Installation Guide for Service Manager

Microsoft Windows 2003 Operating System and the BEA WebLogic Server

Version 5.1

Date Published: 02/28/2006

ORACLE | SIEBEL

Contents

Index

1 Preface

About This Guide

This guide is intended for system administrators and other IT professionals and describes how to install and, configure the third-party platforms that support the Siebel Service Manager production environment and deploy Billing Manager J2EE web applications. See "Service Manager System Requirements" on page 11 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
- Manipulate text files
- Create files and directories
- Use basic network commands
- Transfer files with FTP
- Monitor processes & system resource usage
- Perform system backups and recovery
- Implement system security

Database Administration Requirements

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

- Monitor database performance

Application Server Administration Requirements

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

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

If you are unfamiliar with any of these tasks, please consult the related documentation for your system requirements.

Related Documentation

A PDF version of this guide is also available on SupportWeb.

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

Siebel Billing Manager Developer's Guide	How to customize J2EE web applications for deployment with Service Manager.
Siebel Billing Manager Data Definition (DefTool) Guide	How to create Data Definition Files (DDFs) for use in indexing your application and extracting data for live presentment.
Siebel Billing Manager Presentation Design (Composer) Guide	How to create Application Logic Files (ALFs) to present statement data for dynamic online display.
Siebel Billing Manager Administration Guide	How to set up and run a live Service Manager application in a J2EE environment.

2 Getting Started

Preparing Your Platform

Before installing Service Manager, verify that your platform is ready

- Install and test required hardware and software for your platform.
- Start and test your database server. For details, see your server documentation.
- Start and test your application server. For details, see your 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 Siebel Service Manager includes the following steps

1. Installing Siebel Platform Services and Service Manager on your database and application servers using InstallAnywhere. You can install these individually, running InstallAnywhere twice (once to install each feature), or together using the Custom install feature.

   NOTE: If you are installing Platform Services and Service Manager on different servers, you should install Platform Services and get it running before installing Service Manager.

2. Installing Siebel Tools on a Windows 2000 machine. Run InstallAnywhere here and exclusively install Tools.

3. Configuring the database server.

4. Configuring the application server.

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

Once you successfully install Service Manager 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 Service Manager.
3. Connect to your Service Manager database before configuring your application server.

Configuring Your Application Server

Configuring your application server requires you to

1. Define application server environment variables.
2. Configure JDBC resources for Service Manager on your application server.
3. Configure JMS resources for Service Manager on your application server.
4. Install Windows Services for your application server and the Service Manager Scheduler.

Customizing and Deploying J2EE Applications

After installing Service Manager and configuring your database and application servers, you can

1. Customize your J2EE web application(s) for Service Manager.
2. Deploy J2EE web applications for Service Manager.
3. Deploy your custom J2EE web application.

Service Manager System Requirements

Siebel Platform Services and Service Manager (Windows/SQL Server/WebLogic)

This guide assumes you are installing Service Manager on a Windows 2003 Server operating system, SQL Server 2000 database, and WebLogic application server.

OPERATING SYSTEM

- Microsoft Windows 2003 Server

HARDWARE

- CD-ROM
- Disk space (database) 10 GB
- Disk space (for Service Manager software) 500 MB (in addition to the space required for the application server)
- Intel Pentium or compatible processor at 850 MHz or higher
- Swap space 1 GB per CPU (2 GB recommended)
- RAM 1 GB per CPU (2 GB recommended)

JAVA/C++

- Sun Java 2 SDK Standard Edition 1.4.1 (version shipped with WebLogic)

SUPPORTED DATABASE SERVER

- Microsoft SQL Server 2000 SP3

SUPPORTED APPLICATION SERVERS

- BEA WebLogic Server 8.1 SP4

SUPPORTED BROWSERS

- Netscape Navigator 7.0 or higher
- Microsoft Internet Explorer 6 or higher (on networked PC)
- Firefox 1.0.1

Siebel Tools (Windows)

OPERATING SYSTEM

- Microsoft Windows 2000/Server SP4 or higher

3 Installing Service Manager for Windows

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

Installing Service Manager

InstallAnywhere is a graphical cross-platform wizard that lets you install Service Manager in a distributed environment:

- **Siebel Platform Services** – Install on all database and application servers.
- **Service Manager** - Install on all application servers.
- **Siebel Tools** – Install on a Windows machine accessible to the servers on your network.

Siebel recommends that you install and configure Service Manager in the same top-level directory structure, first on the database server, then on the application servers.

Siebel Platform Services and Service Manager

You must install both Siebel Platform Services and Service Manager using InstallAnywhere. You can install them individually or create a custom install to install at once.

To install Siebel Platform Services and/or Service Manager with InstallAnywhere

1. Obtain and locate the InstallAnywhere installer.
2. Launch InstallAnywhere by double-clicking the `TSMins.exe` icon.
3. INTRODUCTION: InstallAnywhere recommends that you quit all programs before installing.
4. LICENSE AGREEMENT: Review the License Agreement and click "**`I accept the terms of the License Agreement`**" to accept the terms.
5. ENTER THE SERIAL NUMBER provided when you purchased Service Manager.
6. OWNER OF WEB APPLICATION SERVER for example **edxadmin**.
7. GROUP OF WEB APPLICATION SERVER for example **edxadmin.**

8 CHOOSE INSTALL FOLDER: Click **Next** to accept the default or specify another directory. Siebel recommends that you install and configure Service Manager in the same top-level directory structure, first on the database server, then on the application server.

9 CHOOSE PRODUCT FEATURES: Choose which feature you want to install first, or click **Custom** and choose to install Service Manager and Siebel Platform services at the same time.

- **Service Manager** - Installs all components for Service Manager.
- **Platform Services** (default) - Installs the components for Siebel Core Services only, including e-Billing, Hierarchy, Payment, and Reporting.
- **Siebel Tools -** Installs DefTool and Composer (on a Windows machine only)

10 CHOOSE SHORTCUT FOLDER: Click **Next** to install Service Manager in the Siebel program group.

11 PRE-INSTALLATION SUMMARY: Review the screen to confirm your product and version, install folder, product components, and disk space required and available and click **Install.**

12 InstallAnywhere sets up a directory hierarchy on each server and copies files to the appropriate directories.

13 INSTALL COMPLETE: If installation is successful, you see a congratulatory message. Click **Done**.

14 Repeat the installation for other Service Manager servers on your network as necessary.

Tools

To install Siebel Tools on a Windows machine on your network

1 Obtain and locate the InstallAnywhere installer.

2 Launch InstallAnywhere by double-clicking the **TSMins.exe** icon.

3 Follow the steps in the procedure above for installing Service Manager, selecting the Siebel Tools feature to install.

Payment

The Service Manager installation procedure automatically installs and configures Payment with Platform Services.

By default, Payment is installed in: *C:\Siebel\TSM\Payment*.

(The default installation path for Payment is also referred to as PAYMENT_HOME, or *CSM_HOME/Payment*)

The Service Manager Directory Structure

The Service Manager home directory contains all the files you need to create and configure the Service Manager production database. When you install Service Manager components, InstallAnywhere prompts you to specify a destination directory. You can use the default or specify another directory.

The default Service Manager installation directories are:

- Service Manager: C:*Siebel**TSM*\
- Platform Services: C:*Siebel**TSM**estatement*
- Siebel Tools: C:*Siebel**TSM**estatement**bin*

TIP: Siebel recommends that you install Service Manager in the same top-level directory on both the database server and the application servers.

Where to Find Database Components

*Siebel**TSM**estatement**db* contains platform-specific subdirectories for database creation and configuration.

Where to Find Application Server Components

*C:**Siebel**TSM**estatement**J2EEApps* contains platform-specific subdirectories for Siebel J2EE and web applications to be deployed to your application server. Be sure to deploy the correct version for your platform.

Where to Find Input and Output Data

*C:**Siebel**TSM**estatement**AppProfiles* stores information on each new Service Manager application created in the Siebel Command Center. *Siebel**TSM**estatement**Input* is the default input directory used by each Command Center job. *Siebel**TSM**estatement**Data* stores data processed by the Command Center. *Siebel**TSM**estatement**Output* stores the output of jobs.

An additional directory, *Siebel/TSM/estatement/Store*, appears when the first Command Center job runs. The *Store* directory holds temporary files created during job run time. When the job completes, Service Manager automatically cleans up these temporary files.

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.

Siebel recommends that you install and configure Service Manager 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** Siebel database. It includes:

- Windows environment variables for your database server
- Using database partitioning with Siebel

CAUTION: The installation and configuration examples shown in this guide use default Siebel 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 Siebel.

SQL Server Database Server Environment Variables

VARIABLE	DESCRIPTION	WINDOWS	CUSTOM
ADMIN_NAME	Admin name	**sa**	
ADMIN_PASSWD	Admin password	Leave blank	
DB_USERNAME	Database user name	**edx_dba**	
DB_PASSWD	Database password	**edx**	
DB_NAME	Database name	**edx0**	
SERVER_NAME	Server name	**localhost**	
EDX_HOME	Platform Services home	**c:\Siebel\TSM\estatement**	
*CSM*_HOME	Service Manager home path	**c:\Siebel\TSM**	
DB_DATAPATH	Database Data File Path	**c:\sqldir\data**	
LOG	Database Log File Path	**c:\sqldir\log**	

Using Database Partitioning with Service Manager

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 Manager creates and populates a set of partitioned index tables to maintain your dynamic data.

SQL Server supports partitioned views. Index tables are created with a check-constraint, and the view is created on the underlying index tables. Siebel recommends using 4 or 12 partitions for quarterly or monthly index tables.12 partitions are recommended for maximum performance.

For more information on using partitions with your DDNs, see the *Siebel Billing Manager Administration Guide*.

Configuring a New SQL Server Database for Windows

Windows administrators need to complete two steps to create and configure your Platform Services database:

- Increasing the Default Width for SQL Server Numeric Columns
- Using the DBConfigTool

Increasing the Default Width for SQL Server Numeric Columns

The SQL Server default width for numeric columns is 28 characters. However, Service Manager requires a minimum width of **38 characters** for numeric columns. You must increase these defaults **before** creating the database.

To increase the default width of numeric columns

1. From the Start menu, select **Programs** and **Microsoft SQL Server,** and click **Enterprise Manager**. The SQL Enterprise Manager window appears.
2. In the left pane, expand **Microsoft SQL Servers** and **SQL Server Group** to show your server.
3. In the left pane, highlight your SQL server by right clicking on its name, and then click **Properties** from the menu.

   The SQL Server Properties dialog appears.
4. Click **Startup Parameters** at the bottom of the dialog. The Startup Parameters dialog appears showing the name of the server.

5 Enter the startup parameter **-p** in the Parameter text field.

6 Click **Add.** The new startup parameter is added to the list of existing parameters.

7 Click **OK** to close the dialog.

8 Stop and start your SQL server for the new startup parameter to take effect.

Using the DBConfigTool

Windows administrators use the Java database configuration tool, `DBConfigTool.jar`, to create and configure the production database. This wizard guides you to specify information about the database including the username and password needed to access it, the name of the database, the server on which it is installed, and the pathnames for the SQL Data and Log files

The DBConfigTool requires a Java SDK installed on the machine that runs it.

CAUTION: When installing Microsoft SQL Server 2000, select **Mixed Mode (Windows Authentication and SQL Server Authentication)** as your default authentication mode. Selecting only Windows Authentication Mode might cause database creation and configuration to fail.

To configure a new Service Manager database for SQL Server on Windows

1 Open a Command Prompt window and change directory to your Service Manager database home directory. For example:

```
C:\> cd Siebel\TSM\estatement\db\mssql
```

2 Run the Java database creation and configuration tool `DBConfigTool.jar`. For example:

```
\bea\jdk141_05\bin\java –jar DBConfigTool.jar
```

Note that the above java command example uses the JDK provided by WebLogic. Your JDK location may differ.

The Database Configuration screen appears.

3 Select **Create and configure new database** (default), and then click **Next**. The Database Configuration screen appears.

4 Click **Next**. The Database Administrator Information screen appears.

5 Enter the Admin Name and Admin Password (if your database has one) for the SQL database. If you are using the default values provided by SQL Server, the Admin Name is **sa** and the default password is left blank as shown in the example. Please consult for SQL Sever DBA for the correct values for your database server.

Click **Next**. The Database Information screen appears.

6 Enter your database user name, password, database name, and the name of your database server (your server name will differ).

7 Click **Next**. The Database Path Information screen appears.

8 Enter the pathname to the database installation directory, location of the SQL data file, and the location of the SQL log file. By default those values are as follows:

- Siebel DB Home Path: `c:\Siebel\TSM\estatement`
- Database Data File Path: `c:\sqldir\data`
- Database Log File Path: `c:\sqldir\log`

9 Click **Finish**. The DBConfigTool creates your database.

10 A Success screen appears if the database creation process completed without errors.

11 If a Failed screen appears, see the next section.

Creating the Service Manager Database

To create the Service Manager database schema for MSSQL

1 At the command prompt, navigate to the MSSQL directory: *CSM*_HOME\db\mssql.

2 Edit the cbmusr.properties file and change the following property values as appropriate:

```
DB_NAME
DB_USERNAME
DB_PASSWORD
```

3 At the command prompt, run the following commands:

```
set JAVA_HONE=C:\bea81sp4\jdk142_05

set ANT_HOME=C:\Siebel;\TSM\ant
set PATH=%JAVA_HOME%\bin;%ANT_HOME%\bin;%PATH%
```

4 Execute the ant build script:

```
ant –f dbinstall.xml
```

The following menu appears:

```
CBM Enhancement Main Menu
-------------------------
[1]. Create CBM Objects
[2]. Initial data population
[3]. Run all steps at once (Steps 1, 2)
[Q]. Quit

Enter your selection (1, 2, 3, q, Q)
```

5 You can create the database schema in one of two ways:

- Step-by-step: Run options 1-2 in sequence. After completing each option, the system returns to the Command Prompt. Execute the ant build script again and proceed to the next option.
- All steps at once: Choose Option 3, "Run all steps at once (Steps 1 and 2)."

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 SQL Server

1 Make sure your database server is running.

2 From the Start menu, select **Programs** and **Microsoft SQL Server**, and click **Enterprise Manager**. The Enterprise Manager screen opens.

3 Expand the SQL Server Group and click the **Database** folder.

4 Right-click the name of the newly created database (for example, **edx0**) and delete it from the list of installed databases.

5 Scroll down to the **Security** folder and expand it to show **Logins**. Delete the database user (for example, edx_dba) for the database that you just deleted.

6 Open a command line window and run the database configuration tool again, making sure that the values on each screen are correct before proceeding to the next screen.

Creating the Payment Database

To run the database creation script for MSSQL

1. In the directory *%PAYMENT_HOME%\db\mssql*, edit *set_isql_options.bat* and enter the correct information for the database, user name and password (Payment uses the same database as Service Manager) on the line that starts with:

   ```
   set ISQL_OPTIONS......
   ```

2. Run *%PAYMENT_HOME%\db\mssql\create_payment_db.bat* to create payment databases.

   If you run the database creation script from a command prompt, you will see the database creation process and messages.

3. Restart the system.

Creating the Case Management Schema

Log into SQL Query Analyzer as CSM db user and execute these scripts in sequence:

casemgt_db.sql
csm_index.sql
sp_ecs_getNextId.sql

These files are available in `%TSM_HOME%\db\tsm\mssql` folder.

Start and Test Your Database Server

Start and test your database server using the server documentation for your platform. If you encounter any errors, double-check the steps in these chapters before proceeding.

Once your database server starts successfully with the Service Manager database installed, you can proceed to configure your application server.

5 Configuring the WebLogic Server for Windows

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 Service Manager.

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

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

CAUTION: The installation and configuration examples shown in this guide use default Service Manager 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 Service Manager.

About the Sample Windows Domain Used in this Guide

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

CAUTION: When creating WebLogic domains, do not user the default JRockit JVM; Use Sun's JDK.

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

Starting and Stopping WebLogic Server

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

Sourcing Your Configuration

About Sourcing Your Configuration

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

Starting and Stopping an Active Application Server

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

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

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

Capturing Your Windows Environment for Service Manager

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

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

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

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

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

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

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

To edit Windows environment data with edx.config.bat

1. Navigate to %EDX_HOME%\config and edit edx.config.bat.

2 Add the following lines to the :wlMain section. (Add Verisign.jar to the CLASSPATH only if you are implementing credit card payment functionality.)

```
@set CLASSPATH=%CLASSPATH%;C:\Siebel\TSM\lib\commons-logging-1.0.3.jar
@set CLASSPATH=%CLASSPATH%;C:\Siebel\TSM\lib\Configuration.jar
@set CLASSPATH=%CLASSPATH%;C:\Siebel\TSM\lib\javachart.jar
@set CLASSPATH=%CLASSPATH%;C:\Siebel\TSM\lib\ldeprotocol.jar
@set CLASSPATH=%CLASSPATH%;C:\Siebel\TSM\lib\log4j-1.2.8.jar
@set CLASSPATH=%CLASSPATH%;C:\Siebel\TSM\lib\dom4j-1.4.jar
@set CLASSPATH=%CLASSPATH%;C:\Siebel\TSM\config
@set CLASSPATH=%CLASSPATH%;C:\Siebel\TSM\payment\lib\payment_client.jar
@set CLASSPATH=%CLASSPATH%;C:\Siebel\TSM\payment\lib\payment_custom.jar
@set CLASSPATH=%CLASSPATH%;C:\Siebel\TSM\payment\lib\payment_common.jar
@set CLASSPATH=%CLASSPATH%;C:\Siebel\TSM\payment\lib\Verisign.jar
@set CLASSPATH=%CLASSPATH%;C:\Siebel\TSM\payment\lib\jnet.jar
@set CLASSPATH=%CLASSPATH%;C:\Siebel\TSM\payment\lib\jsse.jar
@set CLASSPATH=%CLASSPATH%;C:\Siebel\TSM\payment\lib\jcert.jar
```

3 Still in *edx.config.bat*, add |com.edocs.domain.telco.lde.protocol to the following line:

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

to make:

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

4 Add the following line to edx.config.bat:

```
@set JAVA_OPTIONS=%JAVA_OPTIONS% -
Dcom.edocs.tasks.statementscanner.skipResolver=true

NOTE: This is not used in live production; it is for use with sample data
only.
```

5 Modify the LD_LIBRARY_PATH to:

```
@set LD_LIBRARY_PATH=%EDX_HOME%\bin
```

6 Save and close all files.

To edit Windows environment data with edx.config.bat

1 Navigate to %EDX_HOME%\config and open edx_env.bat.

2 Modify the default settings to reflect your environment. For example:

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

@rem define EDX_HOME
@set EDX_HOME=C:\Siebel\TSM\estatement
@rem define EDX_HOME

@rem define JAVA_HOME
@set JAVA_HOME=C:\bea\jdk141_05
@rem define JAVA_HOME

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

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

To edit Windows Java Options in edx_load.config.bat

1. Navigate to %EDX_HOME%\config and open edx_load.config.bat.
2. Modify the default settings for com.edocs.tasks.loader to reflect your database user, password, and server name. Use the settings for your database as described in the previous chapter. For example:

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

3. Save and close the file.

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

Passing Windows Environment Data to WebLogic

edx.config.bat is a script you call and process in your application server startup script to pass your Service Manager environment (stored in edx_env.bat and edx_load.config.bat) to WebLogic.

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

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

To pass your Service Manager environment to WebLogic (overview)

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

Passing Your Configuration to WebLogic Running as a Windows Service

Siebel recommends installing WebLogic Server as a Windows Service, and modifying the script that calls that service. For WebLogic, this file is `%WL_HOME%\user_projects\domains\mydomain\InstallService.cmd`. This script calls the master startup script `InstallSvc.cmd`.

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

Example InstallService.cmd for WebLogic

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

```
.
.
.

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

set WLS_USER=
set WLS_PW=

.
.
.

set EDX_HOME=C:\Siebel\TSM\estatement
call %EDX_HOME%\config\edx.config.bat
```

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

set JAVA_OPTIONS=%JAVA_OPTIONS%

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

set JAVA_VM=

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

MEM_ARGS="-Xss1m -server -Xms1052m -Xmx3072m -XX:MaxPermSize=184m -
XX:+UseLWPSynchronization -XX:+UseThreadPriorities -Xconcurrentio"

.
.
.

:installSvc

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

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

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

rem *** Install the service

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

:finish

ENDLOCAL
```

To edit InstallService.cmd for WebLogic

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

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

1. Stop WebLogic Server and all application server instances.
2. Navigate to the `%WL_HOME%\user_project\domains\mydomain` subdirectory of your application server home directory.

3 Open `InstallService.cmd` by right clicking on its name, and selecting **Edit**.

4 Before the **JAVA_OPTIONS** definition, set `EDX_HOME` and call `edx.config.bat`. For example:

```
set EDX_HOME=C:\Siebel\TSM\estatement
call %EDX_HOME%\config\edx.config.bat
```

5 Optimize JVM Memory by increasing the memory arguments allocated to the Java Virtual Machine (JVM) on the application server. For example (quotes are optional):

```
set MEM_ARGS="-Xss1m -server -Xms1052m -Xmx3072m -XX:MaxPermSize=184m -
XX:+UseLWPSynchronization -XX:+UseThreadPriorities -Xconcurrentio"
```

6 Add your `CSM\lib` directory to the `EXTRAPATH` setting. See the examples above.

7 (Optional) You can set your application server user and password in the script (to bypass entering it in a console window) by specifying them for `WLS_USER` and `WLS_PW`.

8 Save and close `InstallService.cmd`.

To edit InstallSvc.cmd for WebLogic

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

1 Stop WebLogic Server and all application server instances.

2 Navigate to the `%WL_HOME%\weblogic81\server\bin` subdirectory of your application server home directory.

3 Open `InstallSvc.cmd` by right clicking on its name, and selecting **Edit**.

4 Set your Service Manager home directory and call your Service Manager environment script right after the CLASSPATH setting. For example:

```
set EDX_HOME=C:\Siebel\TSM\estatement
call %EDX_HOME%\config\edx.config.bat
```

5 Save and close `InstallSvc.cmd`.

Passing Your Configuration in a Startup Script for WebLogic

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

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

Example startWebLogic.cmd for WebLogic

```
.
.
.

@REM Initialize the common environment.

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

set PRODUCTION_MODE=true

set JAVA_VENDOR=Sun

set JAVA_HOME=C:\bea\jdk141_05

for %%i in ("%JAVA_HOME%") do set JAVA_HOME=%%~fsi

MEM_ARGS="-Xss1m -server -Xms1052m -Xmx3072m -XX:MaxPermSize=184m -
XX:+UseLWPSynchronization -XX:+UseThreadPriorities -Xconcurrentio"

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

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

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

set SERVER_NAME=myserver

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

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

.
.
.
```

To edit startWebLogic.cmd for WebLogic

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

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

```
set EDX_HOME=C:\Siebel\TSM\estatement
call %EDX_HOME%\config\edx.config.bat
```

6 Add the **classpath** parameter to the **java** command at the end so it uses the paths to the Service Manager classes. For example:

```
%JAVA_HOME%\bin\java %JAVA_VM% %MEM_ARGS% %JAVA_OPTIONS% -classpath
"%CLASSPATH%" -Dweblogic.Name=%SERVER_NAME% -
Dweblogic.ProductionModeEnabled=%PRODUCTION_MODE% -
Djava.security.policy="%WL_HOME%\server\lib\weblogic.policy" weblogic.Server
```

7 Save and close `startWebLogic.cmd`.

Windows Services for Service Manager

Setting Up a WebLogic Server Instance as a Windows Service

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

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

Setting Up the Platform Services Scheduler as a Windows Service

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

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

To install the Scheduler as a Windows Service

1 Navigate to the `bin` directory for Platform Services, or `%EDX_HOME%\bin`.

2 Open the Scheduler template file `SCH.txt` and modify the Java classpath to reflect your active Java environment. For example:

```
classpath=C:\bea\jdk141_05\lib\tools.jar;c:\bea\wlserver\lib\weblogic.jar;c:
\Siebel\TSM\estatement\lib\edx_client.jar;c:\Siebel\TSM\estatement\lib\edx_c
ommon.jar
```

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

3 Confirm that the following line of code is present in the file for your host and port:

```
-Djava.naming.provider.url=t3://localhost:7001
```

4 If you want the Scheduler to log information to a file rather than to the console, add the following value in `SCH.txt`:

```
-Dcom.edocs.pwc.debug=true scheduler_logfile_name
```

5 Confirm that all the directory references in `SCH.txt` are correct.

6 Save and close `SCH.txt`.

7 Open a command prompt window, and then change directory to `%EDX_HOME%\bin`. Use the `schedulersvc` command to install the Scheduler as a Windows Service, for example:

```
C:\> schedulersvc -install C:\Siebel\TSM\estatement\bin\SCH.txt
```

8 If the Scheduler service is installed successfully, a confirmation message appears.

Configuring Payment

To update the Payment and Service Manager configuration files

1 Change your working directory to *%PAYMENT_HOME%\config*, for example:

```
cd \Siebel\TSM\Payment\config
```

2 If you are not using the default Service Manager and Payment directories, then edit the *edx_payment.config* file, and correct the entry that defines PAYMENT_HOME.

3 Copy the updated *edx_payment.config* file to *%EDX_HOME%\config*.

JTA Timeout Configuration

If pmtCheckSubmit will process a large number of checks, the JTA timeout value in WebLogic must be increased to keep the connection open long enough to process all the checks. For every 1500 checks to be processed, the timeout value should be increased by 30 seconds.

The JTA timeout value allows the system to recover a "lost" transaction. Your application code should handle commit and rollback of database transactions. But, if there is a programming error such that transactions are not committed or rolled back, a timeout provides a way for the application server to release those transactions.

The default timeout (30 seconds) is fine for web-based transactions, but it is too small for batch based transactions.

To configure the JTA Timeout setting

1. Enter the URL to log on to the WebLogic console
2. Click on the **JTA** in the left window.
3. Change the Timeout Seconds field to the multiple of 30 seconds that you require, and click the **Apply** button.

Configuring Support for VeriSign Processing

If you are going to use a VeriSign credit card gateway, then you must edit the classpath in your application server startup script, and configure your java security file.

To edit the classpath in the application server startup script

Edit the startup script for your application server to add *jsse.jar*, *jnet.jar* and *jcert.jar*. These files are located in the lib subdirectory of *%CSM_HOME%\payment*.

These JSEE JAR files must be added to the classpath **before** the WebLogic JAR files. For example,

```
set
CLASSPATH=<JSSE_PATH>\jsse.jar;<JSSE_PATH>\jcert.jar;<JSSE_PATH>\jnet.jar;%CLAS
SPATH%
```

Where <JSSE_PATH> is the path to the JSEE JAR files.

Configuring java security

Modify the *java.security* file to configure Java to use the JSSE, which is located in *%CSM_HOME%\payment\lib* to add the following entries:

```
security.provider.1=sun.security.provider.Sun
security.provider.2=com.sun.net.ssl.internal.ssl.Provider
```

If the file already uses the numbers shown above, then use the next available numbers that maintain a contiguous sequence.

CAUTION: Be sure that you edit the correct java security file. Many installations have more than one JDK installed. Edit the one that the application server uses.

Implementing Payment

After configuring the database and the application server for Payment, the following steps are required to implement online bill payment:

1. Configure a payment gateway for online check and/or credit card processing.
2. Enroll customers for online bill viewing and payment.
3. Set up Payment jobs to process payments and optionally send reminders.

See the *Siebel Service Manager Administration Guide* for information about configuring a payment gateway, enrolling customers and other operational issues.

Troubleshooting Tips for WebLogic on Windows

If the WebLogic service is unable to find *edx_load.config.bat* at startup

Add the path `-extrapath:C:\bea\weblogicXX\server\bin;%PATH%` to `installNtService.cmd` (substituting your drive where necessary). For example

```
"C:\bea\weblogic81\server\bin\beasvc" -install -svcname:myserver
-javahome:"%JAVA_HOME%" -execdir:"C:\bea\weblogic81"
-extrapath:"C:\bea\weblogic81\server\bin;%PATH%" -cmdline:%CMDLINE%
-password:lovelyday
```

If you encounter problems when running customized Web pages

1. From your domain in the WebLogic Server Console, select **`Servers`** and **`myserver`** A tabbed dialog with the name of your server appears in the right pane.
2. On the Configuration/Compilers tab, change the Java Compiler value from the default **`javac`** to the location of **`javac`** in the JDK installed **with WebLogic Server**.
3. Restart WebLogic Server.

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.

Editing Properties

On the application server, you must edit hibernate.properties, and hierarchy.hibernate.properties found in CSM_HOME/config for SQL Server:

1. Comment the hibernate.dialect property shown for Oracle.
2. Uncomment the hibernate.dialect property shown for MSSQL.

See the directions inside the properties files for details.

You may need to edit the following files in TSM/config:

app-config.properties

If you are installing a system which contains large hierarchies (that is, many accounts and/or service agreements per company), you can tune the performance of your system by adjusting these settings:

```
#Threshold levels for hierarchy
#maximum size for CCM drop-down menu
tbm.ccm.dropDown.threshold=XX

#number of records per page in paged screens
ps.riconfig.MAX_ENTRIES=XX

#display tree threshold
displayTreeThreshold=XX

#large enterprise user threshold
largeEnterpriseUserThreshold=XX
```

For more information, please refer to the *Developer's Guide*.

hibernate.properties and hierarchy.hibernate.properties

If you are installing a system which contains large hierarchies (that is, many accounts and/or service agreements per company), you can tune the performance of your system by adjusting this setting:

```
hibernate.querythreshold=XX
```

For more information, please refer to the *Developer's Guide*.

log4j.xml

Edit the following line and change the log file path as necessary:

```
<param name="File" value="c:\Siebel\TSM\estatement\logs\log4j.log"\>
```

edocs.tsm.properties

Check that the following property is set for your needs:

edocs.tsm.bizaction.ui_mode=false

This property indicates whether or not the application server is started in UI-Mode. By default, the property is false, which indicates normal (production mode) operation

tsm.hibernate.properties

Check that the database connection parameters match your settings. If you used settings other than the default, you will need to edit the hibernate.connection.url, hibernate.connection.username, and/or hibernate.connection.password parameters.

tsm.xma.xml, sm.xma.xml

Check, and if necessary, replace the following properties:

t3://localhost:7001

t3://localhost

Here, localhost is the server name and 7001 is the port. Please update above based on your server name and WebLogic port number.

sematree.ecs.properties

This contains Oracle specific property settings by default. Make the following changes for MSSQL server.

If you run the application using Oracle, no change is required.

#

If a different RDBMS is used, please adjust the native dialect

property accordingly. For example, for SQLServer, uncomment the

SQLServer native dialect setting and comment Oracle's native dialect setting.

##

Common properties

##

#

Cache Implementation.

#

sematree.ecs.cache.factory_impl=com.sematree.cachewrap.impl.CacheFactoryOSCache

sematree.ecs.sequence.native=true

sematree.ecs.sequence.blocksize=777

Override this default parameter if you want to use a different stored procedure

The default stored procedure is "sp_ecs_getNextId"

sematree.ecs.sequence.native.stored_procedure=sp_ecs_getNextId

##

Oracle native dialect

##

#sematree.ecs.sequence.native.dialect=oracle

##

SQLServer native dialect If using SQLServer, uncomment this and comment other dialect settings

##

sematree.ecs.sequence.native.dialect=mssqlserver

6 Configuring Java Resources for 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 Service Manager.

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 Service Manager.

Configuring Java Database Connectivity (JDBC) for Service Manager

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

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 Value
APP_PORT	app server port	`7001`
ADMIN_PORT	app server admin port	`7002`
JAVA_HOME	Java home directory	`%WLHOME%\jdk141_05`

About JDBC Connections for Service Manager

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.

Service Manager requires three sets of **JDBC Connection Pools** and related **JDBC Data Sources**:

- **edxAdmin** supports the Command Center through the Service Manager web application
- **edxLogger** supports Service Manager logging through the Service Manager web application
- **edxUser** supports user data retrieval through custom web applications

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

CAUTION: **`edxAdmin`** connection pools support concurrency for scheduling multiple jobs. Tuning **`edxAdmin`** connection capacity and threads can improve Service Manager email performance.

Configuring JDBC Connections for WebLogic

You must create three sets of JDBC connection pools and three sets of transaction datasources. Their names are specific to Siebel across all platforms, but JDBC properties vary by both application server and database server.

You must enter the same information six times: one connection pool and one 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 datasource and its properties maps to the connection pool of the same name.

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

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

To configure JDBC Connections for WebLogic

1. Create a JDBC Connection Pool each for **`edxAdmin`**, **`edxLogger`**, and **`edxUser`**. Use the appropriate JDBC values for your database server.

2 Create a JDBC Data Source each for **edxAdmin**, **edxLogger**, and **edxUser**. Use the appropriate JDBC values for your database server.

3 Review your connections. Each data source should target the specified connection pool.

4 When you are finished, proceed to the next section to configure Java Messaging Service (JMS) for Service Manager.

JDBC – WebLogic for Windows

TIP: You may save time by **cloning** additional Java resources. Right-click a resource and select **Clone <name>**, then change the resource name and properties as required.

JDBC Connection Pools for WebLogic

Create three JDBC Connection Pools, using WebLogic Server documentation at http://edocs.bea.com. Use the same **Properties** for all three connection pools. Make sure to deploy them to the server you are configuring for Service Manager (in the examples of this guide, the default **myserver**).

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

- **Database type** = MS SQL Server
- **Database Drive**r = Other

For each of three connection pools, using the following names and properties:

Pool 1: Admin	Pool 2: User	Pool 3: Logger
edxAdminConnectionPool	**edxUserConnectionPool**	**edxLoggerConnectionPool**

General Tab	
URL	**jdbc:inetpool:inetdae7://localhost:1433**
Driver Classname	**com.inet.pool.PoolDriver**
Database User	Enter the database user name. This document uses **edx_dba**.
Password	Enter the password for the database user. This document uses **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**

Connections Tab	
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	**job**

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

JDBC Data Sources

Create the following transaction data sources, using WebLogic Server documentation at http://edocs.bea.com.

	Datasource 1: Admin	Datasource 2: User	Datasource 3: Logger
Name	**edxAdminDataSource**	**edxUserDataSource**	**edxLoggerDataSource**
JNDI Name	**edx.databasePool**	**edx.user.databasePool**	**edx.logger.databasePool**
Pool Name	**edxAdminConnectionPool**	**edxUserConnectionPool**	**edxLoggerConnectionPool**

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

Name:	**ecs**
JNDI Name:	**jdbc.ecs**
Pool Name:	**edxUserConnectionPool**
Row Prefetch Enabled:	**false**
Enable Two Phase Commit:	**false**
Stream Chunk Size:	**256**
Row Prefetch Size:	**48**

Name:	**ecs_serial_trans**
JNDI Name:	**jdbc.ecs_serial_trans**
Pool Name:	**edxUserConnectionPool**
Row Prefetch Enabled:	**false**
Enable Two Phase Commit:	**false**
Stream Chunk Size:	**256**
Row Prefetch Size:	**48**

Name:	**ecs_security**
JNDI Name:	**jdbc.ecs_security**
Pool Name:	**edxUserConnectionPool**
Row Prefetch Enabled:	**false**
Enable Two Phase Commit:	**false**
Stream Chunk Size:	**256**
Row Prefetch Size:	**48**

On the **Targets** tab, select the server that will use this Data Source.

Configuring Java Messaging Service (JMS) for Service Manager

After you have successfully configured JDBC Connections, you must now configure Java Messaging Service (JMS) on the application server. Platform Services requires three sets of JMS resources:

- **edxAnnotation** supports Line Item Dispute and Annotation features.
- **edxDispute** supports Line Item Dispute and Annotation features.
- **edxLogger** supports logging through the web application. It requires **five** JMS consumers and session pools.

  TIP: If your web application does not implement Line Item Dispute and Annotation, you need only configure JMS resources for **`edxLogger`**.

About JMS Resources for Service Manager

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

- **JMS Connection Factories** are data objects that enable Java Messaging Service (JMS) clients to create JMS connections. You define and configure one or more connection factories to create connections with predefined attributes. WebLogic Server adds the connection factories to the JNDI space during startup, and each J2EE web application retrieves a connection factory using the JNDI on the application server.
- **JMS Stores** store persistent messages in a database accessed through a designated JDBC connection pool. The JMS database can be any database that is accessible through a WebLogic-supported JDBC driver. When creating a JMS Store, you must define the name of the **backing store**, and the **JDBC connection pool** and **database table name prefix** for use with multiple instances.
- **JMS Servers** manage connections and message requests on behalf of clients.
- **JMS Topics** can be one of two destinations that you can configure for a JMS server. The other destination is a JMS queue. WebLogic Server allows you to configure one or more destinations for the JMS server. You can configure destinations explicitly or with a **destination template** (useful for multiple destinations with similar attribute values).
- **JMS Session Pools** allow a JMS listener (called a **Consumer** in WebLogic) to have multiple threads that improve performance under heavy load. Each JMS consumer requires its own session pool.

Configuring JMS Resources

You must enter very similar information many times: one set of JMS resources each for annotation and dispute, and FIVE sets for logging. Make sure you have chosen the correct properties for the resource you are creating, and that each resource maps to others of the **same name**.

For general information about configuring Java resources for WebLogic, see WebLogic Server documentation at http://edocs.bea.com.

TIP: If your web application does not use Line Item Dispute and Annotation, you do not need to configure those JMS resources.

TIP: You may **clone** additional Java resources. Right-click a resource and select **Clone <name>**, then change the resource name and properties as required

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

	1: Annotation	2: Dispute	3: Logger
Name	**edxAnnotationTCF**	**edxDisputeTCF**	**edxLoggerTCF**
JNDI Name	**edx/tcf/annotate**	**edx/tcf/dispute**	**edx/tcf/log**

Name	JNDI Name	Description
edxServiceManagerTCF	**edx/tcf/serviceManager**	For asynchronous messaging

On the Targets tab, select the Servers that will use each JMS Connection Factory.

JMS (JDBC) Stores

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

Name (of JMS Store)	**Connection Pool**
edxAnnotationStore	**edxUserConnectionPool**
edxDisputeStore	**edxUserConnectionPool**
edxLoggerStore	**edxLoggerConnectionPool**

Name (of JMS Store)	**Connection Pool**
edxServiceManagerStore	**edxUserConnectionPool**

JMS Servers

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

	1: Annotation	2: Dispute	3: Logger
Name	**edxAnnotationServer**	**edxDisputeServer**	**edxLoggerServer**
(Persistent) Store	**edxAnnotationStore**	**edxDisputeStore**	**edxLoggerStore**

Name	(Persistent) Store
edxServiceManagerServer	**edxServiceManagerStore**

Targets Tab	
Targets-Server	[select **myserver** from drop-down menu]

JMS Topics

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

	1: Annotation	2: Dispute	C3: Logger
Name	**edxAnnotationTopic**	**edxDisputeTopic**	**edxLoggerTopic**
JNDI Name	**edx/jms/annotate**	**edx/jms/dispute**	**edx/jms/log**
Enable Store	**True**		

JMS Session Pools and Consumers for Annotation and Dispute

TIP: If your deployment does not use annotation and dispute, you can skip to configuring session pools and consumers for Logger.

Create one pair of JMS Session Pools and Consumers each for Annotation and Dispute, using WebLogic Server documentation at http://edocs.bea.com. Set **Acknowledge Mode** to **auto** and **Sessions Maximum** to **-1** for all three Session Pools.

Session Pool	1: Annotation	2: Dispute
Name	**edxAnnotationPool**	**edxDisputePool**

Session Pool	1: Annotation	2: Dispute
Connection Factory	**edx/tcf/annotate**	**edx/tcf/dispute**
Listener Class	**com.edocs.services.annotation.Listener**	**com.edocs.services.dispute.Listener**

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

Consumer	1: Annotation	2: Dispute
Name	**edxAnnotationConsumer**	**edxDisputeConsumer**
Messages Maximum	**10**	**10**
Selector	**JMSType='USER'**	**JMSType='USER'**
Destination	**edx/jms/annotate**	**edx/jms/dispute**

JMS Session Pools and Consumers for Logging

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.

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**

Property	Value
Messages Maximum	10
Selector	JMSType='ADM'
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	`edxLoggerMessageLogConsumer`
Messages Maximum	`10`
Selector	`JMSType='MSG'`
Destination	`edx/jms/log`

4) System Activity

JMS session pool- Configuration Tab

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

JMS consumer

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

5) UserActivity

JMS session pool

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

Property	Value
Sessions Maximum	**-1**

JMS consumer

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

Deploying Service Manager

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

- **Application servers**: Deploy the Siebel Platform Services (ear-eStatement.ear) and the Service Manager ears.

The ear files are located at:

Feature	Location	File Name
Self Service CSR	%TSM_HOME%/*J2EEApps/weblogic/*	ear-tbm-csr.ear
Siebel Platform Services	%TSM_HOME%/J2EEApps/weblogic	ear-eStatement.ear
Service Manager and Self-Service Manager	%TSM_HOME%/J2EEApps/weblogic	ear-tbm-b2b.ear

Consult your BEA WebLogic documentation on how to deploy applications.

After successfully deploying the application, you can log into the Command Center:

1. In your browser, point to http://<server name>:<server port>/edocs (where **server name** is your server name and **port** is the WebLogic port for the application). The initial default Administrator ID is **admin** and the Password is **edocs**.

2. Once in the Command Center, change your password. Click the Help button in the Command Center for information changing on passwords.

3. To check Self Service Manager, use the following link, using the name of the application server in place of **localhost**:

http://localhost:7001/tbmb/ and log on to the application. Then select the Self-Service tab.

Starting the Scheduler

You can start the Scheduler from a command line or as a Windows service. Before starting the Scheduler, check the file to make sure it has the correct JAVA_HOME value.

Update wl_scheduler.bat

In **%EDX_HOME%\bin,** edit *wl_scheduler.bat* to set correct value for **JAVA_HOME**.

To start the Scheduler from a Windows command prompt

1. Open a command prompt window and change directory to the **bin** directory of your Service Manager installation, %EDX_HOME%\bin.
2. Run the Scheduler command for WebLogic Server (**wl_scheduler**), host, and port. This example shows the Scheduler command for WebLogic:

   ```
   C:\Siebel\TSM\estatement\bin> wl_scheduler.bat –start –url
   t3://localhost:7001
   ```

3. If the Scheduler starts successfully, a start-up message with the name of the log file appears in the command prompt window. **Do not close this window while** Service Manager **is running,** as closing it will stop the Scheduler. A log file is created in %EDX_HOME%\Logs.
4. You can stop the Scheduler by replacing the -start parameter with the -stop parameter, or simply by closing the command prompt window.

To start the Service Manager Scheduler as a Windows Service

1. Install the Scheduler as a Windows Service if you have not already done so.
2. Start WebLogic Server if it is not already running.
3. From the Start menu, select **Settings**>**Control Panel**.
4. Double-click **Administrative Tools**, then double-click **Services**.
5. Right-click **Scheduler Service** and select **Start**. You may also click the Start icon.
6. To start the Scheduler automatically at startup, right-click the service and select **Properties**. From the Startup Type menu, select **Automatic**.

7 Installing Service Manager Synchronizer

This section describes the CSM Synchronizer installation. The Service Manager Synchronizer loads data into the database using the Service Manager process.

System Software Requirements

CSM Synchronizer requires the software components listed below to be installed and configured.

- JDK 1.4 or greater (use the JDK that comes with WebLogic). If you have multiple versions of the JDK installed, make sure that JDK 1.4 is first in the PATH or configure the Service Manager Synchronizer run script so that it is first in the PATH.

Installation Checklist

To install Service Manager

1. Install required Self Service Manager software
   - Self Service Manager
   - Self Service Manager database
   - Unzip %TSM_HOME%/TSM-zips/*ETL.zip*. This will create the directory structure shown below.

```
ETL/config/
ETL/config/castor/
ETL/config/castor/dtd/
ETL/config/castor/mapping/
ETL/config/etl/
ETL/data/
ETL/ddl/
ETL/lib/
ETL/logs/
ETL/buildrun.xml
```

Directory Name	Description
config	This directory contains the two sub-directories shown below. castor etl The castor directory contains files that are required to map data from the Service Manager Synchronizer load files to Self Service Manager. The ETL load file DTD descriptions are also contained in the DTD directory. The *etl* directory contains the default *tsmproperties.xml* file that provides configuration information for Service Manager Synchronizer. This file should be examined and changed to meet the needs of the installation.
data	The data directory contains sample data files used by Service Manager Synchronizer as defined in the standard tsmproperties.xml file.
lib	The lib directory contains all required JAR files needed to run Service Manager Synchronizer. The default configuration is designed to support Oracle.
logs	The logs directory is initially empty. It will contain error file output.
buildrun.xml	This file contains the commands used to run Service Manager Synchronizer.

2 Update the following files to meet the Self Service Manager database requirements. All these files are found in the ETL/config directory:

hibernate.properties

Change the following parameters to match your environment:

```
#########################################################
#
## Data source information for Oracle jdbc driver
#########################################################
#
#edx.connection.driver_class=oracle.jdbc.OracleDriver
#edx.connection.url=jdbc:oracle:thin:@localhost:1521:edx0

#########################################################
#
## Data source information for SQLServer jdbc driver
#########################################################
#
hibernate.connection.driver_class=com.inet.pool.PoolDriver
hibernate.connection.url=jdbc:inetpool:inetdae7://localhost:1433
hibernate.dialect=net.sf.hibernate.dialect.SQLServerDialect

#########################################################
#
## Database credentials
#########################################################
#
edx.connection.username=edx_dba
edx.connection.password=edx

#########################################################
#
## Hibernate related settings
#########################################################
#
hibernate.dialect=net.sf.hibernate.dialect.Oracle9Dialect

hibernate.show_sql=false
hibernate.schema_auto=false
hibernate.use_outer_join=true
hibernate.max_fetch_depth=3
```

hierarchy.hibernate.properties

Change the following parameters to match your environment:

```
#######################################################
## Data source information for Oracle jdbc driver
#######################################################
#hibernate.connection.driver_class=oracle.jdbc.driver.OracleDriver
#hibernate.connection.url=jdbc:oracle:thin:@localhost:1521:edx0
#hibernate.dialect=net.sf.hibernate.dialect.OracleDialect

#######################################################
## Data source information for SQLServer jdbc driver
#######################################################
hibernate.connection.driver_class=com.inet.pool.PoolDriver
hibernate.connection.url=jdbc:inetpool:inetdae7://localhost:1433
hibernate.dialect=net.sf.hibernate.dialect.SQLServerDialect

#######################################################
## Database credentials
#######################################################
hibernate.connection.username=edx_dba
hibernate.connection.password=edx

#hibernate.cglib.use_reflection_optimizer=true
#hibernate.cache.provider_class=net.sf.hibernate.cache.HashtableCacheProv
ider
#hibernate.show_sql=false
#hibernate.schema_auto=false
hibernate.fetch_size=50

# This property has nothing to do with Hibernate. Needs to be moved
elsewhere
# Eventually
hibernate.querythreshold=500

# This property has nothing to do with Hibernate. Needs to be moved
elsewhere
# Eventually
hibernate.largeResultSetFetchSize=2000

# This property has nothing to do with Hibernate. Needs to be moved
elsewhere
# Eventually.
#
# Statement date format in EDX_TSM_AC_STATEMENTS and
EDX_BSL_ACCT_STATEMENTS tables
#
hibernate.complexqueries.statement_date_format=MM/dd/yyyy

# This property has nothing to do with Hibernate. Needs to be moved
elsewhere
# Eventually.
#
# Payable amount format in EDX_TSM_AC_STATEMENTS and
```

```
EDX_BSL_ACCT_STATEMENTS tables
#
hibernate.complexqueries.payable_amount_format=999G999G999G999D99
```

Persistence.xma.xml

```
You must update the database username, password, and sid in the
persistence.xma.xml file:
```

```
<bean id="edxUserDataSource" class="org.enhydra.jdbc.pool.StandardXAPoolDataSource"
destroy-method="shutdown">
    <property name="dataSource">
        <bean class="org.enhydra.jdbc.standard.StandardXADataSource" destroy-
method="shutdown">
            <property name="transactionManager">
                <ref local="jotm"/>
            </property>
            <property name="driverName">
                <value>com.inet.pool.PoolDriver</value>
            </property>
            <property name="url">
          <value>jdbc:inetpool:inetdae7://localhost:1433</value>
            </property>
        </bean>
    </property>
    <property name="maxSize">
        <value>25</value>
    </property>
    <property name="minSize">
        <value>10</value>
    </property>
    <property name="user">
        <value>edx_dba</value>
    </property>
    <property name="password">
        <value>edx</value>
    </property>
</bean>
```

3 Set ANT_HOME, JAVA_HOME environment variables and PATH into the Shell Command:

```
SET ANT_HOME= ant home
SET JAVA_HOME= java home
SET PATH = %JAVA_HOME%\bin;%ANT_HOME%\bin;%PATH%
```

4 Run ant to update the configuration:

```
ant -f buildrun.xml config-update
```

Note, **`ant -f buildrun.xml`** displays the available options in the Service Manager Synchronizer.

5 Run

```
ant -f buildrun.xml etl-ivp
```

to validate installation/configuration. Check the output for any errors or failures, and correct the configuration to resolve these problems if they exist. If you have made changes to the configuration, repeat the config-update and etl-ivp steps and check again.

6 Run

```
ant -f buildrun.xml etl-run
```

on a clean database to execute Service Manager Synchronizer. Again, check the output for any errors or failures, and correct the configuration to resolve these problems if they exist. If you have made changes to the configuration, repeat the previous steps and check again.

7 You can verify that Service Manager Synchronizer has run successfully by checking that there is data in the following tables:

EDX_TSM_DEVICE_TYPE
EDX_TSM_RATEPLAN
EDX_TSM_RATEPLAN_GROUP
EDX_TSM_RP_FEATURE

The EDX_TSM_SERVICE_AGRMNT table should still have 76 rows (though sometimes there is a blank row mixed in), as it did after running the HierarchySynchronizer job, but each row should now have name, address and rate plan information added.

You can also see summary information at the command prompt after the ETL job runs.

8 After Service Manager Synchronizer runs successfully, log back on to the CSR application and set up your company and admin user(s). The sample data will be under the __default__company.

9 Log into the Service Manager application and verify that the appropriate rate plans, rate plan groups, etc. are available.

Configuring for Performance

The following list shows how to configure and tune the system for the best Synchronizer performance.

1 Service agreement cache in config/ehcache.xml file

The main thing is the cache size. The default value is 300000. This will consume about 1.7 – 1.8 GB memory. The application uses no more than 500MB memory excluding the cache. This should give you a good idea on the memory setting. If you can't allocate over 2GB memory for the application, please reduce this number and run ant –f buildrun.xml config-update.

2 JVM memory usage. At least 2G is recommended. Also use JVM in the server mode. This is configured in the buildrun.xml file. See the sample section from buildrun.xml below on configuring the memory.

3 bulkLoadBatchSize system property. We load all service agreements from the database. Because of memory constraints, we would like to get them in batches. This controls the size of the batch. Any value between 3000 to 10000 should be reasonable. Optional with the default value of 5000.

4 commitSize system property. This control the size of the service agreement operation transactions. There are two reasons against using too large a value. One is that the cost of commit will offset the reduction of the number commit calls. There is a sweet spot. Second, if the quality of the data is not good, there will be lots of rollbacks and we will be doing more work than necessary. In this case, it helps to reduce the size of the commit. On the other hand, if this does occur, the process of generating the input file needs to be looked into. Optional with the default value of 100.

5 etl-temp system property. This allows us to generate temp files in a faster partition. Optional. Will use the system temp directory is none is specified.

Please see the file buildrun.xml for an example how items 2, 3, 4 and 5 are specified.

```
  <!-- ================================================== -->
  <!-- run -run target                                    -->
  <!-- ================================================== -->

  <!-- It's possible to use 3-4 g of memory, for example, changing 2g to 3g.
This improves the performance. -->
  <target name="etl-run" depends="init">
    <java  classname="com.edocs.application.tsm.etl.process.ETLMain" args="-
run" fork="yes" maxmemory="2g">
  <!-- use the server jvm so that it's possible to use bigger heap -->
  <jvmarg value="-server"/>
  <jvmarg value="-DbulkLoadBatchSize=5000"/>
  <jvmarg value="-DcommitSize=100"/>
  <!-- This is commented. If you have a faster partition, such as a ramdisk
based, use that -->
  <!--jvmarg value="-Detl-temp=/tmp/tsm-etl"/-->

     <jvmarg value="-DInitFile=${ETLInitFile}"/>
     <classpath>
       <path refid="run.classpath" />
     </classpath>
    </java>
  </target>
```

Where to Go From Here

Once you have successfully configured the WebLogic Server and deployed the Service Manager application, you can proceed to deploying any custom J2EE applications. This requires customizing each web application's deployment code for your environment and platform. For details, see WebLogic Server documentation at http://edocs.bea.com and *Deploying and Customizing J2EE Applications.*

8 Appendix A: Uninstalling Service Manager

Uninstalling Service Manager

You can uninstall and remove Service Manager components, deployed J2EE applications, and Windows services using the Service Manager Uninstaller.

Uninstall Service Manager 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 Service Manager components, you must:

- Stop your application server.
- Stop your database instance.
- Stop your database server.

To uninstall Service Manager

1. Navigate to the **`Uninstall`** folder of your Service Manager home directory, **`$`***CSM*_**`HOME.`**
2. Windows users may run the command-line script **`Uninstall`**_*TSM*`.exe`, or select Start Menu>Programs>Siebel>Uninstall Service Manager.
3. Click **`Uninstall`**. A second uninstall screen appears showing Service Manager 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 Service Manager 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.

To remove the Billing Manager database for MSSQL

1. Log into SQL Query Analyzer as CSM db user and execute these scripts in sequence:

   drop_bsl_schema.sql

   drop_hierarchy_schema.sql

To remove the Case Management database for MSSQL

1 Log into SQL Query Analyzer as CSM db user and execute the following script:

casemgmt_drop.sql

Uninstalling Windows Services

This information applies to all Windows platforms

Uninstalling WebLogic Server as a Windows Service

Uninstalling this Windows Service will require that you start and stop WebLogic Server from the command line or the administrative console.

To uninstall WebLogic Server as a Windows Service

1 Open a Command Prompt window, and change directory to the `domain` directory of your application server home directory.

```
C:\> cd %WL_HOME%\config\mydomain
```

2 Uninstall WebLogic Server as a Windows Service with the `uninstallNtService` command:

```
C:\> uninstallNtService.cmd
```

You do not have to specify the WebLogic service name on the command line, because it is named in `uninstallNtService.com`.

Uninstalling the Service Manager Scheduler as a Windows Service

Uninstalling this Windows Service will require that you start and stop the Scheduler from the command line.

To uninstall the Scheduler as a Windows Service

1. Open a command prompt window, and change directory to the `\bin` directory of your Service Manager home directory.

   ```
   C:\> cd %CSM_HOME%\bin
   ```

2. Uninstall the Scheduler as a Windows Service with the `–remove` command:

   ```
   C:\> Schedulersvc –remove
   ```

Uninstalling Payment

Removing the Payment Database

Follow the steps below to remove the Payment database tables and indexes. The process involves running an SQL script as the as the owner of the database. You should be aware that this procedure **completely** removes the payment database elements and should be used with care.

CAUTION: If you wish to remove the Payment database tables and indexes, you must do that before removing the Payment database package.

To remove the Payment database for MSSQL

1. Change your working directory to the *%PAYMENT_HOME\db\mssql* directory.
2. Run the file *drop_payment_db.bat*.

Index