Installation Guide for Communications Billing Analytics

Microsoft Windows 2003® Operating System
and the BEA WebLogic® Server

Version 5.1.0.1

Pub Date 02/27/2006

Copyright © 2005, 2006, Oracle and/or its affiliates

All rights reserved

Printed in the United States of America

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.

U.S. GOVERNMENT RESTRICTED RIGHTS. Programs, Ancillary Programs and Documentation, delivered subject to the Department of Defense Federal Acquisition Regulation Supplement, are "commercial computer software" as set forth in DFARS 227.7202, Commercial Computer Software and Commercial Computer Software Documentation, and as such, any use, duplication and disclosure of the Programs, Ancillary Programs and Documentation shall be subject to the restrictions contained in the applicable Siebel license agreement. All other use, duplication and disclosure of the Programs, Ancillary Programs and Documentation by the U.S. Government shall be subject to the applicable Siebel license agreement and the restrictions contained in subsection (c) of FAR 52.227-19, Commercial Computer Software - Restricted Rights (June 1987), or FAR 52.227-14, Rights in Data—General, including Alternate III (June 1987), as applicable. Contractor/licensor is Siebel Systems, Inc., 2207 Bridgepointe Parkway, San Mateo, CA 94404.

Proprietary Information

Siebel Systems, Inc. considers information included in this documentation and in Siebel eBusiness Applications Online Help to be Confidential Information. Your access to and use of this Confidential Information are subject to the terms and conditions of: (1) the applicable Siebel Systems software license agreement, which has been executed and with which you agree to comply; and (2) the proprietary and restricted rights notices included in this documentation.

Contents

1 Preface

About This Guide

This guide is intended for system administrators and other IT professionals. It describes how to install and configure the third-party platforms that support the 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

A PDF version of this guide is also available.

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

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

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

2 Getting Started

Preparing Your Platform

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

- Install and test required hardware and software for your platform.
- Define required user and group permissions for your database server and application server.
- Start and test your database server. For details, see your server documentation.
- Start and test your application server. For details, see your application server documentation.
- For distributed environments, make sure you have any required database client software installed on your application server and any other client machines of your database server.

Overview of the Installation Process

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

1. Installing 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

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

The following 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 9i Release 2 (Oracle 9.2.0)
- Oracle 9i client software (for application server), including the sqlplus and sqlldr packages
- Oracle 9i JDBC driver
- Microsoft SQL Server 2000 SP3

SUPPORTED APPLICATION SERVERS

- BEA WebLogic Server 8.1 SP4

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 (CBA, also referred to as Telco Analytics Manager, or TAM) is distributed as an InstallAnywhere package. Follow the steps below to install Billing Analytics on your system. This document refers to that directory as the CBA_HOME directory, listed as C:\Siebel\CBA.

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

C:\Siebel\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. Siebel recommends you use the default Web Application Server owner and group accounts.

Installing Billing Analytics:

1. Log in as an admin user on the application server.
2. Launch InstallAnywhere by double-clicking the `TAMins.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.

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

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 Billing Analytics for Oracle

Configuring a New Oracle Database

To create and configure your Billing Analytics database, you run the database configuration *ant* script.

TIP: Database clustering is handled by your application server and not by Billing Analytics. Consult your Siebel Technical or 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.

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`: This file creates tablespaces specific for the application. It is invoked from the main shell script.
- `crt_rpt_user.sql`: The file creates a new schema. It is 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 Siebel 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 the 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 = localhost)(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`.

Creating a new Billing Analytics Database

Change the edxadmin.properties file

Provide the following parameters in this properties file

- Database file locations
- OLAP username/password and database name
- OLTP username/password and database name
- System password for olap and oltp
- 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

1 Run the build script by typing **ant** at this location. 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 Select Option **[1] OLTP Setup**. The OLTP main 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)
```

3 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,Q)
```

4 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
```

5 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.

6 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.

7 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. Finally, you should see a success message.

8 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.

9 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.

10 Run `ant` again and navigate back to the OLTP Main menu from step 2. 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)
```

11 Select Option 1, **`Import Initial data set`**.

12 OPTIONAL: If you want to view sample hierarchy data, run `ant` and navigate back to the above menu; select Option 3, **`Build Sample Hierarchy`**.

13 Run `ant`, and from the first menu, select Option 2, **`OLAP Setup`**. The OLAP main menu is displayed:

```
main:
     [echo] Install Siebel Reporting
     [echo] [1]. Create Oracle Instance
     [echo] [2]. Shutdown Database
     [echo] [3]. Startup Database
     [echo] [4]. Create Reporting Tablespaces
     [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)
```

14 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
```

15 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.

16 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.

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

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

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

20 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.

21 OPTIONAL: select step 8, **`Sample Hierarchy Data`** from the OLAP main menu if you want to load sample hierarchies. If you skip this step now, you can load the sample data into the hierarch later by running a sequence of jobs. These steps are described in the *Administration Communications Billing Analytics* guide, in the section *Scheduling Job for Sample Data*.

22 Run `ant` again, and from the main menu, select option 3, **`CREATE DB LINK`**.

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

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 with the manual process, you must still update edxadmin.properties with the appropriate values before running any of the statements 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 connection is established 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 you are successfully connected to the database, 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 olap_dba/edx@olap
   ```

   If the database connection is established 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 you are successfully connected to the database, 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.

Siebel 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 environment for Billing Analytics

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

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. Siebel 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. Siebel recommends using 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.

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

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. Siebel 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.config`, 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 Siebel 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 `config.xml` file, you can corrupt the `config.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.

Throughout this section, %EDX_HOME% refers to the %CBA_HOME%\estatement directory.

Using *edx.config.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 run `edx.config.bat`.

Use the Appendix Quick Reference to enter application server environment variables for your platform for each of the specified parameters. You may want to print the *Environment Variables* sections for easy reference. You can accept the default values, if appropriate, or enter your own.

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%;C:\Siebel\CBA\lib\commons-logging-1.0.3.jar
@set CLASSPATH=%CLASSPATH%;C:\Siebel\CBA\lib\Configuration.jar
@set CLASSPATH=%CLASSPATH%;C:\Siebel\CBA\lib\javachart.jar
@set CLASSPATH=%CLASSPATH%;C:\Siebel\CBA\lib\ldeprotocol.jar
@set CLASSPATH=%CLASSPATH%;C:\Siebel\CBA\lib\log4j-1.2.8.jar
@set CLASSPATH=%CLASSPATH%;C:\Siebel\CBA\lib\ldom4j-1.4.jar
@set CLASSPATH=%CLASSPATH%;C:\Siebel\CBA\config
@set CLASSPATH=%CLASSPATH%;C:\Siebel\CBA\payment\lib\payment_client.jar
@set CLASSPATH=%CLASSPATH%;C:\Siebel\CBA\payment\lib\payment_custom.jar
@set CLASSPATH=%CLASSPATH%;C:\Siebel\CBA\payment\lib\payment_common.jar
@set CLASSPATH=%CLASSPATH%;C:\Siebel\CBA\payment\lib\Verisign.jar
@set CLASSPATH=%CLASSPATH%;C:\Siebel\CBA\payment\lib\jnet.jar
@set CLASSPATH=%CLASSPATH%;C:\Siebel\CBA\payment\lib\jsse.jar
@set CLASSPATH=%CLASSPATH%;C:\Siebel\CBA\payment\lib\jcert.jar
```

3 Still in *edx.config.bat*, add `|com.edocs.domain.telco.lde.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.lde.protocol"
```

4 Save and close all files.

To edit Windows environment data with edx.config.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:\Siebel\CBA\estatement
@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=localhost
   ```

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 shell script that you call 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

Siebel recommends installing 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 that 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:\Siebel\CBA\estatement
call %EDX_HOME%\config\edx.config.bat

@rem Set JAVA_OPTIONS to the java flags you want to pass to the vm. i.e.:
@rem set JAVA_OPTIONS=-Dweblogic.attribute=value -Djava.attribute=value

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

MEM_ARGS="-Xss1m -server -Xms512m -Xmx512m -XX:MaxPermSize=128m -
XX:+UseLWPSynchronization -XX:+UseThreadPriorities -Xconcurrentio"

.
.
.

: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%\server\
bin\oci920_8;%EDX_HOME%\lib

if "%WL_USE_64BITDLL%" == "true" set
EXTRAPATH=%WL_HOME%\server\bin\win64;%WL_HOME%\server\bin;%JAVA_HOME%\jre\bin;%JAVA_
HOME%\bin;%WL_HOME%\server\bin\win64\oci920_8;%EDX_HOME%\lib

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 EDX_HOME=C:\Siebel\CBA\estatement
   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:\Siebel\CBA\estatement
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:\Siebel\CBA\estatement
call %EDX_HOME%\config\edx.config.bat

.
.
.
```

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 Siebel home directory `%EDX_HOME%` and call `edx.config.bat` just after the set CLASSPATH statement. For example:

   ```
   set EDX_HOME=C:\Siebel\CBA\estatement
   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. 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\TAM\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 login to the Billing Manager with the sample user accounts shown below (the username and password are the same for each account).

1. Use following URL to access the application: http://your_host_name:7001/tbmb/
2. Login using the following username/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
British Footwear	Tom	Brown	TBROWN
British Footwear	Micheal	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

3. Once you are logged in, click on the Analytics tab. This takes you to the Billing Analytics report list page.

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 Siebel 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:

1. Navigate to the `bin` directory for Platform Services, or `%EDX_HOME%\bin`.
2. 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:\Siebel\
   CBA\estatement\lib\edx_client.jar;c:\Siebel\CBA\estatement\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:\Siebel\CBA\estatement\bin\SCH.txt
   ```
8. If the Scheduler service is installed successfully, a confirmation message appears.

Testing the Installation

Testing the Billing Analytics Installation

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

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).

Login using the following username/password pairs (username and password are the same) to check the application:

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

Testing Platform Services

Create a new application:

1. Enter the URL http://localhost:7001/edocs (substitute the host:port, if necessary), which displays the Command Center Main Page.
2. Login using the following username and password: admin/edocs.
3. Click the Create New Application button.
4. Enter a name for the new application, for example, **`testApp`**.
5. Use `/edx/ejb/EdocsDataSource` for the Datasource Name.
6. Choose the default for Index Partition Count.
7. Click on the **`Create Application`** button.

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* " on page39.

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_OWNER	app server owner	**`edxadmin`**
APP_GROUP	app server group	**`edxadmin`**
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

To create the first connection pools:

- edxAdminConnectionPool
- edxLoggerConnectionPool
- edxUserConnectionPool

BEA Driver

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

- Database type = Oracle
- Database Driver = BEA's Oracle Driver (Type 4XA)

To create the following connection pools:

- edxXMAConnectionPool
- edxMessagingConnection Pool
- reportConnectionPool

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	
edxXMAConnectionPool	OLTP	online1 and batch1	YES
edxMessagingConnection Pool	OLTP	online1 and batch1	YES
reportConnectionPool	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: **`edx0`**
Database User	Enter the database user name. For example, for OLTP: **`edx_dba`**.
Database Password	Enter the password for the database user. For example, for OLTP: **`edx`**.

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	**`True`** (box checked)
Shrink Frequency	**`15`**
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	**`reportDataSource`**
JNDI Name	**`edx.report.databasePool`**
Pool Name	**`reportConnectionPool`**

Name	**`edxMessagingDataSource`**
JNDI Name	**`edx.messaging.databasePool`**
Pool Name	**`edxMessagingConnectionPool`**

Name	**`XMADataSource`**
JNDI Name	**`edx.xma.databasePool`**
Pool Name	**`edxXMAConnectionPool`**

For all data sources, set the following properties:

Configuration Tab - Advanced Options (use defaults)	
Emulate Two-Phase Commit for non-XA Driver	**`FALSE (checked)`**
Row Prefetch Enabled	**`FALSE (unchecked)`**
Stream Chunk Size: bytes	**`256`**

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 all three connection factories, or consult your application server administrator to tune these values.

Name	**JNDI Name**	**WebLogic Server**
`edxLoggerTCF`	**`edx.tcf.log`**	**`online1`** and **`batch1`**
`edxMessagingConnectionFactory`	**`edx.qcf`**	**`online1`** and **`batch1`**

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:

Name	**`edxForeignJMSServer`**
JNDI Connection URL	**`t3://<reporting_hostname or IP address>:<reporting server port number>`** For example: t3://localhost:7001

Click on create and target to reporting server and apply.

To create the JMS Connection Factory

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

Name	**`edxForeignJMSConnectionFactory`**
Local JNDI Name	**`edx.foreign.qcf`**
Remote JNDI Name	**`edx.qcf`**

To create the JMS Desitination:

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

Name	**`edxForeignJMSDestination`**
Local JNDI Name	**`edx.foreign.queue.outbound`**
Remote JNDI Name	**`edx.queue.outbound`**

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.

Name	**`edxCCLoggerTopic`**	**`edxLoggerTopic`**
JNDI Name	**`edx.jms.log`**	**`edx.jms.log`**

JMS Queues

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

Name	**`xmaEventQueue`**
JNDI Name	**`edx.queue.outbound`**

Repeat the same sequence to create another Destination/JMSQueue:

Name	**`xmaEventErrorQueue`**
JNDI Name	**`edx.queue.errors`**

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

Redeliver delay override	**`1000`**
Redelivery Limit	**`3`**
Error Destination	**`xmaEventErrorQueue`**

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

Expiration Policy	**`Redirect`**

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

Name	**`CCxmaEventQueue`**
JNDI Name	**`edx.queue.outbound`**

Repeat the same sequence to create another Destination/JMSQueue:

Name	**`CCxmaEventErrorQueue`**
JNDI Name	**`edx.queue.errors`**

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

Redeliver delay override	**`1000`**
Redelivery Limit	**`3`**
Error Destination	**`CCxmaEventErrorQueue`**

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

Expiration Policy	**`Redirect`**

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	**`edxCCLoggerAdminActivityPool`**
Connection Factory	**`edx.tcf.log`**
Listener Class	**`com.edocs.fs.logging.sub.AdminActivityListener`**
Acknowledge Mode	**`auto`**
Sessions Maximum	**`-1`**

JMS Consumer- Configuration Tab

Property	**Value**
Name	**`edxCCLoggerAdminActivityConsumer`**
Messages Maximum	**`10`**
Selector	**`JMSType='ADM'`**
Destination	**`edx.jms.log`**

CSR Activity

JMS session pool- Configuration Tab

Property	**Value**
Name	**`edxCCLoggerCSRActivityPool`**

Property	Value
Connection Factory	`edx.tcf.log`
Listener Class	`com.edocs.fs.logging.sub.CSRActivityListener`
Acknowledge Mode	`auto`
Sessions Maximum	`-1`

JMS consumer- Configuration Tab

Property	Value
Name	`edxCCLoggerCSRActivityConsumer`
Messages Maximum	`10`
Selector	`JMSType='CSR'`
Destination	`edx.jms.log`

Message Log

JMS session pool- Configuration Tab

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

JMS consumer- Configuration Tab

Property	Value
Name	`edxCCLoggerMessageLogConsumer`
Messages Maximum	`10`
Selector	`JMSType='MSG'`
Destination	`edx.jms.log`

System Activity

JMS session pool- Configuration Tab

Property	**Value**
Name	`edxCCLoggerSystemActivityPool`
Connection Factory	`edx.tcf.log`
Listener Class	`com.edocs.fs.logging.sub.SystemActivityListener`
Acknowledge Mode	`auto`
Sessions Maximum	`-1`

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'`

Property	**Value**
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	**`edxLoggerAdminActivityPool`**
Connection Factory	**`edx.tcf.log`**
Listener Class	**`com.edocs.fs.logging.sub.AdminActivityListener`**
Acknowledge Mode	**`auto`**
Sessions Maximum	**`-1`**

JMS Consumer- Configuration Tab

Property	**Value**
Name	**`edxLoggerAdminActivityConsumer`**
Messages Maximum	**`10`**
Selector	**`JMSType='ADM'`**
Destination	**`edx.jms.log`**

CSR Activity

JMS session pool- Configuration Tab

Property	**Value**
Name	**`edxLoggerCSRActivityPool`**

Property	**Value**
Connection Factory	`edx.tcf.log`
Listener Class	`com.edocs.fs.logging.sub.CSRActivityListener`
Acknowledge Mode	`auto`
Sessions Maximum	`-1`

JMS consumer- Configuration Tab

Property	**Value**
Name	`edxLoggerCSRActivityConsumer`
Messages Maximum	`10`
Selector	`JMSType='CSR'`
Destination	`edx.jms.log`

Message Log

JMS session pool- Configuration Tab

Property	**Value**
Name	`edxLoggerMessageLogPool`
Connection Factory	`edx.tcf.log`
Listener Class	`com.edocs.fs.logging.sub.MessageLogListener`
Acknowledge Mode	`auto`
Sessions Maximum	`-1`

JMS consumer- Configuration Tab

Property	**Value**
Name	`edxLoggerMessageLogConsumer`
Messages Maximum	`10`
Selector	`JMSType='MSG'`
Destination	`edx.jms.log`

System Activity

JMS session pool- Configuration Tab

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

JMS consumer

Property	Value
Name	`edxLoggerSystemActivityConsumer`
Messages Maximum	`10`
Selector	`JMSType='SYS'`
Destination	`edx.jms.log`

UserActivity

JMS session pool

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

JMS consumer

Property	Value
Name	`edxLoggerUserActivityConsumer`
Messages Maximum	`10`
Selector	`JMSType='USER'`

Property	Value
Destination	`edx.jms.log`

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. Which means, 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`**).
  - reportConnectionPool: 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 cCenter 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 Billing Analytics

Uninstalling Billing Analytics

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

Uninstall Billing Analytics from the **database server** first, then 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.
- Login as the user or administrator who owns 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.