Take1
managerrd	Manager	As assigned in the hierarchy	As assigned in the hierarchy	RD Manager	Take2
managerit	Manager	As assigned in the hierarchy	As assigned in the hierarchy	IT Manager	Default
subscriber1	Subscriber	As assigned in the hierarchy	As assigned in the hierarchy	Subscriber1 Subscriber	Default
subscriber2	Subscriber	As assigned in the hierarchy	As assigned in the hierarchy	Subscriber	Take2
subscriber3	Subscriber	As assigned in the hierarchy	As assigned in the hierarchy	Subscriber3	Take1
subscriber4	Subscriber	As assigned in the hierarchy	As assigned in the hierarchy	Subscriber4	Default

Consumer Users in the Sample Self-Service Application

Table 7 shows a list of the pre-enrolled consumer users created for the sample Self-Service application and the prepopulated data for each.

Table 7. Consumer Users in the Sample Self-Service Application

User ID	Role	Billing Data Types	Account Numbers	Brand Code
johnsmith	B2C User	Post pay accounts; billed and unbilled data	AC-B2C-2014LL AC-B2C-2014Z56	Default
janesmith	B2C User	Post pay account; billed and unbilled data Cable account	AC-B2C-7816385-9 23-B2C-COMCAST0125	Take1
anascott	B2C User	Post pay account; billed and unbilled data Prepaid account	AC-B2C-2014L3A AC-PP-245113-15	Take2
janedoe	B2C User	Prepaid account	AC-PP-120012-55	Default

Table 7. Consumer Users in the Sample Self-Service Application

User ID	Role	Billing Data Types	Account Numbers	Brand Code
johndoe	B2C User	Device data account	AC-B2C-DATA-2661	Default
bobscott	B2C User	Post pay account; billed and unbilled data Cable account	AC-B2C-7836380-1 70-B2C-FIOS-012 05	Take2

Users in the Sample Assisted Service Application

Table 8 shows a list of the preconfigured (enrolled) users created for the sample Assisted Service application.

Table 8. Users in the Sample Assisted Service Application

Sample Assisted Service Application User Name	Role	First Name and Last Name
csradmin	CSRAdministrator	csradmin csradmin
csradminRightNow	CSRAdministrator	csradminRightNow csradminRightNow
csradminSiebel	CSRAdministrator	csradminSiebel csradminSiebel
csradminOSC	CSRAdministrator	csradminOSC csradminOSC
MarketingCloud	CSRAdministrator	MarketingCloud MarketingCloud
ServiceCloud	CSRAdministrator	ServiceCloud ServiceCloud
csruser	CSR	csruser csruser
merchant	Merchant	Tom Johnston

Related Topic

"Running the Sample Oracle Billing Insight Applications on Oracle WebLogic" on page 102

5 Configuring the ODI Data Load Processes

This chapter describes how to install configure the processes for loading data into the Oracle Billing Insight production environment using Oracle Data Integrator. It includes the following topics:

Roadmap for Configuring the Data Load Processes for Oracle Billing Insight

To configure the ODI data load processes for Oracle Billing Insight, perform the following processes and tasks:

1. "Installing a Standalone Version of Oracle Data Integrator" on page 108
2. "Creating an ODI Repository" on page 108
3. "Creating a Standalone Agent" on page 109
4. "Testing ODI Connectivity" on page 109
5. "Setting up the Agent Topology" on page 110
6. "Setting up the Database Topology" on page 110
7. "Configuring Oracle Billing Insight for Your ODI Configuration" on page 111
8. "(Optional) Loading Sample Billing Files" on page 114

Related Topic

"Roadmap for Installing Oracle Billing Insight 7.0" on page 11

Installing a Standalone Version of Oracle Data Integrator

Oracle Billing Insight uses Oracle Data Integrator (ODI) as the tool for loading data from a billing system into the Oracle Billing Insight database. For information about installing and configuring Oracle Data Integrator, see the Oracle Data Integrator documentation on Oracle Technology Network.

This task is a step in "Roadmap for Configuring the Data Load Processes for Oracle Billing Insight" on page 107.

To install a Standalone version of ODI

- Follow the instructions for installing ODI in the Oracle Data Integrator documentation. Follow the installation wizard and, when prompted, choose the Standalone option (not Enterprise).

Creating an ODI Repository

You must create a repository in ODI for use with Oracle Billing Insight using the options and parameters described in this topic. For additional information about installing and configuring Oracle Data Integrator, see the Oracle Data Integrator documentation on Oracle Technology Network.

This task is a step in "Roadmap for Configuring the Data Load Processes for Oracle Billing Insight" on page 107.

To configure the ODI repository

1. Follow the ODI installation wizard for creating an ODI repository.
2. On the Create Repository page, select the following options:
   - Create Repository
   - System Load and Product Load
3. On the Select Components page, specify a prefix and select the following options:
   - As Common Schemas
   - Oracle Data Integrator
4. On the Custom Variables page, specify the following parameters:
   - **Work Repository Type.** Leave the default value (D).
   - **Work Repository Name.** Leave the default name, WORKREP, in all uppercase.

Creating a Standalone Agent

You must create a standalone run-time agent for your production environment.

This task is a step in "Roadmap for Configuring the Data Load Processes for Oracle Billing Insight" on page 107.

To create a standalone agent

1. Launch ODI.
2. On the Learn & Explore page, select Create an Agent.
3. Follow the instructions in the tutorial to run the wizard for creating a Standalone agent. Select the Oracle Data Integrator Standalone Agent checkbox.
4. On the System Components page, within System Components, choose OracleDIAgent1.
5. Configure the component accordingly.

Testing ODI Connectivity

You must test the ODI connections after setting up an ODI repository and a standalone agent.

This task is a step in "Roadmap for Configuring the Data Load Processes for Oracle Billing Insight" on page 107.

To test ODI connectivity

1. Launch ODI.
2. On the ODI landing page, click Connect To Repository.
3. For the Oracle Data Integrator connection, specify the following:
   - **Login Name.** Enter a login name. It is recommended that you use BillingInsight.
   - **User.** Enter SUPERVISOR (all uppercase).
   - **Password.** Enter the value you specified when you created the ODI repository on the Custom Variables page for the SUPERVISOR.
4. For the database connection (Master Repository), specify the following:
   - **User.** Enter the user name that was generated automatically in the Select Components step of the repository creation.
   - **Password.** Enter the value you specified for the Work Repository password on the Custom Variables page.
   - **Driver List.** Choose Oracle JDBC Driver from the drop-down list.
   - **URL.** Enter the credentials for your database server. Click Search, and select jdbc:oracle:thin:*host*:*port*:*SID.* Replace *host*, *port*, and *SID* with the values for your implementation.

5 For the Work Repository, click Work Repository, click the Search icon, and select WORKREP.

6 Click Test.

Setting up the Agent Topology

You must set up the agent topology using ODI.

This task is a step in "Roadmap for Configuring the Data Load Processes for Oracle Billing Insight" on page 107.

To set up the agent topology

1 Run and log into ODI.

2 Connect to the ODI repository.

3 Click the Topology tab, and expand the Physical Architecture tree.

4 Right-click Agents, choose New Agent, and specify the following:

- **Agent Name.** For the name of the agent, use the one you created in the System Component of the standalone agent configuration: use OracleDIAgent1.
- **Host.** Specify the host of the ODI standalone agent.
- **Server Listen Port.** Specify the server listen port used in the ODI server configuration (part of the standalone agent configuration).

5 Click Save.

6 Expand the Logical Architecture tree. Right-click Agents, and choose New Logical Agent. For the agent name, specify ODIAgent.

7 For the Context field, for Global, choose OracleDIAgent from the Physical Agent drop-down list.

8 Click Save.

Setting up the Database Topology

You must set up the database topology for connecting to the Oracle Billing Insight schemas when loading data.

To set up the database topology

1 Run and log into ODI.

2 Connect to the ODI repository.

3 Click the Topology tab, expand the Physical Architecture tree, and then expand Technologies. Right-click Oracle, select New Data Server, and specify the following:

- **Name.** It is recommended that you use db-ebill.

- **Instance and OLAP Schema.** Enter the instance and OLAP schema credentials.
- **Driver List.** Click the JDBC tab, click the Search icon, and search for and choose Oracle JDBC Driver from the drop-down list.
- **JDBC URL.** Enter the credentials for your database server. Click Search, and select jdbc:oracle:thin:*host*:*port*:*SID.* Replace *host*, *port*, and *SID* with the values for your implementation.

4 Click Save.

5 Click Test.

6 Expand the Logical Architecture, expand Technologies. Right-click Oracle, select New Logical Schema, and specify the following:

- **Name.** It is recommended that you use db-ebill.
- **Context.** For Global, choose db-ebill *OLAP schema name* from the Physical Schemas drop-down list.

7 Click Save.

8 In the Physical Architecture tree, expand Technologies tree, and expand the Oracle tree. Right-click the new data server, db-ebill, select New Physical Schema, and specify the following:

- **Schema.** Choose your OLAP schema name.
- **Work Schema.** Choose your OLAP schema name.

9 Click the Context tab, and click the Add icon. In the Logical Schema drop-down list, select db-ebill.

10 Click Save.

Configuring Oracle Billing Insight for Your ODI Configuration

This topic describes how to set up Oracle Billing Insight to connect with your ODI configuration. This topic assumes you have installed Oracle Data Integrator, set up the repository, standalone agent, and topology as described in this chapter.

For Windows, change the slashes (/ or \) and root directory as necessary.

This task is a step in "Roadmap for Configuring the Data Load Processes for Oracle Billing Insight" on page 107.

To configure Oracle Billing Insight for your ODI configuration

1 Open the BillingInsight_etl.properties configuration file in a text editor. This file is located in the following directory:

- **UNIX.** *EDX_HOME*/db/oracle

- **Windows.** *EDX_HOME*\db\oracle

In the path, *EDX_HOME* is the location where Oracle Billing Insight is installed.

2 Modify the following properties to reflect your Oracle Billing Insight environment. The directory of the OUTPUT_DIR directory must *not* be the same as the directory of the INPUT_DIR.

Property	Value
AGENT_BIN_DIR	The location of bin directory where the stand alone agent is installed.
XMLFILES_DIR	The folder location where the Oracle Billing Insight XML files are located, such as: *EDX_HOME*/db/oracle/olap/etl
AGENT_NAME	The default Oracle Data Integrator agent name.
WORKREP_NAME	The default repository name.
SYS_PASSWD	The SYS password of the Oracle Billing Insight database.
OLAP_USER	The user name for the OLAP schema.
OLAP_PASSWD	The password for the OLAP schema.
EBILL_TNS_NAMES	The Oracle Billing Insight database TNS (Transparent Network Substrate) name.
TOPX_REC_CAP	Specify the number of records for a service and period to maintain in the Top X tables. The default is 51 (the first 50 records for a service and period will be inserted into the Top X tables). The higher the value, the greater the impact on performance in the data loading process. It is recommended not to exceed 100.
INPUT_DIR	Specify the input directory path, such as: /export/home/oracle/testdata/OWF_INDATA In a RAC environment, specify a shared location that all RAC nodes can access.
OUTPUT_DIR	Specify the output directory path, such as: /export/home/oracle/testdata/OWF_OUTDATA In a RAC environment, specify a shared location that all RAC nodes can access.
REJECT_DIR	Specify the reject directory path, such as: /export/home/oracle/testdata/OWF_REJDATA In a RAC environment, specify a shared location that all RAC nodes can access.
BAD_FILE_LOCATION	Specify the directory path for bad files.
DISCARD_FILE_LOCATION	Specify the directory path for discarded files.

Property	Value
LOG_FILE_LOCATION	Specify the directory path for log files.
GRP_MOVE_INDIR	Specify the group move input directory path, such as: /export/home/oracle/testdata/GRPMV_INDATA In a RAC environment, specify a shared location that all RAC nodes can access.
GRP_MOVE_OUTDIR	Specify the group move output directory path, such as: /export/home/oracle/testdata/GRPMV_OUTDATA In a RAC environment, specify a shared location that all RAC nodes can access.

3 Open a command prompt, and go to the directory containing the BillingInsight_etl.properties file.

4 Run the following commands to set up your Oracle Data Integrator and Apache Ant environments, substituting the paths of your Oracle Data Integrator and Ant installations and correct slashes (/ or \) for your platform:

```
set ANT_HOME=/opt/apache-ant-1.8.3

set PATH=%PATH%;%ANT_HOME%/bin
```

5 Go to the following directory:

- **UNIX.** *EDX_HOME*/db/oracle
- **Windows.** *EDX_HOME*\db\oracle

6 Enter Ant to run the build script.

By default, the Ant command runs the build.xml file in the current directory.

7 From the Main Menu, select Option 3, Create Database Schema & Objects.

8 Select Option 4, ETL Setup.

9 You have the option to create the packages and import the Oracle Billing Insight ODI processes in one step or in separate steps:

- **One step.** To create the ETL objects and import the ODI processes in one step, select Option 3, Run Steps 1 and 2, from the ETL Setup menu. Ant returns to the current menu when finished. When these steps are complete, go to Step 14.
- **Separate steps.** To create the ETL objects and import the ODI Oracle Billing Insight processes in separate steps, continue with Step 10.

10 Select Option 1, Create ETL Objects.

11 Check the log file, found in the *EDX_HOME*/db/oracle directory, for errors.

12 Review all log files for possible errors

13 Select Option 2, Import ODI Billing Insight Processes.

Ant returns to the current menu when finished.

14 Review all log files found in the *EDX_HOME*/db/oracle directory.

(Optional) Loading Sample Billing Files

Sample data is provided with Oracle Billing Insight for testing purposes.

CAUTION: Loading sample data via scripts and sample billing file data is not supported in the same schema and can cause conflicts.

This task is a step in "Roadmap for Configuring the Data Load Processes for Oracle Billing Insight" on page 107.

To load the sample billing files

1 Run the HierarchyCopy job in the Command Center for the period specified in the billing file.

This job replicates the hierarchy version to the period that the data load file is for. Also run the Hierarchy copy job if the new sample load file is for a new period.

For details on running the HierarchyCopy job, see *Administration Guide for Oracle Billing Insight*.

2 If you are using the Telco version of Oracle Billing Insight, then copy the following sample data files from the following directory to the input directory for your data load process. In the path, *EDX_HOME* is the location where Oracle Billing Insight is installed.

- **UNIX.** *EDX_HOME*\db\oracle\olap\etl\sample_data\Telco
- **Windows.** *EDX_HOME*/db/oracle/olap/etl/sample_data\Telco

Move the following sample Telco data files:

- MASTER-DATA-FILE-001.DAT
- EBILLING_B2B-DATA-FILE-001.DAT
- EBILLING_B2B-DATA-FILE-002.DAT
- EBILLING_B2B-DATA-FILE-003.DAT
- EBILLING_B2B-DATA-FILE-004.DAT
- EBILLING_B2B-DATA-FILE-005.DAT
- EBILLING_B2B-DATA-FILE-006.DAT
- EBILLING_B2C-DATA-FILE-001.DAT
- EBILLING_B2C-DATA-FILE-002.DAT
- EBILLING_B2C-DATA-FILE-003.DAT
- EBILLING_B2C-DATA-FILE-004.DAT
- EBILLING_B2C-DATA-FILE-005.DAT

- EBILLING_B2C-DATA-FILE-006.DAT

3 If you are using the Utility version of Oracle Billing Insight, then copy the sample data files from the following directory to the input directory for your data load process:

- **UNIX.** *EDX_HOME*\db\oracle\olap\etl\sample_data\Utilities
- **Windows.** *EDX_HOME*/db/oracle/olap/etl/sample_data/Utilities

Move the following sample Utility data files:

- MASTER-UTILITIES-001.DAT
- EBILLING_B2C-UTILITIES-001.DAT
- EBILLING_B2C-UTILITIES-002.DAT
- EBILLING_B2C-UTILITIES-003.DAT

4 Use Oracle Data Integrator to run the data file load processes. For details, see *Administration Guide for Oracle Billing Insight*.

5 You must remove sample data before going live. For information about the process of purging sample data, see *Administration Guide for Oracle Billing Insight*.

CAUTION: You must remove sample data from your production environment to comply with the Payment Card Industry Data Security Standard.

6 Migrating to Oracle Billing Insight 7.0

This chapter describes how to migrate to Oracle Billing Insight version 7.0. It includes the following topics:

Process of Migrating Oracle Self-Service E-Billing Version 6.2 to Oracle Billing Insight Version 7.0

This topic describes the process required to migrate Oracle Self-Service E-Billing version 6.2 to Oracle Billing Insight version 7.0.

To migrate from Oracle Self-Service E-Billing version 6.2 to Oracle Billing Insight version 7.0, perform the following tasks:

1. Back up your existing OLTP and OLAP Oracle Self-Service E-Billing databases.

   For additional information, see Oracle Database 11*g* documentation on Oracle Technology Network.

2. Start the database instance that accesses the database you are upgrading. Check the status of all user objects. If any of them indicate an INVALID status, then contact the database administrator to correct this problem.

   For additional information, see Oracle Database 11*g* documentation on Oracle Technology Network.

3. Have any existing Oracle Billing Insight database passwords available.

   Check with your database administrator.

4. Process all pending notifications in Oracle Self-Service E-Billing version 6.2.

   For details on how to run the Notifier job, see *Administration Guide for Oracle Self-Service E-Billing*, version 6.2.

5. Process all pending batch reports in Oracle Self-Service E-Billing version 6.2

   For details on how to run the batch report jobs, see *Administration Guide for Oracle Self-Service E-Billing*, version 6.2.

6. Write down your payment gateway and job configuration settings as shown in the Command Center. You must reenter the settings after migrating.

7 Write down your Command Center admin user names. You must recreate these users after migrating.

8 Upgrade your Oracle Database 11*g* to Oracle Database 12*c*.

For additional information, see Oracle Database 12*c* documentation on Oracle Technology Network.

9 Perform the following installation tasks:

a "Checking the Integrity of the Oracle Billing Insight Installer Package" on page 12

b "Installing the Oracle Billing Insight Software Using InstallAnywhere" on page 13.

c Go to My Oracle Support (MOS) and open a Service Request (SR) to receive the Oracle Billing Insight v7.0.0.1 patch.

d Follow the steps in the Patch 1 release notes to install the patch.

e "Configuring Log File Paths for Log4j" on page 15

10 Perform the following migration tasks:

a "Migrating Oracle Self-Service E-Billing Version 6.2 OLTP to Oracle Billing Insight Version 7.0" on page 119

b "Migrating Oracle Self-Service E-Billing Version 6.2 OLAP to Oracle Billing Insight Version 7.0" on page 120

c "Migrating Batch Reports from Oracle Self-Service E-Billing Version 6.2 to Oracle Billing Insight Version 7.0" on page 121

11 "Prepopulating and Migrating Product Catalog Tables" on page 128.

12 Run the following command on both the OLTP and OLAP schemas to verify that all objects are valid. The correct result is zero. Connect as the OLTP and OLAP schema owner.

```
select count(*) from user_objects where status ='INVALID';
```

13 If you ran the master key update process on Oracle Billing Insight version 6.2, then you must copy the version 6.2 *EDX_HOME*/keystore directory to version 7.0. Otherwise, encryption and decryption will not work.

14 Configure your application server for Oracle Billing Insight version 7.0.

Follow "Roadmap for Configuring Oracle WebLogic for Oracle Billing Insight" on page 47.

15 Follow the tasks and processes in "Roadmap for Configuring the Data Load Processes for Oracle Billing Insight" on page 107.

16 Uninstall Oracle Self-Service E-Billing version 6.2.

For information about uninstalling, see "Uninstalling Oracle Billing Insight" on page 21.

17 Enable password auto-migration. Open the cryptography.xma.xml file, located in the following directory, and set the passwordMigrationEnabled property to true:

- **UNIX.** *EDX_HOME*\xma\config\modules\cryptography

- **Windows.** *EDX_HOME*/xma/config/modules/cryptography

Migrating passwords enables the Self-Service application and the Assisted Service application users to log in with their passwords from version 6.2.

18 Recreate the Command Center admin users, using the bootstrap user.

For details, see *Administration Guide for Oracle Billing Insight*.

19 Recreate the payment gateway and jobs, using the Command Center.

For details, see *Administration Guide for Oracle Billing Insight*.

Migrating Oracle Self-Service E-Billing Version 6.2 OLTP to Oracle Billing Insight Version 7.0

Follow these steps to migrate Oracle Self-Service E-Billing version 6.2 OLTP to Oracle Billing Insight version 7.0.

This task is a step in "Process of Migrating Oracle Self-Service E-Billing Version 6.2 to Oracle Billing Insight Version 7.0" on page 117.

To migrate Oracle Self-Service E-Billing version 6.2 OLTP to Oracle Billing Insight version 7.0

1 On UNIX, log in as the ORACLE user for migration activity.

2 Go to the following directory:

- **UNIX.** *EDX_HOME*/db/ebilling/oracle/oltp/migration/ebilling6.2_to_7.0
- **Windows.** *EDX_HOME*\db\ebilling\oracle\oltp\migration\ebilling6.2_to_7.0

In the path, EDX_HOME is the location where Oracle Billing Insight is installed.

3 Open the migrate_oltp_6.2_to_7.0.properties file, and set the correct value for each property in the file.

Property	What to Set
OLAP_USER	OLAP schema user name
OLAP_PASSWD	OLAP schema password
EBILL_SID	Your existing EBILL SID (instance name)
OLTP_USER	OLTP schema user name
OLTP_PASSWD	OLTP schema password
SYS_PASSWD	sys password

4 Save and close the migrate_oltp_6.2_to_7.0.properties file.

5 Run the following command to start the migration:

```
ant -f migrate_oltp_6.2_to_7.0.xml
```

6 Select Option 1, Migrate OLTP schema.

7 Select Option 2, Migrate OLTP data.

8 Review all log files for possible errors.

Migrating Oracle Self-Service E-Billing Version 6.2 OLAP to Oracle Billing Insight Version 7.0

Follow these steps to migrate Oracle Billing Insight version 6.2 OLAP to version 7.0 for both UNIX and Windows.

This task is a step in "Process of Migrating Oracle Self-Service E-Billing Version 6.2 to Oracle Billing Insight Version 7.0" on page 117.

To migrate Oracle Billing Insight version 6.2 OLAP to version 7.0

1 On UNIX, log in as the ORACLE user for migration activity.

2 Go to the following directory:

- **UNIX.** *EDX_HOME*/db/oracle/olap/migration/ebilling6.2_to_7.0
- **Windows.** *EDX_HOME*\db\oracle\olap\migration\ebilling6.2_to_7.0

In the path, EDX_HOME is the location where Oracle Billing Insight is installed.

3 Open the migrate_olap_6.2_to_7.0.properties file, and set the correct value for each property in the file.

Property	What to Set
OLAP_USER	OLAP schema user name
OLAP_PASSWD	OLAP schema password
EBILL_SID	Your existing EBILL SID (instance name)
TOPX_REC_CAP	The cap on the number of records to be calculated for TopX tables. The default is 51. Changing this value impacts the processing time for the data load process (post load process). Do Not Exceed 100.
OLTP_USER	OLTP schema user name
OLTP_PASSWD	OLTP schema password
SYS_PASSWD	sys password

4 Save and close the migrate_olap_6.2_to_7.0.properties file.

5 Run the following command to start the migration:

```
ant -f migrate_olap_6.2_to_7.0.xml
```

6 Select Option 1, Migrate OLAP Schema.

7 Select Option 2, Migrate OLAP data.

8 Review all log files for possible errors.

Migrating Batch Reports from Oracle Self-Service E-Billing Version 6.2 to Oracle Billing Insight Version 7.0

After you have successfully migrated from Oracle Self-Service E-Billing version 6.2 to Oracle Billing Insight version 7.0, you must copy the batch reports to the directory you want to use for reports in the Oracle Billing Insight version 6.2.

If you copy the reports to a different location than you used in version 6.2, then you must update the batch report file location in the OLTP database. Also, if you installed Oracle Billing Insight version 7.0 in a location other than the default EDX_HOME, or if you create a custom subdirectory for storing batch reports (the default directory is output/reportapp), then you must update the related properties in the reporting.batch.xma.xml file.

This task is a step in "Process of Migrating Oracle Self-Service E-Billing Version 6.2 to Oracle Billing Insight Version 7.0" on page 117.

To migrate your batch reports from Oracle Self-Service E-Billing version 6.2 to Oracle Billing Insight version 7.0

1 On the application server, copy your existing batch reports to the following directory:

- **UNIX.** *EDX_HOME*\output\reportapp
- **Windows.** *EDX_HOME*/output/reportapp

In the directory, *EDX_HOME* is the directory where you installed Oracle Self-Service E-Billing version 6.2.

2 If you copy your batch reports to a different file location in version 7.0 than you used in version 6.2, then you must update the batch report file location in the OLTP database. Log on to the OLTP schema, using SQL*Plus, and run the following script:

```
update edx_rpt_batch_report set file_location = 'NEW_LOCATION' ||
substr(file_location,length('OLD_LOCATION')+1) where file_location is not null;

SQL>commit;
```

where:

- *NEW_LOCATION* is the new location where you will store batch reports in Oracle Billing Insight version 7.0.

- *OLD_LOCATION* is the old location where you stored batch reports in Oracle Billing Insight version 6.2.

For example:

```
SQL> update edx_rpt_batch_report set file_location = '/export/home/oracle/
eBilling62/output/reportapp' || substr(file_location,length('/export/home/
oracle/eBilling/output/reportapp')+1) where file_location is not null;

SQL>commit;
```

3 If you installed Oracle Billing Insight version 7.0 in a directory other than the default EDX_HOME directory, or if you plan to use a customized batch report location (the default batch report location is output/reportapp), then you must update the properties in the reporting.batch.xma.xml file, located in the following directory:

- **UNIX.** *EDX_HOME*/xma/config/com/edocs/common/reporting/
- **Windows.** *EDX_HOME*\xma\config\com\edocs\common\reporting\

Specify your installation root directory in the rootDir property, and specify the custom report subdirectory in the path property (report files are stored in the rootDir/path directory, or the rootDir/path directory on Windows):

```
<!-- use this property to override the default base, the system property for
edx.home

<property name="rootDir"><value>C:/edocs</value></property>

-->

<property name="path">

<value>/output/reportapp</value>

</property>
```

Process of Migrating Oracle Self-Service E-Billing Version 6.1 to Oracle Billing Insight Version 7.0

This topic describes the process required to migrate Oracle Self-Service E-Billing version 6.1 to Oracle Billing Insight version 7.0.

NOTE: The process of migrating from version 6.1 to 7.0 is a single migration (direct path); no incremental migrations are necessary.

To migrate from Oracle Self-Service E-Billing version 6.1 to Oracle Billing Insight version 7.0, perform the following tasks:

1 Back up your existing OLTP and OLAP Oracle Self-Service E-Billing databases.

For additional information, see Oracle Database 11*g* documentation on Oracle Technology Network.

2. Start the database instance that accesses the database you are upgrading. Check the status of all user objects. If any of them indicate an INVALID status, then contact the database administrator to correct this problem.

   For additional information, see Oracle Database 11*g* documentation on Oracle Technology Network.

3. Have any existing Oracle Billing Insight database passwords available.

   Check with your database administrator.

4. Process all pending notifications in Oracle Self-Service E-Billing version 6.1.

   For details on how to run the Notifier job, see *Administration Guide for Oracle Self-Service E-Billing*, version 6.1.

5. Process all pending batch reports in Oracle Self-Service E-Billing version 6.1.

   For details on how to run the batch report jobs, see *Administration Guide for Oracle Self-Service E-Billing*, version 6.1.

6. Write down your payment gateway and job configuration settings as shown in the Command Center. You must reenter the settings after migrating.

7. Write down your Command Center admin user names. You must recreate these users after migrating.

8. Upgrade your Oracle Database 11*g* to Oracle Database 12*c*.

   For additional information, see Oracle Database 12*c* documentation on Oracle Technology Network.

9. If you implemented Transparent Data Encryption with the the Oracle Self-Service E-Billing version 6.1 OLTP database, make sure the Oracle Wallet is open.

10. Perform the following installation tasks:

    a. "Checking the Integrity of the Oracle Billing Insight Installer Package" on page 12

    b. "Installing the Oracle Billing Insight Software Using InstallAnywhere" on page 13.

    c. Go to My Oracle Support (MOS) and open a Service Request (SR) to receive the Oracle Billing Insight v7.0.0.1 patch.

    d. Follow the steps in the Patch 1 release notes to install the patch.

    e. "Configuring Log File Paths for Log4j" on page 15

11. Perform the following migration tasks:

    a. "Migrating Oracle Self-Service E-Billing Version 6.1 OLTP to Oracle Billing Insight Version 7.0" on page 124

    b. "Migrating Oracle Self-Service E-Billing Version 6.1 OLAP to Oracle Billing Insight Version 7.0" on page 125

    c. "Migrating Batch Reports from Oracle Self-Service E-Billing Version 6.1 to Oracle Billing Insight Version 7.0" on page 127

12 If you implemented Transparent Data Encryption with the the Oracle Self-Service E-Billing version 6.1 database, then reimplement it on the version 7.0 database. To implement Transparent Data Encryption on the EBILL (OLAP) instance, see "Choosing a Database Encryption Method" on page 32.

13 "Prepopulating and Migrating Product Catalog Tables" on page 128.

14 Run the following command on both the OLTP and OLAP schemas to verify that all objects are valid. The correct result is zero. Connect as the OLTP and OLAP schema owner.

```
select count(*) from user_objects where status ='INVALID';
```

15 If you ran the master key update process on Oracle Billing Insight version 6.1, then you must copy the version 6.1 *EDX_HOME*/keystore directory to version 7.0. Otherwise, encryption and decryption will not work.

16 Configure your application server for Oracle Billing Insight version 7.0.

Follow "Roadmap for Configuring Oracle WebLogic for Oracle Billing Insight" on page 47.

17 Follow the tasks and processes in "Roadmap for Configuring the Data Load Processes for Oracle Billing Insight" on page 107.

18 Uninstall Oracle Self-Service E-Billing version 6.1.

For information about uninstalling, see "Uninstalling Oracle Billing Insight" on page 21.

19 Enable automatic migration of user passwords. Open the cryptography.xma.xml file, located in the following directory, and set the passwordMigrationEnabled property to true:

- **UNIX.** *EDX_HOME*\xma\config\modules\cryptography
- **Windows.** *EDX_HOME*/xma/config/modules/cryptography

Migrating passwords enables the Self-Service and Assisted Service application users to log in with their passwords from version 6.1.

20 Recreate the Command Center admin users, using the bootstrap user.

For details, see *Administration Guide for Oracle Billing Insight*.

21 Recreate the payment gateway and jobs, using the Command Center.

For details, see *Administration Guide for Oracle Billing Insight*.

Migrating Oracle Self-Service E-Billing Version 6.1 OLTP to Oracle Billing Insight Version 7.0

Follow these steps to migrate Oracle Self-Service E-Billing version 6.1 OLTP to Oracle Billing Insight version 7.0.

This task is a step in "Process of Migrating Oracle Self-Service E-Billing Version 6.1 to Oracle Billing Insight Version 7.0" on page 122.

To migrate Oracle Self-Service E-Billing version 6.1 OLTP to Oracle Billing Insight version 7.0

1. On UNIX, log in as the ORACLE user for migration activity.
2. Go to the following directory:
   - **UNIX.** *EDX_HOME*/db/ebilling/oracle/oltp/migration/ebilling6.1_to_7.0
   - **Windows.** *EDX_HOME*\db\ebilling\oracle\oltp\migration\ebilling6.1_to_7.0

   In the path, EDX_HOME is the location where Oracle Billing Insight is installed.
3. Open the migrate_oltp_6.1_to_7.0.properties file, and set the correct value for each property in the file.

Property	What to Set
OLTP_USER	OLTP schema user name
OLTP_PASSWD	OLTP schema password
OLTP_SID	OLTP SID (system identifier)
OLTP_SYS_PASSWD	OLTP SYS password
OLAP_USER	OLAP schema user name
OLAP_PASSWD	OLAP schema password
EBILL_SID	Your existing EBILL SID (instance name)
SYS_PASSWD	sys password

4. Save and close the migrate_oltp_6.1_to_7.0.properties file.
5. Run the following command to start the migration:

```
ant -f migrate_oltp_6.1_to_7.0.xml
```

6. Select Option 1, Export OLTP schema.
7. Select Option 2, Create OLTP tablespaces and OLTP user.
8. Select Option 3, Import OLTP schema into OLAP instance.
9. Select Option 4, Migrate OLTP schema.
10. Select Option 5, Migrate OLTP data.
11. Review all log files for possible errors.

Migrating Oracle Self-Service E-Billing Version 6.1 OLAP to Oracle Billing Insight Version 7.0

Follow these steps to migrate Oracle Billing Insight version 6.1 OLAP to version 7.0 for both UNIX and Windows.

This task is a step in "Process of Migrating Oracle Self-Service E-Billing Version 6.1 to Oracle Billing Insight Version 7.0" on page 122.

To migrate Oracle Billing Insight version 6.1 OLAP to version 7.0

1 On UNIX, log in as the ORACLE user for migration activity.

2 Go to the following directory:

- **UNIX.** *EDX_HOME*/db/oracle/olap/migration/ebilling6.1_to_7.0
- **Windows.** *EDX_HOME*\db\oracle\olap\migration\ebilling6.1_to_7.0

In the path, EDX_HOME is the location where Oracle Billing Insight is installed.

3 Open the migrate_olap_6.1_to_7.0.properties file, and set the correct value for each property in the file.

Property	What to Set
OLAP_USER	OLAP schema user name
OLAP_PASSWD	OLAP schema password
EBILL_SID	Your existing EBILL SID (instance name)
TOPX_REC_CAP	The cap on the number of records to be calculated for TopX tables. The default is 51. Changing this value impacts the processing time for the data load process (post load process). Do Not Exceed 100.
OLTP_USER	OLTP schema user name
OLTP_PASSWD	OLTP schema password
SYS_PASSWD	sys password

4 Save and close the migrate_olap_6.1_to_7.0.properties file.

5 Run the following command to start the migration:

```
ant -f migrate_olap_6.1_to_7.0.xml
```

6 Select Option 1, Migrate OLAP Schema.

7 Select Option 2, Migrate OLAP data.

8 Review all log files for possible errors.

Migrating Batch Reports from Oracle Self-Service E-Billing Version 6.1 to Oracle Billing Insight Version 7.0

After you have successfully migrated from Oracle Self-Service E-Billing version 6.1 to Oracle Billing Insight version 7.0, you must copy the batch reports to the directory you want to use for reports in the Oracle Billing Insight version 7.0.

If you copy the reports to a different location than you used in version 6.1, then you must update the batch report file location in the OLTP database. Also, if you installed Oracle Billing Insight version 7.0 in a location other than the default EDX_HOME, or if you create a custom subdirectory for storing batch reports (the default directory is output/reportapp), then you must update the related properties in the reporting.batch.xma.xml file.

This task is a step in "Process of Migrating Oracle Self-Service E-Billing Version 6.1 to Oracle Billing Insight Version 7.0" on page 122.

To migrate your batch reports from Oracle Self-Service E-Billing version 6.1 to Oracle Billing Insight version 7.0

1. On the application server, copy your existing batch reports to the following directory:
   - **UNIX.** *EDX_HOME*\output\reportapp
   - **Windows.** *EDX_HOME*/output/reportapp

   In the directory, *EDX_HOME* is the directory where you installed Oracle Self-Service E-Billing version 6.1.

2. If you copy your batch reports to a different file location in version 7.0 than you used in version 6.1, then you must update the batch report file location in the OLTP database. Log on to the OLTP schema, using SQL*Plus, and run the following script:

   ```
   update edx_rpt_batch_report set file_location = 'NEW_LOCATION' ||
   substr(file_location,length('OLD_LOCATION')+1) where file_location is not null;

   SQL>commit;
   ```

   where:
   - *NEW_LOCATION* is the new location where you will store batch reports in Oracle Billing Insight version 7.0.
   - *OLD_LOCATION* is the old location where you stored batch reports in Oracle Billing Insight version 6.1.

   For example:

   ```
   SQL> update edx_rpt_batch_report set file_location = '/export/home/oracle/
   eBilling62/output/reportapp' || substr(file_location,length('/export/home/
   oracle/eBilling/output/reportapp')+1) where file_location is not null;

   SQL>commit;
   ```

3 If you installed Oracle Billing Insight version 7.0 in a directory other than the default EDX_HOME directory, or if you plan to use a customized batch report location (the default batch report location is output/reportapp), then you must update the properties in the reporting.batch.xma.xml file, located in the following directory:

- **UNIX.** *EDX_HOME*/xma/config/com/edocs/common/reporting/
- **Windows.** *EDX_HOME*\xma\config\com\edocs\common\reporting\

Specify your installation root directory in the rootDir property, and specify the custom report subdirectory in the path property (report files are stored in the rootDir/path directory, or the rootDir/path directory on Windows):

```
<!-- use this property to override the default base, the system property for
edx.home

<property name="rootDir"><value>C:/edocs</value></property>

-->

<property name="path">

<value>/output/reportapp</value>

</property>
```

Prepopulating and Migrating Product Catalog Tables

You must prepopulate the product catalog tables with new data required for Oracle Billing Insight Version 7.0, and then migrate your existing data. The product catalog tables that you must update contain the following types of information:

- **EDX_RPT_PLAN_TYPE_DIM**. Brief definitions of plan types. For example, some organizations have Family and Individual plan types, or options such as Road-Side Assistance.
- **EDX_RPT_PRODUCT_DIM.** Brief product definitions.
- **EDX_RPT_SUB_PRODUCT_DIM.** Details of what the product comprises.
- **EDX_RPT_SERVICE_PRODUCT_FACT.** Costs of the plans, or products, add-on products acquired.

Specifications for the Oracle Billing Insight database tables are described in the Data Dictionary, provided in your product. For details on accessing the Data Dictionary as well as additional details on file specifications required with the ETL process, see *Implementation Guide for Oracle Billing Insight*.

To prepopulate and migrate product catalog tables

1 Prepopulate the product catalog tables with required data. Create a Master data file with the following records, and load it into the Oracle Billing Insight database using the ETL process.

- For each plan type, add a record type 140 with the following columns in the EDX_RPT_PLAN_TYPE_DIM table.

  A default entry with a value of UNK appears in the column PLAN_TYPE_CD after installation. This value represents Unknown plan types.

EDX_RPT_PLAN_TYPE_DIM Column Name	Description
PLAN_TYPE_KEY	A surrogate key that uniquely identifies a record that uses the sequence table, SEQ_EDX_RPT_PLAN_TYPE_DIM.
PLAN_TYPE_CD	A unique code that identifies the plan type.
PLAN_TYPE_NAME	The name of the plan type, such as Current Calling Plan, Insurance Plan, and so on.

- For each product, add a record type 150 with the following columns in the EDX_RPT_PRODUCT_DIM table.

  A default entry with a value of UNK appears in the column PRODUCT_CD after installation. This value represents Unknown products.

EDX_RPT_PRODUCT_DIM Column Name	Description
PRODUCT_KEY	A surrogate key that uniquely identifies a record that uses the sequence table SEQ_EDX_RPT_PRODUCT_DIM.
PRODUCT_CD	A unique code that identifies the product.
PRODUCT_NAME	The product name, such as Anywhere Anytime 1500.
ADD_ON_FLAG	Identifies add-on products. *Add-ons* are products that consumers purchase in addition to standard plans, such as Roadside Assistance, Insurance, and so on.
PRODUCT_DESC	The product description that displays in the applications, such as Anywhere Anytime 1500, 20$ per month, contains 1500 free minutes.

- For each subproduct, add a record type 150 with the following columns in the EDX_RPT_SUB_PRODUCT_DIM table.

  A default entry with a value of UNK appears in the column SUB_PRODUCT_CD after installation. This value represents Unknown products.

EDX_RPT_SUB_PRODUCT_DIM Column Name	Description
PRODUCT_KEY	A surrogate key that uniquely identifies a record that uses the sequence table, SEQ_EDX_RPT_PRODUCT_DIM.
SUB_PRODUCT_KEY	A surrogate key that uniquely identifies a record that uses the sequence table, SEQ_EDX_RPT_SUB_PRODUCT_DIM. The combination of PRODUCT_KEY and SUB_PRODUCT_KEY make up the primary key for the EDX_RPT_SUB_PRODUCT_DIM table.
SUB_PRODUCT_CD	A code that identifies the subproduct.
SUB_PRODUCT_NAME	The name of the subproduct.
SUB_PRODUCT_DESC	The description of the subproduct.

- For each product, add a record type 3200 with the following columns in the EDX_RPT_SERVICE_PRODUCT_FACT table.

EDX_RPT_SERVICE_PRODUCT_FACT Column Name	Description
PRODUCT_CHILD_KEY	A reference to the SUB_PRODUCT_KEY column in the EDX_RPT_SUB_PRODUCT_DIM table.
SERVICE_KEY	A reference to the SERVICE_KEY column in the EDX_OMF_SERVICEAGREEMENT table in the OLTP schema.
ACCOUNT_KEY	A reference to the ACCOUNT_KEY column in the EDX_OMF_SERVICEAGREEMENT table in the OLTP schema.
PERIOD_KEY	A reference to the PERIOD_KEY column in the EDX_RPT_PERIOD_DIM table.
SERVICE_TYPE_KEY	A reference to the SERVICE_TYPE_KEY column in the EDX_RPT_SERVICE_TYPE_DIM table (if used).
PRODUCT_PARENT_KEY	A reference to the PRODUCT_KEY column in the EDX_RPT_PRODUCT_DIM table.
CHARGE_AMT	The product cost.
STATEMENT_KEY	A reference to the STATEMENT_KEY column in the EDX_RPT_STATEMENT_FACT table.

EDX_RPT_SERVICE_PRODUCT _FACT Column Name	Description
ETL_KEY	A reference to the ETL_FILE_KEY column in the EDX_RPT_ETL_FILE_DIM table.
PLAN_TYPE_KEY	A reference to the PLAN_TYPE_KEY column in the EDX_RPT_PLAN_TYPE_DIM table.
PRODUCT_NOTE	A product note.
DISPUTE_DETAIL_ID	A surrogate key to uniquely identify the dispute detail. This is a reference to the EDX_RPT_DISPUTE_DETAIL table.
GROUP_KEY	A group key reference to the EDX_RPT_PARTN_MGMT table.

2 Migrate your existing product catalog data. You must build a SQL script to ensure that your usage of the product catalog tables prior to Oracle Billing Insight version 7.0 properly conforms to the following standards:

- **EDX_RPT_PLAN_TYPE_DIM.** Data stored in this table can remain the same if you used this table to associate plans to a plan type.
- **EDX_RPT_PRODUCT_DIM.** The ADD_ON_FLAG column was added in Oracle Self-Service E-Billing Version 6.2, with a default value of N. Determine which of your products are add-on's and update those records as follows:
  - Use the following statement to update the product key value: UPDATE EDX_RPT_PRODUCT_DIM SET ADD_ON_FLAG = 'Y' WHERE PRODUCT_KEY = *product_key_value*)
  - Populate the PRODUCT_DESC column with the description to display in applications. This column was added in Oracle Billing Insight version 7.0.

  You can optionally use an ETL load to populate this table. Column 11 (position 12) on record type 150 is where the ADD ON FLAG value is located. Column 12 (position 13) on record type 150 is where the PRODUCT_DESC value is located. You can reload all of your products with a value for the ADD_ON_FLAG column and the PRODUCT_DESC column.
- **EDX_RPT_SUB_PRODUCT_DIM.** Each product in the EDX_RPT_PRODUCT_DIM table must have its own set of records in the EDX_RPT_SUB_PRODUCT_DIM table:
  - The ALLOWANCE_ID, SUB_PRODUCT_USAGE, SUB_PRODUCT_UNIT columns have been deprecated in Oracle Billing Insight version 7.0.
  - Populate the SUB_PRODUCT_DESC column with the description to display in applications. This column was added in Oracle Billing Insight version 7.0.

  You can optionally use an ETL load to populate this table. Column 13 (position 14) on record type 150 is where the SUB_PRODUCT_DESC column is located. You can reload all of your subproducts with a value for the SUB_PRODUCT_DESC column.
- **EDX_RPT_SERVICE_PRODUCT_FACT.** Data must be stored in the columns as follows:

- **PRODUCT_CHILD_KEY.** The value of the SUB_PRODUCT_KEY from the EDX_RPT_SUB_PRODUCT_DIM table.
- **PRODUCT_PARENT_KEY.** The value of the PRODUCT_KEY from the EDX_RPT_PRODUCT_DIM table.
- **PLAN_TYPE_KEY.** The value of the PLAN_TYPE_KEY from the EDX_RPT_PLAN_TYPE_DIM table.

Examples of Populated Product Catalog Tables

The following examples show sample data records in the product catalog tables.

Example 1- Family Circle 750 Calling Plan

This example shows how data can be set up in the product catalog tables for a calling plan called Family Circle 750.

Table 9 shows a sample data record for a plan type called Family Plan in the EDX_RPT_PLAN_TYPE_DIM table.

Table 9. Sample Data Record in the EDX_RPT_PLAN_TYPE_DIM Table for Family Plan

Record	Column Name	Value	Description
1	PLAN_CD	fam_plan	Fam_plan is the code specified for this plan type.
1	PLAN_NAME	Family Plan	Family Plan is the name specified for this plan type.

Table 10 shows a sample data record for the Family Circle 750 product in the EDX_RPT_PRODUCT_DIM table.

Table 10. Sample Data Record in the EDX_RPT_PRODUCT_DIM Table for the Family Circle 750 Product

Record	Column Name	Value	Description
1	PRODUCT_CD	Family circle 750	Family circle 750 is the code specified for this product, or plan.
1	PRODUCT_NAME	Family Circle 750	Family circle 750 is the name specified for this product, or plan.
1	ADD_ON_FLAG	N	N indicates that this not an add-on product.
1	PRODUCT_DESC	Family Circle 750, $20 per month, 750 free minutes per month	This is the description of the product that displays in applications.

Table 11 shows three sample data records for the subproduct components of the Family Circle 750 plan in the EDX_RPT_SUB_PRODUCT_DIM table.

Table 11. Sample Records in the EDX_RPT_SUB_PRODUCT_DIM Table for the Family Circle 750 Product Components

Record	Column Name	Value	Description
1	PRODUCT_KEY	Family circle 750	Family circle 750 is the code value from the EDX_RPT_PRODUCT_DIM table.
1	SUB_PRODUCT_CD	Access Charge	Access Charge is the code specified for this subproduct.
1	SUB_PRODUCT_NAME	Month Access Charges	Month Access Charges is the name specified for this subproduct.
1	SUB_PRODUCT_DESC	74.99 per month	This is the description for the subproduct that displays in applications.
2	PRODUCT_KEY	Family circle 750	Family circle 750 is the code value from the EDX_RPT_PRODUCT_DIM table.
2	SUB_PRODUCT_CD	Roaming	Roaming is the code specified for this subproduct.
2	SUB_PRODUCT_NAME	Roaming Charges	Roaming Charges is the name specified for this subproduct.
2	SUB_PRODUCT_DESC	Charge per 0.15 minute	This is the description of the subproduct that displays in applications.
3	PRODUCT_KEY	Family circle 750	Family circle 750 is the code value from the EDX_RPT_PRODUCT_DIM table.
3	SUB_PRODUCT_CD	ADDTL Line	ADDTL Line is the code specified for this subproduct.
3	SUB_PRODUCT_NAME	Additional Line Monthly Access Charge	Additional Line Monthly Access Charge is the name specified for this subproduct.
3	SUB_PRODUCT_DESC	$9.99 one additional line per month	This is the description of the subproduct that displays in applications.

Example 2- Insurance Plan

This example shows how data can be set up in the tables for additional insurance options.

Table 12 shows a sample data record for a plan type called Insurance in the EDX_RPT_PLAN_TYPE_DIM table.

Table 12. Sample Data Record in the EDX_RPT_PLAN_TYPE_DIM Table for the Insurance Plan Type

Record	Column Name	Value	Description
1	PLAN_CD	Ins_plan	Ins_plan is the code specified for this plan type.
1	PLAN_NAME	Insurance	Insurance is the name specified for this plan type.

Table 13 shows a sample data record for the Insurance plan product in the EDX_RPT_PRODUCT_DIM table.

Table 13. Sample Data Record in the EDX_RPT_PRODUCT_DIM Table for the Insurance Product

Record	Column Name	Value	Description
1	PRODUCT_CD	Insurance	Insurance is the code specified for this product, or plan.
1	PRODUCT_NAME	Insurance Option	Insurance Options is the name specified for this product, or plan.
1	ADD_ON_FLAG	Y	Y indicates that this is an add-on product.
1	PRODUCT_DESC	Phone insurance add-on plan	This is the description of the product that displays in applications.

Table 14 shows a sample data record for the subproduct components of the Insurance plan in the EDX_RPT_SUB_PRODUCT_DIM table.

Table 14. Sample Records in the EDX_RPT_SUB_PRODUCT_DIM Table for the Insurance Product Components

Record	Column Name	Value	Description
1	PRODUCT_KEY	Insurance	Insurance is the code value from the EDX_RPT_PRODUCT_DIM table.
1	SUB_PRODUCT_CD	Phone Basic	Phone Basic is the code specified for this subproduct.
1	SUB_PRODUCT_NAME	Insurance Basic Charges	Insurance Basic Charges is the name specified for this subproduct.
1	SUB_PRODUCT_DESC	$4.99 per month	This is the description of the subproduct that displays in applications.

Example 1 - Customer Service Plan

This example shows how to populate the EDX_RPT_SERVICE_PRODUCT_FACT table for a customer's chosen service plan.

In this example, the customer chose the Family Circle 750 plan for $74.99 along with the Basic Phone insurance option for $4.99. This customer did not choose an additional line with their family plan. Table 15 shows two records for these two service plan products in the EDX_RPT_SERVICE_PRODUCT_FACT table.

Table 15. Sample Data Records for a Customer Service Plan in the EDX_RPT_SERVICE_PRODUCT_FACT Table

Record	Column Name	Value	Description
1	PLAN_TYPE_KEY	fam_plan	Fam_plan is the code value from the EDX_RPT_PLAN_TYPE_DIM table.
1	PRODUCT_PARENT_KEY	Family circle 750	Family circle 750 is the code value from the EDX_RPT_PRODUCT_DIM table.
1	PRODUCT_CHILD_KEY	Access Charge	Access Charge is the code value from the EDX_RPT_SUB_PRODUCT_DIM table.
1	CHARGE_AMT	74.99	74.99 is the charged value, which is the usage value in the EDX_RPT_SUB_PRODUCT_DIM table.
1	SERVICE_KEY	555-555-5555	555-555-5555 is the service number value from the EDX_RPT_SERVICE_DIM table.
1	ACCOUNT_KEY	ACCT123	ACCT123 is the account number value from the EDX_RPT_ACCOUNT_DIM table.
2	PLAN_TYPE_KEY	ins_plan	Ins_plan is the code value from the EDX_RPT_PLAN_TYPE_DIM table.
2	PRODUCT_PARENT_KEY	Insurance	Insurance is the code value from the EDX_RPT_PRODUCT_DIM table.
2	PRODUCT_CHILD_KEY	Phone Basic	Access Charge is the code value from the EDX_RPT_SUB_PRODUCT_DIM table.
2	CHARGE_AMT	4.99	4.99 is the charged value, which is the usage value in the EDX_RPT_SUB_PRODUCT_DIM table.

Table 15. Sample Data Records for a Customer Service Plan in the EDX_RPT_SERVICE_PRODUCT_FACT Table

Record	Column Name	Value	Description
2	SERVICE_KEY	555-555-5555	555-555-5555 is the service number value from the EDX_RPT_SERVICE_DIM table.
2	ACCOUNT_KEY	ACCT123	ACCT123 is the account number value from the EDX_RPT_ACCOUNT_DIM table.

Index

O

P

R

S

T

U

W