Installation Guide for Communications Billing Analytics

Sun Solaris Operating Environment Software / AIX, Oracle database, the IBM WebSphere® Server

Version 5.1.0.2

Pub Date 6/1/2006

ORACLE®

Contents

8   Uninstalling Communications Billing Analytics

1 Preface

About This Guide

This guide is intended for system administrators and other IT professionals. It describes how to install and configure the third-party platforms that support the production environment of Oracle's Communications Billing Analytics product (often referred to in this guide as Billing Analytics or CBA) and deploy Billing Analytics J2EE web applications. See "Communications Billing Analytics System Requirements" on page10 for details on the platforms this guide is intended for.

This guide assumes in-depth understanding of and practical experience with system administrator responsibilities, including the items under the headings that follow.

Operating System Administration Requirements

- Start up and shut down of the system
- Log in and out of the system
- Determine software patch/pack levels
- Install software & patches/packs
- Navigate the file system
- Manipulate text files
- Create files and directories
- Change permissions of files and directories
- Use basic network commands
- Transfer files with FTP
- Monitor processes & system resource usage
- Perform system backups and recovery
- Implement system security

Database Administration Requirements

- Install and configure your database server
- Start and stop your database server and database instances
- Use administrative tools
- Manage users, privileges, and resources
- Create an operational database

- Manage database files

- Manage tables and indexes

- Back up and restore databases

- Monitor database performance

Application Server Administration Requirements

- Install and configure your application server

- Start and stop your application server

- Use administrative tools

- Manage users, privileges, and resources

- Configure Java resources

- Package and deploy web applications

- Monitor application server performance

This guide does *not* describe general UNIX system administration. See the appropriate UNIX user documentation.

Related Documentation

A PDF version of this guide is also available.

Online	How to Access
A PDF of this guide	A PDF of this guide is available on the product CD-ROM.

This guide is part of the Billing Analytics documentation set. For more information about using Billing Analytics, see the following guides:

Deploying and Customizing J2EE Applications	How to customize J2EE Web applications for deployment with Billing Analytics.
Billing Analytics Administration Guide	How to set up and run a live Billing Analytics application in a J2EE environment.

2 Getting Started

Preparing Your Platform

Before installing Communications Billing Analytics, verify that your platform is ready:

- Install and test required hardware and software for your platform.

- Define required user and group permissions for your database server and application server.

- Start and test your database server. For details, see your server documentation.

- Start and test your application server. For details, see your application server documentation.

- For distributed environments, make sure you have any required database client software installed on your application server and any other client machines of your database server.

Overview of the Installation Process

The process of installing and setting up Communications Billing Analytics includes the following steps:

1 Installing Oracle's Siebel Platform Services and Communications Billing Analytics products on your database and application servers.

2 Configuring the database server.

3 Configuring the application server.

Follow the chapters in this guide in sequence, consulting your third-party documentation as needed.

You must use the same user to install Communications Billing Analytics that you used to install WebLogic.

Once you successfully install Communications Billing Analytics and configure your database and application servers, you can customize and deploy your J2EE application.

To configure your database server

1 Define database server environment variables.

2 Create and configure the Communications Billing Analytics database with *edx_main_admin.sh*

3 Connect to your Communications Billing Analytics database before configuring your application server.

To configure your application server

- Configure JDBC resources for Communications Billing Analytics on your application server.

Deploying the Billing Analytics J2EE Application

After installing Communications Billing Analytics and configuring your database and application servers, you can:

- Deploy the J2EE web application for Communications Billing Analytics.

Communications Billing Analytics System Requirements

Platform Services and Communications Billing Analytics

This guide assumes you are installing Communications Billing Analytics on a Sun Solaris operating system, Oracle database, and WebLogic application server.

The following table lists the specific combinations supported for Communications Billing Analytics. **Required JDK versions**, **system patches, fix packs and other updates are not listed in this section**.

Be sure to check the Release Notes for any updates to these requirements.

Server O/S	Database	App Server	JRE
Solaris 9/AIX 5.1	Oracle 9.205	WebSphere 5.1	Sun JDK 1.4.1_05.

OPERATING SYSTEM

- Sun Solaris 9 or AIX

Note: the installation procedures for Solaris and AIX are virtually identical. However, where Solaris uses /opt, AIX uses /usr.

HARDWARE

- CD-ROM

- Disk space (database) 2.6 GB

- Disk space (software) 60 MB

- Swap space 512 MB per CPU (1 GB recommended)

- RAM 512 MB per CPU (1 GB recommended)

JAVA/C++

- Sun Java 2 SDK Standard Edition 1.4.1_05 (version shipped with WebSphere 5.1)

SUPPORTED DATABASE SERVERS

New Installation of Communications Billing Analytics

- Oracle 9i Release 2 (Oracle 9.2.0)

- Oracle 9i client software (for application server), including the sqlplus and sqlldr packages

- Oracle 9i JDBC driver

Supported Application Servers

- IBM WebSphere 5.1

Supported Browsers

- Netscape Navigator 7 or higher

- Microsoft Internet Explorer 6.0 or later (on networked PC)

3 Installing Communications Billing Analytics

This chapter provides a step-by-step guide to installing Communications Billing Analytics. It assumes that you have an in-depth understanding of and practical experience with administering your operating system. Consult your system documentation as necessary.

UNIX Permissions for Installation

You may use the **root** privilege on each server to install and uninstall Communications Billing Analytics components, for tasks such as creating pre-requisite users and install directories.

Recommended users for installing and running the CBA application are as listed below:

User	Group	2nd Group	Shell	Home directory
edxadmin	bea	dba	ksh	/export/home/edxadmin
oracle	dba	oinstall	ksh	/export/home/oracle

For your application and database servers, you also need the owner (user) and group permissions specified during installation. For details on user and group permissions, see "UNIX Permissions for Your Database Server" on page 17.

CAUTION: Oracle does not recommend administering Communications Billing Analytics with the user and group nobody:nobody.

Installing Communications Billing Analytics

Communications Billing Analytics (CBA, also referred to as Telco Analytics Manager, or TAM) is distributed as an InstallAnywhere package. Follow the steps below to install Billing Analytics on your system. This document refers to the installation directory as the CBA_HOME directory, which defaults to */opt/siebel/CBA*. Oracle recommends the following directory structure:

```
/opt/<projectname>/Siebel/CBA
/opt/<projectname>/Siebel/dist
/opt/<projectname>/Siebel/DROPS

/opt/<projectname>/bea/domains
/opt/<projectname>/bea/jdk142_08

/opt/<projectname>/apache
```

You can change the default installation directory when prompted during the installation procedure. This guide uses the generic term CBA_HOME to define the installation directory in the examples.

$CBA_HOME/db/oracle contains platform-specific subdirectories for database creation and configuration.

$CBA_HOME/J2EEApps contains the web applications to be deployed to your application server.

During the installation procedure, you are prompted to enter the user and group identifier of the Web Application Server owner. Oracle recommends that you use the default Web Application Server owner and group accounts.

To install Billing Analytics

1	Log in as the relevant Unix user on the application server.

2	After you obtain and locate the Communications Billing Analytics software installer as described in the Preface of this guide, you can run the following from a command prompt at the directory location where the installer resides:

	`$ ./TAMins.bin -i console`

3	On the Introduction screen, read the introductory information. Click **Next** to continue.

4	On the License Agreement screen, carefully read the licensing agreement, select the acceptance button, and then click **Next**.

5	On the Enter Serial Number screen, enter your product serial number. It is stapled to the inside front cover of this guide (if your serial number has been misplaced, contact Oracle Technical Support). Then click **Next**.

6	On the Owner of Web Application Server screen, enter the name of the application server owner (if you have installed other applications in Oracle's Siebel Self-Service product family, use the same owner here that you used for those product installations). Then click **Next**.

7	On the Group of Web Application Server screen, enter the name of the group for the application server (if you have installed other applications in Oracle's Siebel Self-Service product family, use the same group here that you used for those product installations). Then click **Next**.

8	On the Choose Install Folder screen, accept the default installation folder or click **Choose** and enter the directory where you want to install the Billing Analytics files and directories. This document refers to that directory as CBA_HOME. Click the **Next** button to continue.

9	On the Choose Product Features screen, click **CBA Group**. Then click **Next**.

10	On the Pre-Installation Summary screen, verify that the information is correct, and click on **Install**. To correct any entries, click **Previous**, and then return here.

	At this point, the Billing Analytics database server components are copied to the designated installation folder. A status bar on the bottom of the screen shows each database server component being installed. No user intervention is required.

11	The release notes display inside the installer window.

12	The Install Complete screen reports a successful installation and the directory that contains the Billing Analytics components.

13	Click **Next** to view the release notes. Then click Next when you are done reading the release notes.

14	 Click **Done** to exit the installer.

If the installation fails, determine the cause of the problem and run InstallAnywhere again to reinstall Communications Billing Analytics.

4   Configuring the Database Server

Overview

This chapter assumes in-depth understanding of and practical experience with database administration. Consult your database documentation as necessary. For distributed environments, make sure you have any required database client software installed on your application server and any other client machines of your database server.

Oracle recommends that you install and configure Billing Analytics in the same top-level directory structure, first on the database server, then the application server.

This chapter provides instructions for configuring your database server to support a **new** Billing Analytics database. It includes:

- ■   UNIX permissions for your database server

- ■   Starting and stopping your database server

- ■   UNIX environment variables for your database server

- ■   Using database partitioning with Billing Analytics

CAUTION: The installation and configuration examples shown in this guide use default Billing Analytics pathnames, privileges, and permissions. If you choose not to accept the default values, make sure your values are consistent on all servers across your installation of Billing Analytics.

UNIX Permissions for Your Database Server

Before creating the Billing Analytics databases, you should verify that the owner and group permissions (`userid:groupid`) of the Billing Analytics database directory, including all subfolders, are set to the **DB Admin user** defined during database installation.

You should install Billing Analytics database components with the default owner and group for your platform. After installation, change the user and group ownership of Billing Analytics database server components to that of the **DB Admin user**.

User	Group	2nd Group	Shell	Home directory
oracle	dba	oinstall	ksh	/export/home/oracle

CAUTION: The DB Admin user has special privileges on Oracle. For details on owner and group permissions for your database server, please consult the database documentation for your platform.

If your database administrator uses custom user and group permissions, you can reset these permissions with the `chown` command.

Developers and system administrators need to be familiar with how to stop and start a database server and an active Billing Analytics database instance for your platform.

For details on starting and stopping your database server and instances, please consult the database documentation for your platform.

UNIX Environment Variables for Your Database Server

Before you can configure the database to support Billing Analytics, you must define environment variables for your database server.

Oracle Database Server Environment Variables

Below are some sample Oracle user variables that are required to be set in the shell.

Variable	Definition	Example
CBA_HOME	CBA home path	/opt/Siebel/CBA
ORACLE_BASE	Mount point base path	/u01/oracle
ORACLE_HOME	Data File Path	$ORACLE_BASE/product/9.2.0
PATH	Database Path	.:$ORACLE_HOME/bin:$PATH
LD_LIBRARY_PATH	Shared Library Path	$ORACLE_HOME/lib:/usr/lib:/usr/ucblib
SHLB_PATH	Shared Library Path	$ORACLE_HOME/lib:$ORACLE_HOME/lib64:$ORACLE_HOME/lib32
ORACLE_DATA	Data File Path	/u01/oradata
ORACLE_SID	Database instance name	oltp

The syntax used to define environment variables depends on which UNIX shell you are using.

NOTE: Be sure to export each variable after setting it. It is recommended these be set in the user's .profile file. For example, in the Korn Shell:

```
CBA_HOME=/opt/Siebel/CBA
ORACLE_BASE=/u01/oracle
ORACLE_HOME=$ORACLE_BASE/product/9.2.0
PATH=.:$ORACLE_HOME/bin:$PATH
LD_LIBRARY_PATH=$ORACLE_HOME/lib:/usr/lib:/usr/ucblib:/usr/ccs/lib
SHLIB_PATH=$ORACLE_HOME/lib:$ORACLE_HOME/lib64:$ORACLE_HOME/lib32
ORACLE_DATA=/u01/oradata
ORACLE_SID=oltp

export CBA_HOME ORACLE_BASE ORACLE_HOME PATH LD_LIBRARY_PATH
```

```
export SHLIB_PATH ORACLE_DATA ORACLE_SID
```

Using Database Partitioning

Database partitioning (partition splitting) reduces the number of tables the system must scan when indexing your data. You specify the number of partitions when you create a DDN in the Command Center. At the first run of the Indexer job, Billing Analytics creates and populates a set of partitioned index tables to maintain your dynamic data.

Oracle no longer supports partitioned views. Native partitioning can be applied to a single index table depending on your Oracle software license. For an Oracle database, we recommend you create one index table per DDN, and use Oracle's native table partitioning functionality for higher performance.

5 Configuring Communications Billing Analytics for Oracle

Configuring a New Oracle Database for UNIX

Before configuring Oracle, you must first complete the instructions in the previous chapter to define UNIX environment variables for your database server.

To create and configure your Billing Analytics database, you can either run the database configuration shell script `edx_main_admin.sh` or the ant script `build.xml`. This topic describes each alternative's steps in detail.

TIP: Database clustering is handled by your application server and not by Billing Analytics. Consult your Oracle Technical Services or Professional Services representative for clustered installations.

About edx_main_admin.sh for Oracle

To create and configure the Billing Analytics production database, you run the `edx_main_admin.sh` script for database configuration. If you have to abort database setup before it completes successfully, see "What to Do if Database Configuration Fails" on page 31.

Before running `edx_main_admin.sh`, you should:

- Upgrade your database server software **as necessary**.

- Make a full backup of your current database.

- 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, contact the database administrator to correct this problem.

- Confirm that all login sessions using the Billing Analytics database user have logged out of the instance.

- Have any required database passwords available. Check with your database administrator for custom passwords.

- Check the *Release Notes* for disk space requirements and confirm that you have sufficient disk space on your database server. Insufficient disk space can cause database configuration to fail.

- Configure `tnsnames.ora`, as described in the next section.

Included Files:

- `edx_rpt_admin.sh`: The main file to be run as "oracle" user on a UNIX machine to set up a new instance. The file will prompt for various options before finally creating a new instance. Please make sure that a shell file has "execute privileges" which can be granted by running the command "chmod +x edx_rpt_admin.sh".

- `configure_ts.sql`: This file creates tablespaces specific for the application and is being invoked from the main shell script.

- `crt_rpt_user.sql`: The file creates a new schema and is being invoked from the main shell script.

- `init.ora`: Base initialization file used for setting up a new database.

Configuring Oracle Services

The next step in setting up the database server is to edit two Oracle configuration files that control access to the Billing Analytics production database.

TIP: Always consult with your onsite DBA and your Oracle Professional Services representative to configure database connectivity, to make sure you comply with client standards for the enterprise.

- **listener.ora** includes 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 listener.ora requires the SID (SID_NAME) of a database server in tnsnames.ora.

  You need to modify listener.ora on the database server machine.

- **tnsnames.ora** includes a list of service names of network databases that are mapped to connect descriptors. It is used by clients and distributed database servers to identify potential server destinations. The address of a given database server in tnsnames.ora matches the address of a listener for that server in listener.ora.

  You need to modify tnsnames.ora on the database client machine.

By default, these files are installed to the network administration directory of your database server, $ORACLE_HOME/network/admin.

NOTE: You need to manually set up the listener.ora and tnsnames.ora connectivity files before running the database creation scripts.

To configure Oracle services:

1 Switch user to the Oracle admin user.

   $ su - oracle

2 Change directory to the network administration directory of your database server. For example:

   cd /export/home/oracle/product/9.2.0/network/admin

3 Open listener.ora and edit the SID_LIST_LISTENER section to reflect your Oracle SID and database home directory. For example:

   (SID_DESC =
    (SID_NAME = oltp)

```
   (ORACLE_HOME = /export/home/oracle/product/9.2.0)
  )

  (SID_DESC =
   (SID_NAME = olap)
   (ORACLE_HOME = /export/home/oracle/product/9.2.0)
  )
```

4   Save and close listener.ora.

5   Change directory to the network administration directory of your database client. For example:

```
$  cd $ORACLE_HOME/network/admin
```

6   Open tnsnames.ora and edit the database service that identifies your protocol, host, and port. This example uses the service name edx.db (your service name might be different), installed on the database server localhost.

```
oltp =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SID = oltp)
    )
   )

olap =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SID = olap)
    )
   )
```

If you paste this into your *tnsnames.ora* file, be sure to update the HOST!

7   Save and close tnsnames.ora.

8   *(Single machine environments only)* Repeat Step 5 for the tnsnames.ora file on your application server. This file is installed with your database client software. Distributed environments may skip this step.

9   Stop and restart the Oracle listener with the listener control commands.

```
$  lsnrctl reload
```

10 After the Oracle listener has been restarted, you should see a service handler for the Billing Analytics instance.

```
Instance "olap", status READY, has 1 handler(s) for this service...
Instance "oltp", status READY, has 1 handler(s) for this service...
```

This service handler should match the name you entered for the Oracle SID during database configuration, in this example **olap** and **oltp**.

To use a database instance block size other than the default 4k (RECOMMENDED)

1   Open the template the template init.ora files for the OLAP and OLTP instances:

      a. $CBA_HOME/db/oracle/oltp/init.ora

      b. $CBA_HOME/db/oracle/olap/init.ora

2   Edit the **db_block_size** parameter to be 8k or 16k (which should match your file system):

    db_block_size = 8192

Configuring CBA OLTP and OLAP database instances for Oracle

1   Switch user to the Oracle admin user, in this example `oracle`. Oracle requires the administrative user in order to create files.

      $ su – oracle

2   Change directory to your Billing Analytics database home directory. For example:

      $ cd /opt/siebel/TAM/db/oracle

3   Start the `edx_main_admin` shell script with the command:

      $ ./edx_main_admin.sh

This will bring up the main menu page:

```
Siebel Server Administration Main Menu Version 1.0
--------------------------------------------------------
        [1] OLTP Setup
        [2] OLAP Setup
        [3] CREATE DB LINK
        [Q] Quit
--------------------------------------------------------
```

From this menu you should run through the steps in order. Each step may have its own sub-steps, which you should also run in order before proceeding to the next step.

4   Select Option **[1] OLTP Setup**. The OLTP set page appears:

```
Siebel e-Statement Server Administration Main Menu Version 1.0
--------------------------------------------------------
 [1] Sign in Menu
 [2] Capture Database File Locations
 [3] Install Siebel e-Statement
 [4] Initial Data Population
 [Q] Quit
--------------------------------------------------------
 Enter Your Selection: 1
```

5   Select Option 1, **Sign in Menu**. A second sign-in screen appears.

6   Enter the username, password, and database name for the Billing Analytics database.

CAUTION: When creating an Oracle database, limit its name to eight characters. Defining or entering an Oracle SID with more than eight characters causes the Oracle database configuration to fail.

The following example shows sample values for **Oracle** on **Solaris**. See "Oracle Database
Server Environment Variables" on page 18 and your database administrator to specify suitable
values for your platform.

```
SIGN IN MENU
---------------------------------------------
 [1] Enter Database USERNAME...> cba_dba
 [2] Enter Database PASSWORD...> cba
 [3] Enter ORACLE_SID ...> oltp
 [4] Enter the password for SYS user...> change_on_install
---------------------------------------------
Enter Your Selection:

Press Enter to return to the main menu.
```

7 Select option 2, **Capture Database File Locations**. This option allows you to specify the
location of database files.

TIP: Oracle installations should adhere to the Optimal Flexible Architecture (OFA) for database
file locations.

The system prompts you to provide absolute pathnames for a variety of database files.

Database files can reside wherever you choose. For a single database server, you can use, for
example:

```
/u01/oradata
```

NOTE: Make sure this oradata directory exists and is owned by the oracle user. The scripts
will automatically create a folder called <sid_name> under the oradata directory. Using the
above example, this folder would be created:

```
/u01/oradata/oltp
```

Depending on your platform, you may then be prompted to enter multiple locations for your
database control file. This file allows access to the database, so you should specify backup
locations in case a database control file is deleted or corrupted. Consult your database
documentation for details.

You might also then be prompted to enter suggested **mount points** for your database.
Consult your database documentation for details.

The script then validates the locations you specify. If all are valid, this message appears:

```
Capture of Database file locations completed.
```

Press **Enter** to return to the Siebel Server Administration Main Menu.

8 Select Option 3, **Install.** The Install menu appears:

```
Install
--------------------------------------
[1] Create Oracle Instance
[2] Shutdown Database
[3] Startup Database
[4] Install Application Database I
[5] Install Application Database II
[6] Install Application Database III
```

```
[7] View Status Log Directory
---------------------------------------
[R] Return to previous menu
SELECT YOUR OPTION: 1
```

9   Select Option 1, **Create Oracle Instance**.

This step creates a database instance for Billing Analytics, defines a data dictionary and stored procedure for the new database, and modifies the stored procedure to contain the absolute pathnames you defined in **Capture Database File Locations**. No user input is required, although several progress messages appear.

If this step is successful, the following message appears:

```
Database created.  Configuration in process…
```

Press **Enter** to return to the Install menu.

10   Select Option 2, **Shutdown Database**. If this step is successful, you will see a success message. Press **Enter** to return to the Install menu.

11   Select Option 3, **Startup Database**. If this step is successful, you will see a success message. Press **Enter** to return to the Install menu.

12   Select Option 4, **Install Application Database I**.

This option creates new Billing Analytics database tablespaces, users, and rollback segment data files. This option may take up to 20 minutes to complete.

Toward the end of this process, you should see messages that the utility scripts and stored procedures are executing. When you see the final success message, press **Enter** to return to the Install menu.

13   Select Option 5, **Install Application Database II**.

This option creates database tables and indexes. No user input is required. The error messages at the start of this step are an expected part of the process and can be ignored.

When this step completes, press **Enter** to return to the Install menu.

14   Select Option 6, **Install Application Database III**.

This option compiles stored procedures to support database processing. No user input is required. When this step completes, press **Enter** to return to the main menu.

15   (Optional) If you want to view log files for your database configuration, select Option **[7] View Status Log Directory**. Log files are copied to the database subdirectory of your home directory, for example:

```
$CBA_HOME/db/oracle
```

16   Select option **R** to return to the OLTP main menu:

```
Siebel e-Statement Server Administration Main Menu Version 1.0
--------------------------------------------------------
 [1] Sign in Menu
 [2] Capture Database File Locations
 [3] Install Siebel e-Statement
 [4] Initial Data Population
 [Q] Quit
```

```
          --------------------------------------------------------
          Enter Your Selection:  1
```

17 Select Option 4, **Initial Data Population**, to bring up this menu:

```
     Initial Data Population
     ─────────────────────────────────────────────────────
      [1] Import initial data set
      [2] Export Siebel database data
      [3] Build Sample Hierarchy (Optional)
      ----------------------------------------------------
      [R] Return to previous menu
```

Select **1**. Then, if you want to load sample hierarchy data, select **3**.

18 Select **R** to return to the OLTP main menu.

19 Select option **Q** to return to the top level main menu, and select Option **[2] OLAP Setup** from
there. The OLAP main menu is displayed:

```
     Siebel Reporting Server Administration Main Menu Version 1.0
     ----------------------------------------------------------
          [1] Sign in Menu
          [2] Capture Database File Locations
          [3] Install Siebel Reporting
     ----------------------------------------------------------
     Enter Your Selection:
```

20 Select Option **1** for the sign in page for the OLAP database. The following screen appears:

```
                    SIGN IN MENU
     ----------------------------------------------------
        [1] Enter Database USERNAME ...>cba_dba
        [2] Enter Database PASSWORD ...>cba
        [3] Enter ORACLE_SID        ...>olap
        [4] Enter the password for SYS user ...>change_on_install
     ----------------------------------------------------
```

Enter the appropriate username, password, SID, and sys password.

21 Select option 2, **Capture Database File Locations**, and enter values using the same rules
you used when you defined locations for the OLTP database.

22 Select option 3, **Install Siebel Reporting**. The following screen appears:

```
     Install Siebel Reporting
     ----------------------------------------------------
     [1] Create Oracle Instance
     [2] Shutdown Database
     [3] Startup Database
     [4] Create Reporting Tablespaces
     [5] Create Reporting schema
     [6] Install Reporting schema
     [7] Sample Reporting Data Population
     [8] Sample Hierarchy Data Population (optional)
     ----------------------------------------------------
     [R] Return to previous menu
```

Select Option 1, **Create Oracle Instance**.

This step creates a database instance for Billing Analytics reporting, defines a data dictionary and stored procedure for the new database, and modifies the stored procedure to contain the absolute pathnames you defined in **Capture Database File Locations**. No user input is required, although several progress messages appear.

If this step is successful, the following message appears:

    Database created. Configuration in process…

TIP: This step can take anywhere from 20 minutes to 2 hours depending on the speed of your platform.

Press **Enter** to return to the Install menu.

23 Select Option 2, **Shutdown Database**. If this step is successful, you will see a success message. Press **Enter** to return to the Install menu.

24 Select Option 3, **Startup Database**. If this step is successful, you will see a success message. Press **Enter** to return to the Install menu.

25 Select Step 4, **Create Reporting Tablespaces**. Press enter when this step completes.

26 Select Step 5, **Create Reporting schema**. Press enter when this step completes.

27 Select Step 6, **Install Reporting schema**. Press enter when this step completes.

28 Select Step 7, **Sample Reporting Data Population**. Press enter when this step completes.

29 OPTIONAL: select step 8, **Sample Hierarchy Data Population**. If you choose to skip this step, you can load the sample hierarchy data later by running a sequence of jobs. These steps are described in the *Administration Communications Billing Analytics* guide, in the section *Scheduling Job for Sample Data*.

30 Select option **R** to return to the OLAP main menu.

NOTE: You must update tnsnames.ora before running this next step.

31 Select option 3, **CREATE DB LINK**. This brings up the following menu, which shows example entries in bold:

```
      --------------------------------------------------
        [1] Enter OLAP Database USERNAME ...>cba_dba
        [2] Enter OLAP Database PASSWORD ...>cba
        [3] Enter OLTP Database USERNAME ...>cba_dba
        [4] Enter OLTP Database PASSWORD ...>cba
        [5] Enter OLAP TNS NAME        ...>olap
        [6] Enter OLTP_SID ...>oltp
        [7] Enter OLAP SYS Password ...>change_on_install
        [8] Enter OLTP SYS Password ...>change_on_install
      --------------------------------------------------
```

NOTE: These entries must match the entries you entered previously on the OLTP and OLAP menus.

32 Return to the main menu, and select Option **Q** (Quit), then press **Enter**.

About build.xml for Oracle

You can also use ant to create and configure the Billing Analytics production database. If you have to abort database setup before it completes successfully, see "What to Do if Database Configuration Fails" on page 31.

Before running ant, you should:

- Complete all the prerequisites specified previously in "About edx_main_admin.sh for Oracle" on page 21.

- Install Apache ant version 1.6.5 or higher. This software can be downloaded from http://ant.apache.org/. The installation directions can be found on that site as well.

- Configure *edxadmin.properties*, as described in the next section.

Configuring edxadmin.properties

The next step in setting up the database server is to edit the properties file that controls the Billing Analytics production database ant installation.

TIP: Always consult with your onsite DBA and your Oracle Professional Services representative to configure database connectivity, to make sure you comply with client standards for the enterprise

CAUTION: When creating an Oracle database, limit its name to eight characters. Defining or entering an Oracle SID with more than eight characters causes Oracle database configuration to fail

The following example for *edxadmin.properties* shows sample values that should be replaced with the appropriate paths, usernames, passwords, SIDs, and TNS settings:

```
ORACLE_HOME==/u01/oracle/product/9.2.0/ora92
ORACLE_ADMIN=/u01/oracle/product/9.2.0/admin
OLAP_USER=cba_dba
OLAP_PASSWD=cba
OLAP_SID=olap
OLAP_TNS_NAME=olap
DB_USER=cba_dba
DB_PASSWD=cba
DB_SID=oltp
SYS_PASSWD=change_on_install
META_USER=meta
META_PASSWD=meta
L_DB_EDX_DATA_TB_FILE_LOC=/u01/oradata
L_DB_APP_DATA_TB_FILE_LOC=/u01/oradata
L_DB_EDX_INDX_TB_FILE_LOC=/u01/oradata
L_DB_APP_INDX_TB_FILE_LOC=/u01/oradata
L_DB_LOAD_DATA_TB_FILE_LOC=/u01/oradata
L_DB_LOAD_INDX_TB_FILE_LOC=/u01/oradata
L_DB_FS_DATA_TB_FILE_LOC=/u01/oradata
L_DB_FS_INDX_TB_FILE_LOC=/u01/oradata
L_DB_STG_DATA_TB_FILE_LOC=/u01/oradata
L_DB_STG_INDX_TB_FILE_LOC=/u01/oradata
DB_CONTROL_FILE_LOCN1=/u01/oradata
```

```
DB_CONTROL_FILE_LOCN2=/u01/oradata
DB_CONTROL_FILE_LOCN3=/u01/oradata
REDO_LOG_FILE_LOCN=/u01/oradata
SYSTEM_FILE_LOCN=/u01/oradata
TRACE_FILE_LOCN=/u01/oradata
L_BACKUP_FILE=/u01/oradata/expoltp.dmp
L_LOG_FILE=/u01/oradata/expoltp.log
L_OLAP_BACKUP_FILE=/u01/oradata/expolap.dmp
L_OLAP_LOG_FILE=/u01/oradata/expolap.log
```

To configure a new Billing Analytics database for Oracle on UNIX

1   Switch user to the DB admin user. Oracle requires the administrative user in order to create files.

    `$ su – oracle`

2   Change directory to your Billing Analytics database home directory. For example:

    `cd /opt/siebel/TAM/db/oracle`

3   There are multiple ant targets you can use that will automate the installation process. The `install-new` target will create new OLTP and OLAP instances with the SIDs specified in the properties file:

    `ant install-new`

4   The `install-existing` target will create new OLTP and OLAP schemas on an existing instance with the usernames/passwords specified in the properties file:

    `ant install-existing`

5   If you wish to install sample data, add the `–DloadSampleData=true` argument to the ant call. For example:

    `ant install-existing –DloadSampleData=true`

6   There are also OLTP- and OLAP- specific ant build files, `buildedxadmin.xml` and `buildedxolapadmin.xml`, respectively. You can run any of the above targets and flags using these instance-specific files by using the `–f <filename>` flag. For example, to install just a new OLAP instance with sample data, you would run the following:

    `ant install-new –f buildedxolapadmin.xml –DloadSampleData=true`

7   If you wish to manually run each install step, simply start the ant script with the command:

    `ant`

    The main menu will appear:

    `main:`

```
        [echo] Siebel Server Administration Main Menu Version 1.0
        [echo] [1] OLTP Setup
        [echo] [2] OLAP Setup
        [echo] [3] CREATE DB LINK
        [echo] [Q] Quit
        [input] Enter your selection (1,2,3,q,Q)
```

    Choose the steps you wish to complete and refer to the *edx_main_admin.sh* menu descriptions starting on page 25 for more info about each of the menu options.

What to Do if Database Configuration Fails

If you encounter errors during database creation and configuration, you must first remove the partially configured database before configuring the database again.

To recover from a failed database configuration for Oracle

1   Switch user to the database **admin** user, in this example `oracle.` Oracle requires the administrative user in order to remove files.

```
$ su – oracle
```

2   Check the database instance name with the **echo** command. For example:

```
$ echo $ORACLE_SID
oltp
```

3   If the database instance name is incorrect, set it as follows:

```
$ ORACLE_SID=oltp
$ export ORACLE_SID
```

4   Shut down the database server.

5   Locate all files associated with the database instance you defined. For example:

```
$ find $ORACLE_BASE -name '*oltp*'
$ find /u01/oradata –name "*oltp*"
$ find /u02/oradata –name "*oltp*"
$ find /u03/oradata –name "*oltp*"
$ find /u04/oradata –name "*oltp*"
$ find /u05/oradata –name "*oltp*"
$ find /u06/oradata –name "*oltp*"
```

6   Recursively remove the resulting folders and any contained files. For example:

```
$ rm –rf <found_files>
```

7   Repeat this step for other root directories containing database files for your OLTP database instance.

8   Repeat for the OLAP instance.

Connecting to Your Oracle Database

Once you have configured Oracle services, you should now be able to connect to your Billing Analytics database.

1   Switch user to the **DB Admin** user.

```
$ su – oracle
```

2   Run the **sqlplus** command on your Billing Analytics database, with arguments for your database username, password, and connection string (database alias). For example:

```
$ sqlplus cba_dba/cba@oltp
```

3   If the database connection is successful, a connection message appears.

```
Connected to: Oracle9i Enterprise Edition Release 9.2.0.0.0
```

4   At the SQL prompt, enter a database query command, for example:

```
SQL> show parameters db_name
```

If the database connection is successful, you see output for your database instance.

```
NAME TYPE VALUE
--------- ------- -----
db_name string oltp
SQL>
```

5   Repeat for OLAP Database

Once your database server tests successfully with the TAM databases installed, you can proceed to configure your application server.

6   Configuring the Application Server

Overview

This chapter assumes in-depth understanding of and practical experience with application server administration. It is designed for experienced WebSphere administrators and primarily presents only the steps and settings specific to Billing Analytics.

See the WebSphere documentation for detailed step-by-step instructions on Java resource configuration, performance, and tuning. You must also consult your application server administrator for settings that may be specific to your configuration.

You must start your WebSphere Server instance and bring up the Administrative Console before you begin this chapter.

CAUTION: If you cannot bring up the WebSphere Console, you will be unable to proceed with configuring your application server for Billing Analytics.

CAUTION: Click "Save" on the menu bar on the left side or the "Save" hyperlink in the messages table to save your changes to "Master Configuration" before you log out or exit.

CAUTION: The installation and configuration examples shown in this guide use default Billing Analytics pathnames, privileges, and permissions. If you choose not to accept the default values, make sure your values are consistent on all servers across your installation of Billing Analytics.

This chapter provides instructions for configuring WebSphere Server to support Billing Analytics. It includes:

- Unix Permissions for WebSphere Server
- Configure WebSphere Servers
- Java Virtual Machine (JVM) Settings for WebSphere
- Edit startServer.sh to Set Environment Data
- Use *edx_config* to Capture Environment Data for WebSphere
- Start the Server

Unix Permissions for WebSphere Server

Application servers running CBA will not function correctly without access to CBA configuration files, storage directories, and related resources. When installing CBA on WebSphere Server, you were asked to specify the owner and group permissions (userid:groupid) of the application server. If you made a mistake during installation, you must change the owner and group permissions for the directory, including all subfolders, to the application server owner.

This guide uses the example username and password edxadmin:edxadmin as the application server owner and group.

For details on owner and group permissions for WebSphere, please consult the documentation for your platform.

If your application server administrator uses custom user and group permissions, you can reset these permissions with the chown command.

To reset user and group permissions

1 Switch user to the default owner of your application server home directory.

2 Recursively change the user and group permissions of the application server installation directory and all subdirectories to the application server owner.

```
chown –R edxadmin:edxadmin /opt/WebSphere
```

3 Switch user to your application server owner and configure your application server with your new owner.

Configure WebSphere Servers

You must create two WebSphere server instances, one for Command Center (ccserver) and the other one for the customer facing application, tam-tbm (tamserver)

- **cbaserver** – WAS instance for the Billing Analytics customer facing application

- **ccserver** – WAS instance for estatement application

To create a WAS instance (WebSphere server)

1 In the WebSphere Administrative console, click **Servers**, then click `Application Servers`, then click New and give the server name.

2 Click Next and then click Finish.
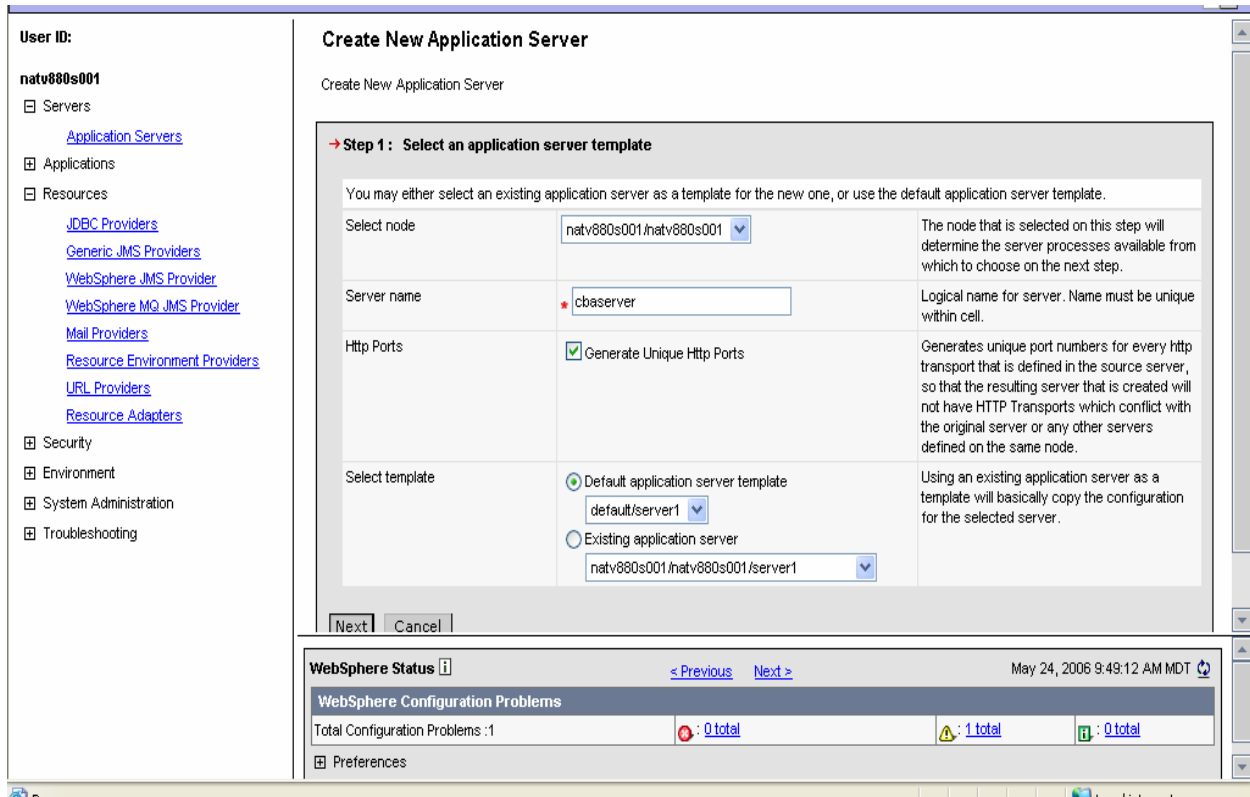
Java Virtual Machine (JVM) Settings for WebSphere

To configure JVM settings for WebSphere:

1 In the WebSphere Administrative console, click **Servers**, then `Application Servers`, then the server for Communications Billing Analytics (cba server), then **Process Definition**, then `Java Virtual Machine`.

2 On the General Properties page, for Classpath, specify the following:

Property Name	Value
Classpath	<EDX_HOME>/lib/edx_common.jar
	<EDX_HOME>/lib/edx_system.jar
	<EDX_HOME>/lib/edx_client.jar
	<EDX_HOME>/lib/jakarta-oro-2.0.7.jar
	<EDX_HOME>/lib/xpp3_1_1_2.jar
	<TAM_HOME>/lib/javachart.jar
	<TAM_HOME>/lib/log4j.jar

	<TAM_HOME>/lib/commons-logging.jar
	<TAM_HOME>/lib/Configuration.jar
	<TAM_HOME>/lib/ldeprotocol.jar
	<TAM_HOME>/config
	
	Where TAM_HOME is the directory where you installed Communications Billing Analytics; and EDX_HOME is TAM_HOME/estatement, for example:
	        EDX_HOME=/opt/Siebel/CBA/estatement
	        TAM_HOME=/opt/Siebel/CBA

Your screen should resemble the following:

3   In Generic JVM arguments, enter the following:

`-Dorg.apache.commons.logging.Log=com.edocs.common.logging.plugins.jcl.Logger`

4   Click Apply.

5   Under **Additional Properties**, click on **Custom Properties** and create the following properties using the values described below:

Property Name	Value
com.edocs.tasks.loader.alias	OLTP Database alias; for example, TAMOLTP
com.edocs.tasks.loader.password	Database password; for example, tamtest

com.edocs.tasks.loader.user	Database user name; for example, tamtest
edx.home	estatement directory under TAM installation directory; for example: `/opt/Siebel/CBA/estatement`
java.protocol.handler.pkgs	com.edocs.protocol

Your screen should resemble the following:

6 For server for estatement application (ccserver), follow steps 1 through 5. Change the edx.home property in step 5 to the following:

`/opt/Siebel/CBA/estatement_cc`

Edit startServer.sh to Set Environment Data

You must add some settings to the script that is used to start your WebSphere server, in `startServer.sh`, which is located under $WAS_Home/AppServer (where $WAS_Home stands for WebSphere installation directory).

To edit startServer.sh for Billing Analytics and Platform Services:

1 Switch user to your application server owner, for example edxadmin:

su – edxadmin

2 Stop your application server.

3   Change directory to the location of your application server startup script and open the file for editing. For example,

```
cd $WAS_HOME/bin vi startServer.sh
```

4   Declare and initialize the variable $EDX_HOME near the beginning of the file with other variable declarations. For example:

```
binDir='dirname $0' . $binDir/setupCmdLine.sh
EDX_HOME=/opt/Siebel/CBA/estatement
export EDX_HOME
```

5   In the same file, source edx.config just before the command to start the JVM. The dot and space preceding the pathname are a required part of the syntax. For example:

```
. $EDX_HOME/config/edx.config
```

6   Save and close the file.

Use *edx_config* to Capture Environment Data for WebSphere

The script edx_config prompts you to enter values for your Java and database installation, including absolute directory pathnames or user identification information. It stores these values in the configuration file **edx_env.**

You should run **edx_config** any time you need to modify your CBA environment. Do not modify **edx_env** directly.

CAUTION: Do not confuse **edx_config** (underscore) with **edx.config** (dot), which sources your configuration at server startup.

To capture environment data with edx_config for WebSphere:

1   Switch user to the $EDX_HOME owner, in this example **edxadmin**.

```
su - edxadmin
```

2   Navigate to the **bin** directory for eStatement on your application server, for example

```
cd $EDX_HOME/bin
```

3   Run the script **edx_config**.

```
./edx_config
```

4   Enter values as prompted by the script for your OLTP database home, database username and password, application server, Java home, and application server home. The WebSphere home directory pathname is where its application server files were installed, usually someplace like */opt/WebSphere/AppServer*. Make sure that the database values you enter in this session are the same values specified during database configuration. Consult your DBA for any custom settings specific to your platform.

Start the Server

When you have finished the steps to pass your Communications Billing Analytics environment to WebSphere, restart the server. For example:

```
./startServer.sh server1
```

7 Configuring IBM WebSphere

Overview

This chapter provides instructions for configuring WebSphere Server resources to support Billing Analytics. These include:

- JDBC Providers

- JDBC Data Sources

- JMS Servers

- JMS Queue, MQ Queue, and Topic Connection Factories

- JMS Queue, MQ Queue, and Topic Destinations

This chapter assumes in-depth understanding of and practical experience with application server administration. It is designed for experienced WebSphere administrators and primarily presents only the steps and settings specific to Billing Analytics.

See the WebSphere documentation for detailed step-by-step instructions on Java resource configuration, performance, and tuning. You must also consult your application server administrator for settings that may be specific to your configuration.

You must start your WebSphere Server instance and bring up the Administrative Console before you begin this chapter.

CAUTION: If you cannot bring up the WebSphere Console, you will be unable to proceed with configuring your application server for Billing Analytics.

Starting and Stopping WebSphere

Developers and system administrators will need to be familiar with how to stop and start your WebSphere server and any active web applications for your platform.

For details on starting and stopping your WebSphere server and for verifying startup, please consult the server documentation for your platform.

Starting and Stopping an Active Application Server

Improperly starting or stopping an application server in an active CBA production environment can produce unexpected and unintended results. You can create custom startup and shutdown scripts that include all your command parameters, as well as the command used to start or stop the Scheduler.

Configure Java Database Connectivity (JDBC)

After you have successfully configured the Billing Analytics database, you must configure Java Database Connectivity (JDBC) resources on the Billing Analytics application server. JDBC Connections on the application server support data retrieval from relational databases and other data sources.

About JDBC Connections for Communication Billing Analytics

JDBC connection pools contain named groups of JDBC Connections that are created when the connection pool is registered, usually when starting up the application server. The application server opens JDBC Connections to the database during startup and adds these connections to the pool. A J2EE web application borrows a connection from the pool, uses it, and then returns it to the pool by closing it.

JDBC data sources enable JDBC clients to obtain a connection to a Database Management System (DBMS). Each data source points to the value specified for the Name attribute when a JDBC connection pool was configured.

For more details on configuring JDBC Connections, please see the JDBC documentation for your application and database servers.

Configuring JDBC Connections for WebSphere

Create the following jdbc providers for both the cba server and the ccserver:

- Oracle JDBC Driver
- Oracle JDBC Driver (XA)

Configuring JDBC Providers

Configure a new **JDBC Provider**:

1 Navigate to Resources > JDBC Providers > New.

2 choose `Oracle JDBC Driver` from the JDBC Providers drop-down list.

3 Click OK.

4 In the JDBC Provider configuration screen, modify the Classpath property to point to the path of **ojdbc14.jar** in the server directory (for example: /opt/oracle/OraHome1/jdbc/lib/ojdbc14.jar) as shown below.

Similarly, create a second JDBC Provider: Oracle JDBC Driver (XA).

JDBC Data Sources for eStatement Server (ccserver)

Create the following **Data Sources (Version 4)** for **Oracle JDBC Driver**, using the steps described below.

Name	JNDI Name	Database
edxCCAdminDataSource	edx.databasePool	OLTP
edxLoggerDataSource	edxLoggerDataSource	OLTP
edxUserDataSource	edx.user.databasePool	OLTP

To create a data source

1   Click the "`Oracle JDBC Driver`" link and select Data Sources (Version 4).

2   Click the "New" button.

3   In the Data Source creation page, enter the Data Source name, JNDI name, database name, database username, and password. Then click OK, as shown in the picture below.

4   Select the created data source by clicking the data source name.

5   Select Custom properties.

6   Within custom properties, select the URL property and provide the appropriate value.

URL	jdbc:oracle:thin:@<DB_host>:<DB_port>:<DB_SID>.
	For example, jdbc:oracle:thin:@localhost:1521:TAMOLTP

7 Create the following **Data Sources (Version 4)** for **Oracle JDBC Driver (XA**) in the same manner described above. Note, however, that the driver is different!

Name	JNDI Name	Database
edx.report.databasepool	edx.report.databasePool	OLAP
XMADataSource	edx.xma.databasePool	OLTP
edxMessagingDataSource	edx.messaging.databasePool	OLTP

JDBC Data Sources for CBA Server (cbaserver)

Create the following **Data Sources (Version 5)** for **Oracle JDBC Driver**. To create version five data sources, follow the same configuration steps as specified above. However, you must additionally configure the **J2C Authentication Data Entries** property.

Name	JNDI Name	Database
edxCCAdminDataSource	edx.databasePool	OLTP
edxLoggerDataSource	edx.logger.databasePool	OLTP
edxUserDataSource	edx.user.databasePool	OLTP

Create an Authentication alias for the OLAP and OLTP databases.

To create an authentication alias, use the following instructions:

Additional Properties	
Connection Pool	An optional set of connection pool settings.
Custom Properties	Properties that may be required for Resource Providers and Resource Factories. For example, most database vendors require additional custom properties for data sources that will access the database.

Related Items	
J2C Authentication Data Entries	Specifies a list of userid and password for use by Java 2 Connector security.

In J2C Authentication Data Entries screen enter the following values and click the OK button.

Alias	Database alias name; for example, TOLAP
User ID	Database user; for example, **OLAP1**
Password	Database password; for example, **OLAP1**

JDBC Providers > Oracle JDBC Driver > Data Sources > reportDataSource > J2C Authentication Data Entries >

demeter/TOLAP

Specifies a list of userid and password for use by Java 2 Connector security. [i]

Configuration

General Properties		
Alias	★ demeter/TOLAP	[i] Specifies the name of the authentication data entry.
User ID	★ olap1	[i] Specifies the J2C authentication data user ID.
Password	★ *****	[i] Specifies the password to use for the target Enterprise Information System.
Description		[i] Specifies an optional description of the authentication data entry. For example, this authentication data entry is used to connect to DB2.

Apply | OK | Reset | Cancel

Now, select the name of the relevant Authentication Alias created above from the drop-down list for the component-managed and container-managed authentication alias fields when creating data sources.

JDBC Providers > Oracle JDBC Driver > Data Sources >

XMADataSource

Data Source is used by the application to access the data from the database. A data source is created under a JDBC provider which provides the specific JDBC driver implementation class. ⓘ

Configuration

Test Connection		
General Properties		
Scope	★ cells:demeter:nodes:demeter	ⓘ The scope of the configured resource. This value indicates the configuration location for the configuration file.
Name	★ XMADataSource	ⓘ The required display name for the resource.
JNDI Name	edx.xma.databasePool	ⓘ The JNDI name for the resource.
Container managed persistence	☐ Use this Data Source in container managed persistence (CMP)	ⓘ Enable if this data source will be used for container managed persistence of EJBs. This will cause a corresponding CMP connection factory which corresponds to this datasource to be created for the relational resource adapter.
Category		ⓘ An optional category string which can be used to classify or group the resource.
Statement Cache Size	10 statements	ⓘ Number of free prepared statements per connection. This is different from the old datasource which is defined as number of free prepared statements per data source.
Datasource Helper Classname	com.ibm.websphere.rsadapter.Orac	ⓘ The datastore helper that is used to perform specific database functions.
Component-managed Authentication Alias	demeter/TOLTP ▾	ⓘ References authentication data for component-managed signon to the resource.
Container-managed Authentication Alias	demeter/TOLTP ▾	ⓘ References authentication data for container-managed signon to the resource.
Mapping-Configuration Alias	DefaultPrincipalMapping ▾	ⓘ Select a suitable JAAS login configuration from the security-JAAS configuration panel to map the user identity and credentials to a resource principal and credentials that is required to open a connection to the back-end server.

Apply	OK	Reset	Cancel

Create the following **Data Source (Version 5)** for **Oracle JDBC Driver (XA)**. To create a version five data source, follow the same configuration steps described earlier in this section.

Name	JNDI Name	Database
edxMessagingDataSource	edx.messaging.databasePool	OLTP
XMADataSource	edx.xma.databasePool	OLTP
reportDataSource	edx.report.databasePool	OLAP

About JMS Resources for Communications Billing Analytics

JMS enable web application components to asynchronously send and receive messages.

- **JMS Connection Factories** are data objects that enable Java Messaging Service (JMS) clients to create JMS connections. You define and configure one or more connection factories to create connections with predefined attributes. Your application server adds the connection factories to the JNDI space during startup, and each J2EE web application retrieves a connection factory using the JNDI on the application server.

- **JMS Stores** store persistent messages in a database accessed through a designated JDBC connection pool. The JMS database can be any database that is accessible through a WebSphere-supported JDBC driver. When creating a JMS Store, you must define the name of the backing store, and the JDBC connection pool and database table name prefix for use with multiple instances.

- **JMS Servers** manage connections and message requests on behalf of clients.

- **JMS Topics** can be one of two destinations that you can configure for a JMS server. The other destination is a JMS queue.

- **JMS Session Pools** allow a JMS listener to have multiple threads that will improve performance under heavy load.

Do the following JMS configuration for both the CBA (cbaserver) and the estatement (ccserver):

WebSphere Queue Connection Factories

Create a WebSphere Queue connection factory:

1   Navigate to Resources-> WebSphere JMS Provider -> WebSphere Queue Connection Factories -> new.

2   Provide the following information:

Queue Name	JNDI Name
edxMessagingConnectionFactory	edx.qcf

WebSphere JMS Provider > WebSphere Queue Connection Factories >

edxMessagingConnectionFactory

A queue connection factory is used to create connections to the associated JMS provider of JMS queue destinations, for point-to-point messaging. Use W administrative objects to manage queue connection factories for the internal WebSphere JMS provider. ⓘ

Configuration

General Properties

Scope	＊cells:natv880s001:nodes:natv880s001:servers:cba	ⓘ The scope of t the configuration
Name	＊edxMessagingConnectionFactory	ⓘ The required d
JNDI Name	＊edx.qcf	ⓘ The JNDI name
Description		ⓘ An optional de
Category		ⓘ An optional ca group the resourc
Node	natv880s001 ▾	ⓘ The WebSpher where the JMS se Connections crea
Component-managed Authentication Alias	(none) ▾	ⓘ References au signon to the reso
Container-managed Authentication Alias	(none) ▾	ⓘ References au signon to the reso
Mapping-Configuration Alias	DefaultPrincipalMapping ▾	ⓘ Select a suitab JAAS configuratio credentials to a re required to open a
XA Enabled	☑ Enable XA	ⓘ Attribute to ind enabled or not. Th

WebSphere Queue Destinations

Create a WebSphere Queue Destination:

1 Navigate to Resources-> WebSphere JMS Provider -> WebSphere Queue Destinations -> new.

2 Provide the following information:

Topic Name	JNDI Name
xmaEventErrorQueue	edx.queue.errors
xmaEventQueue	edx.queue.outbound

WebSphere Topic Connection Factories

Create a WebSphere Topic Connection Factory:

1   Navigate to Resources-> WebSphere JMS Provider -> WebSphere Topic Connection Factories -> new.

2   Provide the following information:

Topic Name	JNDI Name
edxLoggerTCF	edx.tcf.log
edxDisputeTCF	edx.tcf.dispute
edxAnnotationTCF	edx.tcf.annotate

WebSphere Topic Destinations

Create a WebSphere Topic Destination:

1   Navigate to Resources-> WebSphere JMS Provider WebSphere Topic Destinations -> new.

2   Provide the following information:

Name	JNDI Name	Topic
edxLoggerTopic	edx.jms.log	edxLoggerTopic
edxDisputeTopic	edx.jms.dispute	edxDisputeTopic
edxAnnotationTopic	edx.jms.annotate	edxAnnotationTopic

WebSphere MQ Queue Connection Factories

Create a WebSphere MQ Queue connection factory:

1   Navigate to Resources-> WebSphere MQ JMS Provider -> WebSphere MQ Queue Connection Factories -> new.

2   Provide the following information:

Queue Name	JNDI Name
edxForeignJMSConnectionFactory	edx.foreign.qcf

Property	Comments
Queue Manager	Name of the default MQ Queue Manager in cbaserver. This can be obtained by running the dspmq command in the /opt/mqm/bin directory.
Host	Name of Host in which the Server2 resides.

Port	Default MQ Queue Managers port in cbaserver. This can be obtained by running the following command:
	
	`ps -ef | grep <Default Queue Manager of cbaserver>`
	
	When you run this command, you will get an output, and the port can be found from the following output:
	
	`runmqlsr -t tcp -p 5571 -m WAS_natv880s001_cba`

Channel	Channel of the default Queue Manager of cbaserver. To find this, run the following command: `runmqsc < cbaserver Default Queue Manager>`

Then run the following command:

`DISPLAY CHANNEL(*)`

You will receive an output similar to the following:

`CHANNEL(WAS.JMS.SVRCONN)  CHLTYPE(SVRCONN)`

The channel name is **WAS.JMS.SVRCONN.**

Transport Type   CLIENT

WebSphere MQ Queue Destinations

Create a WebSphere MQ Queue Destination:

1   Navigate to Resources-> WebSphere MQ JMS Provider -> WebSphere MQ Queue Destinations -> new.

2   Provide the following information:

Topic Name	JNDI Name	Base Queue Name	Base Queue Manager Name
edxForeignJMSDestination	edx.foreign.queue.outbound	WQ_xmaEventQueue	Name of the default MQ Queue Manager in cbaserver. This can be obtained by running the **dspmq** command in the following directory: `/opt/mqm/bin`

Property	Comments

Base Queue Name	Name of the local queue defined in the cbaserver. To obtain the queue name, navigate to the /opt/mqm/bin directory and run the following command:
	
	`runmqsc < cbaserver default Queue Manager>`
	
	Then run the following command:
	
	`DISPLAY QL(WQ*)`

	This command will display a list of queue names. Copy the following: **WQ_xmaEventQueue**
Base Queue Manager Name	Name of the default MQ Queue Manager in cbaserver. This can be obtained by running the **dspmq** command in the /opt/mqm/bin directory.

Message Driven Beans

Configure the JMS server and listeners for both servers (cbaserver and ccserver).

To configure message driven beans (MDBs):

1   Navigate to cbaserver > Server Components > JMS Server

2   Add the queues created above to the JMS server by adding the following queue names:

   edxForeignJMSDestination

   xmaEventErrorQueue

   xmaEventQueue

3   Create a JMS Listener by navigating to Application Servers > server1 > Message Listener Service > Listener Ports.

4   Click on new.

5   Enter the "Name", "Connection factory JNDI name" and "Destination JNDI name" as shown below.

Name	eventListener
Connection factory JNDI name	edx.qcf
Destination JNDI name	edx.queue.outbound

Edit Hibernate Properties Files

On the application server, you must edit both hibernate.properties and hierarchy.hibernate.properties in /opt/Siebel/TBM/Config (or TAM_HOME/Config). Edit both the property files as shown below. The changes that need to be done are marked in boldface text.

```
###################################################
## Data source information for Oracle jdbc driver
###################################################

hibernate.connection.driver_class=oracle.jdbc.driver.OracleDriver
#hibernate.connection.url=jdbc:oracle:thin:@diablo:1521:edx43
hibernate.dialect=net.sf.hibernate.dialect.OracleDialect
hibernate.connection.datasource=edx.databasePool

###################################################
## Data source information for SQLServer jdbc driver
###################################################
```

#hibernate.connection.driver_class=com.inet.pool.PoolDriver
#hibernate.connection.url=jdbc:inetpool:inetdae7://localhost:1433
#hibernate.dialect=net.sf.hibernate.dialect.SQLServerDialect

hibernate.jta.UserTransaction=java:comp/UserTransaction
hibernate.cglib.use_reflection_optimizer=true
hibernate.cache.provider_class=net.sf.hibernate.cache.HashtableCacheProvider
hibernate.show_sql=false

Edit the following line and change the installation path as necessary in TAM_HOME/Config/log4j.xml
(if log4j.xml does not exist, rename log4j.xml.bak to log4j.xml).

<param name="File" value="/opt/Siebel/TBM/estatement/logs/log4j.log"/>

Changing the JTA User Transaction JNDI name for estatement

To change the JTA user transaction JNDI name for estatement, follow this procedure.

To change the JTA user transaction JNDI name for estatement

1 Navigate to the <CBA_Home>/estatement_cc/xma/config/modules directory.

2 Open the persistence.xma.xml file.

In the persistence.xma.xml file, change the JTA user transaction jndi name to
jta/usertransaction by changing the following tag:

```
<!-- WebSphere Transaction Manager -->
    <bean id="TransactionManager"
    class="org.springframework.transaction.jta.JtaTransactionManager">
            <property name="userTransactionName">
                    <value>java:comp/UserTransaction</value>
            </property>
    </bean>
```

To this:

```
<!-- WebSphere Transaction Manager -->
    <bean id="TransactionManager"
    class="org.springframework.transaction.jta.JtaTransactionManager">
            <property name="userTransactionName">
                    <value>jta/usertransaction</value>
            </property>
    </bean>
```

Deploying the Billing Analytics Application

After configuring your WebSphere server instance, you can deploy the EAR files to the appropriate servers. The EAR files are located at:

Feature	Location	File Name
Billing Analytics Consumer	%TAM_HOME%\J2EEApps\TAM\tam-tbm	tam-tbm.ear
Platform Services	%TAM_HOME%\J2EEApps\estatement	ear-eStatement.ear

Consult your WebSphere documentation on how to deploy applications.

Testing the Billing Analytics Installation

After successfully deploying the EARs, you can log into the Billing Analytics application.

In your browser, point to http://localhost:<port_no>/tbmb (where <port_no> is the port_no configured for the WAS instance on which tam-tbm.ear is installed).

Log in using the following username/password pairs to check the application.

NOTE: In each case, the username and password are identical.

Company	First Name	Last Name	Login/Password
American HighTech1	John	Smith	JSMITH
American HighTech1	Aaron	Bush	ABUSH
American HighTech1	Charles	Andrews	CANDREWS
American HighTech1	James	Book	JBOOK
British Footwear	Tim	Burr	TBURR
British Footwear	Tom	Brown	TBROWN
British Footwear	Michael	Law	MLAW
Dutch Home Insurance	Tim	Walsh	TWALSH
Dutch Home Insurance	Frank	Town	FTOWN
Dutch Home Insurance	Kevin	Laracey	KLARACEY
Dutch Home Insurance	Lisa	Green	LGREEN

To test Platform Services

Create a new application:

1   Enter the URL http://localhost:<port_no>/edocs, which displays the Command Center Main Page. Note: <port_no> is the port_no configured for the WAS instance on which ear-estatement.ear is installed.

2   Login using the following User Name and Password: admin/edocs.

3   In the Main Console, select Create New Application.

4   Enter a name for the new application; for example, **testApp**.

5   Use **/edx/ejb/EdocsDataSource** for the Datasource Name.

6   Choose the default for Index Partition Count.

7   Click on the **Create Application** button.

Install X-Terminal

To display charts, your Web server must have access to an X display device. Do not expect font styles if your server does not have them. The following sections deal with the individual issues involved.

Display Permission

The command in UNIX environments that allows X displays on a particular machine from other machines is: xhost +. Without arguments beyond the +, this results in allowing X displays on all machines, as opposed to named machines only. Generally xhost can be found in */usr/openwin/bin*. You can make things easier for yourself by adding it to your environment PATH variable. At the command prompt, type:

```
xhost +
```

Display Device

Java's graphic packages are based on java.awt, which requires X libraries and an X display. That means that the Web server that is rendering the charts—the web tier—must have access to a real or virtual X display device. Therefore, the necessary X Libraries must be loaded.

NOTE: It is not necessary for the Web server's system to have a physical X Display device.

In a development environment, the Web server may actually have a real physical display device attached to it with the X Libraries loaded. In deployment environments, especially at server hosting sites where typically there are racks of machines with few if any connected to display devices, a virtual X display is common. Even in situations where there are attached display devices, it may be preferable not to load X libraries if they are not already present. The X Virtual Frame Buffer (Xvfb) software provides a virtual X display device.

If you do not already have Xvfb installed on your machine, you can obtain a tar-ed version at ftp://www.ferret.noaa.gov/special_request/xvfb/solaris/, along with further usage information. (You may need to use gunzip and tar xvf to unzip and extract the contents of the archive,

if you do not have a specialized utility.) Untar the file to have it installed under */usr/X11R6*. The command to start the virtual frame buffer, Xvfb, will then be located in */usr/X11R6/bin*. Execute the following command at the command prompt:

```
/usr/X11R6/bin/Xvfb :2 -screen  0 800x600x24  &
```

Using the "&" allows you to close the command window and leave the task running in the background.

Server Display Awareness

If you are running an application or Web server from the command line, it will use the `DISPLAY` environment variable in effect. If you want it to use something else, ensure that you either `setenv` or `export DISPLAY` (depending on your shell) to the value you are interested in. If you are likely to stick to a certain configuration, you may prefer to incorporate the information below in your server start script:

```
DISPLAY=MyServer:2.0
export DISPLAY
```

Setting up the server's charting environment

To set up the server's charting environment manually

1 `# gzip —d X11R6.tar.gz` (if gzip is installed)
   `# tar xvf X11R6.tar`

2 Start the X Virtual Frame Buffer, as shown below. This will create a virtual display at :2.0 with a size of 800 by 600 pixels and a color depth of 24 bits. To ensure that Java draws to this display, you must set the DISPLAY environment variable to :2.0 before invoking Java. If Java encounters any X11 Environment exceptions, try adjusting the color depth, modifying the screen size, or changing –screen 0 to –screen 1.

   `# /usr/X11R6/bin/Xvfb :2.0 -screen 0 800x600x24 &`
   `# ps —ef | grep Xvfb (to see if it's running)`

3 Change to the WebLogic owner user profile.

   `# su — weblogic`

4 Set the DISPLAY parameter in the weblogic .profile by adding the following line:

   `$ DISPLAY=:2.0; export DISPLAY`

5 Allow open access to the machine's X Display:

   `$ xhost +`

6 In the file *startWebLogic.sh* add the following line at the top:

   `DISPLAY=:2.0; export DISPLAY`

7 Restart the WebLogic server instance.

This process must be repeated EVERY time the physical web server is restarted (from root uid).

```
# /usr/X11R6/bin/Xvfb :2.0 -screen 0 800x600x24 &
# su – weblogic
$ DISPLAY=:2.0; export DISPLAY (not necessary if in .profile)
$ xhost +
```

To set up the server's charting environment automatically on system reboot

The following procedure shows a convenient way of automating the charting environment setup if the application server is restarted:

1
```
# cd /etc/init.d
# vi chart
```

2 Add the following to the file:

```
#!/bin/sh
PATH=/usr/X11R6/bin:$PATH; export PATH
nohup Xvfb :2.0 -screen 1 800x600x24 &
DISPLAY=:2.0; export DISPLAY
xhost +
```

3 Update file permissions:

```
# chmod 777 chart
```

4 Change your working directory to /etc/rc3.d, and create a link.

```
cd /etc/rc3.d
ln –s /etc/init.d/chart S99chart
```

5 Make sure that both the WebLogic owner .profile file and the startWebLogic.sh file have the following line:

```
DISPLAY=:2.0; export DISPLAY
```

8 Uninstalling Communications Billing Analytics

Uninstalling Billing Analytics

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

Uninstall Billing Analytics from the **database server** first, then from the **application server**.

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

Before uninstalling Billing Analytics components, you must:

- Stop your application server.
- Stop your database instance.
- Stop your database server.
- Switch user to `root`, which is the default owner of the Uninstall directory.

To uninstall Billing Analytics

1 Navigate to the `Uninstall` folder of your Billing Analytics home directory, `$CBA_HOME`.

2 Launch the Billing Analytics Uninstaller with the following command:

```
./Uninstall_TAM.exe
```

The dot and slash are required, and there is no space after the slash.

The Uninstall screen appears.

3 Click `Uninstall`. A second uninstall screen appears showing Billing Analytics components being removed from your machine.

When the uninstaller is finished, a screen appears listing any items that could not be removed.

4 Change the directory to Oracle's Billing Analytics product home directory and manually remove any remaining files and directories as necessary.

5 Click `Done` to close the uninstaller.

6 Repeat this procedure on your application server and any other installations.