



Installation Guide

Comunications Billing Analytics

**Hewlett Packard HP-UX™ Operating System
and the BEA WebLogic® Server**

Siebel Systems, Inc., 2207 Bridgepointe Parkway, San Mateo, CA 94404
Copyright © 2005 Siebel Systems, Inc.
All rights reserved.
Printed in the United States of America

No part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way, including but not limited to photocopy, photographic, magnetic, or other record, without the prior agreement and written permission of Siebel Systems, Inc.

Siebel, the Siebel logo, UAN, Universal Application Network, Siebel CRM OnDemand, TrickleSync, Universal Agent, and other Siebel names referenced herein are trademarks of Siebel Systems, Inc., and may be registered in certain jurisdictions.

Other product names, designations, logos, and symbols may be trademarks or registered trademarks of their respective owners.

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

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

Proprietary Information

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

Contents

1 Preface

About Customer Self-Service and Tools 5

About This Guide 6

2 Getting Started

Before You Install; Preparing Your Platform 13

Overview of the Installation Process 13

Communications Billing Analytics System Requirements 14

3 Installing CBA for UNIX

UNIX Permissions for Installation 17

Installing Communications Billing Analytics 17

4 Configuring Your Database Server

Overview 21

UNIX Permissions for Your Database Server 21

UNIX Environment Variables for Your Database Server 22

Oracle Database Server Environment Variables 23

Using Database Partitioning with CBA 23

5 Configuring CBA for Oracle

Configuring a New Oracle Database for UNIX 25

Connecting to Your Oracle Database 34

6 Configuring the WebLogic Application Server for UNIX

- Overview 37
- UNIX Permissions for WebLogic Server 38
- Starting and Stopping WebLogic Server 39
- WebLogic Domain Start Scripts 39
- Setting up Xvfb 40

7 Configuring Java Resources for WebLogic for UNIX

- Overview 43
- Configuring WebLogic 43
- Configuring Java Database Connectivity (JDBC) for CBA 46
- Deploying the CBA Application 47
- Starting the Platform Services scheduler 48
- Testing the Installation 48
- Install X-Terminal 49

Appendix A: WebLogic Reference

- Java Database Connectivity (JDBC) 53
- WebLogic Environment Variables 53
- JDBC Resources for CBA 54
- JTA Timeout 66
- Extending XAResource Transaction Timeout 66
- Performance related settings 68

Appendix B: Uninstalling CBA

- Uninstalling CBA 71

Index

1 Preface

About Customer Self-Service and Tools

Siebel's Self-Service for Communications includes every application that communications service providers need to enable a complete online customer-self service experience at their website. The suite includes software applications for:

- e-Billing and Payment
- Service and Order Management
- Point-of-Sale
- Reporting and Analytics
- Rate Plan Advice

Siebel's Self-Service applications for the telecommunications industry combine Siebel's unrivaled Customer Self-Service and e-Billing software suite with its extensive industry domain expertise. The packaged, out-of-the-box applications are tailored to solve communications service providers' distinct business problems and to meet communications industry-specific process requirements.

Siebel's Self-Service for Communications includes:

Communications Billing Manager

Communication Billing Manager is a complete e-billing application for communications service providers that gives business and consumer customers valuable and convenient access to their communications bills along with the ability to easily make online payments.

Communications Self-Service Manager

Communications Self-Service Manager enables customers of communications service providers to manage every aspect of their service relationship online. From a single convenient interface, customers can easily activate and manage subscriptions, change rate plans and features, and modify subscriber profile settings. Business customers are able to complete these activities for individual employees, as well as company departments and divisions, across their entire organization.

Communication Billing Analytics

Communication Billing Analytics is a reporting solution for business customers that empowers both individual employees and business managers to analyze and understand their communications costs and usage by investigating and identifying trends and patterns across multiple views of their own unique organization.

Rate Plan Advisor

Rate Plan Advisor is a web-based application that recommends the ideal rate plan for wireless subscribers in real-time. Individual consumers as well as large businesses can analyze their actual historical voice/mobile/data usage, find the best-fit rate plans, and compare the features offered by those plans. With its intuitive wizard user interface, Rate Plan Advisor quickly guides end-customers or customer service representatives through the entire analysis process. In addition, a service provider's customer care and marketing groups can also use Rate Plan Advisor to identify pre-churn subscribers, simulate new rate plans, and run predictive analytics.

About This Guide

This guide is intended for system administrators and other IT professionals and describes how to install and configure the third-party platforms that support the CBA production environment and deploy CBA J2EE web applications. See “*Communications Billing Analytics System Requirements*” on page 14 for details on the platforms this guide is intended for.

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

Operating System Administration Requirements

- Start up and shut down the system
- Log in and out of the system
- Determine software patch/pack levels
- Install software & patches/packs
- Navigate the file system
- Manipulate text files
- Create files and directories
- Change permissions of files and directories
- Use basic network commands

- Transfer files with FTP
- Monitor processes & system resource usage
- Perform system backups and recovery
- Implement system security

Database Administration Requirements

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

Application Server Administration Requirements

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

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

Related Documentation

A PDF version of this guide is also available.

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

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

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

Obtaining Siebel Software and Documentation

You can download edocs software and documentation directly from Customer Central at <https://support.edocs.com>. After you log in, click the Downloads button on the left. When the next page appears, a table displays all of the available downloads. To search for specific items, select the Version and/or Category and click the Search Downloads button. If you download software, an email from edocs Technical Support will automatically be sent to you (the registered owner) with your license key information.

If you received a Siebel product installation CD, load it on your system and navigate from its root directory to the folder where the software installer resides for your operating system. You can run the installer from that location, or you can copy it to your file system and run it from there. The product documentation included with your CD is in the Documentation folder located in the root directory. The license key information for the products on the CD is included with the package materials shipped with the CD.

If You Need Help

Technical Support is available to customers who have an active maintenance and support contract with edocs. Technical Support engineers can help you install, configure, and maintain your Siebel application.

This guide contains general troubleshooting guidelines intended to empower you to resolve problems on your own. If you are still unable to identify and correct an issue, contact Technical Support for assistance.

Information to provide

Before contacting Siebel Technical Support, try resolving the problem yourself using the information provided in this guide. If you cannot resolve the issue on your own, be sure to gather the following information and have it handy when you contact technical support. This enables your edocs support engineer to more quickly assess your problem and get you back up and running more quickly.

Please be prepared to provide Technical Support the following information:

Contact information:

- Your name and role in your organization.
- Your company's name
- Your phone number and best times to call you
- Your e-mail address

Product and platform:

- In which edocs product did the problem occur?
- What version of the product do you have?
- What is your operating system version? RDBMS? Other platform information?

Specific details about your problem:

- Did your system crash or hang?
- What system activity was taking place when the problem occurred?
- Did the system generate a screen error message? If so, please send us that message. (Type the error text or press the Print Screen button and paste the screen into your email.)
- Did the system write information to a log? If so, please send us that file. For more information, see the *Troubleshooting Guide*.
- How did the system respond to the error?
- What steps have you taken to attempt to resolve the problem?
- What other information would we need to have (supporting data files, steps we'd need to take) to replicate the problem or error?
- **Problem severity:**
 - Clearly communicate the impact of the case (Severity I, II, III, IV) as well as the Priority (Urgent, High, Medium, Low, No Rush).
 - Specify whether the problem occurred in a production or test environment.

Contacting Siebel Technical Support

You can contact Technical Support online, by email, or by telephone.

edocs provides global Technical Support services from the following Support Centers:

US Support Center

Natick, MA
Mon-Fri 8:30am – 8:00pm US EST
Telephone: 508-652-8400

Europe Support Center

London, United Kingdom
Mon-Fri 9:00am – 5:00 GMT
Telephone: +44 20 8956 2673

Asia Pac Rim Support Center

Melbourne, Australia
Mon-Fri 9:00am – 5:00pm AU
Telephone: +61 3 9909 7301

Customer Central

<https://support.edocs.com>

Email Support

<mailto:support@edocs.com>

Escalation process

edocs managerial escalation ensures that critical problems are properly managed through resolution including aligning proper resources and providing notification and frequent status reports to the client.

edocs escalation process has two tiers:

1. **Technical Escalation** - edocs technical escalation chain ensures access to the right technical resources to determine the best course of action.
2. **Managerial Escalation** - All severity 1 cases are immediately brought to the attention of the Technical Support Manager, who can align the necessary resources for resolution. Our escalation process ensures that critical problems are properly managed to resolution, and that clients as well as edocs executive management receive notification and frequent status reports.

By separating their tasks, the technical resources remain 100% focused on resolving the problem while the Support Manager handles communication and status.

To escalate your case, ask the Technical Support Engineer to:

1. Raise the severity level classification
2. Put you in contact with the Technical Support Escalation Manager
3. Request that the Director of Technical Support arrange a conference call with the Vice President of Services
4. Contact VP of Services directly if you are still in need of more immediate assistance.

2

Getting Started

Before You Install; Preparing Your Platform

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

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

Overview of the Installation Process

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

1. Installing Siebel Platform Services and Communications Billing Analytics on your database and application servers.
2. Configuring the database server.
3. Configuring the application server.

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

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

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

Configuring Your Database Server

Configuring your database server requires you to:

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

Configuring Your Application Server

Configuring your application server requires you to:

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

Deploying the CBA 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

edocs' Platform Services and Communications Billing Analytics Consumer (HP-UX/Oracle/WebLogic)

This guide assumes you are installing Communications Billing Analytics on an HP/UX 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
HP-UX 11i	Oracle 9.20	WebLogic 8.1 SP3	Sun JDK 1.4.1

OPERATING SYSTEM

- Hewlett Packard HP-UX 11i for PA-RISC systems, December 2002 release

HARDWARE

- CD-ROM
- Disk space (database) 2.6 GB
- Disk space (software) 60 MB
- Swap space 512 MB per CPU (1 GB recommended)
- RAM 512 MB per CPU (1 GB recommended)

JAVA/C++

- Sun Java 2 SDK Standard Edition 1.4.1 (version shipped with WebLogic 8)
- IBM Java 2 SDK Standard Edition 1.3.1 (version shipped with WebSphere)

SUPPORTED DATABASE SERVERS

New installation of Communications Billing Analytics

- Oracle 9i Release 2 (Oracle 9.2.0)
- Oracle 9i client software (for application server)
- Oracle 9i JDBC driver

SUPPORTED APPLICATION SERVERS

- BEA WebLogic Server 8.1 SP2

SUPPORTED BROWSERS

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

3

Installing CBA for UNIX

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 must have **root** privilege on each server to install and uninstall Communications Billing Analytics components.

DEFAULT	EXAMPLE	CUSTOM
<code>root:other</code>	<code>edxadmin:edxadmin</code>	

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 21.



edocs does not recommend administering Communications Billing Analytics with the user and group **nobody:nobody**.

Installing Communications Billing Analytics

Communications Billing Analytics (also referred to as **CBA**) is distributed as an InstallAnywhere package. Follow the steps below to install CBA on your system. This document refers to that directory as the TAM_HOME directory, listed as *siebel/TAM*.

siebel/TAM/db/oracle contains platform-specific subdirectories for database creation and configuration.

siebel/TAM contains the web applications to be deployed to your application server.

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

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

Installing CBA:

1. Log in as the root 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 it as follows:

Enter `./TAMins.bin` from a command prompt at the directory location where the installer resides.
3. On the Introduction screen, read the CBA 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 edocs Technical Support). Then click **Next**.
6. On the Owner of Web Application Server screen, enter the name of the application server owner (the same one you used when installing Billing Manager). Then click **Next**.
7. On the Group of Web Application Server screen, enter the name of the group for the application server (the same one you used when installing Billing Manager). 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 CBA files and directories. This document refers to that directory as TAM_HOME. Click the **Next** button to continue.
9. On the Choose Product Features screen, click **TAM 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 CBA 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 CBA 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 CBA. If you need to contact edocs Technical Support, see the section *If You Need Help* on page 9.

4

Configuring Your 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.

edocs recommends that you install and configure CBA 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** CBA 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 CBA



Caution

The installation and configuration examples shown in this guide use default CBA 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 CBA.

UNIX Permissions for Your Database Server

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

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

DATABASE	DB ADMIN USER	DB USER	CUSTOM
Oracle	oracle:dba	edx_dba:edx	

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.

To reset user and group permissions for Oracle:

1. Switch user to **root**.

```
su - root
```

2. Recursively change the user and group permissions of your **TAM_HOME** directory and all subdirectories to the **database instance owner**.

```
chown -R edxadmin:edxadmin siebel/TAM
```

3. Recursively change the user and group permissions of your **TAM_HOME database** directory and all subdirectories to the **database instance owner**.

```
chown -R oracle:dba siebel/TAM/db
```

Tip

You should also verify the owner information in any profile files used by the database server owner and application server owner. See your server documentation for details.

Developers and system administrators need to be familiar with how to stop and start a database server and an active CBA 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 CBA, you must define **environment variables for your database server**. See “*Oracle Database Server Environment Variables*” on page 23.

Example Oracle environment variables for UNIX:

```
ORACLE_BASE=/apps/oracle
ORACLE_HOME=$ORACLE_BASE/product/9.2.0
LD_LIBRARY_PATH=$ORACLE_HOME/lib:/usr/lib:/usr/ucblib
PATH=$ORACLE_HOME/bin:$PATH
export ORACLE_BASE ORACLE_HOME LD_LIBRARY_PATH
```

The syntax used to define environment variables depends on which UNIX shell you are using, as shown in the following examples.

To define environment variables in the Bourne or Korn shell:

```
PATH=$PATH:$HOME/bin
export PATH
```

To define environment variables in the C shell:

```
setenv PATH=$PATH:$HOME/bin
```

Oracle Database Server Environment Variables

VARIABLE	DEFINITION	HP/UX	CUSTOM
TAM_HOME	CBA home path	siebel/TAM/estatement	
ORACLE_BASE	Mount point base path	apps/oracle	
ORACLE_HOME	Data File Path	\$ORACLE_BASE/product/9.2.0 OR 8.1.7	
LD_LIBRARY_PATH	Shared Library Path	\$ORACLE_HOME/lib:/usr/lib:/usr/ucblib	
PATH	Database Path	\$ORACLE_HOME/bin:\$PATH	
ORACLE_DATA	Data File Path	\$ORACLE_HOME/oradata	
ORACLE_PASSWD	Database password	tamtest	
ORACLE_SID	Database instance name	TAMOLTP	
ORACLE_USER	Database user name	tamtest	
ORACLE_DBALIAS	Database alias	TAMOLTP.db	

Using Database Partitioning with CBA

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, CBA 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 CBA 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 CBA database, you run the database configuration shell script `edx_main_admin.sh`. This topic describes each step in detail.



Tip

Database clustering is handled by your application server and not by CBA. Consult your edocs Technical or Professional Services representative for clustered installations.

About `edx_main_admin.sh` for Oracle

To create and configure the CBA 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 33.

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 CBA 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`: The 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 CBA production database.

**Tip**

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

- `listener.ora` includes service names and address of all listeners on a computer, the instance names of the databases for which they listen, and listener control parameters. The address for a server in `listener.ora` requires the SID (SID_NAME) of a database server in `tnsnames.ora`.
You need to modify `listener.ora` on the **database server** machine.
- `tnsnames.ora` includes a list of service names of network databases that are mapped to connect descriptors. It is used by clients and distributed database servers to identify potential server destinations. The address of a given database server in `tnsnames.ora` matches the address of a listener for that server in `listener.ora`.

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

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

To configure Oracle services for UNIX:

1. Switch user to the **DB admin** user.
`su - oracle`
2. Change directory to the network administration directory of your **database server**. For example:
`cd /export/home/oracle/product/9.2.0/network/admin`

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

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

(SID_DESC =
  (SID_NAME = TAMOLAP)
  (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 /export/home/oracle/product/9.2.0/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**.

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

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

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

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 CBA instance.

```
Services Summary...
PLSExtProc has 1 service handler(s)
TAMOLTP has 1 service handler(s)
```

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

To configure a new CBA database for Oracle on UNIX:

1. Switch user to the **DB Admin** user, in this example **oracle**. Oracle requires the administrative user in order to create files.

```
su - oracle
```

2. Change directory to your CBA 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
```

```
edocs Server Administration Main Menu Version 1.0
```

```
-----
[1] OLTP Setup
[2] OLAP Setup
[3] META Setup
[4] CREATE DB LINK
[Q] Quit
-----
```

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

```
edocs e-Statement Server Administration Main Menu
Version 1.0
```

```
-----
[1] Sign in Menu
[2] Capture Database File Locations
[3] Install edocs e-Statement
[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 CBA 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 23 and your database administrator to specify suitable values for your platform.

```
SIGN IN MENU
-----
[1] Enter Database USERNAME...> tamtest
[2] Enter Database PASSWORD...> tamtest
[3] Enter ORACLE_SID ...> TAMOLTP
[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:

```
$ORACLE_HOME/oradata
```

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 edocs Server Administration Main Menu.

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

```

Install edocs
-----
[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

```

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

This step creates a database instance for CBA, 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

Now would be a good time to get a cup of coffee or take a brisk walk.

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

9. Next, the script stops and starts your database server. No user input is required although you see several progress messages. If this step is successful you see a success message. Press **Enter** to return to the Install menu.

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

This option creates the Billing Manager 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.

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

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

12. (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 Billing Manager home directory, for example

```
$EDX_HOME/db/oracle
```

13. Select Option 6, **Initial Data Population**.

This option loads sample data into the CBA schema.

14. Return to the main menu, and select Option [2] **OLAP Setup**. This following screen displays:

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

Enter Your Selection:

15. Select Option 1 for sign in page for OLAP. The following screen appears:

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

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

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

17. Select option 3, **Install edocs Reporting**. The following screen appears:

```
Install edocs 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 CBA 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 you skip step 8, *Sample Hierarchy Data Population*, you can load the sample data into the hierarch later by running a sequence of jobs. These steps are described in the *Administration Communications Billing Analytics* guide, in the section *Scheduling Job for Sample Data*.

If this step is successful, the following message appears:

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



Tip

Now would be a good time to get another cup of coffee or do some other work.

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

18. Next, the scripts stop and start your database server. No user input is required although you see several progress messages. If of these steps is successful you see a success message. Press **Enter** to return to the Install menu.
19. Select Step 4, **Create Reporting Tablespaces**. Press enter when this step completes.
20. Select Step 5, **Create Reporting schema**. Press enter when this step completes.
21. Select Step 6, **Install Reporting schema**. Press enter when this step completes.
22. Select Step 7, **Sample Reporting Data Population**. Press enter when this step completes.
23. Select Option **R** to return to main **OLAP Setup** page, then select option **Q** to return **Main Admin** page. From the main menu, select Option [3] **Meta Setup**, to install the META tools in the OLAP database. The following page appears.

```
Install edocs META Tool
```

```
-----
[1] Shutdown Database
[2] Startup Database
[3] Create META Tool Tablespaces
[4] Create META Tool schema
[5] Install META Tool schema
-----
```

```
[R] Return to previous menu
```

```
SELECT YOUR OPTION:
```


24. Select Option [1] to shutdown the database instance. Once completed Press Enter to return to META tools page.
25. Select Option [2] to Start the database instance. Once completed Press Enter to return to META tools page.
26. Select Option [3] to create table spaces. Once completed Press Enter to return to META tools page.
27. Select Option [4] to create META tool schema. Once completed Press Enter to return to META tools page.
28. Select Option [5] to install META tool schema. Once completed Press Enter to return to META tools page.
29. Select [R] to return to main META page.
30. Select [Q] to return to main admin page.
31. Select option 4, [4] **CREATE DB LINK**, which brings up the following menu, which shows example entries in bold:

```

-----
[1] Enter OLAP Database USERNAME ...>tamtest
[2] Enter OLAP Database PASSWORD ...>tamtest
[3] Enter OLTP Database USERNAME ...>oltptest
[4] Enter OLTP Database PASSWORD ...>oltptest
[5] Enter OLAP TNS NAME           ...>TIAPQA01
[6] Enter OLTP_SID ...>OLTPQA01
[7] Enter OLAP SYS Password ...>sysdba
[8] Enter OLTP SYS Password ...>sysdba
-----

```

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

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

What to Do if Database Configuration Fails

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

To recover from a failed database configuration for Oracle:

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

```
su - oracle
```

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


```
echo $ORACLE_SID
TAMOLTP
```
3. If the database instance name is incorrect, set it as follows:


```
ORACLE_SID=TAMOLTP
export ORACLE_SID
```
4. Shut down the database server.
5. Locate all files associated with the database instance you defined. For example:


```
find . -name '*TAMOLTP*' -print
```
6. Change directory to the root directory for your database and recursively remove the folder and its containing files. For example:


```
cd ./product/oradata
rm -rf TAMOLTP
```
7. Repeat this step for other root directories containing database files for your database instance.

Connecting to Your Oracle Database

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

To test the oltp database for UNIX:

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


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


```
sqlplus tamtest/tamtest@TAMOLTP
```

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

```
Connected to: Oracle9i Enterprise Edition Release
9.2.0.0.0
```
3. At the SQL prompt, enter a database query command, for example:


```
SQL> show parameters db_name
```

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

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

To test the OLAP database for UNIX:

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

```
su - oracle
```

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

```
sqlplus olap_dba/edx@olap
```

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

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

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

```
SQL> show parameters db_name
```

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

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

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

6

Configuring the WebLogic Application Server for UNIX

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 CBA.

Siebel recommends that you install and configure CBA 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 CBA, 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 CBA. It includes:

- UNIX permissions for WebLogic Server
- Starting and Stopping WebLogic Serve
- Capturing your Unix environment for CBA

Caution The installation and configuration examples shown in this guide use default CBA 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 CBA.

About the Sample UNIX Domain Used in this Guide

This guide uses the following example of a UNIX domain:
`$WL_HOME/user_projects/domains/mydomain`

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



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

UNIX Permissions for WebLogic Server

Application servers running CBA do not function correctly without access to CBA configuration files, storage directories, and related resources. When installing CBA 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 CBA.

DEFAULT	EXAMPLE	CUSTOM
Specified during installation.	<code>edxadmin:edxadmin</code>	

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

To reset user and group permissions:

1. Switch user to the default owner of WebLogic Server home directory, for example `root`.

```
su - root
```
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 /export/home/boa
```
3. Switch user to WebLogic Server owner and configure WebLogic Server with your new owner.

```
su - edxadmin
```



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

Starting and Stopping WebLogic Server

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

Starting and Stopping an Active Application Server

Improperly starting or stopping an application server in an active CBA production environment can produce unexpected and unintended results. You can create custom startup and shutdown scripts that include all your command parameters, as well as the command used to start or stop the Scheduler, to schedule and run jobs in the CBA 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. edocs recommends using the web console and/or the SHUTDOWN command to ensure a graceful shutdown.

To start WebLogic in an active CBA production environment, edocs 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.

WebLogic Domain Start Scripts

When you use the Configuration Wizard to create a domain, the wizard also creates a script that you can use to start an Application Server for the domain. To use the script, enter the following command at a command prompt:

```
mydomain/startWebLogic.sh
```

Where **mydomain** is the directory in which you located your domain.

The script sets values for some domain-specific variables and then calls the master startup script, `$WL_HOME/weblogic##/server/bin/startWLS.sh`, where `$WL_HOME` is the location where you installed WebLogic Server and `##` is the WebLogic version installed. The master startup script sets environment variables, such as the location of the JVM, and then starts the JVM with WebLogic Server arguments.

This guide uses the example domain `$WL_HOME/user_projects/domains/mydomain`. WebLogic users may use the Domain Configuration Wizard to create this domain or replace these pathnames with a custom domain created by your system administrator.



Caution If you use a custom domain, substitute the name accordingly in examples in this guide or they will not work.

Setting up Xvfb

Java's graphic packages that are part of CBA 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. To simplify X configuration, charting can use the virtual frame buffers provided by Xvfb.

If you do not already have Xvfb installed on your machine, you can get it from any internet site for X11. Un-tar the archive and install it under */usr/X11R6*.

To Set Display Permission

The command in UNIX environments that allows X displays on a particular machine from other machines is: **xhost +**. Without arguments, commands following the + implies all machines (as opposed to named machines only).

If xhost is already on your path, type the following command. (xhost may exist in the */usr/openwin/bin* folder, which you can add to your PATH in your *.profile*).

At the command prompt, type: **xhost +**

To Set the Display Device

The Xvfb command starts the virtual frame buffer, and is located in `/usr/X11R6/bin`. To use Xvfb:

1. Open an Xterm window on the main console of the server.
2. Set the `DISPLAY` environment variable for Xvfb to use as follows:

```
export DISPLAY=10.2.1.125:0.0
```
3. Assuming that Display 2 will be used for Xvfb, then start the Xvfb server as follows:

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

Tip

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

You can edit the `/etc/profile` file to set the `DISPLAY` environment variable for all sessions and all users, and create a startup script in the `/etc/rc3.d` directory to automatically startup the Xvfb server when the system is rebooted.

4. Restart the application server, after setting up Xvfb.

7

Configuring Java Resources for WebLogic for UNIX

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 CBA.

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 CBA.

Configuring WebLogic

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

- **admin** - the administrative server
- **tamservice** - a managed server for the CBA customer facing application
- **ccservice** - a managed server for Platform Services

Editing *setEnv.sh* to Set Environment Data

Add the following lines to the file *setEnv.sh*, which is located under mydomain:

```
EDX_HOME=/opt/siebel/TAM/estatement
export EDX_HOME
. /opt/siebel/TAM/estatement/config/edx.config (the dot and the
space are important)
```

```

CLASSPATH="${WEBLOGIC_CLASSPATH}:${POINTBASE_CLASSPATH}:${JAVA_
HOME}/jre/lib/rt.jar:${WL_HOME}/server/lib/webservices.jar:${CL
ASSPATH}:/opt/siebel/TAM/config:/opt/siebel/TAM/lib/javachart.
jar:/opt/siebel/TAM/lib/Configuration.jar:/opt/siebel/TAM/lib/l
deprotocol.jar:/opt/siebel/TAM/estatement/lib/edx_common.jar"

export CLASSPATH

JAVA_VM="-
Dorg.apache.commons.logging.Log=com.edocs.common.logging.plugin
s.jcl.Logger -Dedx.home=${EDX_HOME}"

export JAVA_VM

export CLASSPATH

```

Using `edx_config` to Capture Environment Data

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

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



Do not confuse `edx_config` (underscore) with `edx.config` (dot), which sources your configuration at server startup. For details, see [Using `edx.config` to source your configuration](#).

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.



Be sure the time zone (TZ) for your server is set to your system time zone. Billing Manager jobs can fail if 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 - edxadmin
```
2. Navigate to the `bin` directory for Billing Manager 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/bee/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.

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.

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

To source `setEnv.sh` for CBA 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](#).
3. Navigate to the application server startup script **for your domain** and open the file for editing. For example:


```
cd $WL_HOME/user_projects/domains/mydomain
vi startWebLogic.sh
```
4. Add a line to source `setEnv.sh` in the script. For example:


```
# Call setEnv.sh
. ./setEnv.sh
```
5. Define and export the environment variable for your Platform Services home directory, then source the Platform Services configuration. For example:


```
EDX_HOME=/opt/siebel/TAM/estatement
export EDX_HOME
. /opt/siebel/TAM/estatement/config/edx.config
```
6. Save and close `startWebLogic.sh`.
7. Repeat the same edits to `startManagedWebLogic.sh`.

Configuring Java Database Connectivity (JDBC) for CBA

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

About JDBC Connections for CBA

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

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

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

Configuring JDBC Connections for CBA

You must create JDBC connection pools and transaction data sources for the CBA and Platform Services WebLogic servers.

See Appendix A for appropriate WebLogic JDBC configuration settings for each server.

For more details on how to configure JDBC connections, see WebLogic Server documentation at <http://edocs.bea.com>.

To configure JDBC Connections for CBA:

1. Create a JDBC Connection Pool each for **ConnectionPool1**. Use the appropriate JDBC values for your database server.
2. Create a JDBC Data Source each for **DataSource**. Use the appropriate JDBC values for your database server.
3. Review your connections. The data source should target the connection pool.
4. You can clone JDBC resources to save time.

Deploying the CBA Application

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

- **Application servers:** Deploy the CBA EAR file.

The EAR files are located at:

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

Consult your BEA WebLogic documentation on how to deploy applications.

Testing the CBA Installation

After successfully deploying the application, you can log into the CBA 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).

Testing the Billing Manager Installation

Once deployed, you should be able to successfully log-in to the Billing Manager application with the following username/password..

Use following URL to access the application: `http://your_host_name:7001/tbmb/`

Login using the following username/passwords

twhite/twhite
 jsmith/jsmith
 gsnyder/gsnyder
 klaracey/klaracey
 jblack/jblack
 jthomas/jthomas
 asameuls/asameuls

Once you are logged in, click on Analytics tab. This takes to the CBA reporting list page.

Starting the Platform Services 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 with the Scheduler not running, the job will not 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 EAR file.

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

1. Switch user to the application server owner.
`su - edxadmin`
2. Change directory to the **bin** directory of your Billing Manager installation, **\$EDX_HOME/bin**.
3. Run the Scheduler command for your application server, host, and port. For example:
`./wl_scheduler -start -url t3://localhost:7001`

Tip

The example uses the default port for the administration server. You can specify another port number if necessary, but it must match the port for the admin server.

4. You can stop the Scheduler by replacing the **-start** parameter with the **-stop** parameter.

Testing the Installation

Testing the CBA Installation

After successfully deploying the application, you can log into the CBA 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:

jsmith/jsmith

klaracey/klaracey

Testing Platform Services

Create a new application:

1. Enter the URL <http://localhost:7001/edocs> (substitute the host:port, if necessary), which displays the Command Centers Main Page.
2. The User Name and Password is admin/edocs.
3. Enter the new Application Name , for example, **testApp**.
4. Use **/edx/ejb/EdocsDataSource** for Datasource Name
5. Choose the default for Index Partition Count.
6. Click on the **Create Application** button.

Install X-Terminal

X Displays

To display charts, your webserver must have access to an X display device to render charts. Do not expect font styles if you 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 `+`, 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 webserver which is rendering the charts -- the web tier -- must have access to a real or virtual X display device, and that the necessary X Libraries are loaded. Note: it is not necessary for the webserver'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 and 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 will then be located in `/usr/X11R6/bin` and is called Xvfb. 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 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
```

Manual steps for setting up the server's charting environment

1. Install X libraries on the Webserver. Untar `X11R6.tar.gz` into the `/` (root) directory (as the root user):


```
# gzip -d X11R6.tar.gz (if gzip is installed)
# tar xvf X11R6.tar
```
2. Start the X Virtual Frame Buffer. 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 X11Environment 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 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 webserver 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 +
```

Setting the environment automatically on system reboot

The following show a convenient way of automating the charting environment set-up if the app servers are restarted...

1. After installing the X11R6 libraries, change your working directory to */etc/init.d*, and edit the file *chart*:

```
# 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
```

Appendix A: WebLogic Reference


Java Database Connectivity (JDBC)

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

For details of how to configure JDBC connections, see your application server documentation. For the procedure to create connections for CBA, see “*JDBC Resources for*” on page54 .

 **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	HP/UX
APP_OWNER	app server owner	<code>edxadmin</code>
APP_GROUP	app server group	<code>edxadmin</code>
APP_PORT	app server port	<code>7001</code>
ADMIN_PORT	app server admin port	<code>7002</code>
JAVA_HOME	Java home directory	<code>\$WLHOME/jdk141_05</code>

JDBC Resources for CBA

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, **ccserver**.

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:

- edxCCAdminConnectionPool
- edxLoggerConnectionPool
- edxUserConnectionPool
- olapOracleConnectionPool

Oracle's Thin XA 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 XA) Versions:8.1.7,9.0.1,9.2.0

To create the first connection pools:

- edxCCXMAConnectionPool

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, the the WebLogic server to target when the connection pool wizard completes.

Connection Pool	Database	Targeted Server	Supports Local Transactions
edxCCAdminConnectionPool	OLTP	tamserver	
edxLoggerConnectionPool	OLTP	tamserver and ccserver	
edxUserConnectionPool	OLTP	tamserver and ccserver	
edxXMAConnectionPool	OLTP	tamserver	YES
edxCCXMAConnectionPool	OLTP	ccserver	YES
edxMessagingConnection Pool	OLTP	tamserver and ccserver	
reportConnectionPool	OLAP	tamserver and ccserver	YES
olapOracleConnectionPool	OLAP	ccserver	

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):

Connections Tab	
Initial Capacity	1
Maximum Capacity	20
Capacity Increment	5
Login Delay	1
Statement Cache Size	300
Test Frequency	60
Allow Shrinking	True (box checked)
Shrink Frequency	15
Test Reserved Connections	TRUE (checked)
Test Released Connections	FALSE (unchecked)
Test Table Name	dual
Supports Local Transaction	see the Connection Pool Settings table

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

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



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 clone that to create the others.

Name	<code>edxCCAdminDataSource</code>
JNDI Name	<code>edx.databasePool</code>
Pool Name	<code>edxCCAdminConnectionPool</code>

Name	<code>edxUserDataSource</code>
JNDI Name	<code>edx.user.databasePool</code>
Pool Name	<code>edxUserConnectionPool</code>

Name	<code>edxLoggerDataSource</code>
JNDI Name	<code>edx.logger.databasePool</code>
Pool Name	<code>edxLoggerConnectionPool</code>

Name	<code>reportDataSource</code>
JNDI Name	<code>edx.report.databasePool</code>
Pool Name	<code>reportConnectionPool</code>

Name	<code>edxMessagingDataSource</code>
JNDI Name	<code>edx.messaging.databasePool</code>
Pool Name	<code>edxessagingConnectionPool</code>

Name	<code>XMADataSource</code>
JNDI Name	<code>edx.xma.databasePool</code>
Pool Name	<code>edxXMAConnectionPool</code>

Name	<code>XMACCDataSource</code>
JNDI Name	<code>edx.xma.databasePool</code>
Pool Name	<code>edxCCXMAConnectionPool</code>

Name	<code>OLAPOracleDataSource</code>
JNDI Name	<code>edx.olap.databasePool</code>
Pool Name	<code>olapOracleConnectionPool</code>

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	tamserver and ccserver
edxMessagingConnectionFactory	edx.qcf	tamserver and ccserver

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

JMS (JDBC) Stores

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

Name (of JMS Store)	Connection Pool	Directory
edxCCLoggerStore	edxLoggerConnectionPool	
MessageCCFileStore		rmfilestore
xmaEventFileStore		xmafilestore
CCxmaEventFileStore		xmafilestore

JMS Servers

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

Name	(Persistent) Store	Targets Tab
edxCCLoggerServer	edxCCLoggerStore	ccserver
edxLoggerServer	edxLoggerStore	tamserver
CCxmaEventServer	CCxmaEventFileStore	ccserver
xmaEventServer	xmaEventFileStore	tamserver

JMS Topics

Create three 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

xmaEventErrorQueue also requires the following settings:

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
--------------------------	----------

Create these topics for both the **tamserver** and **ccserver**.

JMS Session Pools and Consumers for Logging for Command Center

Create FIVE pairs of **JMS Session Pools and Consumers** for **Logger**, 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.

1) Admin Activity

JMS session pool - Configuration Tab

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

JMS Consumer- Configuration Tab

Property	Value
Name	edxCCLoggerAdminActivityConsumer
Messages Maximum	10
Selector	JMSType= 'ADM'

Property	Value
Destination	<code>edx/jms/log</code>

2) CSR Activity

JMS session pool- Configuration Tab

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

JMS consumer- Configuration Tab

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

3) Message Log

JMS session pool- Configuration Tab

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

JMS consumer- Configuration Tab

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

4) 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>

5) 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

Property	Value
Sessions Maximum	-1

JMS consumer

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

JMS Session Pools and Consumers for Logging for CBA

Create FIVE pairs of **JMS Session Pools and Consumers** for **Logger**, 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.

2) Admin Activity

JMS session pool - Configuration Tab

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

JMS Consumer- Configuration Tab

Property	Value
Name	edxLoggerAdminActivityConsumer
Messages Maximum	10
Selector	JMSType= 'ADM'

Property	Value
Destination	edx/jms/log

2) CSR Activity

JMS session pool- Configuration Tab

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

JMS consumer- Configuration Tab

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

3) Message Log

JMS session pool- Configuration Tab

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

JMS consumer- Configuration Tab

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

4) 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	-1

JMS consumer

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

5) 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>

Property	Value
Sessions Maximum	-1

JMS consumer

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

JTA Timeout

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

Extending XAResource Transaction Timeout

The WebLogic Server Transaction Manager now supports setting a transaction branch timeout value on a participating XA resource if the resource manager supports the `javax.transaction.xa.XAResource.setTimeout()` 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 JDBC connection pool tag in the `config.xml` file:

- **XASetTransactionTimeout**-A boolean property. When set to true, the WebLogic Server Transaction Manager calls `XAResource.setTimeout()` 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 `setTimeout()`. The default value is **false**.

- **XATransactionTimeout**-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.

The `XASetTransactionTimeout` and `XATransactionTimeout` properties are **not** available in the Administration Console. You must add them to the `config.xml` file while the domain is not active. For example, the following section from `config.xml` shows the two new parameters added to the `edxXMAConnectionPool`:

```
<JDBCConnectionPool
  DriverName="oracle.jdbc.xa.client.OracleXADataSource"
  Name="edxXMAConnectionPool"
  Password="{3DES}8YdvP4FQW3k="
  Properties="user=edx_dba"
  URL="jdbc:oracle:thin:@server:port:sid"
  XASetTransactionTimeout="true"
  XATransactionTimeout="4800"/>
```

You must manually apply these settings to `edxXMAConnectionPool` and `reportConnectionPool` in WebLogic's `config.xml`.

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 command center and CBA servers)

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

For a machine with more RAM, it's 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 source 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 (for CBA application server, 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.

Appendix B: Uninstalling CBA

Uninstalling CBA

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

Uninstall CBA 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 CBA 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 CBA:

1. Navigate to the **uninstall** folder of your CBA home directory, **\$TAM_HOME**.
2. Launch the CBA 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 CBA 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 CBA 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.

Index

- A**
 - administration, 26
- C**
 - CBA, 6, 38
 - uninstalling, 71
 - Checking .log files for errors, 31
 - Command Center, 39
 - Connecting to the CBA Database
 - Oracle, 34
- D**
 - Database Configuration Fails, 33
 - Database Connectivity, 46
 - Database Server, 14
 - documentation, 22
- E**
 - edx_main_admin.sh, 14, 21, 25
 - Environment Data, 44
 - Database Server, 22
- H**
 - Help
 - documentation, 7
- J**
 - technical support, 8
- O**
 - JDBC, 46
 - Oracle, 21, 33
 - Oracle Services, 26
- P**
 - Permissions
 - Application Server, 38
 - Database Server, 21
 - Installation, 17
- S**
 - Scheduling jobs, 39
 - Setting up Xvfb, 40
 - sqlplus, 34, 36
 - Starting and Stopping
 - Application Server, 39
- U**
 - Uninstalling CBA, 71
 - Using edx_config, 44
- W**
 - WebLogic, 46
- X**
 - Xvfb setup, 40