



Installation Guide for Communications Billing Analytics

Sun Solaris Operating Environment Software
and the Oracle database

Version 5.1.0.1

Pub Date 2/27/2006

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 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 7

Operating System Administration Requirements 7

2 Getting Started

Preparing Your Platform 9

Overview of the Installation Process 9

Deploying the Billing Analytics J2EE Application 10

Communications Billing Analytics System Requirements 10

Platform Services and Communications Billing Analytics 10

3 Installing Communications Billing Analytics

UNIX Permissions for Installation 13

Installing Communications Billing Analytics 13

4 Configuring the Database Server

Overview 17

UNIX Permissions for Your Database Server 17

UNIX Environment Variables for Your Database Server 18

Oracle Database Server Environment Variables 18

Using Database Partitioning 19

5 Configuring Billing Analytics for Oracle

Configuring a New Oracle Database for UNIX 21

About `edx_main_admin.sh` for Oracle 21

Configuring Oracle Services 22

About `build.xml` for Oracle 29

Configuring `edxadmin.properties` 29

What to Do if Database Configuration Fails 31

Connecting to Your Oracle Database 32

6 Configuring the Application Server

Overview 33

About the Sample UNIX Domain Used in This Guide 33

UNIX Permissions for WebLogic Server 34

Application Server Environment Variables 35

Setting up the Environment to Support Charting 36

Configuring CBA for Weblogic 37

Using *edx_config* to capture environment Data 37

Add extra environment variables 38

Sourcing Your Configuration 39

Performance related JVM changes 39

Performance related *edx.config* changes 42

7 Configuring BEA Weblogic

Overview 43

Starting and Stopping WebLogic Server 43

Starting the CBA Batch scheduler 44

Configuring WebLogic Servers 45

Configuring Weblogic Services 46

JDBC Connection Pools 47

About JDBC Connections for Billing Analytics 47

Configuring JDBC Connections for Billing Analytics 47

JDBC Data Sources 50

JMS Connection Factories 51

JMS Stores 52

JMS Servers 53

JMS Topics 55

JMS Queues 55

- JMS Session Pools and Consumers for Logging for Command Center 56
- JMS Session Pools and Consumers for Logging for Billing Analytics 59
- JMS Foreign JMS Servers 62
- Configuring WebLogic JTA Timeout 63
- Deploying the Billing Analytics Application 64
 - Testing the Billing Analytics Installation 64
- Install X-Terminal 65
 - Display Permission 66
 - Display Device 66
 - Server Display Awareness 66
 - Setting up the server's charting environment 67

8 WebLogic Reference

- Java Database Connectivity (JDBC) 69
- WebLogic Environment Variables 69
- JDBC Resources for Billing Analytics 69
 - JDBC Connection Pools 69
 - JDBC Data Sources 72
 - JMS Connection Factories 73
 - JMS (JDBC) Stores 73
 - JMS Servers 74
 - Foreign JMS Server 74
 - JMS Topics 75
 - JMS Queues 75
 - JMS Session Pools and Consumers for Logging for Command Center 76
 - JMS Session Pools and Consumers for Logging for Billing Analytics 79
- JTA Timeout 82
- Extending XAResource Transaction Timeout 82
- Performance-Related Settings 83

9 Uninstalling Billing Analytics

Uninstalling Billing Analytics 85

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

Database Administration Requirements

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

- Back up and restore databases
- Monitor database performance

Application Server Administration Requirements

- Install and configure your application server
- Start and stop your application server
- Use administrative tools
- Manage users, privileges, and resources
- Configure Java resources
- Package and deploy web applications
- Monitor application server performance

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

Related Documentation

A PDF version of this guide is also available.

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

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

<i>Deploying and Customizing J2EE Applications</i>	How to customize J2EE Web applications for deployment with Billing Analytics.
<i>Billing Analytics Administration Guide</i>	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.

To configure your database server

- 4 Define database server environment variables.
- 5 Create and configure the Communications Billing Analytics database with `edx_main_admin.sh`
- 6 Connect to your Communications Billing Analytics database before configuring your application server.

To configure your application server

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

Deploying the Billing Analytics J2EE Application

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

- Deploy the J2EE web application for Communications Billing Analytics.

Communications Billing Analytics System Requirements

Platform Services and Communications Billing Analytics

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

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

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

Server O/S	Database	App Server	JRE
Solaris 9	Oracle 9.205	WebLogic 8.1 SP4	Sun JDK 1.4.2

OPERATING SYSTEM

- Sun Solaris 9

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 8)

SUPPORTED DATABASE SERVERS

New Installation of Communications Billing Analytics

- Oracle 9i Release 2 (Oracle 9.2.0)
- Oracle 9i client software (for application server), including the sqlplus and sqlldr packages
- Oracle 9i JDBC driver

Supported Application Servers

- BEA WebLogic Server 8.1 SP4

Supported Browsers

- Netscape Navigator 7 or higher
- Microsoft Internet Explorer 6.0 or later (on networked PC)

3

Installing Communications Billing Analytics

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

UNIX Permissions for Installation

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

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

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

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

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

Installing Communications Billing Analytics

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

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

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

```
/opt/<projectname>/apache
```

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

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

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

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

To install Billing Analytics

- 1 Log in as the relevant Unix user on the application server.
- 2 After you obtain and locate the Communications Billing Analytics software installer as described in the Preface of this guide, you can run the following from a command prompt at the directory location where the installer resides:

```
$ ./TAMins.bin -i console
```
- 3 On the Introduction screen, read the introductory information. Click **Next** to continue.
- 4 On the License Agreement screen, carefully read the licensing agreement, select the acceptance button, and then click **Next**.
- 5 On the Enter Serial Number screen, enter your product serial number. It is stapled to the inside front cover of this guide (if your serial number has been misplaced, contact Siebel Technical Support). Then click **Next**.
- 6 On the Owner of Web Application Server screen, enter the name of the application server owner (if you have installed other Siebel Self-Service products, use the same owner here that you used for those product installations). Then click **Next**.
- 7 On the Group of Web Application Server screen, enter the name of the group for the application server (if you have installed other Siebel Self-Service products, use the same group here that you used for those product installations). Then click **Next**.
- 8 On the Choose Install Folder screen, accept the default installation folder or click **Choose** and enter the directory where you want to install the Billing Analytics files and directories. This document refers to that directory as `CBA_HOME`. Click the **Next** button to continue.
- 9 On the Choose Product Features screen, click **CBA Group**. Then click **Next**.
- 10 On the Pre-Installation Summary screen, verify that the information is correct, and click on **Install**. To correct any entries, click **Previous**, and then return here.

At this point, the Billing Analytics database server components are copied to the designated installation folder. A status bar on the bottom of the screen shows each database server component being installed. No user intervention is required.
- 11 The release notes display inside the installer window.
- 12 The Install Complete screen reports a successful installation and the directory that contains the Billing Analytics components.
- 13 Click **Next** to view the release notes. Then click **Next** when you are done reading the release notes.
- 14 Click **Done** to exit the installer.

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

4 Configuring the Database Server

Overview

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

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:

- UNIX permissions for your database server
- Starting and stopping your database server
- UNIX environment variables for your database server
- Using database partitioning with Billing Analytics

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

UNIX Permissions for Your Database Server

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

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

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

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

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

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

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

UNIX Environment Variables for Your Database Server

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

Oracle Database Server Environment Variables

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

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

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

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

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

export CBA_HOME ORACLE_BASE ORACLE_HOME PATH LD_LIBRARY_PATH
export SHLIB_PATH ORACLE_DATA ORACLE_SID
```

Using Database Partitioning

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

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

5

Configuring Billing Analytics for Oracle

Configuring a New Oracle Database for UNIX

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

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

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

About `edx_main_admin.sh` for Oracle

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

Before running `edx_main_admin.sh`, you should:

- Upgrade your database server software **as necessary**.
- Make a full backup of your current database.
- Start the database instance that accesses the database you are upgrading.
- Check the status of all user objects. If any of them indicate an INVALID status, contact the database administrator to correct this problem.
- Confirm that all login sessions using the Billing Analytics database user have logged out of the instance.
- Have any required database passwords available. Check with your database administrator for custom passwords.
- Check the *Release Notes* for disk space requirements and confirm that you have sufficient disk space on your database server. Insufficient disk space can cause database configuration to fail.
- Configure `tnsnames.ora`, as described in the next section.

Included Files:

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

- `configure_ts.sql`: This file creates tablespaces specific for the application and is being invoked from the main shell script.
- `crt_rpt_user.sql`: The file creates a new schema and is being invoked from the main shell script.
- `init.ora`: Base initialization file used for setting up a new database

Configuring Oracle Services

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

TIP: Always consult with your onsite DBA and your 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`.

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

To configure Oracle services:

- 1 Switch user to the Oracle admin user.

```
$ su - oracle
```
- 2 Change directory to the network administration directory of your database server. For example:

```
cd /export/home/oracle/product/9.2.0/network/admin
```
- 3 Open `listener.ora` and edit the `SID_LISTENER` section to reflect your Oracle SID and database home directory. For example:

```
(SID_DESC =  
  (SID_NAME = o1tp)  
  (ORACLE_HOME = /export/home/oracle/product/9.2.0)  
)
```

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

4 Save and close `listener.ora`.

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

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

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

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

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

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

7 Save and close `tnsnames.ora`.

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

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

```
$ lsnrctl reload
```

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

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

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

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

- 1 Open the template the template init.ora files for the OLAP and OLTP instances:
 - a. \$CBA_HOME/db/oracle/oltp/init.ora
 - b. \$CBA_HOME/db/oracle/olap/init.ora
- 2 Edit the **db_block_size** parameter to be 8k or 16k (which should match your file system)


```
db_block_size = 8192
```

Configuring CBA OLTP and OLAP database instances for Oracle

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

```
$ su - oracle
```

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

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

- 3 Start the edx_main_admin shell script with the command:

```
$ ./edx_main_admin.sh
```

This will bring up the main menu page:

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

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

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

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

- 5 Select Option 1, **Sign in Menu**. A second sign-in screen appears.
- 6 Enter the username, password, and database name for the Billing Analytics database.

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

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

```
SIGN IN MENU
-----
[1] Enter Database USERNAME...> cba_dba
[2] Enter Database PASSWORD...> cba
[3] Enter ORACLE_SID ...> o1tp
[4] Enter the password for SYS user...> change_on_install
-----
Enter Your Selection:

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

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

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

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

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

```
/u01/oradata
```

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

```
/u01/oradata/o1tp
```

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

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

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

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

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

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

```

Install
-----
[1] Create Oracle Instance
[2] Shutdown Database
[3] Startup Database
[4] Install Application Database I
[5] Install Application Database II
[6] Install Application Database III
[7] View Status Log Directory
-----
[R] Return to previous menu
SELECT YOUR OPTION: 1
    
```

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

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

If this step is successful, the following message appears:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
Initial Data Population
-----

[1] Import initial data set
[2] Export Siebel database data
[3] Build Sample Hierarchy (Optional)
-----

[R] Return to previous menu
```

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

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

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

```
Siebel Reporting Server Administration Main Menu Version 1.0
-----

[1] Sign in Menu
[2] Capture Database File Locations
[3] Install Siebel Reporting
-----

Enter Your Selection:
```

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

```
SIGN IN MENU
-----

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

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

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

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

```

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

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

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

If this step is successful, the following message appears:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

About build.xml for Oracle

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

Before running ant, you should:

- Complete all the prerequisites specified previously in “*About edx_main_admin.sh for Oracle*” on page 21.
- Install Apache ant version 1.6.5 or higher. This software can be downloaded from <http://ant.apache.org/>. The installation directions can be found on that site as well.
- Configure *edxadmin.properties*, as described in the next section.

Configuring edxadmin.properties

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

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

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

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

```
ORACLE_HOME==/u01/oracle/product/9.2.0/ora92
ORACLE_ADMIN=/u01/oracle/product/9.2.0/admin
OLAP_USER=cba_dba
OLAP_PASSWD=cba
```

```

OLAP_SID=olap
OLAP_TNS_NAME= olap
DB_USER=cba_dba
DB_PASSWD=cba
DB_SID=oltp
SYS_PASSWD=change_on_install
META_USER=meta
META_PASSWD=meta
L_DB_EDX_DATA_TB_FILE_LOC=/u01/oradata
L_DB_APP_DATA_TB_FILE_LOC=/u01/oradata
L_DB_EDX_INDX_TB_FILE_LOC=/u01/oradata
L_DB_APP_INDX_TB_FILE_LOC=/u01/oradata
L_DB_LOAD_DATA_TB_FILE_LOC=/u01/oradata
L_DB_LOAD_INDX_TB_FILE_LOC=/u01/oradata
L_DB_FS_DATA_TB_FILE_LOC=/u01/oradata
L_DB_FS_INDX_TB_FILE_LOC=/u01/oradata
L_DB_STG_DATA_TB_FILE_LOC=/u01/oradata
L_DB_STG_INDX_TB_FILE_LOC=/u01/oradata
DB_CONTROL_FILE_LOCN1=/u01/oradata
DB_CONTROL_FILE_LOCN2=/u01/oradata
DB_CONTROL_FILE_LOCN3=/u01/oradata
REDO_LOG_FILE_LOCN=/u01/oradata
SYSTEM_FILE_LOCN=/u01/oradata
TRACE_FILE_LOCN=/u01/oradata
L_BACKUP_FILE=/u01/oradata/expoltp.dmp
L_LOG_FILE=/u01/oradata/expoltp.log
L_OLAP_BACKUP_FILE=/u01/oradata/expolap.dmp
L_OLAP_LOG_FILE=/u01/oradata/expolap.log
    
```

To configure a new Billing Analytics database for Oracle on UNIX

- 1 Switch user to the DB admin user. Oracle requires the administrative user in order to create files.
`$ su - oracle`
- 2 Change directory to your Billing Analytics database home directory. For example:
`cd /opt/siebel/TAM/db/oracle`
- 3 There are multiple ant targets you can use that will automate the installation process. The `install-new` target will create new OLTP and OLAP instances with the SIDs specified in the properties file:
`ant install-new`
- 4 The `install-existing` target will create new OLTP and OLAP schemas on an existing instance with the usernames/passwords specified in the properties file:
`ant install-existing`
- 5 If you wish to install sample data, add the `-DloadSampleData=true` argument to the ant call. For example:
`ant install-existing -DloadSampleData=true`


```
$ find /u06/oradata -name "*oltp*"
```

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

```
$ rm -rf <found_files>
```

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

Connecting to Your Oracle Database

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

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

```
$ su - oracle
```

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

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

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

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

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

```
SQL> show parameters db_name
```

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

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

- 5 Repeat for OLAP Database

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

6 Configuring the Application Server

Overview

This chapter assumes in-depth understanding of and practical experience with application server administration. 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:

- UNIX permissions for WebLogic Server
- Starting and Stopping WebLogic Server
- Capturing your Unix 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 UNIX Domain Used in This Guide

This guide uses the following example of a UNIX domain:

```
$BEA_HOME/domains/mydomain
```

WebLogic users can use the Domain Configuration Wizard to create the UNIX domain `$BEA_HOME/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 including `/user_projects/`

UNIX Permissions for WebLogic Server

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

This guide uses the example username and password **edxadmin:edxadmin** as the application server owner and group for Billing Analytics:

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

NOTE: The oracle user is also needed as the Oracle 9i client must be installed on the application tier.

For details on owner and group permissions for WebLogic Server, please consult the application server documentation for your platform.

Siebel recommends the following directory structure:

```

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

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

/opt/<projectname>/apache

```

To reset user and group permissions

- 1 Switch the user to the default owner of WebLogic Server home directory, for example **weblogic**.

```
$ su - weblogic
```

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

```
$ chown -R edxadmin:edxadmin /opt/bea
```

- 3 Switch the user to WebLogic Server owner and configure WebLogic Server with your new owner.

```
$ su - edxadmin
```

TIP: Verify the owner information in any profile files used by the database server owner and application server owner. See your server documentation for details.

Application Server Environment Variables

Siebel recommends the following additions to the **edxadmin** user's `.profile` file:

Variable	Description	Value
CBA_HOME	CBA home directory	/opt/Siebel/CBA
BEA_HOME	BEA Home directory	/opt/bea
ORACLE_BASE	Oracle Base	/u01/oracle
EDX_HOME	estatement home directory	\$CBA_HOME/estatement
WL_DOMAIN_DIR	BEA WebLogic Domain directory	\$BEA_HOME/domains/mydomain
JAVA_HOME	Java home directory	\$BEA_HOME/jdk142_08
TNS_ADMIN	TNS Admin directory	\$WL_DOMAIN_DIR
ORACLE_HOME	Oracle Home	\$ORACLE_BASE/9205
LD_LIBRARY_PATH	Shared Libraries	\$ORACLE_HOME/lib:/usr/lib:/usr/ucblib:/usr/ccs/lib
SHLIB_PATH	Shared Libraries	\$ORACLE_HOME/lib:\$ORACLE_HOME/lib64:\$ORACLE_HOME/lib32
PATH	PATH	./usr/local/bin:/usr/bin:/usr/ccs/bin:/usr/contrib/bin:\$ORACLE_HOME/bin:\$JAVA_HOME/bin:\$PATH

For example in the Korn shell:

```
CBA_HOME=/opt/Siebel/CBA
BEA_HOME=/opt/bea
ORACLE_BASE=/u01/oracle

EDX_HOME=$CBA_HOME/estatement
WL_DOMAIN_DIR=$BEA_HOME/domains/mydomain
JAVA_HOME=$BEA_HOME/jdk142_08
TNS_ADMIN=$WL_DOMAIN_DIR
ORACLE_HOME=$ORACLE_BASE/9205
LD_LIBRARY_PATH=$ORACLE_HOME/lib:/usr/lib:/usr/ucblib:/usr/ccs/lib
SHLIB_PATH=$ORACLE_HOME/lib:$ORACLE_HOME/lib64:$ORACLE_HOME/lib32
PATH=./usr/local/bin:/usr/bin:/usr/ccs/bin:/usr/contrib/bin:$ORACLE_HOME/bin:$JAVA_HOME/bin:$PATH

export CBA_HOME BEA_HOME ORACLE_BASE EDX_HOME WL_DOMAIN_DIR
export JAVA_HOME TNS_ADMIN ORACLE_HOME LD_LIBRARY_PATH SHLIB_PATH
export PATH
```

Setting up the Environment to Support Charting

Java's graphic packages that are part of Billing Analytics require X libraries and an X display. To support this, the web server rendering charts must have the X libraries installed and must have access to an X server. Even for sites where the server has an attached display device, there may be difficulty loading X libraries if they are not already present.

The X11 Libraries can be downloaded from the following site:

ftp://www.ferret.noaa.gov/special_request/xvfb/solaris/

Once downloaded, unzip with `gzip` and `untar` to create the directory: `/usr/X11R6`

Now please choose one of the following two options:

OPTION 1: Sun JDK 1.4 headless option

Instead of running `Xvfb` (see option 2 below), charting can also be enabled through utilization of the JDK 'headless' option:

Sun JDK 1.4 implementation provides a new Headless graphics operation mode. The headless feature is activated by setting the Sun JDK 1.4-specific "java.awt.headless" system property to "true", for example with an appropriate Java virtual machine command line option:

```
java -Djava.awt.headless=true
```

You can add this as a `JAVA_OPTION` in the WebLogic start scripts (eg. `startManagedWebLogic.sh`)

```
JAVA_OPTIONS=" \
-XX:+DisableExplicitGC \
-XX:+UseParallelGC \
-XX:+UseAdaptiveSizePolicy \
-XX:ParallelGCThreads=1 \
-Djava.awt.headless=true
"
```

This should be all that is needed, but unfortunately Sun JDK 1.4 versions up to at least 1.4.1.1 have certain problems that limit the usefulness of this feature. In particular, based on our own experiences, the presence of X11 runtime libraries is still needed. This means that an X11 environment must be installed into the system, but the X11 server itself need not to be started.

NOTE: No installation is required at this stage for the JDK headless option. This will be configured during the main application installation.

OPTION 2: Setting Up X Virtual Frame Buffer (Xvfb)

After installing the X11R6 libraries (under `/usr/X11R6`) set-up the following: -

As **root**, edit the `/etc/init.d` file

```
$ cd /etc/init.d
$ vi chart
```

Add the following to the file:

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

Save the file. Then run the following from the command prompt:

```
$ chmod 755 chart
$ cd /etc/rc2.d
$ ln -s /etc/init.d/chart S99chart
```

Make sure that in your WebLogic Server start scripts (eg. startManagedWebLogic.sh) you have the following line present:

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

Configuring CBA for Weblogic

Using `edx_config` to capture environment Data

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

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

CAUTION: Do not confuse `edx_config` (underscore) with `edx.config` (dot), which sources your configuration at server startup. For details on the latter, see “*Sourcing Your Configuration*” on page 39.

The *Quick Reference Appendix* can help you determine the values 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.

CAUTION: Be sure the time zone (TZ) for your server is set to your system time zone. Billing Analytics jobs can fail if the Java system time does not match actual system time.

To capture environment data with `edx_config` for WebLogic:

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

```
$ su - edocs
```

- 2 Navigate to the `bin` directory for CBA on your application server, for example

```
$ cd $EDX_HOME/bin
```

- 3 Run the script `edx_config`.

```
$ ./edx_config
```

- 4 Enter values as prompted by the script for your **OLTP** database home, database username and password, application server, Java home, and application server home. The WebLogic home directory pathname is where its application server files were installed, usually someplace like `/opt/BEA/weblogicXX` where XX is your WebLogic version number. See the *Quick Reference Appendix* for additional version and platform information.

CAUTION: Make sure that the database values you enter in this session are the same values specified during database configuration. Consult your DBA for any custom settings specific to your platform.

Add extra environment variables

- 1 Create a new file named: `$EDX_HOME/config/edx_cba.config`

- 2 Add the following to the file:

```
#####
# Supplement to edx.config for Core CBA

CBA_HOME=/opt/Siebel/CBA
EDX_HOME=$CBA_HOME/estatement

CLASSPATH=$CLASSPATH:$EDX_HOME/lib/edx_client.jar
CLASSPATH=$CLASSPATH:$EDX_HOME/lib/jakarta-oro-2.0.7.jar
CLASSPATH=$CLASSPATH:$CBA_HOME/lib/Configuration.jar
CLASSPATH=$CLASSPATH:$CBA_HOME/lib/commons-logging.jar
CLASSPATH=$CLASSPATH:$CBA_HOME/lib/javachart.jar
CLASSPATH=$CLASSPATH:$CBA_HOME/lib/ldprotocol.jar
CLASSPATH=$CLASSPATH:$CBA_HOME/lib/log4j.jar
CLASSPATH=$CLASSPATH:$CBA_HOME/config

JAVA_OPTIONS="$JAVA_OPTIONS -
Djava.protocol.handler.pkgs=com.edocs.protocol|com.edocs.domain.telco.lde.protocol"
JAVA_OPTIONS="$JAVA_OPTIONS -
Dcom.edocs.tasks.statementscanner.skipResolver=true"

export EDX_HOME CBA_HOME CLASSPATH JAVA_OPTIONS
```

- 3 Enter the correct values for the CBA_HOME parameter.
- 4 Save and close the file.
- 5 Make the file executable (so that it is detected by `edx.config`):


```
$ chmod 755 edx_cba.config
```
- 6 Open the core CBA file, `$EDX_HOME/config/edx.config`
- 7 Since the `edx_cba.config` file sets `JAVA_OPTIONS`, the protocol handler line towards the end of the file is unnecessary and should be commented out. For example:


```
#JAVA_OPTIONS="$JAVA_OPTIONS -Djava.protocol.handler.pkgs=com.edocs.protocol"
```
- 8 Save and close the file.

Sourcing Your Configuration

The `startWebLogic.sh` and `startManagedWebLogic.sh` scripts for your domain sets values for some domain-specific variables and then calls the master startup script, `startWLS.sh`. The master startup script sets environment variables, such as the location of the JVM, and then starts the JVM with WebLogic Server arguments.

Siebel recommends that WebLogic users source your Platform Services and Billing Analytics configuration directly in `startWebLogic.sh` for your domain, for consistency with this feature of WebLogic.

To source `setEnv.sh` for Billing Analytics and Platform Services:

- 1 Switch user to your application server owner, for example `edxadmin`.


```
$ su - edxadmin
```
- 2 Stop WebLogic Server and all application server instances. For details, see “Starting and Stopping WebLogic Server” on page 43.
- 3 Navigate to the application server startup script **for your domain** and open the file for editing. For example:


```
$ cd $BEA_HOME/domains/mydomain
$ vi startWebLogic.sh
```
- 4 Define and export the environment variable for your Platform Services home directory, then source the Platform Services configuration. For example, insert the following text into `startWebLogic.sh`:


```
EDX_HOME=/opt/Siebel/CBA/estatement
export EDX_HOME
. $EDX_HOME/config/edx.config

${JAVA_HOME}/bin/java ${JAVA_VM} ${MEM_ARGS} ${JAVA_OPTIONS} .....
```
- 5 Save and close `startWebLogic.sh`.
- 6 Repeat the same edits to `startManagedWebLogic.sh`.

Performance related JVM changes

Make the following changes to optimize the Java Virtual Machine for running Communications Billing Analytics.

To optimize performance for Billing Analytics and Platform Services:

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


```
$ su - edxadmin
```
- 2 Stop the WebLogic Server and all application server instances.

- 3** Navigate to the application server startup script **for your domain** and open the file for editing. For example:

```
$ cd $BEA_HOME/domains/mydomain
$ vi startWebLogic.sh
```

- 4** Insert JVM parameters into the file. For example:

```
# Set JAVA_VM to the java virtual machine you want to run.
JAVA_VM="-server"

# Set MEM_ARGS to the memory arguments you want to pass to java.
MEM_ARGS="\
-Xms512m \
-Xmx1024m \
-XX:MaxPermSize=128m \
-XX:NewSize=128m \
-XX:MaxNewSize=128m \
-Xconcurrentio \
"

# Set JAVA_OPTIONS to those you want to pass to java.
JAVA_OPTIONS="\
-XX:+DisableExplicitGC \
-XX:+UseParallelGC \
-XX:+UseAdaptiveSizePolicy \
-XX:ParallelGCThreads=1 \
-Djava.awt.headless=true \
"

CLASSPATH="${WEBLOGIC_CLASSPATH}:${POINTBASE_CLASSPATH}:${JAVA_HOME}/jre/lib/rt
.jar:${WL_HOME}/server/lib/webservices.jar:${CLASSPATH}"
export CLASSPATH
```

NOTE: These MEM_ARGS should be tailored to the amount of memory and the number of CPUs you have on your system.

- 5** Save and close **startWebLogic.sh**.
- 6** Repeat the same edits to **startManagedWebLogic.sh**

NOTE: The startManagedWebLogic.sh script is typically used for the batch and online applications and as such requires more memory allocated to its JVM. The startWebLogic.sh script is typically only used to run the BEA Admin Console, so its memory requirements are minimal (e.g. -Xms128m -Xmx256m)

What these options do:

JAVA_VM="-server" - Produces more efficient run-time execution over extended periods

-XX:+UseAdaptiveSizePolicy and -XX:+UseParallelGC - Considers three goals:

- A desired maximum GC pause goal
- A desired application throughput goal
- Minimum footprint

The implementation checks (in this order):

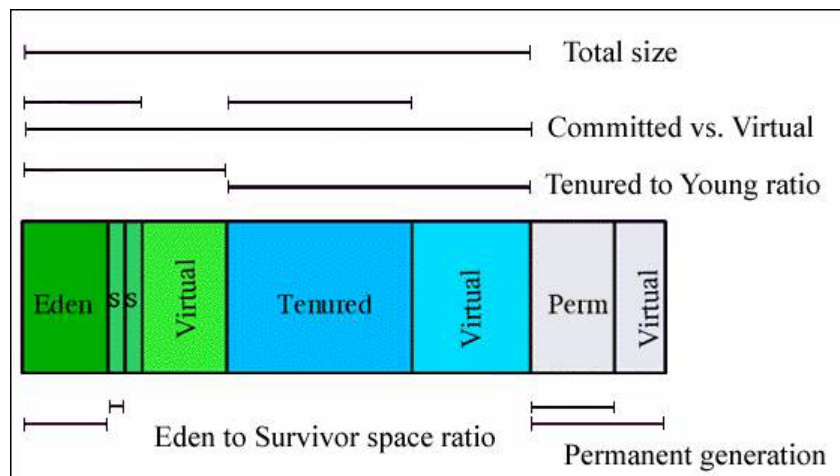
- If the GC pause time is greater than the pause time goal then reduce the generation's sizes to better attain the goal.
- If the pause time goal is being met then consider the application's throughput goal. If the application's throughput goal is not being met, then increase the sizes of the generations to better attain the goal.

If both the pause time goal and the throughput goal are being met, then the sizes of the generations are decreased to reduce pauses.

-XX:+DisableExplicitGC - Another way apps can interact with garbage collection is by invoking GCs explicitly, such as through the System.gc() call. These calls force major collection and inhibit scalability on large systems. This flag prevents that.

-XX:ParallelGCThreads=1 - Forces the use of all processors on the server for Parallel Garbage Collection. This number is set equal to the number of CPUs on the server.

Below is some background on JVM tuning: - <http://java.sun.com/docs/hotspot/gc/index.html>



Performance related edx.config changes

For Oracle 9i, the following change is more efficient. Change the line in **\$EDX_HOME/config/edx.config** from the following:

```
CLASSPATH=$ORACLE_HOME/jdbc/lib/classes12.zip:$CLASSPATH
```

To:

```
CLASSPATH=$BEA_HOME/weblogic81/server/ext/jdbc/oracle/920/ojdbc14.jar:$CLASSPATH
```

7 Configuring BEA Weblogic

Overview

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

See WebLogic Server documentation at <http://edocs.bea.com> for detailed step-by-step instructions on Java resource configuration, performance, and tuning. You must also consult your application server administrator for settings that may be specific to your configuration.

You must start your 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.

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.

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, which schedules and runs 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.

- 1 For example, if you created the following sample environment file in the WebLogic domain directory:

```
$BEA_HOME/domains/mydomain/wl_env
```

- 2 And entered these contents:

```
BEA_HOME=/opt/bea
WL_DOMAIN_DIR=/opt/bea/domains/mydomain
WL_DOMAIN=mydomain
CBA_HOME=/opt/Siebel/CBA
EDX_HOME=/opt/Siebel/CBA/estatement

WL_ADMIN_SERVER=ADMIN
WL_ADMIN_MACHINE=localhost
WL_ADMIN_PORT=7001
```

```

WL_SERVER=batch1
WL_LISTEN_ADDRESS=localhost
WL_PORT=8001

WLS_USER=weblogic
WLS_PW=edocsdocs

TZ=US/Eastern

export WL_DOMAIN_DIR EDX_HOME CBA_HOME WL_ADMIN_MACHINE BEA_HOME
export WL_DOMAIN WL_ADMIN_SERVER WL_ADMIN_PORT WL_SERVER WL_PORT
export WL_LISTEN_ADDRESS WLS_USER WLS_PW TZ

```

NOTE: The Timezone Parameter (TZ) can be found in the Solaris file `/etc/TIMEZONE`

- 3** After sourcing the file in your environment, you do the following to stop the BEA Admin Server:

```

$ . ./wl_env
$ . $WL_DOMAIN_DIR/setEnv.sh > /dev/null
$ java weblogic.Admin -url $WL_ADMIN_MACHINE:$WL_ADMIN_PORT SHUTDOWN -username
  $WLS_USER -password $WLS_PW

```

- 4** To stop a managed server:

```

$ . ./wl_env
$ . $WL_DOMAIN_DIR/setEnv.sh > /dev/null
$ java weblogic.Admin -url $WL_LISTEN_ADDRESS:$WL_PORT SHUTDOWN -username
  $WLS_USER -password $WLS_PW

```

To start WebLogic in an active Billing Analytics production environment, Siebel recommends that you use the **nohup** command to ignore hang-ups. This leaves the server running in the background even if you end your terminal session or try to force a hard shutdown, providing a more stable production environment.

- 5** To start the BEA Admin server:

```

$ . ./wl_env
$ . $EDX_HOME/config/edx_env
$ /usr/xpg4/bin/nohup $WL_DOMAIN_DIR/startWebLogic.sh >>
  $WL_DOMAIN_DIR/logs/$WL_ADMIN_SERVER.log &

```

- 6** To start a managed server:

```

$ . ./wl_env
$ . $EDX_HOME/config/edx_env
$ /usr/bin/nohup $WL_DOMAIN_DIR/startManagedWebLogic.sh $WL_SERVER
  http://$WL_ADMIN_MACHINE:$WL_ADMIN_PORT >> $WL_DOMAIN_DIR/logs/$WL_SERVER.log &

```

Starting the CBA Batch scheduler

You must start the Platform Services Scheduler in order to schedule and run jobs in the Command Center. If you attempt to run a new job while the Scheduler is not running, the job will not be run and you will see 'Not yet started' as its status.

The Platform Services scheduler does not need to be started until after you have deployed the Command Center EAR file.

To start the Platform Services Scheduler from the UNIX command line:

- 1 Switch user to the application server owner.

```
$ su - edxadmin
```
- 2 The scheduler script can be found under the **bin** directory of your CBA installation, **\$EDX_HOME/bin**.
- 3 Run the Scheduler command for your application server, host, and port. For example:

```
$ $EDX_HOME/bin/wl_scheduler -start -url t3://$WL_LISTEN_ADDRESS:$WL_PORT
```

NOTE: WL_LISTEN_ADDRESS and WL_PORT should match the WebLogic Server instance running the batch application.
- 4 You can stop the Scheduler by replacing the **-start** parameter with the **-stop** parameter.

```
$ $EDX_HOME/bin/wl_scheduler -stop -url t3://$WL_LISTEN_MACHINE:$WL_PORT
```

Configuring WebLogic Servers

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

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

The screenshot shows the 'mydomain > Servers' page. It includes a navigation tree on the left with 'Servers' expanded to show 'ADMIN', 'batch1', and 'online1'. The main content area shows a table of servers:

Name	Listen Address	Listen Port	Cluster	Machine	State	
ADMIN	n/a	7001	n/a	n/a	RUNNING	
batch1	natv280s008	8001	n/a	n/a	RUNNING	
online1	natv280s008	9001	n/a	n/a	RUNNING	

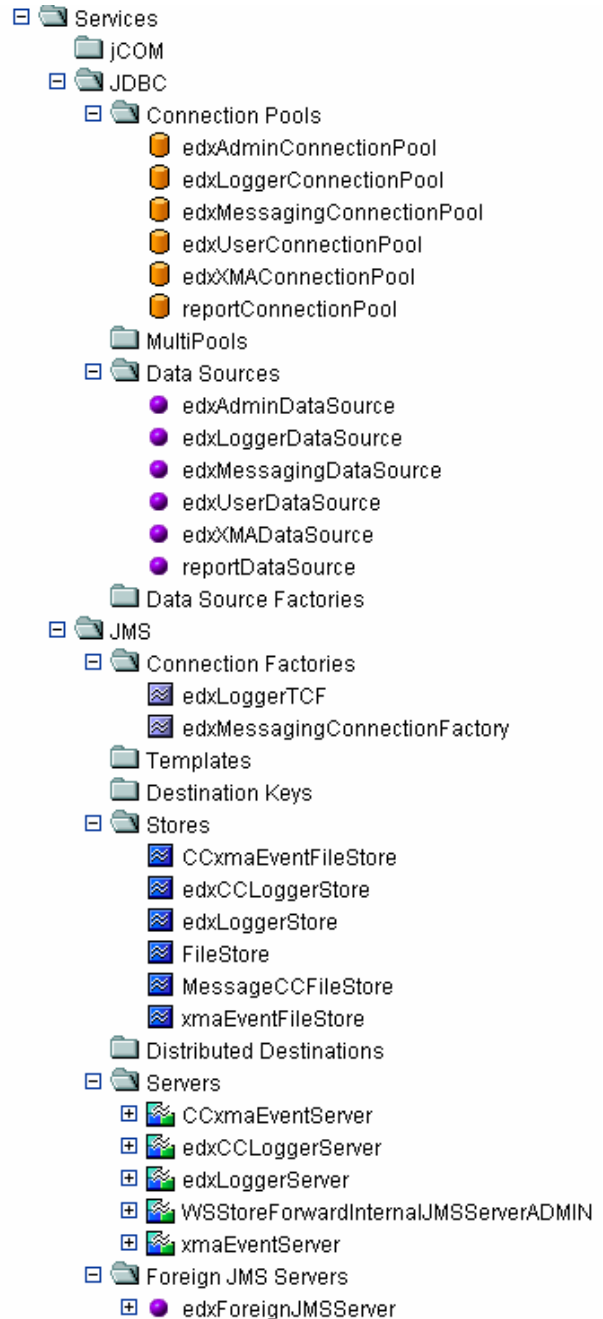
NOTE: In the above screenshot, natv280s008 is the name of the server on which the admin server lives.

Configuring Weblogic Services

This section outlines the Weblogic Services configurations that are required for CBA. These include:

- JDBC Connection Pools
- JDBC Data Sources
- JMS Connection Factories
- JMS Stores
- JMS Servers
- JMS Foreign JMS Servers

After completion of the configuration, your WebLogic Services should look something like the illustration here:



JDBC Connection Pools

JDBC Connections on the application server support data retrieval from relational databases and other data sources.

The screenshot displays the 'mydomain > JDBC Connection Pools' configuration page. It includes a navigation tree on the left and a main content area with a table of connection pools.

Name	URL	Driver Classname
edxAdminConnectionPool	jdbc:oracle:thin:@natv280s010:1521:oltp	oracle.jdbc.OracleDriver
edxLoggerConnectionPool	jdbc:oracle:thin:@natv280s010:1521:oltp	oracle.jdbc.OracleDriver
edxMessagingConnectionPool	jdbc:bea:oracle:/natv280s010:1521	weblogic.jdbcx.oracle.OracleDataSource
edxUserConnectionPool	jdbc:oracle:thin:@natv280s010:1521:oltp	oracle.jdbc.OracleDriver
edxXMACConnectionPool	jdbc:bea:oracle:/natv280s010:1521	weblogic.jdbcx.oracle.OracleDataSource
reportConnectionPool	jdbc:bea:oracle:/natv280s010:1521	weblogic.jdbcx.oracle.OracleDataSource

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.

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.

TIP: You can clone JDBC resources to save time.

NOTE: In the below tables, "localhost" must be changed to the IP of your database server if it's a different machine than your weblogic server.

You will create six connection pools, using the following settings:

TAB: Configuration > General			
Name	edxAdminConnectionPool	edxLoggerConnectionPool	edxUserConnectionPool
Database	OLTP	OLTP	OLTP
Database Driver	*Oracle's driver (Thin) Versions:8.1.7,9.0.1,9.2.0	*Oracle's driver (Thin) Versions:8.1.7,9.0.1,9.2.0	*Oracle's driver (Thin) Versions:8.1.7,9.0.1,9.2.0
URL	jdbc:oracle:thin:@localhost:1521:oltp	jdbc:oracle:thin:@localhost:1521:oltp	jdbc:oracle:thin:@localhost:1521:oltp
Driver Classname	oracle.jdbc.OracleDriver	oracle.jdbc.OracleDriver	oracle.jdbc.OracleDriver
Properties	user=cba_dba	user=cba_dba	user=cba_dba
Password	cba	cba	cba

TAB: Configuration > Connections			
Initial Capacity	1	10	10
Maximum Capacity	20	25	25
Capacity Increment	5	5	5
Statement Cache Size	300	50	150
Login Delay	1	1	1
Test Frequency	60	60	60
Test Table Name	SQL SELECT 1 FROM DUAL WHERE ROWNUM < 1	SQL SELECT 1 FROM DUAL WHERE ROWNUM < 1	SQL SELECT 1 FROM DUAL WHERE ROWNUM < 1
Allow Shrinking	True (box checked)	True (box checked)	True (box checked)
Shrink Frequency	15	15	15
Test Reserved Connections	True (checked)	True (checked)	True (checked)
Test Released Connections	False (unchecked)	False (unchecked)	False (unchecked)
Enable XA Transaction Timeout			
XA Transaction Timeout			
Supports Local Transaction			

TAB: Targets and Deploy			
Targets	Batch1, Online1	Batch1, Online1	Batch1, Online1

Continued on the next page...

TAB: Configuration > General			
Name	edxMessagingConnectionPool	edxXMAConnectionPool	reportConnectionPool
Database	OLTP	OLTP	OLAP
Database Driver	BEA's Oracle Driver (Type 4XA)	BEA's Oracle Driver (Type 4XA)	BEA's Oracle Driver (Type 4XA)
URL	jdbc:bea:oracle://localhost:1521	jdbc:bea:oracle://localhost:1521	jdbc:bea:oracle://localhost:1521
Driver Classname	weblogic.jdbcx.oracle.OracleDataSource	weblogic.jdbcx.oracle.OracleDataSource	weblogic.jdbcx.oracle.OracleDataSource
Properties	user=cba_dba portNumber=1521 SID=oltp serverName=localhost	user=cba_dba portNumber=1521 SID=oltp serverName=localhost PinnedToThread=true	user=cba_dba portNumber=1521 SID=olap serverName=localhost PinnedToThread=true
Password	cba	cba	cba

TAB: Configuration > Connections			
Initial Capacity	10	40	40
Maximum Capacity	25	40	40
Capacity Increment	5	1	1
Statement Cache Size	200	300	300
Login Delay	1	1	1
Test Frequency	60	60	60
Test Table Name	SQL SELECT 1 FROM DUAL WHERE ROWNUM < 1	SQL SELECT 1 FROM DUAL WHERE ROWNUM < 1	SQL SELECT 1 FROM DUAL WHERE ROWNUM < 1
Allow Shrinking	True (box checked)	True (box checked)	True (box checked)
Shrink Frequency	15	15	15
Test Reserved Connections	True (checked)	True (checked)	True (checked)
Test Released Connections	False (unchecked)	False (unchecked)	False (unchecked)
Enable XA Transaction Timeout	YES	YES	YES
XA Transaction Timeout	6000	6000	6000
Supports Local Transaction	YES	YES	YES

TAB: Targets and Deploy			
Targets	Batch1, Online1	Batch1, Online1	Batch1, Online1

JDBC Data Sources

Create the following 6 transaction data sources. Set the properties for all data sources as below:

Name	JNDIName	JNDIName Separator	Pool Name	Row Prefetch Enabled	Enable Two Phase Commit	Stream Chunk Size	Row Prefetch Size
edxAdminDataSource	edx.databasePool	;	edxAdminConnectionPool	false	false	256	48
edxLoggerDataSource	edx.logger.databasePool	;	edxLoggerConnectionPool	false	false	256	48
edxMessagingDataSource	edx.messaging.databasePool	;	edxMessagingConnectionPool	false	false	256	48
edxUserDataSource	edx.user.databasePool	;	edxUserConnectionPool	false	false	256	48
edxXMADDataSource	edx.xma.databasePool	;	edxXMADConnectionPool	false	false	256	48
reportDataSource	edx.report.databasePool	;	reportConnectionPool	false	false	256	48

TIP: After creating the first data source, and you can clone it for use in creating the others.

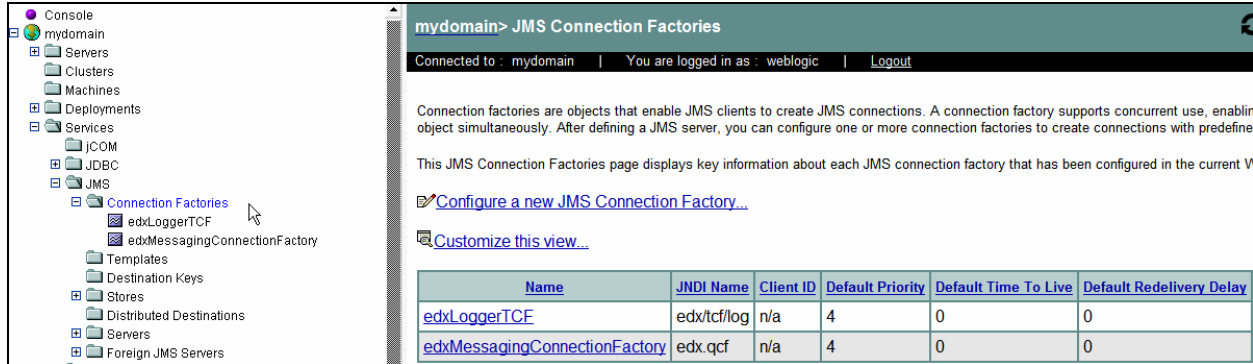
TAB: Configuration			
Name	edxAdminDataSource	edxLoggerDataSource	edxMessagingDataSource
JNDIName	edx.databasePool	edx.logger.databasePool	edx.messaging.databasePool
JNDIName Separator	;	;	;
Pool Name	edxAdminConnectionPool	edxLoggerConnectionPool	edxMessagingConnectionPool
Enable Two Phase Commit	FALSE	FALSE	FALSE
Row Prefetch Enabled	FALSE	FALSE	FALSE
Row Prefetch Size	48	48	48
Stream Chunk Size	256	256	256
TAB: Targets and Deploy			
Targets	Batch1, Online1	Batch1, Online1	Batch1, Online1

The first three pools are configured above. The second three are configured below:

TAB: Configuration			
Name	edxUserDataSource	edxXMADDataSource	reportDataSource
JNDIName	edx.user.databasePool	edx.xma.databasePool	edx.report.databasePool
JNDIName Separator	;	;	;
Pool Name	edxUserConnectionPool	edxXMADConnectionPool	reportConnectionPool
Enable Two Phase Commit	FALSE	FALSE	FALSE
Row Prefetch Enabled	FALSE	FALSE	FALSE
Row Prefetch Size	48	48	48
Stream Chunk Size	256	256	256
TAB: Targets and Deploy			
Targets	Batch1, Online1	Batch1, Online1	Batch1, Online1

JMS Connection Factories

Create the following JMS connection factories. You may accept the default **Properties** for all three connection factories, or consult your application server administrator to tune these values.



TAB: Configuration > General		
Name	edxLoggerTCF	edxMessagingConnectionFactory
JNDI Name	edx/tcf/log	edx.qcf
Acknowledge Policy	All	All
Allow Close In On Message	FALSE	FALSE
Client ID	n/a	n/a
Default Delivery Mode	Persistent	Persistent
Default Priority	4	4
Default Redelivery Delay	0	0
Default Time To Deliver	0	0
Default Time To Live	0	0
Messages Maximum	10	10
Overrun Policy	Keep Old	Keep Old

TAB: Configuration > Transactions		
Transaction Timeout	3600	3600
XA Connection Factory Enabled	FALSE	TRUE

TAB: Targets and Deploy		
Targets	Batch1, Online1	Batch1, Online1

JMS Stores

Create four JMS Stores. You may accept the default **Prefix Name=<NULL>** for all three stores, or consult your application server administrator to tune these values.

The screenshot shows the BEA Weblogic console interface. On the left, a navigation tree is visible with 'Stores' expanded, showing five entries: CCxmaEventFileStore, edxCCLoggerStore, edxLoggerStore, FileStore, and xmaEventFileStore. On the right, the 'mydomain > JMS Stores' page is displayed. It includes a header 'Connected to : mydomain | You are logged in as : we', a descriptive paragraph about JMS stores, and three links: 'Configure a new JMS JDBC Store...', 'Configure a new JMS File Store...', and 'Customize this view...'. Below these links is a table with the following data:

Name	Type	
CCxmaEventFileStore	JMSFileStore	
edxCCLoggerStore	JMSJDBCStore	
edxLoggerStore	JMSJDBCStore	
FileStore	JMSFileStore	
xmaEventFileStore	JMSFileStore	

Name	Type	Connection Pool	Directory
CCxmaEventFileStore	JMSFileStore		xmafilestore
edxCCLoggerStore	JMSJDBCStore	edxLoggerConnectionPool	
edxLoggerStore	JMSJDBCStore	edxLoggerConnectionPool	
xmaEventFileStore	JMSFileStore		xmafilestore

NOTE: You need to pre-create the directory \$WL_DOMAIN_DIR/xmafilestore first.

JMS Servers

Create three JMS Servers. You may accept the default **Properties** for all three servers, or consult your application server administrator to tune these values.

mydomain > JMS Servers

Connected to : mydomain | You are logged in as : weblogic | [Logout](#)

A JMS server manages connections and message requests on behalf of JMS clients.

This JMS Servers page displays key information about each JMS server that has been configured in the current WebLogic Server domain.

[Configure a new JMS Server...](#)

[Customize this view...](#)

Name	Persistent Store	Temporary Template	Bytes Maximum	Messages Maximum
CCxmaEventServer	CCxmaEventFileStore	n/a	-1	-1
edxCCLoggerServer	edxCCLoggerStore	n/a	-1	-1
edxLoggerServer	edxLoggerStore	n/a	-1	-1
WSStoreForwardInternalJMSServerADMIN	FileStore	n/a	-1	-1
xmaEventServer	xmaEventFileStore	n/a	-1	-1

TAB: Configuration > General

Name	CCxmaEventServer	edxCCLoggerServer	edxLoggerServer	xmaEventServer
Persistent Store	CCxmaEventFileStore	edxCCLoggerStore	edxLoggerStore	xmaEventFileStore

TAB: Configuration > Thresholds & Quotas

Bytes Maximum	-1	-1	-1	-1
Bytes Threshold High	-1	-1	-1	-1
Bytes Threshold Low	-1	-1	-1	-1
Maximum Message Size	2147483647	2147483647	2147483647	2147483647
Messages Maximum	-1	-1	-1	-1
Messages Threshold High	-1	-1	-1	-1
Messages Threshold Low	-1	-1	-1	-1

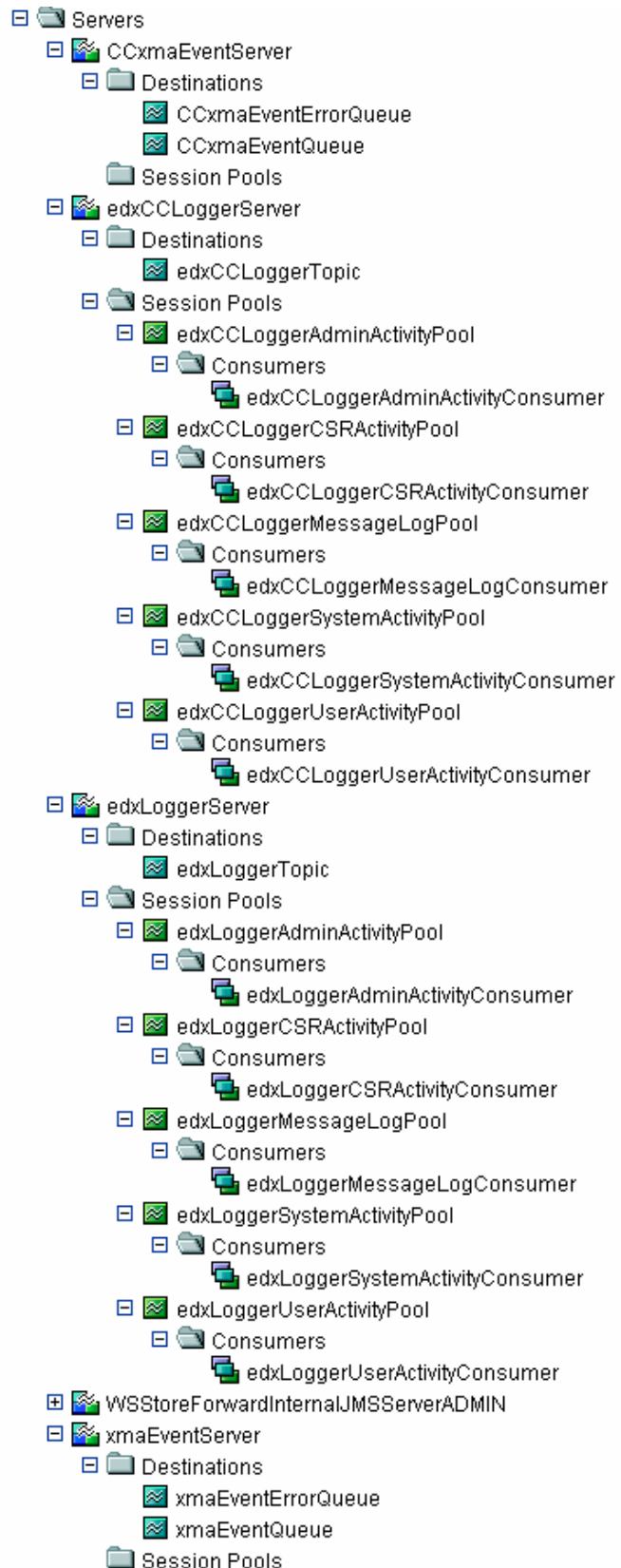
TAB: Targets and Deploy

Targets	Batch1	Batch1	Online1	Online1

Next you must configure JMS Server Destinations and JMS Server Session Pools as detailed below.

After completing the addition of the Destinations and Session Pools your WebLogic Configuration tree for JMS Servers should look like the following:

The next few sections describe the creation of this tree in detail.



JMS Topics

Create two JMS Topics. Select **Destinations** under each defined Server, then click on **Configure a new JMSTopic**. Make sure to create the matching topic for each server.

Name	<code>edxCCLoggerTopic</code>	<code>edxLoggerTopic</code>
JNDI Name	<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**:

Name	<code>xmaEventQueue</code>
JNDI Name	<code>edx.queue.outbound</code>

Repeat the same sequence to create another Destination/JMSQueue:

Name	<code>xmaEventErrorQueue</code>
JNDI Name	<code>edx.queue.errors</code>

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

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

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

Expiration Policy	<code>Redirect</code>
--------------------------	-----------------------

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

Name	<code>CCxmaEventQueue</code>
JNDI Name	<code>edx.queue.outbound</code>

Repeat the same sequence to create another Destination/JMSQueue:

Name	<code>CCxmaEventErrorQueue</code>
-------------	-----------------------------------

Name	<code>CCxmaEventErrorQueue</code>
JNDI Name	<code>edx.queue.errors</code>

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

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

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

Expiration Policy	<code>Redirect</code>
--------------------------	-----------------------

JMS Session Pools and Consumers for Logging for Command Center

Create FIVE pairs of **JMS Session Pools and Consumers** for the `edxCCLoggerServer`. 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>

Property	Value
Selector	JMSType= 'ADM'
Destination	edx.jms.log

CSR Activity

JMS session pool- Configuration Tab

Property	Value
Name	edxCCLoggerCSRActivityPool
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	<code>edxCCLoggerMessageLogConsumer</code>
Messages Maximum	10
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	auto
Sessions Maximum	-1

JMS consumer

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

UserActivity

JMS session pool

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

JMS consumer

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

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>
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	<code>10</code>
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	<code>-1</code>

JMS consumer

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

JMS Foreign JMS Servers

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

Name	JNDI Connection URL	JNDI Initial Context Factory	Targets
edxForeignJMSServer	t3://natv280s008:8001	weblogic.jndi.WLInitialContextFactory	Batch1

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 JMSConnectionFactory and enter the following values:

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

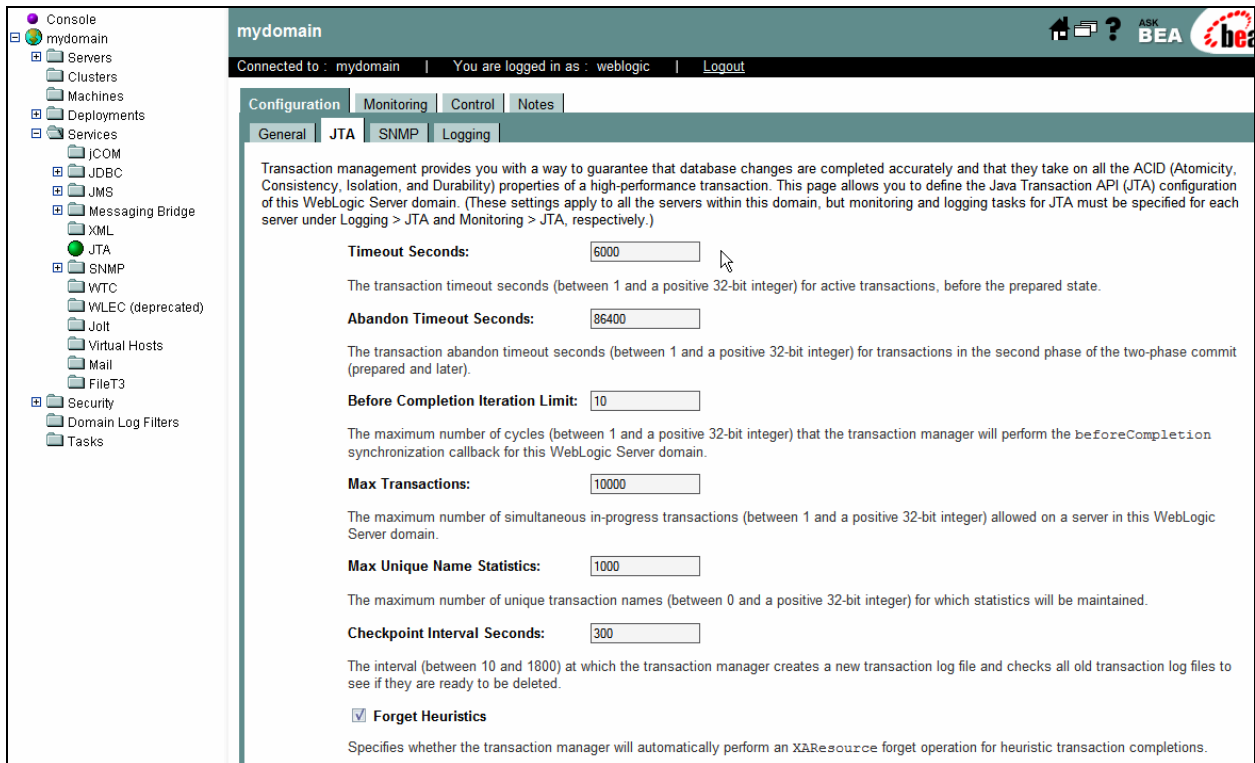
To create the JMS Destination:

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

Configuring WebLogic 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.



Deploying the Billing Analytics Application

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

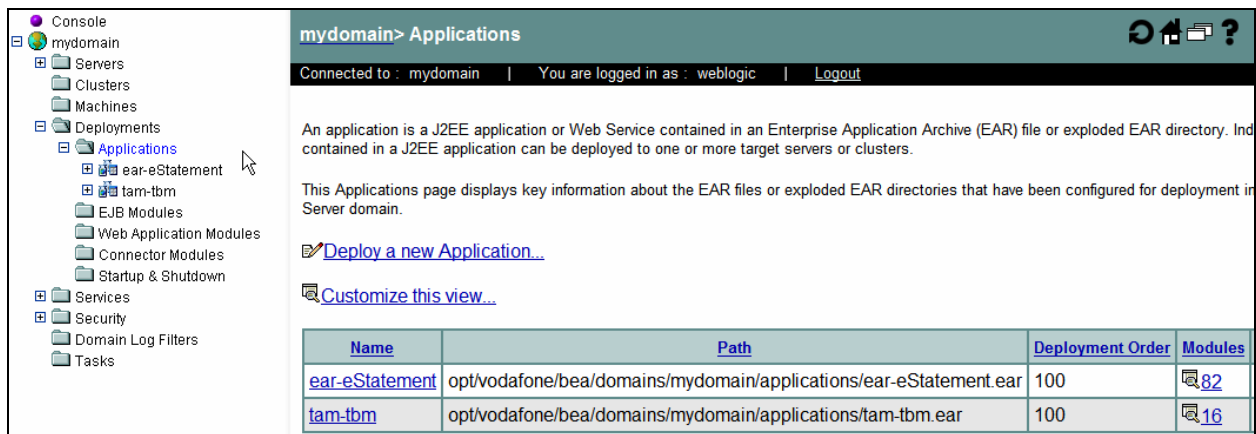
- **Application servers:** Deploy the Billing analytics application EAR files.

The EAR files are located at:

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

It is recommended that you copy these ear files to the WebLogic domain 'applications' directory and deploy from there. For example:

```
$ cp $CBA_HOME/J2EEApps/tam-tbm/tam-tbm.ear $WL_DOMAIN_DIR/applications
```



Target the tam-tbm.ear to the online1 WebLogic server and the ear-eStatement.ear to the batch1 WebLogic server.

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 to check the application (note: in each case, the username and password are identical).

Company	First Name	Last Name	Login/Password
American HighTech1	John	Smith	JSMITH
American HighTech1	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

To test Platform Services

NOTE: Make sure you have started the scheduler as described in “Starting the CBA Batch scheduler”, on page 44, before running any Command Center jobs.

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 User Name and Password: admin/edocs.
- 3 In the Main Console, select Create New Application.
- 4 Enter a name for the new application , for example, **testApp**.
- 5 Use **/edx/ejb/EdocsDataSource** for the Datasource Name
- 6 Choose the default for Index Partition Count.
- 7 Click on the **Create Application** button.

Install X-Terminal

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

Display Permission

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

```
xhost +
```

Display Device

Java's graphic packages are based on `java.awt`, which requires X libraries and an X display. That means that the Web server that is rendering the charts -- the web tier -- must have access to a real or virtual X display device. Therefore, the necessary X Libraries must be loaded. Note: it is not necessary for the Web server's system to have a physical X Display device.

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

If you do not already have Xvfb installed on your machine, you can obtain a tar'd version at ftp://www.ferret.noaa.gov/special_request/xvfb/solaris/, along with further usage information. (You may need to use `gunzip` and `tar xvf` to unzip and extract the contents of the archive, if you don't have a specialized utility.) Untar the file to have it installed under `/usr/X11R6`. The command to start the virtual frame buffer, Xvfb, will then be located in `/usr/X11R6/bin`. Execute the following command at the command prompt:

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

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

Server Display Awareness

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

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

Setting up the server's charting environment

To set up the server's charting environment manually

- 1 # gzip -d X11R6.tar.gz (if gzip is installed)
tar xvf X11R6.tar
- 2 Start the X Virtual Frame Buffer, as shown below. This will create a virtual display at :2.0 with a size of 800 by 600 pixels and a color depth of 24 bits. To ensure that Java draws to this display, you must set the DISPLAY environment variable to :2.0 before invoking Java. If Java throws any X11 Environment exceptions, either try changing the color depth or screen size, or change `-screen 0` to `-screen 1`.

/usr/X11R6/bin/Xvfb :2.0 -screen 0 800x600x24 &
ps -ef | grep Xvfb (to see if it's running)
- 3 Change to the WebLogic owner user profile
su - weblogic
- 4 Set the DISPLAY parameter in the weblogic .profile by adding the following line
\$ DISPLAY=:2.0; export DISPLAY
- 5 Allow open access to the machine's X Display
\$ xhost +
- 6 In the file *startWebLogic.sh* add the following line at the top
DISPLAY=:2.0; export DISPLAY
- 7 Restart the WebLogic server instance

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

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

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

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

- 1 # cd /etc/init.d
vi chart
- 2 Add the following to the file:f

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

- 3 Update file permissions:

```
# chmod 777 chart
```

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

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

- 5 Also double check that both the WebLogic owner .profile and startWebLogic.sh have the following line:

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

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 on how to configure JDBC connections, see your application server documentation. For the procedure to create connections for Billing Analytics, see "About JDBC Connections for Billing Analytics" on page 69.

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	Example
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/jdk141_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 batch 1	
edxLoggerConnectionPool	OLTP	online1 and batch 1	
edxUserConnectionPool	OLTP	online1 and batch 1	

Connection Pool	Database	Targeted Server	Supports Local Transactions
edxXMAConnectionPool	OLTP	online1 and batch 1	YES
edxMessagingConnection Pool	OLTP	online1 and batch 1	YES
reportConnectionPool	OLAP	online1 and batch 1	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

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	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
<code>edxCCLoggerServer</code>	<code>edxCCLoggerStore</code>	<code>batch1</code>
<code>edxLoggerServer</code>	<code>edxLoggerStore</code>	<code>online1</code>
<code>CCxmaEventServer</code>	<code>CCxmaEventFileStore</code>	<code>batch1</code>
<code>xmaEventServer</code>	<code>xmaEventFileStore</code>	<code>online1</code>

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

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 JMSConnectionFactory and enter the following values:

Name	<code>edxForeignJMSConnectionFactory</code>
Local JNDI Name	<code>edx.foreign.qcf</code>
Remote JNDI Name	<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:

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

Property	Value
Sessions Maximum	-1

JMS Consumer- Configuration Tab

Property	Value
Name	<code>edxCCLoggerAdminActivityConsumer</code>
Messages Maximum	10
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	auto
Sessions Maximum	-1

JMS consumer- Configuration Tab

Property	Value
Name	<code>edxCCLoggerCSRActivityConsumer</code>
Messages Maximum	10
Selector	<code>JMSType= 'CSR'</code>
Destination	<code>edx.jms.log</code>

Message Log

JMS session pool- Configuration Tab

Property	Value
Name	<code>edxCCLoggerMessageLogPool</code>

Property	Value
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	<code>edxCCLoggerUserActivityPool</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	<code>-1</code>

JMS consumer

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

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>

Property	Value
Sessions Maximum	-1

JMS Consumer- Configuration Tab

Property	Value
Name	<code>edxLoggerAdminActivityConsumer</code>
Messages Maximum	10
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	auto
Sessions Maximum	-1

JMS consumer- Configuration Tab

Property	Value
Name	<code>edxLoggerCSRActivityConsumer</code>
Messages Maximum	10
Selector	<code>JMSType= 'CSR'</code>
Destination	<code>edx.jms.log</code>

Message Log

JMS session pool- Configuration Tab

Property	Value
Name	<code>edxLoggerMessageLogPool</code>

Property	Value
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	<code>10</code>
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	<code>-1</code>

JMS consumer

Property	Value
Name	<code>edxLoggerUserActivityConsumer</code>
Messages Maximum	<code>10</code>
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. 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 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 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.
- 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.