



Installation Guide for Siebel Communications Billing Analytics

For Microsoft Windows 2003 Operating System,
BEA WebLogic Server, and Oracle® Database

Version 5.1.1

Pub Date 2/2/2007

ORACLE®

Copyright © 2005, 2006, 2007 Oracle. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

PRODUCT MODULES AND OPTIONS. This guide contains descriptions of modules that are optional and for which you may not have purchased a license. Siebel's Sample Database also includes data related to these optional modules. As a result, your software implementation may differ from descriptions in this guide. To find out more about the modules your organization has purchased, see your corporate purchasing agent or your Siebel sales representative.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS. Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Contents

1 Preface

About This Guide 7

Operating System Administration Requirements 7

2 Getting Started

Preparing Your Platform 9

Overview of the Installation Process 9

Communications Billing Analytics System Requirements 10

3 Installing Communications Billing Analytics

Installing Communications Billing Analytics 13

4 Configuring the Database Server

Overview 15

Using Database Partitioning 15

5 Configuring Communication Billing Analytics for Oracle

Configuring a New Oracle Database 17

About Creating the Billing Analytics Database 17

Configuring Oracle Services 18

Creating a New Billing Analytics Database 20

What to Do if Database Configuration Fails 23

Connecting to Your Oracle Database 23

6 Configuring the Application Server

Overview 27

Starting and Stopping WebLogic Server 27

7 Configuring Java Resources for WebLogic

Overview 29

About the Sample Windows Domain Used in this Guide 29

About the Sample WebLogic Servers Used in this Guide 30

Starting and Stopping WebLogic Server 30

Sourcing Your Configuration 30

About Sourcing Your Configuration 30

Starting and Stopping an Active Application Server 30

Capturing Your Windows Environment for Billing Analytics 31

Using *edx.config.bat*, *edx_env.bat*, and *edx_load.config.bat* to Store Environment Data 31

Passing Windows Environment Data to WebLogic 33

Passing Your Configuration to WebLogic Running as a Windows Service 33

Passing Your Configuration in a Startup Script for WebLogic 36

Configuring Java Database Connectivity (JDBC) for Billing Analytics 37

Deploying the Billing Analytics Application 38

Windows Services for Billing Analytics 39

Setting up a WebLogic Server Instance as a Windows Service 39

Setting Up the Platform Services Scheduler as a Windows Service 39

Testing the Installation 40

8 WebLogic Reference

Java Database Connectivity (JDBC) 43

WebLogic Environment Variables 43

JDBC Resources for Billing Analytics 43

JDBC Connection Pools 43

JDBC Data Sources 46

JMS Connection Factories 47

JMS (JDBC) Stores 47

JMS Servers 48

Foreign JMS Server 48

JMS Topics 49

JMS Queues 49

JMS Session Pools and Consumers for Logging for Command Center 50

JMS Session Pools and Consumers for Logging for Billing Analytics 53

JTA Timeout 56

Extending XAResource Transaction Timeout 57

Performance-Related Settings 57

9 Uninstalling Communication Billing Analytics

Uninstalling Communication Billing Analytics 59

Before uninstalling Billing Analytics components, you must: 59



1

Preface

About This Guide

This guide is intended for system administrators and other IT professionals and describes how to install and configure the third-party platforms that support the Billing Analytics production environment and deploy Billing Analytics J2EE Web applications. See “**Communications Billing Analytics System Requirements**” on Page 10 for details on the platforms this guide is intended for.

It assumes in-depth understanding of and practical experience with system administrator responsibilities, including:

Operating System Administration Requirements

- Start up and shut down 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 Windows system administration. See the appropriate Windows user documentation.

Related Documentation

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

Title of Guide	Description
<i>Developer's Overview Guide for Siebel Communications Billing Analytics</i>	An Overview of how to use CBA as a development platform and customize it for a particular deployment.
<i>Reporting Developer's Guide for Siebel Communications Billing Analytics</i>	How to customize Billing Analytics application for bill presentment and reporting.
<i>Hierarchy Developer's Guide for Siebel Communications Billing Analytics</i>	How to customize Billing Analytics application to optimize use of your enterprise's structures and data for rapid searches and queries across hierarchies.
<i>Applications Guide for Siebel Communications Billing Analytics</i>	Describes out-of-the-box functionality of the Billing Analytics foundation application.
<i>Administration Guide for Siebel Communications Billing Analytics</i>	How to set up and run a live Billing Analytics application in a J2EE environment.

2 Getting Started

Preparing Your Platform

Before installing Communication 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 Siebel Platform Services and Communications Billing Analytics 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.

Configuring Your Database Server

Configuring your database server requires you to:

- 1 Define database server environment variables.
- 2 Create and configure the Communications Billing Analytics database with ant.
- 3 Connect to your Communications Billing Analytics database before configuring your application server.

Configuring Your Application Server

Configuring your application server requires you to:

- 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

Siebel's Platform Services and Communications Billing Analytics Consumer

This guide assumes you are installing Communications Billing Analytics on a Windows 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.

OPERATING SYSTEM

- Microsoft Windows 2003 Server

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 1.4.2 (version shipped with WebLogic)

Other Software

- Ant version 1.65 or later

SUPPORTED DATABASE SERVERS

New installation of Communications Billing Analytics

- Oracle 9.2.0.7

SUPPORTED APPLICATION SERVERS

- BEA WebLogic Server 8.1 SP6

SUPPORTED BROWSERS

- Netscape Navigator 7 or higher
- Microsoft Internet Explorer 5.5 or later (on networked PC)
- Firefox 1.0



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 administrating your operating system. Consult your system documentation as necessary.

Installing Communications Billing Analytics

Communications Billing Analytics is distributed as an InstallAnywhere package. Follow the steps below to install Billing Analytics on your system. This document refers to that directory as the CBA_HOME directory, listed as C:\Oracle\CBA.

C:\Oracle\CBA\db\mssql contains platform-specific subdirectories for database creation and configuration.

C:\Oracle\CBA\J2EEApps\weblogic contains the Web applications to be deployed to your application server.

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.

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

To install Billing Analytics:

- 1 Log in as an admin user on the application server.
- 2 Launch InstallAnywhere by double-clicking the TAMi ns. exe icon.
- 3 On the Introduction screen, read the Billing Analytics 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. Then click **Next**.
- 6 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.
- 7 On the Choose Product Features screen, click **CBA Group**. Then click **Next**.
- 8 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.

- 9 The Install Complete screen reports a successful installation and the directory that contains the Billing Analytics components.
- 10 Click **Done** to exit the installer.

If the installation fails, determine the cause of the problem and run InstallAnywhere again to reinstall 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:

- Starting and stopping your database server
- Using database partitioning with Billing Analytics

CAUTION: The installation and configuration examples 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.

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 Communication Billing Analytics for Oracle

Configuring a New Oracle Database

To create and configure your Communication Billing Analytics database, you run the database configuration ant script. Be sure to use ant script version 1.6.5.

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

About Creating the Billing Analytics Database

To create and configure the Billing Analytics production database, you run the ant script for database configuration. Be sure to use ant script version 1.6.5.

Before running the script, 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.
- 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:

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

To configure Oracle services:

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

```
cd \Oracle\ora92
```

- 2 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 = TAMOLTP)
  (ORACLE_HOME = C:\Oracle\ora92)
)
```

```
(SID_DESC =
  (SID_NAME = TAMOLAP)
  (ORACLE_HOME = C:\Oracle\ora92)
)
```

- 3 Save and close listener.ora.
 - 4 Change directory to the network administration directory of your database client. For example:
- ```
cd \Oracle\ora92
```
- 5 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.

```
TAMOLTP.db =
 (DESCRIPTION =
 (ADDRESS_LIST =
 (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
```

```

)
(CONNECT_DATA =
 (SID = TAMOLTP)
)
)

TAMOLAP.db =
 (DESCRIPTION =
 (ADDRESS_LIST =
 (ADDRESS = (PROTOCOL = TCP)(HOST = local host)(PORT = 1521))
)
)
(CONNECT_DATA =
 (SID = TAMOLAP)
)
)

```

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

- 6 Save and close *tnsnames.ora*.
- 7 (*multiple machine environments only*) Repeat Step 5 for the *tnsnames.ora* file on your application server. This file is installed with your database client software. Single machine environments may skip this step.

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

```
lsnrctl reload
```

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

```

Services Summary...
 TAMOLAP has 1 service handler(s)
 TAMOLTP has 1 service handler(s)

```

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

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

- 1 Open the template *init.ora* files for the OLAP and OLTP instances:
  - a. `%CBA_HOME%\db\oracle\oltp\init<oraclesid>.ora`
  - b. `%CBA_HOME%\db\oracle\olap\init<oraclesid>.ora`

Note: You can change the settings in the files above before or after installing CBA instances. The new database settings are to conform to Oracle 9i and to replace some obsolete 8i settings. You can reduce the settings down to the lower levels dynamically by logging on as *dba sys* user and executing the following:

```

ALTER SYSTEM SET pga_aggregate_target = 256M;
ALTER SYSTEM SET db_cache_size = 256M;

```

- 2 Edit the **db\_block\_size** parameter to be 8k or 16k (which should match your file system)



db\_block\_size = 8192

## Creating a New Billing Analytics Database

### Changing the edxadmin.properties file

*Provide the following parameters in the edxadmin.properties file:*

- database file locations
- OLAP username/password and database name
- OLTP username/password and database name
- System password for olap and oltp
- Database file locations
- Redo file locations
- ORACLE HOME and ADMIN locations
- Trace file location

**NOTE:** Make sure all target directories exist before running the ant scripts.

### Creating the Database

**Note:** Before running ant, you must configure edxadmin.properties as described above.

*To create the Billing Analytics database:*

- 1 Run the build script by typing **ant** at this location (be sure to use ant script version 1.6.5). By default ant picks up the build.xml in the current directory. The Server Administration Main Menu appears:

```

Main:
 [echo] Siebel DB 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)

```

- 2 From the top level Main Menu, select Option 2, **[2] OLAP Setup** to start. The OLAP menu appears:

```

Main:
 [echo] Install Siebel Reporting
 [echo] [1]. Create Oracle Instance
 [echo] [2]. Shutdown Database
 [echo] [3]. Startup Database
 [echo] [4]. Create Reporting Tablespace
 [echo] [5]. Create Reporting Schema

```

```
[echo] [6]. Install Reporting Schema
[echo] [7]. Sample Reporting Data Population
[echo] [8]. Sample Hierarchy Data (optional)
[echo] [Q]. Quit
[input] Enter your selection(1, 2, 3, 4, 5, 6, 7, 8, q, Q)
```

**3** 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 procedures to contain the absolute pathnames you defined in **edxadmin.properties**. No user input is required, although several progress messages appear.

**NOTE:** This step can take anywhere from 20 minutes to 2 hours to complete, depending on the speed of your platform

If this step is successful, you will see no error messages and ant will exit with the following:

```
Build Successful
```

**4** Run ant again and navigate back to the OLAP main menu from the previous step. This time, select Option 2, **Shutdown Database**. If this step is successful, you will see no error messages and the "Build Successful" message.

**5** Run ant again and navigate back to the OLAP main menu from the previous step. This time, select Option 3, **Startup Database**. If this step is successful, you will see no error messages and the "Build Successful" message.

**6** Run ant again and navigate back to the OLAP main menu from the previous step. Select Step 4, **Create Reporting Tablespaces**.

**7** Run ant again and navigate back to the OLAP main menu from the previous step. Select Step 5, **Create Reporting schema**.

**8** Run ant again and navigate back to the OLAP main menu from the previous step. Select Step 6, **Install Reporting schema**.

**9** Run ant again and navigate back to the OLAP main menu from the previous step. Select Step 7, **Sample Reporting Data Population**.

**NOTE:** This step may take up to 20 minutes to complete.

**10** (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 Guide for Communications Billing Analytics*.

**11** Run ant again, and from the top level Main Menu, select Option 1, **[1] OLTP Setup**. The OLTP menu appears:

```
Main:
[echo] [1]. Install Siebel e-Statement
[echo] [2]. Initial Data Population
[echo] [Q]. Quit
[input] Enter your selection (1, 2, q, Q)
```

**12** Select Option 1, **Install Siebel e-Statement**. The OLTP Install menu appears:

```
CreateInitDatabaseMenu:
[echo] Install Siebel eStatement
```



```
[echo] [1]. Create Oracle Instance
[echo] [2]. Shutdown Database
[echo] [3]. Startup Database
[echo] [4]. Install Application Database I
[echo] [5]. Install Application Database II
[echo] [6]. Install Application Database III
[echo] [Q]. Quit
[input] Enter your selection (1, 2, 3, 4, 5, 6, q, 0)
```

**13** 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 **edxadmin.properties**. No user input is required, although several progress messages appear.

**NOTE:** This option may take up to 20 minutes to complete.

If this step is successful, you will see no error messages and ant will exit with the following:

```
Build Successful
```

**14** Run ant again and navigate back to the OLTP Install menu from the previous step. This time, select Option 2, **Shutdown Database**. If this step is successful, you will see no error messages and the “Build Successful” message.

**15** Run ant again and navigate back to the OLTP Install menu from the previous step. This time, select Option 3, **Startup Database**. If this step is successful, you will see no error messages and the “Build Successful” message.

**16** Run ant again and navigate back to the OLTP Install menu from the previous step. Select Option 4, **Install Application Database I**.

This option creates new Billing Analytics database tablespaces, users, and rollback segment data files.

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.

**17** Run ant again and navigate back to the OLTP Install menu from the previous step. 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.

**18** Run ant again and navigate back to the OLTP Install menu from the previous step. Select Option 6, **Install Application Database III**.

This option compiles stored procedures to support database processing. No user input is required.

**19** Run ant again, and from the top level Main Menu, select Option 3, **[3] CREATE DB LINK**.

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

**20** Run ant again and navigate back to the OLTP menu shown in Step 12. This time, select Option 2, **Initial Data Population**. The following menu appears:

OtherOperationsMenu:

```
[echo] [1]. Import initial data set
[echo] [2]. Export Siebel database data
[echo] [3]. Build Sample Hierarchy (Optional)
[echo] [Q]. Quit
[input] Enter your selection (1, 2, 3, q, Q)
```

- 21** Select Option 1, **Import Initial data set**.
- 22** (Optional) If you want to view sample hierarchy data, run ant and navigate back to the above menu; select Option 3, **Build Sample Hierarchy**.

## Using the automated Ant targets

Instead of running manually through each of the steps above, you can use the following ant targets to automate the process. As for the manual process, you must still update edxadmin.properties with the appropriate values before running the below.

- 1** Run the **install-new** target to create new OLTP and OLAP instances with the SIDs specified in the properties file:

```
ant install-new
```

- 2** Use the **install-existing** target to create new OLTP and OLAP schemas on an existing instance with the usernames/passwords specified in the properties file:

```
ant install-existing
```

- 3** If you wish to install sample data, add the **-DloadSampleData=true** argument to the ant call. For example:

```
ant install-existing -DloadSampleData=true
```

- 4** 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
```

## 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.

## Connecting to Your Oracle Database

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



### *To test the OLTP database:*

- 1 Open a command prompt window, and run the **sqlplus** command on your Billing Analytics database, with arguments for your database username, password, and connection string (database alias). For example:

```
sqlplus oltp_dba/edx@TAMOLTP
```

- 2 If the database is connected successfully, a connection message appears.

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

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

```
SQL> show parameters db_name
```

If the database is connected successfully, you see output for your database instance.

```
NAME TYPE VALUE

db_name string TAMOLTP
SQL>
```



### *To test the OLAP database:*

- 1 Open a command prompt window, and run the **sqlplus** command on your Billing Analytics database, with arguments for your database username, password, and connection string (database alias). For example:

```
sqlplus ol_ap_dba/edx@ol_ap
```

If the database is connected successfully, a connection message appears.

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

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

```
SQL> show parameters db_name
```

If the database is connected successfully, you see output for your database instance.

```
NAME TYPE VALUE

db_name string TAMOLAP
SQL>
```

Once your database server tests successfully with the CBA 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. Consult WebLogic Server documentation at <http://edocs.bea.com> as necessary.

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

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

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

If you have not already installed database server components and configured the database server for Billing Analytics, do so now.

For distributed environments, ensure that you have any required database client software installed on WebLogic Server and any other client machines of your database server.

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

- Starting and Stopping WebLogic Server
- Capturing your environment for Billing Analytics

**CAUTION:** The installation and configuration examples 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.

## About the Sample Domain Used in this Guide

This guide uses the following example of a WebLogic domain:  
`%WL_HOME%\user_projects\domains\mydomain`

WebLogic users can use the Domain Configuration Wizard to create the WebLogic domain `%WL_HOME%\user_projects\domains\mydomain`, or replace these pathnames with a custom domain created by your system administrator.

**CAUTION:** If you use a custom domain, be sure to substitute the pathnames accordingly throughout the procedures in this guide. Oracle does not recommend that you accept the default path of `\user_projects`.

## Starting and Stopping WebLogic Server

Developers and system administrators need to be familiar with how to stop and start WebLogic Server and any active Web applications for your platform. Consult your BEA WebLogic documentation for instructions on how to do this.

## Starting and Stopping an Active Application Server

Improperly starting or stopping an application server in an active Billing Analytics 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, to schedule and run jobs in the Command Center.

The default command-line startup shell scripts are fine for an inactive production environment where there are no running jobs. However, the startup process stops immediately if you enter a **Ctrl+C** (often used to force a hard shutdown of the server) in the startup directory, or if you close the terminal session. This can damage your configuration file. It is recommended that you use the Web console and/or the SHUTDOWN command to ensure a graceful shutdown.

# 7

## Configuring Java Resources for WebLogic

### Overview

This chapter assumes in-depth understanding of and practical experience with application server administration. Consult WebLogic Server documentation at <http://edocs.bea.com> as necessary.

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

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

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.

If you have not already installed database server components and configured the database server for Billing Analytics, do so now.

For distributed environments, ensure that you have any required database client software installed on WebLogic Server and any other client machines of your database server.

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

- Starting and Stopping WebLogic Server
- Capturing your Windows environment for Billing Analytics
- Passing Windows Environment Data to WebLogic
- Windows Services for Billing Analytics

**CAUTION:** The installation and configuration examples 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.

### About the Sample Windows Domain Used in this Guide

This guide uses the default WebLogic domain `%WL_HOME%\user_projects\domains\mydomain` (`%WL_HOME%` is the directory path where you installed WebLogic). WebLogic users may use the Domain Configuration Wizard to create this domain or replace these pathnames with a custom domain created by your system administrator.

**CAUTION:** If you use a custom domain, the examples in this guide must be changed accordingly or they may not work. Oracle does not recommend that you accept the default path that includes the `\user_projects\` folder.

## About the Sample WebLogic Servers Used in this Guide

You will create two WebLogic servers in one domain for Billing Analytics. These servers can be on the same or different systems. This document uses the following two servers for examples:

- **online1** - the administrative server; also used for the Billing Analytics customer facing application
- **batch1** - a managed server for Platform Services

## Starting and Stopping WebLogic Server

Developers and system administrators need to be familiar with how to stop and start WebLogic Server and any active Web applications for your platform. Consult your BEA WebLogic documentation for instructions on how to do this.

# Sourcing Your Configuration

## About Sourcing Your Configuration

Before you start your WebLogic server instance, you must edit its startup script to source your customized version of the configuration file `edx.conf`, which passes your Billing Analytics environment to WebLogic Server at startup.

## Starting and Stopping an Active Application Server

Improperly starting or stopping an application server in an active Billing Analytics 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, to schedule and run jobs in the Command Center.

The default command-line startup shell scripts are fine for an inactive production environment where there are no running jobs. However, the startup process will stop immediately if you enter a **Ctrl+C** (often used to force a hard shutdown of the server) in the startup directory, or if you close the terminal session.

By default, if you use the Windows Control Panel to stop a server instance, the Windows Service Control Manager (SCM) kills the server's Java Virtual Machine (JVM). If you kill the JVM, the server immediately stops all processing. Any session data is lost. If you kill the JVM for an Administration Server while the server is writing to the `conf`.xml file, you can corrupt the `conf`.xml file. See BEA Documentation for Enabling Graceful Shutdowns from the Control Panel.

# Capturing Your Windows Environment for Billing Analytics

Billing Analytics installs several configuration files that you use to define your Billing Analytics environment. These configuration scripts are required **only on the application server**:

|                                       |                                                                                                                                               |
|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| %EDX_HOME%\config\edx_load.config.bat | Editable configuration file stores database Java options required by your application server                                                  |
| %EDX_HOME%\config\edx.config.bat      | Shell script passes the environment data in edx_env.bat and edx_load.config.bat to your application server when called in your startup script |

This section describes how to run `edx.config` to capture your environment variables.

## Using `edx.config.bat`, `edx_env.bat`, and `edx_load.config.bat` to Store Environment Data

You must edit the configuration file `edx_load.config.bat` to set values for your database user, password, and server name. Then you must also manually edit `edx.config.bat`.

You may want to make backup copies of the following files:

- `edx.config.bat`
- `edx_env.bat`
- `edx_load.config.bat`

### To edit Windows environment data with `edx.config.bat`:

- 1 Navigate to `%EDX_HOME%\config` and edit `edx.config.bat`.
- 2 Add the following lines to the `:wlMain` section. (Add `Verisign.jar` to the `CLASSPATH` only if you are implementing credit card payment functionality.)

```
@set CLASSPATH=%CLASSPATH%; %EDX_HOME%\lib\edx_client.jar
@set CLASSPATH=%CLASSPATH%; %EDX_HOME%\lib\jakarta-oro-2.0.7.jar
@set CLASSPATH=%CLASSPATH%; %EDX_HOME%\lib\edx_common.jar
@set CLASSPATH=%CLASSPATH%; %EDX_HOME%\lib\edx_system.jar
@set CLASSPATH=%CLASSPATH%; %CBA_HOME%\lib\commons-logging.jar
@set CLASSPATH=%CLASSPATH%; %CBA_HOME%\lib\Configuration.jar
@set CLASSPATH=%CLASSPATH%; %CBA_HOME%\lib\javachart.jar
@set CLASSPATH=%CLASSPATH%; %CBA_HOME%\lib\deprotocol.jar
@set CLASSPATH=%CLASSPATH%; %CBA_HOME%\lib\log4j.jar
```



- 3 Comment out the memory arguments (we use a different memory arguments default):

```
@rem set JAVA_OPTIONS=%JAVA_OPTIONS% -ms128m -mx128m -Xnoclassgc
```

- 4 Still in *edx.config.bat*, add `|com.edocs.domain.telco.ide.protocol` to the following line:

```
@set JAVA_OPTIONS=%JAVA_OPTIONS% -Dedx.home="%EDX_HOME%" -
Djava.protocol.handler.pkgs="com.edocs.protocol "
```

to make:

```
@set JAVA_OPTIONS=%JAVA_OPTIONS% -Dedx.home="%EDX_HOME%" -
Djava.protocol.handler.pkgs="com.edocs.protocol |com.edocs.domain.telco.ide.protocol
| "
```

- 5 Save and close all files.

### **Removing the old JCL JAVA setting:**

If you are upgrading from an older version of CBA, you must remove the previous JCL logger JAVA setting from `JAVA_OPTIONS`. CBA now uses `log4j`.

### **To edit Windows environment data with *edx\_env.bat*:**

- 1 Navigate to `%EDX_HOME%\config` and open *edx\_env.bat*.
- 2 Modify the default settings to reflect your environment. For example:

```
@rem define APP_SERVER
@set APP_SERVER=wl
@rem define APP_SERVER

@rem define EDX_HOME
@set EDX_HOME=C:\Oracle\CBA\statement
@rem define EDX_HOME

@rem define JAVA_HOME
@set JAVA_HOME=C:\bea\jdk142_05
@rem define JAVA_HOME

@rem define WL_HOME
@set WL_HOME=C:\bea\weblogic81
@rem define WL_HOME
```

**CAUTION:** Make sure you set all paths to the appropriate point releases/patches for WebLogic Server and JDK, if necessary. Check the Release Notes and your system documentation for updated requirements.

### **To edit Windows Java Options in *edx\_load.config.bat*:**

- 1 Navigate to `%EDX_HOME%\config` and open *edx\_load.config.bat*.



- 2 Modify the default settings for `com.edocs.tasks.loader` to reflect your database user, password, and server name. Use the settings for your database as described in the previous chapter. For example:

```
@set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.edocs.tasks.loader.user=edx_dba
@set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.edocs.tasks.loader.password=edx
@set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.edocs.tasks.loader.alias=local host
```

- 3 Save and close the file.

**CAUTION:** Make sure you set all paths to the appropriate point releases/patches for WebLogic Server and JDK, if necessary. Check the Release Notes and your system documentation for updated requirements.

## Passing Windows Environment Data to WebLogic

`edx.config.bat` is a batch file that you call and process in your application server startup script to pass your Billing Analytics environment (stored in `edx_env.bat` and `edx_load.config.bat`) to WebLogic.

**CAUTION:** Do not confuse `edx.config.bat` with `edx_env.bat`, in which you enter the environment data to pass to the server.

This section describes how to use `edx.config.bat` to pass your environment data to WebLogic at server startup.

### *To pass your Billing Analytics environment to WebLogic (overview):*

- 1 Determine whether you wish to start WebLogic as a Windows Service or directly from the startup script. Use the appropriate procedure for your service or startup scripts.
- 2 In your **domain** service or startup script, set your Billing Analytics home directory, `%CBA_HOME%`.
- 3 In your **domain** service or startup script, call and process the configuration script `edx.config.bat`. This procedure is called **sourcing** your configuration.
- 4 In the **master** service or startup script, set your `CLASSPATH`, to use the classpath defined in `edx.config.bat`.

## Passing Your Configuration to WebLogic Running as a Windows Service

It is recommended that you install WebLogic Server as a Windows Service, and modifying the script that calls that service. For WebLogic, this file is:

```
%WL_HOME%\user_projects\domains\mydomain\InstallService.cmd.
```

This script calls the master startup script `InstallSvc.cmd`.

Before editing either of these files, be sure to save a backup copy **in a different directory**.



### Example - InstallService.cmd for WebLogic

Bold indicates text you should add or change from the default.

```

.
.
.
@rem Set WLS_USER equal to your system username and WLS_PW equal
@rem to your system password for no username and password prompt
@rem during server startup. Both are required to bypass the startup
@rem prompt.

set WLS_USER=
set WLS_PW=

.
.
.

set EDX_HOME=C:\Oracl e\CBA\estatement
call %EDX_HOME%\confi g\edx. confi g. bat

@rem Set JAVA_OPTIONS to the java flags you want to pass to the vm. i.e.:
@rem set JAVA_OPTIONS=-Dweblogi c.attri bute=val ue -Dj ava.attri bute=val ue

set JAVA_OPTIONS=%JAVA_OPTIONS%

@rem Set JAVA_VM to the java virtual machine you want to run. For instance
@rem set JAVA_VM=-server

set JAVA_VM=

@rem Set MEM_ARGS to the memory args you want to pass to java. For instance
@rem set MEM_ARGS=-Xms32m -Xmx200m

Set MEM_ARGS="-Xss1m -server -Xms512m -Xmx512m -XX:MaxPermSi ze=128m -
XX: +UseLWPSynchroni zati on -XX: +UseThreadPri ori ti es -Xconcurrenti o"

.
.
.

: installSvc

rem *** Set up extra path for win32 and win64 platform separately

if not "%WL_USE_64BITDLL%" == "true" set
EXTRAPATH=%WL_HOME%\server\bin;%JAVA_HOME%\jre\bin;%JAVA_HOME%\bin;%WL_HOME%\serve
r\bin\oci920_8;%EDX_HOME%\iib

if "%WL_USE_64BITDLL%" == "true" set
EXTRAPATH=%WL_HOME%\server\bin\win64;%WL_HOME%\server\bin;%JAVA_HOME%\jre\bin;%JAV
A_HOME%\bin;%WL_HOME%\server\bin\win64\oci920_8;%EDX_HOME%\iib

rem *** Install the service

```

```
"%WL_HOME%\server\bin\beasvc" -install -svcname:"beasvc"
%DOMAIN_NAME%_SERVER_NAME%" -javahome:"%JAVA_HOME%" -execdir:"%USERDOMAIN_HOME%"
-extrapath:"%EXTRAPATH%" -password:"%WLS_PW%" -cmdline:%CMDLINE%

: finish

ENDLOCAL
```

### *To edit InstallService.cmd for WebLogic:*

See the example above for default settings. Make sure to change these as needed for your environment.

**CAUTION:** Make sure you set all paths to the appropriate point releases/patches for WebLogic Server and JDK, if necessary. Check the Release Notes and your system documentation for updated requirements.

- 1 Stop WebLogic Server and all application server instances.
- 2 Navigate to the %WL\_HOME%\user\_project\domains\mydomain subdirectory of your application server home directory.
- 3 Open InstallService.cmd by right clicking on its name, and selecting **Edit**.
- 4 Before the **JAVA\_OPTIONS** definition, set EDX\_HOME and call edx.config.bat. For example:  

```
set CBA_HOME= C:\Oracle\CBA

set ORACLE_DRIVER_PATH=%CBA_HOME%\lib\jdbc14.jar

set EDX_HOME=%CBA_HOME%\statement
call %EDX_HOME%\config\edx.config.bat
```
- 5 Set **JAVA\_OPTIONS** to %JAVA\_OPTIONS%.
- 6 Set **JAVA\_VM** to null.
- 7 Optimize JVM Memory by increasing the memory arguments allocated to the Java Virtual Machine (JVM) on the application server. For example (quotes are optional):  

```
set MEM_ARGS="-Xss1m -server -Xms512m -Xmx512m -XX:MaxPermSize=256m -
XX:+UseLWPSynchronization -XX:+UseThreadPriorities -Xconcurrentio"
```
- 8 Add your CBA\lib directory to the EXTRAPATH setting. See the examples above.
- 9 (Optional) You can set your application server user and password in the script (to bypass entering it in a console window) by specifying them for WLS\_USER and WLS\_PW.
- 10 Save and close InstallService.cmd.

### *To edit InstallSvc.cmd for WebLogic:*

**CAUTION:** This procedure is required by a defect in WebLogic that does not correctly pass classpath settings from the domain to the master when the master script is called. Consult your WebLogic administrator when editing scripts that control multiple domains.

- 1 Stop WebLogic Server and all application server instances.



- 2 Navigate to the %WL\_HOME%\weblogic81\server\bin subdirectory of your application server home directory.
- 3 Open InstallSvc.cmd by right clicking on its name, and selecting **Edit**.
- 4 Set your Billing Analytics home directory and call your Billing Analytics environment script right after the CLASSPATH setting. For example:
 

```
set EDX_HOME=C:\Oracle\CBA\statement
call %EDX_HOME%\config\edx.config.bat
```
- 5 Save and close InstallSvc.cmd.

## Passing Your Configuration in a Startup Script for WebLogic

You can also choose to start WebLogic Server directly by modifying the server startup script to source your configuration. WebLogic recommends that you start up the server from your domain, using %WL\_HOME%\user\_projects\domains\mydomain\startWebLogic.cmd.

Before editing either of these files, be sure to save a backup copy **in a different directory**.

### Example - startWebLogic.cmd for WebLogic

```
.
.
.

@REM Initialize the common environment.

set WL_HOME=C:\bea\weblogic81
for %%i in ("%WL_HOME%") do set WL_HOME=%%~fsi

set PRODUCTION_MODE=true

set JAVA_VENDOR=Sun

set JAVA_HOME=C:\bea\jdk142_05
for %%i in ("%JAVA_HOME%") do set JAVA_HOME=%%~fsi

MEM_ARGS="-Xss1m -server -Xms128m -Xmx512m -XX:MaxPermSize=256m -
XX:+UseLWPSynchronization -XX:+UseThreadPriorities -Xconcurrentio"

@REM Call commEnv here AFTER setting the java_vendor to get common environmental
settings.

call "%WL_HOME%\common\bin\commEnv.cmd"

@REM Set SERVER_NAME to the name of the server you wish to start up.

set SERVER_NAME=myserver

set CLASSPATH=%WEBLOGIC_CLASSPATH%; %POINTBASE_CLASSPATH%; %JAVA_HOME%\jre\
lib\rt.jar; %WL_HOME%\server\lib\webservices.jar; %CLASSPATH%

set EDX_HOME=C:\Oracle\CBA\statement
call %EDX_HOME%\config\edx.config.bat

set CLASSPATH="%ORACLE_DRIVER_PATH%; %CLASSPATH%"
```

**To edit startWebLogic.cmd for WebLogic:**

(See the example above for default settings. Make sure to change these as needed for your environment.)

- 1 Stop WebLogic Server and all application server instances.
- 2 Edit %WL\_HOME%\user\_projects\domains\mydomain\startWebLogic.cmd.
- 3 Optimize JVM Memory by increasing the memory arguments allocated to the Java Virtual Machine (JVM) on the application server. For example (quotes are optional):
- 4 set MEM\_ARGS=-ms128m -mx128m -Xss1m -noclassgc
- 5 Set your home directory %EDX\_HOME% and call edx.config.bat just after the set CLASSPATH statement. For example:
 

```
set CBA_HOME= C:\Oracle\CBA
set ORACLE_DRIVER_PATH=%CBA_HOME%\lib\jdbc14.jar
set EDX_HOME=%CBA_HOME%\statement
call %EDX_HOME%\config\edx.config.bat
```
- 6 Add the **classpath** parameter to the **java** command at the end so it uses the paths to the Billing Analytics classes. For example:
 

```
%JAVA_HOME%\bin\java %JAVA_VM% %MEM_ARGS% %JAVA_OPTIONS% -classpath "%CLASSPATH%"
-Dweblogic.Name=%SERVER_NAME% -Dweblogic.ProductionModeEnabled=%PRODUCTION_MODE% -
Djava.security.policy="%WL_HOME%\server\lib\weblogic.policy" weblogic.Server
```
- 7 At the bottom of the file before the Java command to start the WebLogic Server is issued insert the following
 

```
set CLASSPATH="%ORACLE_DRIVER_PATH%:%CLASSPATH%"
```
- 8 Save and close startWebLogic.cmd.

## Configuring Java Database Connectivity (JDBC) for Billing Analytics

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 Billing Analytics

**JDBC connection pools** contain named groups of JDBC Connections that are created when the connection pool is registered, usually when starting up WebLogic Server. WebLogic 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 Billing Analytics

You must create JDBC connection pools and transaction data sources for the Billing Analytics and Platform Services WebLogic Servers.

See Chapter 8, *WebLogic Reference*, for the appropriate WebLogic JDBC configuration settings for each server.

**NOTE: You must implement all the settings in Chapter 8 before you can deploy the Billing Analytics Application.**

## Deploying the Billing Analytics Application

After configuring your WebLogic domain server, you can deploy the EAR file to the appropriate servers:

- **Application servers:** Deploy the Billing Analytics EAR files to the appropriate servers.

The EAR files are located at:

| Feature                    | Location                        | File Name          | Server Target |
|----------------------------|---------------------------------|--------------------|---------------|
| Billing Analytics Consumer | %CBA_HOME%\J2EEApps\CBA\tam-tbm | tam-tbm.ear        | online1       |
| Platform Services          | %CBA_HOME%\J2EEApps\estatement  | ear-eStatement.ear | batch1        |

Consult your BEA WebLogic documentation on how to deploy applications.

## Testing the Billing Analytics Installation

Once deployed, you should be able to successfully log in to Billing Manager with the following usernames/passwords.

- 1 Use following URL to access the application: `http://your_host_name:7001/tbmb/`
- 2 Login using the following usernames/passwords:

| Company            | First Name | Last Name | Login/Password |
|--------------------|------------|-----------|----------------|
| American HighTech1 | John       | Smith     | JSMITH         |
| American HighTech1 | Aron       | Bush      | ABUSH          |
| American HighTech1 | Charles    | Andrews   | CANDREWS       |
| American HighTech1 | James      | Book      | JBOOK          |
| British Footwear   | Tim        | Burr      | TBURR          |

| Company              | First Name | Last Name | Login/Password |
|----------------------|------------|-----------|----------------|
| 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         |

- Once you are logged in, click on the **Analytics** tab. The Billing Analytics Reporting List page appears.

## Windows Services for Billing Analytics

### Setting up a WebLogic Server Instance as a Windows Service

If you want a WebLogic Server instance to start automatically when you boot a Windows host computer, you can set up the server as a Windows service.

For detailed instructions on setting up WebLogic Server as a Windows Service, see the BEA documentation.

### Setting Up the Platform Services Scheduler as a Windows Service

After all Billing Analytics EAR files have been deployed to the application server **and** WebLogic is running, you must start the Billing Analytics Scheduler in order to schedule and run jobs in the Command Center. If you attempt to run a new job with the Scheduler not running, the job will not run and you will see 'Not yet started' as its status.

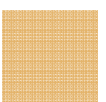
To install the Scheduler as a Windows Service, you must modify the Scheduler template file **SCH.txt**, installed to the **bin** directory for Billing Analytics.

#### *To install the Scheduler as a Windows Service:*

- Navigate to the **bin** directory for Platform Services, or `%EDX_HOME%\bin`.
- Open the Scheduler template file **SCH.txt** and modify the Java classpath to reflect your active Java environment, for example:

```
classpath=C:\bea\jdk142_05\lib\tools.jar;c:\bea\wlserver\lib\weblogic.jar;c:\Oracl
e\CBA\statement\lib\edx_client.jar;c:\Oracl e\CBA\statement\lib\edx_common.jar
```

**CAUTION:** Make sure you set all paths to the appropriate point releases/patches for WebLogic Server and JDK, if necessary. Check the Release Notes and your system documentation for updated requirements.



- 3 Confirm that the following line of code is present in the file for your host and port:  
`-Djava.naming.provider.url=t3://localhost:7001`
- 4 If you want the Scheduler to log information to a file rather than to the console, add the following value in SCH.txt:  
`-Dcom.edocs.pwc.debug=true scheduler_logfile_name`
- 5 Confirm that all the directory references in SCH.txt are correct.
- 6 Save and close SCH.txt.
- 7 Open a command prompt window, and then change directory to %EDX\_HOME%\bin. Use the schedulerSvc command to install the Scheduler as a Windows Service, for example:  
`C:\> schedulerSvc -install C:\Oracle\CBA\estatement\bin\SCH.txt`
- 8 If the Scheduler service is installed successfully, a confirmation message appears.

## Testing the Installation

### Testing Billing Analytics

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

#### *To log into the sample Billing Analytics application:*

- 1 In your browser, point to <http://localhost:7001/tbmb> (where localhost:7001 is your server name and application port number if you are on a different machine).
- 2 Log in using the following username/password pairs to check the application:

| Username and Password | Role       |
|-----------------------|------------|
| JSMITH                | ADMIN      |
| ABUSH                 | ADMIN      |
| CANDREWS              | MANAGER    |
| JBOOK                 | SUBSCRIBER |

### Testing Platform Services

#### *To create a new application:*

- 1 Enter the URL <http://localhost:7001/edocs> (substitute the host:port, if necessary), which displays the Command Center.
- 2 The User Name and Password is admin/edocs.
- 3 Enter the new Application Name, for example, **testApp**.



- 4 Use /edx/ej b/EdocsDataSource for Datasource Name
- 5 Choose the default for Index Partition Count.
- 6 Click on the **Create Application** button.





# 8

## WebLogic Reference

### Java Database Connectivity (JDBC)

You must enter the same information six times: one connection pool and one Tx data source each for **Admin**, **User**, and **Logger**. Make sure you have chosen the correct properties for your application server and database server, and that each data source and its properties maps to the connection pool of the same name.

For details of how to configure JDBC connections, see your application server documentation. For the procedure to create connections for Billing Analytics, see "JDBC Resources for Billing Analytics" on page 43.

**CAUTION:** Make sure you are using the correct properties for your application server, database, and JDBC resource.

### WebLogic Environment Variables

**CAUTION:** Make sure you set all paths to the appropriate point releases/patches for WebLogic Server and JDK, if necessary. Check the Release Notes and your system documentation for updated requirements to these environment variables.

| VARIABLE   | DESCRIPTION           | Default Value (used in this doc) |
|------------|-----------------------|----------------------------------|
| APP_PORT   | app server port       | 7001                             |
| ADMIN_PORT | app server admin port | 7002                             |
| JAVA_HOME  | Java home directory   | \$WLHOME/jdk142_05               |

### JDBC Resources for Billing Analytics

#### JDBC Connection Pools

You will create six connection pools, using different drivers. Set the target for all the connection pools to the Platform Services server, **batch1**.

Create the following JDBC Connection Pools, using WebLogic Server documentation at <http://edocs.bea.com>.

#### Oracle's Thin Driver

WebLogic creates a new JDBC Connection Pool using a wizard. Follow the prompts, and enter:

- Database type = Oracle
- Database Driver = \*Oracle's driver (Thin) Versions: 8.1.7, 9.0.1, 9.2.0, 10

To create the first connection pools:

- edxAdminConnectionPool
- edxLoggerConnectionPool
- edxUserConnectionPool

### Oracle's XA Thin Driver

To create the remaining connection pools, follow the prompts, and enter:

- Database type = Oracle
- Database Driver = \*Oracle's driver XA (Thin) Versions: 8.1.7, 9.0.1, 9.2.0, 10

To create the following connection pools:

- edxXMAConnectionPool
- edxMessagingConnection Pool
- edxReportConnectionPool

### Connection Pool Settings

For each connection pool, you will enter the database information and target the server(s) that will use each connection pool. The following table lists the database whose values you will use for each server and the WebLogic Server to target when the connection pool wizard completes.

| Connection Pool         | Database | Targeted Server    | Supports Local Transactions |
|-------------------------|----------|--------------------|-----------------------------|
| edxAdminConnectionPool  | OLTP     | online1 and batch1 |                             |
| edxLoggerConnectionPool | OLTP     | online1 and batch1 |                             |
| edxUserConnectionPool   | OLTP     | online1 and batch1 |                             |

| Connection Pool            | Database | Targeted Server    | Supports Local Transactions |
|----------------------------|----------|--------------------|-----------------------------|
| edxXMAConnectionPool       | OLTP     | online1 and batch1 | YES                         |
| edxMessagingConnectionPool | OLTP     | online1 and batch1 | YES                         |
| edxReportConnectionPool    | OLAP     | online1 and batch1 | YES                         |

The following table shows an **example** of the database settings for each connection pool:

|                   |                                                                               |
|-------------------|-------------------------------------------------------------------------------|
| Database name     | Enter the Oracle SID. For example, for OLTP: <b>edx0</b>                      |
| Database User     | Enter the database user name. For example, for OLTP: <b>edx_dba</b> .         |
| Database Password | Enter the password for the database user. For example, for OLTP: <b>edx</b> . |

After the wizard completes, go to the Configuration page to make adjustments using the values shown in the following table (on the Connections tab, click **Show** for Advanced Options):

| TAB: Connections     |                           |
|----------------------|---------------------------|
| Initial Capacity     | 1                         |
| Maximum Capacity     | 20                        |
| Capacity Increment   | 5                         |
| Login Delay          | 1                         |
| Statement Cache Size | 300                       |
| Test Frequency       | 60                        |
| Allow Shrinking      | <b>True</b> (box checked) |
| Shrink Frequency     | 15                        |



| TAB: Connections           |                                        |
|----------------------------|----------------------------------------|
| Test Reserved Connections  | TRUE (checked)                         |
| Test Released Connections  | FALSE (unchecked)                      |
| Test Table Name            | dual                                   |
| Supports Local Transaction | see the Connection Pool Settings table |

Click **Apply** to save these values for each connection pool.

Set the target for each connection pool according to the Connection Pool Settings table.

**TIP:** After creating the first datasource for each database driver, you can save time by cloning that datasource to create the next one

## JDBC Data Sources

Create the following transaction data sources, using WebLogic Server documentation at <http://edocs.bea.com>. Set the properties for all data sources as shown in the last table. You can create the first datasource, and then clone it to create the others.

|           |                        |
|-----------|------------------------|
| Name      | edxAdminDataSource     |
| JNDI Name | edx.databasePool       |
| Pool Name | edxAdminConnectionPool |

|           |                       |
|-----------|-----------------------|
| Name      | edxUserDataSource     |
| JNDI Name | edx.user.databasePool |
| Pool Name | edxUserConnectionPool |

|           |                         |
|-----------|-------------------------|
| Name      | edxLoggerDataSource     |
| JNDI Name | edx.logger.databasePool |
| Pool Name | edxLoggerConnectionPool |

|           |                         |
|-----------|-------------------------|
| Name      | edxreportDataSource     |
| JNDI Name | edx.report.databasePool |
| Pool Name | edxreportConnectionPool |

|           |                                         |
|-----------|-----------------------------------------|
| Name      | <code>edxMessagingDataSource</code>     |
| JNDI Name | <code>edx.messaging.databasePool</code> |
| Pool Name | <code>edxMessagingConnectionPool</code> |

|           |                                   |
|-----------|-----------------------------------|
| Name      | <code>edxXMADataSource</code>     |
| JNDI Name | <code>edx.xma.databasePool</code> |
| Pool Name | <code>edxXMAConnectionPool</code> |

For all data sources, set the following properties:

| Configuration Tab - Advanced Options (use defaults) |                                |
|-----------------------------------------------------|--------------------------------|
| Emulate Two-Phase Commit for non-XA Driver          | <code>FALSE</code> (unchecked) |
| Row Prefetch Enabled                                | <code>FALSE</code> (unchecked) |
| Stream Chunk Size: bytes                            | <code>256</code>               |

On the **Targets** tab, select the same servers you specified for the connection pools associated with each data source.

## JMS Connection Factories

Create the following JMS connection factories, using WebLogic Server documentation at <http://edocs.bea.com>. You may accept the default **Properties** for both connection factories, or consult your application server administrator to tune these values.

| Name                                       | JNDI Name                | WebLogic Server                              |
|--------------------------------------------|--------------------------|----------------------------------------------|
| <code>edxLoggerTCF</code>                  | <code>edx.tcf.log</code> | <code>online1</code> and <code>batch1</code> |
| <code>edxMessagingConnectionFactory</code> | <code>edx.qcf</code>     | <code>online1</code> and <code>batch1</code> |

For `edxMessagingConnectionFactory`, also select the **Transactions Tab**, and check **XA Connection Factory Enabled**.

## JMS (JDBC) Stores

Create four JMS JDBC Stores, using WebLogic Server documentation at <http://edocs.bea.com>. You may accept the default **Prefix Name=<NULL>** for all four stores, or consult your application server administrator to tune these values. Also accept the default **Synchronous Write Policy=Cache-Flush**.



| Name (of JMS Store) | Connection Pool         | Directory    |
|---------------------|-------------------------|--------------|
| edxCCLoggerStore    | edxLoggerConnectionPool |              |
| edxLoggerStore      | edxLoggerConnectionPool |              |
| xmaEventFileStore   |                         | xmafilestore |
| CCxmaEventFileStore |                         | xmafilestore |

## JMS Servers

Create four JMS Servers, using WebLogic Server documentation at <http://edocs.bea.com>. You may accept the default **Properties** for all four servers, or consult your application server administrator to tune these values.

| Name              | (Persistent) Store  | Targets Tab |
|-------------------|---------------------|-------------|
| edxCCLoggerServer | edxCCLoggerStore    | batch1      |
| edxLoggerServer   | edxLoggerStore      | online1     |
| CCxmaEventServer  | CCxmaEventFileStore | batch1      |
| xmaEventServer    | xmaEventFileStore   | online1     |

## Foreign JMS Server

Create a Foreign JMS server, along with its JMSConnectionFactory and JMSDestination.

### *To create the Foreign JMS server*

Click on Service -> JMS -> Foreign JMS Servers -> Configure a new Foreign JMS Server and enter the following values:

|                            |                                                                                                                   |
|----------------------------|-------------------------------------------------------------------------------------------------------------------|
| <b>Name</b>                | edxForeignJMSServer                                                                                               |
| <b>JNDI Connection URL</b> | t3://<reporting_hostname or IP address>:<reporting server port number><br><br>For example:<br>t3://localhost:7001 |

Click on Create and target to both the online1 and batch1 servers and apply; the JNDI port should match online1.

### *To create the JMS Connection Factory:*

Click on Services -> JMS -> Foreign JMS Servers -> edxForeignJMSServer -> Configure Foreign JMSConnectionFactory and enter the following values:



|                         |                                             |
|-------------------------|---------------------------------------------|
| <b>Name</b>             | <code>edxForeignJMSConnectionFactory</code> |
| <b>Local JNDI Name</b>  | <code>edx.foreign.qcf</code>                |
| <b>Remote JNDI Name</b> | <code>edx.qcf</code>                        |

*To create the JMS Destination:*

Click on Services -> JMS -> Foreign JMS Servers -> edxForeignJMSServer -> Configure Foreign JMSDestination and enter the following values:

|                         |                                         |
|-------------------------|-----------------------------------------|
| <b>Name</b>             | <code>edxForeignJMSDestination</code>   |
| <b>Local JNDI Name</b>  | <code>edx.foreign.queue.outbound</code> |
| <b>Remote JNDI Name</b> | <code>edx.queue.outbound</code>         |

## JMS Topics

Create two JMS Topics, using WebLogic Server documentation at <http://edocs.bea.com>. Select **Destinations** under each defined Server, then click on **Configure a new JMSTopic**. Make sure to create the matching topic for each server.

|                  |                               |                             |
|------------------|-------------------------------|-----------------------------|
| <b>Name</b>      | <code>edxCCLoggerTopic</code> | <code>edxLoggerTopic</code> |
| <b>JNDI Name</b> | <code>edx.jms.log</code>      | <code>edx.jms.log</code>    |

## JMS Queues

Under JMS, Servers, expand the xmaEventServer, right click on Destinations and choose **Configure a new JMSQueue**:

|                  |                                 |
|------------------|---------------------------------|
| <b>Name</b>      | <code>xmaEventQueue</code>      |
| <b>JNDI Name</b> | <code>edx.queue.outbound</code> |

Repeat the same sequence to create another Destination/JMSQueue:

|                  |                                 |
|------------------|---------------------------------|
| <b>Name</b>      | <code>xmaEventErrorQueue</code> |
| <b>JNDI Name</b> | <code>edx.queue.errors</code>   |

For the xmaEventQueue, click on the **Redelivery** tab, and set the following values:



|                                 |                                 |
|---------------------------------|---------------------------------|
| <b>Redeliver delay override</b> | 1000                            |
| <b>Redelivery Limit</b>         | 3                               |
| <b>Error Destination</b>        | <code>xmaEventErrorQueue</code> |

Click on the Expiration Policy tab, and set the following values:

|                          |                 |
|--------------------------|-----------------|
| <b>Expiration Policy</b> | <b>Redirect</b> |
|--------------------------|-----------------|

Repeat the same steps for the `CCxmaEventServer`. Under JMS, Servers, expand the `CCxmaEventServer`, right click on Destinations and choose **Configure a new JMSQueue**:

|                  |                                 |
|------------------|---------------------------------|
| <b>Name</b>      | <code>CCxmaEventQueue</code>    |
| <b>JNDI Name</b> | <code>edx.queue.outbound</code> |

Repeat the same sequence to create another Destination/JMSQueue:

|                  |                                   |
|------------------|-----------------------------------|
| <b>Name</b>      | <code>CCxmaEventErrorQueue</code> |
| <b>JNDI Name</b> | <code>edx.queue.errors</code>     |

For the `CCxmaEventQueue`, click on the **Redelivery** tab, and set the following values:

|                                 |                                   |
|---------------------------------|-----------------------------------|
| <b>Redeliver delay override</b> | 1000                              |
| <b>Redelivery Limit</b>         | 3                                 |
| <b>Error Destination</b>        | <code>CCxmaEventErrorQueue</code> |

Click on the Expiration Policy tab, and set the following values:

|                          |                 |
|--------------------------|-----------------|
| <b>Expiration Policy</b> | <b>Redirect</b> |
|--------------------------|-----------------|

## JMS Session Pools and Consumers for Logging for Command Center

Create FIVE pairs of **JMS Session Pools and Consumers** for the `edxCCLoggerServer`, using WebLogic Server documentation at <http://edocs.bea.com>. Set **Acknowledge Mode** to **auto** and **Sessions Maximum** to **-1** for all five Session Pools.

**TIP:** For each session pool, -1 specifies no session maximum. Tune each Session Maximum to the maximum number of threads for each pool.

## Admin Activity

### JMS Session Pool - Configuration Tab

| Property           | Value                                                       |
|--------------------|-------------------------------------------------------------|
| Name               | <code>edxCCLoggerAdminActivityPool</code>                   |
| Connection Factory | <code>edx.tcf.log</code>                                    |
| Listener Class     | <code>com.edocs.fs.logging.sub.AdminActivityListener</code> |
| Acknowledge Mode   | <code>auto</code>                                           |
| Sessions Maximum   | <code>-1</code>                                             |

### JMS Consumer- Configuration Tab

| Property         | Value                                         |
|------------------|-----------------------------------------------|
| Name             | <code>edxCCLoggerAdminActivityConsumer</code> |
| Messages Maximum | <code>10</code>                               |
| Selector         | <code>JMSType= 'ADM'</code>                   |
| Destination      | <code>edx.jms.log</code>                      |

## CSR Activity

### JMS Session Pool- Configuration Tab

| Property           | Value                                                     |
|--------------------|-----------------------------------------------------------|
| Name               | <code>edxCCLoggerCSRActivityPool</code>                   |
| Connection Factory | <code>edx.tcf.log</code>                                  |
| Listener Class     | <code>com.edocs.fs.logging.sub.CSRActivityListener</code> |
| Acknowledge Mode   | <code>auto</code>                                         |
| Sessions Maximum   | <code>-1</code>                                           |

### JMS Consumer- Configuration Tab

| Property         | Value                                       |
|------------------|---------------------------------------------|
| Name             | <code>edxCCLoggerCSRActivityConsumer</code> |
| Messages Maximum | <code>10</code>                             |
| Selector         | <code>JMSType= 'CSR'</code>                 |



| Property    | Value                    |
|-------------|--------------------------|
| Destination | <code>edx.jms.log</code> |

## Message Log

### JMS Session Pool- Configuration Tab

| Property           | Value                                                    |
|--------------------|----------------------------------------------------------|
| Name               | <code>edxCCLoggerMessageLogPool</code>                   |
| Connection Factory | <code>edx.tcf.log</code>                                 |
| Listener Class     | <code>com.edocs.fs.logging.sub.MessageLogListener</code> |
| Acknowledge Mode   | <code>auto</code>                                        |
| Sessions Maximum   | <code>-1</code>                                          |

### JMS Consumer- Configuration Tab

| Property         | Value                                      |
|------------------|--------------------------------------------|
| Name             | <code>edxCCLoggerMessageLogConsumer</code> |
| Messages Maximum | <code>10</code>                            |
| Selector         | <code>JMSType= 'MSG'</code>                |
| Destination      | <code>edx.jms.log</code>                   |

## System Activity

### JMS Session Pool- Configuration Tab

| Property           | Value                                                        |
|--------------------|--------------------------------------------------------------|
| Name               | <code>edxCCLoggerSystemActivityPool</code>                   |
| Connection Factory | <code>edx.tcf.log</code>                                     |
| Listener Class     | <code>com.edocs.fs.logging.sub.SystemActivityListener</code> |
| Acknowledge Mode   | <code>auto</code>                                            |
| Sessions Maximum   | <code>-1</code>                                              |

**JMS Consumer**

| Property         | Value                             |
|------------------|-----------------------------------|
| Name             | edxCCLoggerSystemActivityConsumer |
| Messages Maximum | 10                                |
| Selector         | JMSType= 'SYS'                    |
| Destination      | edx.jms.log                       |

**UserActivity****JMS Session Pool**

| Property           | Value                                         |
|--------------------|-----------------------------------------------|
| Name               | edxCCLoggerUserActivityPool                   |
| Connection Factory | edx.tcf.log                                   |
| Listener Class     | com.edocs.fs.logging.sub.UserActivityListener |
| Acknowledge Mode   | auto                                          |
| Sessions Maximum   | -1                                            |

**JMS Consumer**

| Property         | Value                           |
|------------------|---------------------------------|
| Name             | edxCCLoggerUserActivityConsumer |
| Messages Maximum | 10                              |
| Selector         | JMSType= 'USER'                 |
| Destination      | edx.jms.log                     |

**JMS Session Pools and Consumers for Logging for Billing Analytics**

Create FIVE pairs of **JMS Session Pools and Consumers** for the **edxLoggerServer**, using WebLogic Server documentation at <http://edocs.bea.com>. Set **Acknowledge Mode** to **auto** and **Sessions Maximum** to **-1** for all five Session Pools.

**TIP:** For each session pool, -1 specifies no session maximum. Tune each Session Maximum to the maximum number of threads for each pool.



## Admin Activity

### JMS Session Pool - Configuration Tab

| Property           | Value                                                       |
|--------------------|-------------------------------------------------------------|
| Name               | <code>edxLoggerAdminActivityPool</code>                     |
| Connection Factory | <code>edx.tcf.log</code>                                    |
| Listener Class     | <code>com.edocs.fs.logging.sub.AdminActivityListener</code> |
| Acknowledge Mode   | <code>auto</code>                                           |
| Sessions Maximum   | <code>-1</code>                                             |

### JMS Consumer- Configuration Tab

| Property         | Value                                       |
|------------------|---------------------------------------------|
| Name             | <code>edxLoggerAdminActivityConsumer</code> |
| Messages Maximum | <code>10</code>                             |
| Selector         | <code>JMSType= 'ADM'</code>                 |
| Destination      | <code>edx.jms.log</code>                    |

## CSR Activity

### JMS Session Pool- Configuration Tab

| Property           | Value                                                     |
|--------------------|-----------------------------------------------------------|
| Name               | <code>edxLoggerCSRActivityPool</code>                     |
| Connection Factory | <code>edx.tcf.log</code>                                  |
| Listener Class     | <code>com.edocs.fs.logging.sub.CSRActivityListener</code> |
| Acknowledge Mode   | <code>auto</code>                                         |
| Sessions Maximum   | <code>-1</code>                                           |

### JMS Consumer- Configuration Tab

| Property         | Value                                     |
|------------------|-------------------------------------------|
| Name             | <code>edxLoggerCSRActivityConsumer</code> |
| Messages Maximum | <code>10</code>                           |
| Selector         | <code>JMSType= 'CSR'</code>               |

| Property    | Value                    |
|-------------|--------------------------|
| Destination | <code>edx.jms.log</code> |

## Message Log

### JMS Session Pool- Configuration Tab

| Property           | Value                                                    |
|--------------------|----------------------------------------------------------|
| Name               | <code>edxLoggerMessageLogPool</code>                     |
| Connection Factory | <code>edx.tcf.log</code>                                 |
| Listener Class     | <code>com.edocs.fs.logging.sub.MessageLogListener</code> |
| Acknowledge Mode   | <code>auto</code>                                        |
| Sessions Maximum   | <code>-1</code>                                          |

### JMS Consumer- Configuration Tab

| Property         | Value                                    |
|------------------|------------------------------------------|
| Name             | <code>edxLoggerMessageLogConsumer</code> |
| Messages Maximum | <code>10</code>                          |
| Selector         | <code>JMSType= 'MSG'</code>              |
| Destination      | <code>edx.jms.log</code>                 |

## System Activity

### JMS Session Pool- Configuration Tab

| Property           | Value                                                        |
|--------------------|--------------------------------------------------------------|
| Name               | <code>edxLoggerSystemActivityPool</code>                     |
| Connection Factory | <code>edx.tcf.log</code>                                     |
| Listener Class     | <code>com.edocs.fs.logging.sub.SystemActivityListener</code> |
| Acknowledge Mode   | <code>auto</code>                                            |
| Sessions Maximum   | <code>-1</code>                                              |



**JMS Consumer**

| Property         | Value                                        |
|------------------|----------------------------------------------|
| Name             | <code>edxLoggerSystemActivityConsumer</code> |
| Messages Maximum | 10                                           |
| Selector         | <code>JMSType= 'SYS '</code>                 |
| Destination      | <code>edx.jms.log</code>                     |

**UserActivity**

**JMS Session Pool**

| Property           | Value                                                      |
|--------------------|------------------------------------------------------------|
| Name               | <code>edxLoggerUserActivityPool</code>                     |
| Connection Factory | <code>edx.tcf.log</code>                                   |
| Listener Class     | <code>com.edocs.fs.logging.sub.UserActivityListener</code> |
| Acknowledge Mode   | <code>auto</code>                                          |
| Sessions Maximum   | -1                                                         |

**JMS Consumer**

| Property         | Value                                      |
|------------------|--------------------------------------------|
| Name             | <code>edxLoggerUserActivityConsumer</code> |
| Messages Maximum | 10                                         |
| Selector         | <code>JMSType= 'USER '</code>              |
| Destination      | <code>edx.jms.log</code>                   |

## JTA Timeout

On the WebLogic console, click on your domain in the tree (the default domain is mydomain). Click on the JTA tab, and increase the Timeout Seconds (first parameter on the page) to 6000 seconds.



## Extending XAResource Transaction Timeout

The WebLogic Server Transaction Manager now supports setting a transaction branch timeout value on a participating XA resource if the resource manager supports the `javax.transaction.xa.XAResource.setTransactionTimeout()` method. You may want to set a transaction branch timeout if you have long-running transactions that exceed the default timeout value on the XA resource.

To direct the WebLogic Server Transaction Manager to set the transaction timeout on a JDBC XA resource, specify a value for the following properties in the **Advanced Options** section on the **Connections** tab for each JDBC connection pool:

- **Enable XA Transaction Timeout**-A boolean checkbox. When set to true, the WebLogic Server Transaction Manager calls `XAResource.setTransactionTimeout()` before calling `XAResource.start`, and passes either the `XATransactionTimeout` or the global transaction timeout in seconds. When set to false, the Transaction Manager does not call `setTransactionTimeout()`. The default value is **false**.
- **XA Transaction Timeout**-The number of seconds to pass as the transaction timeout value in the `XAResource.setTransactionTimeout()` method. When this property is set to 0, the WebLogic Server Transaction Manager passes the global WebLogic Server transaction timeout in seconds in the method. The default value for this parameter is **0**. If set, this value should be greater than or equal to the global WebLogic Server transaction timeout. This means that if you've already set JTA Transaction to a larger timeout value, this value should be set to 0.

These properties apply to connection pools that use an XA JDBC driver to create database connections only. They are ignored if a non-XA JDBC driver is used.

When these values are set, the WebLogic Server Transaction Manager calls `XAResource.setTransactionTimeout()` as described above. The implementation of the method in the XA resource manager (for example, an XA JDBC driver) or the XA resource determines how the value is used. For example, for Oracle, the `setTransactionTimeout()` method sets the Session Timeout (SesTm), which acts as a maximum idle time for a transaction. The behavior may be different for other XA Resources.

## Performance-Related Settings

Please note the following settings are only a **recommendation**. These settings have been tested on a server with the following configuration: 4 CPU (1.2Ghz), 8GB memory, Sun-V880 server. The testing was done using one WebLogic server instance that supported 100 concurrent users with 15 seconds think time.

If you are using different and/or newer versions of the OS, Application Server, and/or JDK, these settings may not be optimal. Tuning is essential for every application.

- Memory in the WebLogic startup script (for both Platform Services and Billing Analytics servers)

```
MEM_ARGS="-Xms1024M -Xmx1024M -XX:MaxPermSize=256M -XX:PermSize=128M"
```

For a machine with more RAM, it is better to create multiple clustered instances rather than increase the heap size. 1 GB is generally the upper limit that the JVM can utilize efficiently.



- Memory for the admin server

```
MEM_ARGS="-Xms512M -Xmx512M"
```

Not much memory is required for the admin server unless the server also runs an application, which is **not** recommended.

- Connection pool settings

- edxLoggerConnectionPool: init 10, max 25, Capacity Increment 5, Statement Cache Type Fixed, Statement Cache Size 50.

- edxMessagingConnectionPool: init 10, max 25, Capacity Increment 5, Statement Cache Type Fixed, Statement Cache Size 200.

- edxUserConnectionPool: init 10, max 25, Capacity Increment 5, Statement Cache Type Fixed, Statement Cache Size 150.

- edxXMAConnectionPool: init 40, max 40, Capacity Increment 1, Statement Cache Type Fixed, Statement Cache Size 300. Also add the following property: **PinnedToThread=true** (add this line in the multi-select list **Properties**).

- edxreportConnectionPool: init 40, max 40, Capacity Increment 1, Statement Cache Type Fixed, Statement Cache Size 300. Also add the following property: **PinnedToThread=true** (add this line in the multi-select list **Properties**).

- For command center pool settings, no tuning is required, since the command center does not use a lot of connections. At most, 10 connections should be sufficient for the pools with Statement Cache Type Fixed, and Statement Cache Size 400. You can conserve more database resources by using smaller pools.

- Data source setting

All data sources will have Row Prefetch Enabled in the advanced option. Row Prefetch Size 48 and Stream Chunk Size 256 bytes (all default values).

- Set the size of the execute queue. On the WebLogic Console, click the server of interest (the Billing Analytics application server, there is no need to do this for the command center server); click **General**; click **(Advanced Options) Show**; click **Configure Execute Queues**; click the queue **weblogic.kernel.Default** (make sure you are in the Configuration tab); change the Thread Count to **40**.

All these changes require you to restart the server.

# 9

## Uninstalling Communication Billing Analytics

### Uninstalling Communication 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 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 command `./Uninstall_TAM.exe`. The dot and slash are required, and there is no space after the slash.

```
./Uninstall_TAM.exe
```

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 your Billing Analytics 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.