Installation Guide for Siebel Service Manager

For Sun Solaris Operating Environment,
IBM WebSphere Server, and Oracle® Database

Version 5.1.1
Data Published: 12.13.2006

ORACLE

Contents

1 Preface

About This Guide

This guide is intended for system administrators and other IT professionals and describes how to install Service Manager, configure the third-party platforms that support the Service Manager production environment, and deploy Service Manager J2EE Web applications.

It assumes in-depth understanding of and practical experience with system administration responsibilities, listed here.

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.

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

Related Documentation

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

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

2 Getting Started

Before You Install; Preparing Your Platform

Before installing Service Manager, 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. See your server documentation.
- Start and test your application server. 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.
- Install Xwindows software on your database, application servers, and Windows machine (for Tools) to support the InstallAnywhere GUI.
- Do not install Service Manager on the same server as the Oracle Communications Billing Analytics product.

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 Siebel 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.

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 requires you to:

1. Define database server environment variables.
2. Create and configure the Service Manager database with edx_admin.sh.
3. Connect to your Service Manager database before 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.

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's Platform Services and Service Manager (Solaris/Oracle/WebSphere)

This guide assumes you are installing Service Manager on a Solaris operating system, Oracle database, and WebSphere application server.

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

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

OPERATING SYSTEM

- Sun Solaris 9. This release supports the use of the Solaris multi-threaded MT-MALLOC library on Solaris 9 with patch 115697-02.

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)
- Sun SPARC platform
- Swap space 1GB per CPU (2 GB recommended)
- RAM 1GB per CPU (2 GB recommended)

JAVA/C++

- IBM Java 2 SDK Standard Edition 1.4.1 or higher (use the Java version packaged with WebSphere)
- Sun C++ runtime packages **`SUNWesu`** and **`SUNWlibC`**

SUPPORTED DATABASE SERVERS

New installation of Service Manager

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

SUPPORTED APPLICATION SERVERS

- IBM WebSphere 5.1.1.9 plus the iFix for PK31999 available from the following IBM Web site: http://www-1.ibm.com/support/docview.wss?uid=swg24013681

SUPPORTED BROWSERS

- Netscape Navigator 7.0 and higher
- Microsoft Internet Explorer 6.0 and higher (on networked PC)

- Firefox 1.0.1

Siebel Tools (Windows)

OPERATING SYSTEM

- Microsoft Windows 2000/Server SP4

3 Installing Service Manager

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.

UNIX Permissions for Installation

This section applies to all UNIX platforms. You must have **root** privilege on each server to install and uninstall Service Manager components.

- Default: **`root:other`**
- Example: **`edxadmin:edxadmin`**

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

Caution: Siebel does not recommend administering Service Manager with the user and group `nobody:nobody`

Installing Service Manager Using InstallAnywhere

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 UNIX 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**.

You have the option to install Service Manager on UNIX in either of two installation modes:

- GUI Mode (default)
- Console Mode

The instructions in this chapter assume you are using GUI mode. Console Mode is an interactive character-based installation that prompts you to respond to several installation questions. If you prefer to use Console mode, see “Installing Service Manager in Console Mode for UNIX” on page 13 for details.

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 both at once.

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

1. Ask your system administrator for the user and group name of the application server owner, such as **edxadmin:edxadmin**.
2. Set and export the DISPLAY environment variable for your machine, for example **DISPLAY=localhost:0.0 export DISPLAY.**
3. Obtain and locate the **InstallAnywhere** installer.
4. Launch InstallAnywhere by typing **TSMins.bin.**
5. INTRODUCTION: InstallAnywhere recommends that you quit all programs before installing.
6. LICENSE AGREEMENT: Review the License Agreement and click “**I accept the terms of the License Agreement**” to accept the terms.
7. ENTER SERIAL NUMBER provided when you purchased Service Manager.
8. OWNER OF WEB APPLICATION SERVER for example **edxadmin**.
9. GROUP OF WEB APPLICATION SERVER for example **edxadmin.**
10. 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.
11. CHOOSE PRODUCT FEATURES: Choose which feature you want to install first, or click Custom and choose to install Siebel Service Manager and Siebel Platform services at the same time.

    Service Manager - Installs all components for the selected Service Manager edition.

    Platform Services (Default) - Installs the components for Siebel Core Services only, including Billing, Hierarchy, Payment, and Reporting.

    Siebel Tools - Installs DefTool and Composer (on a Windows machine only).

12. CHOOSE SHORTCUT FOLDER: Click **Next** to install Service Manager in the Siebel program group.
13. PRE-INSTALLATION SUMMARY: Review the screen to confirm your product and version, install folder, product components, and disk space required and available, then click **Install**.

    Note that the Service Manager install folder is also referred to as TSM_HOME (default is /opt/Siebel/TSM). EDX_HOME refers to the platform services directory, $ TSM_HOME/estatement.

    InstallAnywhere sets up a directory hierarchy on each server and copies files to the appropriate directories.
14. INSTALL COMPLETE: If installation is successful, you see a congratulatory message. Click **Done**.
15. Repeat the installation for other Service Manager servers on your network as necessary.

Tools (Windows)

To install Siebel Tools on a Windows 2000-based machine on your network:

1. Obtain and locate the InstallAnywhere installer.
2. Launch InstallAnywhere by typing TSMins.bin.
3. Follow the steps in the procedure above for installing Service Manager, selecting the Siebel Tools feature to install.

Installing Service Manager in Console Mode for UNIX

Console Mode is an interactive character-based installation which prompts you to respond to several installation questions. Follow the procedures in this section if you prefer to install in Console Mode.

To install Service Manager in Console Mode for UNIX:

1. Give full permission to TSMins.bin:

   ```
   chmod 777 TSMins.bin
   ```
2. Navigate to the InstallAnywhere directory for your platform and run the command to start InstallAnywhere, using the **`–i console`** flag. For example:

   ```
   ./TSMins.bin –i console
   ```

InstallAnywhere displays the banner:

```
Preparing CONSOLE Mode Installation...
```

3. Enter the Serial Number.
4. Enter owner of the Web application server files.
5. Enter group of the Web application server files.
6. Enter application install folder.
7. Choose to install **`Platform Services`**. Respond to each prompt to proceed to the next step as directed. If you want to change something on a previous step, type **`back`**.
8. Run InstallAnywhere again (Steps 2 – 7) and install your edition of Service Manager.

   A successful installation displays a congratulatory message indicating that the software has been installed to the directory you specified as the Siebel home directory **(**`$TSM_HOME`**)** in response to the installation prompts.

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 (TSM_HOME): /opt/Siebel/TSM/
- Platform Services (EDX_HOME): /opt/Siebel/TSM/estatement
- Tools: \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.

By default, Payment is installed in /opt/Siebel/TSM/Payment. This guide refers to the default installation path for Payment as PAYMENT_HOME, or TSM_HOME/Payment.

Where to Find Database Components

/opt/Siebel/TSM/estatement/db and **/opt/Siebel/TSM/payment/db** contain platform-specific subdirectories for database creation and configuration.

/opt/Siebel/TSM/db contains TSM subdirectories for database creation and configuration

Where to Find Application Server Components

/opt/Siebel/TSM/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

`/opt/Siebel/TSM/estatement/AppProfiles` stores information on each new Service Manager application created in the Command Center.

`/opt/Siebel/TSM/estatement/Input` is the default input directory used by each Command Center job. **`/opt/Siebel/TSM/estatement/Data`** stores data processed by the Siebel Command Center. **`/opt/Siebel/TSM/estatement/Output`** stores the output of jobs.

An additional directory, **`/opt/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** Service Manager 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 Service Manager

Caution: The installation and configuration examples 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.

UNIX Permissions for Your Database Server

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

This guide uses the example username and password edx_dba:edx as the owner and group for the Siebel database user. This is the user for your database instance. This guide also uses the example database instance name edx0.

Siebel recommends that you install Service Manager database components with the default owner and group for your platform. After installation, change the user and group ownership of Service Manager database server components to that of the DB Admin user.

DATABASE	DB ADMIN USER	DB USER
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 EDX_HOME directory and all subdirectories to the Service Manager instance owner.

   ```
   chown -R edxadmin:edxadmin /opt/Siebel/TSM/estatement
   ```

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

   ```
   chown -R oracle:dba /opt/Siebel/TSM/estatement/db
   ```

4. Switch user to your database instance owner and run **edx_admin.sh** to create the Service Manager database with your new owner. 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 Service Manager 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 Service Manager, you must define **environment variables for your database server**.

Example: Setting 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	SOLARIS
EDX_HOME	Platform services home path	**/opt/Siebel/TSM/estatement**
ORACLE_BASE	Mount point base path	**apps/oracle**
ORACLE_HOME	Data File Path	**$ORACLE_BASE/product/9.2.0**
LD_LIBRARY_PATH	Shared Library Path	**$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	**edx**
ORACLE_SID	Database instance name	**edx0**
ORACLE_USER	Database user name	**edx_dba**
ORACLE_DBALIAS	Database alias	**edx.db**

Editing Oracle Initialization Parameters

Before running the Oracle database install script for Billing Manager, follow these steps to add the following parameters to the init<oraclesid>.ora file, located in $ORACLE_HOME/dbs.

1. In init<oraclesid>.ora, set the following database parameters:

   QUERY_REWRITE_INTEGRITY = TRUSTED

   QUERY_REWRITE_ENABLED = TRUE

   COMPATIBLE = 8.1.0.0.0 (or higher)

2. Shut down and restart the database.

3. Connect as sysdba and execute the following command:

   GRANT CREATE ANY INDEX, QUERY REWRITE TO <tbmuser>

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, Service Manager 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. We recommend choosing the range partition on the **`Z_DOC_DATE`** column.

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

5 Configuring Service Manager for Oracle

Configuring a New Oracle Database for UNIX

This section applies to all UNIX platforms. 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 Service Manager database, you run the database configuration shell script edx_admin.sh.

Tip: Database clustering is handled by your application server and not by Service Manager. Consult your Siebel Technical or Professional Services representative for clustered installations.

About edx_admin.sh for Oracle

To create and configure the Service Manager production database, you run the **`edx_admin.sh`** script for database configuration. If you have to abort database setup before it completes successfully, see “If the Database Configuration Fails” on page 25.

Before running `edx_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 Service Manager 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.

To configure a new Service Manager 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. Set the environment variable for ORACLE_SID, for example:

```
ORACLE_SID=edx0
export ORACLE_SID
```

3. Change directory to your Service Manager database home directory. For example:

```
cd /opt/Siebel/TSM/estatement/db/oracle
```

4. Start the **edx_admin** shell script with the command:

```
./edx_admin.sh
```

The Server Administration Main Menu 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
 [4] Initial Data Population
 [5] Database Version Migration
 [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 Service Manager 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 19 and your database administrator to specify suitable values for your platform.

```
SIGN IN MENU
 ---------------------------------------------
 [1] Enter Database USERNAME...> edx_dba
 [2] Enter Database PASSWORD...> edx
 [3] Enter ORACLE_SID ...> edx0
 [4] Enter the password for SYS user...> change_on_install
```

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 Siebel Server Administration Main Menu.

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

```
Install edocs e-Statement
 ------------------------------------
 [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 Service Manager, 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. Select Option 2, to shutdown the database.
10. Select Option 3, to restart the database.
11. Select Option 4, **Install Application Database I**.

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

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

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

    This option creates the Service 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.

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

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

14. (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 Service Manager home directory, for example

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

15. Select **Return to Previous Menu**. The Main menu appears.

16. Select Option 4, **Initial Data Population**.

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

    The Initial Data Population Menu appears.

    (If this is your first installation of Service Manager, you can ignore Option 5, Database Version Migration. You select this option only if migrating an existing Service Manager database to a newer version.)

17. Select Option 1, **Import Initial Data Set**.

    This option populates the newly defined database with an administrative dataset that supports the Siebel Command Center and related features. No user input is required, although you see several progress messages. If this step is successful, a success message appears.

(Step 2, Export Siebel database data, is used for migration; it is not necessary for new installations because the database is still empty.)

18. Select Option **R** (Return to Previous Menu), and then press **Enter**. The Main menu appears.
19. Select Option **Q** (Quit), and then press **Enter.**

If the 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 using the echo command. For example:

   ```
   echo $ORACLE_SID
   edx0
   ```

3. If the database instance name is incorrect, set it as follows:

   ```
   ORACLE_SID=edx0
   export ORACLE_SID
   ```

4. Shut down the database server.
5. Locate all files associated with the database instance you defined. For example:

   ```
   find . -name ‘*edx*’ –print
   ```

6. Change directory to the root directory for your database and recursively remove the folder and its files. For example:

   ```
   cd ./product/oradata
   rm –rf edx0
   ```

7. Repeat this step for other root directories containing database files for your database instance.

Configuring Oracle Services

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

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

- You need to modify **`listener.ora`** on the **database server** machine**.**

  `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 **`tnsnames.ora`** on the **database client** 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`**.

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 = edx0)
    (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.

```
edx.db =
 (DESCRIPTION =
 (ADDRESS_LIST =
 (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
)
(CONNECT_DATA =
 (SID = edx0)
 )
 )
```

7. Save and close **tnsnames.ora**.
8. Repeat Steps 5-7 if the application server is not also the database server.
9. Stop and restart the Oracle listener with the listener control commands.

   ```
   lsnrctl stop
   lsnrctl start
   ```

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

    Services Summary...
    PLSExtProc has 1 service handler(s)
    edx0 has 1 service handler(s)

    This service handler should match the name you entered for the Oracle SID during database configuration, in this example edx0.

Connecting to Your Oracle Database

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

To connect to an Oracle database for UNIX:

1. Switch user to the DB Admin user.

   ```
   su – oracle
   ```

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

   ```
   sqlplus  edx_dba/edx@edx.db
   ```

   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 edx0
SQL>
```

Creating a New Payment Database

Configuring the Payment database consists of running a script that creates tables and indexes in the Service Manager database that Payment uses.

To run the database creation script for Oracle in UNIX:

1. Recursively change the user and group permissions of your PAYMENT_HOME database directory and all subdirectories to the database instance owner.

   ```
   chown -R oracle:dba /opt/Siebel/TSM/payment/db
   ```

2. Log on as the Oracle user. For example:

   ```
   su - oracle
   ```

3. You may have to set the environment variable for the Oracle SID to the Service Manager Oracle database instance before starting *sqlplus*. For example:

   ```
   export ORACLE_SID=edx0
   ```

4. Change your working directory to the *db/oracle* directory under *$PAYMENT_HOME*. For example:

   ```
   cd /opt/Siebel/TSM/Payment/db/oracle
   ```

5. Run the Payment database configuration script *payment_admin.sh*.

6. The Payment database installation menu displays:

   ```
   Siebel Payment Server Administration Main Menu Version 1.0
   ----------------------------------------------------------
     [1] Sign in Menu
     [2] Install Application Database I
     [3] Install Application Database II
     [4] Initial Data Population
     [Q] Quit
     Enter Your Option :
   ```

7. Enter 1 Sign in Menu. A second sign-in screen appears, where you enter the following database sign-in information:

```
          SIGN IN MENU
----------------------------------------
 [1] Enter Database USERNAME       ...>edx_dba
 [2] Enter Database PASSWORD       ...>edx
 [3] Enter the ORACLE_SID ...>edx0
```

8. After you enter the ORACLE_SID value, the main menu reappears. Select **[2] Install Application Database**.
9. When the script finishes, enter **[3] Install Application Database II**.
10. When the script finishes, enter **[4] Initial Data Population**. This step populates the Payment database with data required to run Payment.
11. When the script finishes, enter **[Q] Quit** to end the script. The Payment database installation is completed. Make sure the Oracle listener is running, and the Payment database is ready to use.

Creating a Service Manager Database

On the database server, follow these instructions to use the setup_schema.sh script to set up a Service Manager database in an existing Oracle database instance.

To set up the Service Manager database:

1. Recursively change the user and group permissions of your TSM_HOME database directory and all subdirectories to the database instance owner.

   ```
   chown -R oracle:dba /opt/Siebel/TSM/db
   ```

2. Log on as the Oracle user. For example:

   ```
   su – oracle
   ```

3. Set the ORACLE_SID

   ```
   ORACLE_SID=edx0
   export ORACLE_SID
   ```

4. Change directory to your Service Manager database home directory. For example:

   ```
   cd /opt/Siebel/TSM/db/oracle
   ```

5. Execute the database setup script:

   ```
   ./setup_schema.sh
   ```
   The script displays the Main Menu:

   ```
         CBM Enhancement Main Menu
   ----------------------------------------

   [1] Sign in Menu
   [2] Apply CBM enhancement
   [3] Initial data population
   [4] CBM Migration
   [Q] Quit

   ----------------------------------------
   Enter Your Selection:
   ```

6. On the Main Menu, Enter Option 1, “Sign in Menu.” The Sign-In Menu appears:

   ```
               SIGN IN MENU
   ----------------------------------------
   Please enter Oracle SID        -->
   Enter Database Username        -->
   Enter Database Password        -->
   Please enter the password for SYS user ...>
   ```

7. Enter the appropriate Oracle SID, Platform Services database user name and password, and the SYS user password.

8. On the Main menu, choose Option 2, “Apply CBM Enhancement.”

   The script prompts you to execute the database scripts one at a time. Enter Yes or No to run each script (default is “Y”) as shown here:

   ```
   Task creation started....

   Run pwc_tasks.sql ([Y/N]-default option Y)--> y
   Running pwc_tasks.sql...

   Run pwc_job_types.sql ([Y/N]-default option Y)--> y
   Running pwc_job_types.sql...

   Run drop_bsl_schema ([Y/N]-default option Y)--> y
   Running drop_bsl_schema.sql....

   Run drop_hierarchy_schema ([Y/N]-default option Y)--> y
   Running drop_hierarchy_schema.sql....

   Run create_bsl_schema ([Y/N]-default option Y)-->Y
   Running create_bsl_schema.sql....

   Run create_bsl_const ([Y/N]-default option Y)--> y
   Running create_bsl_const.sql…..
   ```

```
Run create_hierarchy_schema ([Y/N]-default option Y)--> y
Running create_hierarchy_schema.sql

Run create_hierarchy_const ([Y/N]-default option Y)--> y
Running create_hierarchy_const.sql….

Run add_payment_indexes ([Y/N]-default option Y)--> Y
Running add_payment_indexes.sql...

Run create_addr_book_tables ([Y/N]-default option Y)--> Y
Running create_addr_book_tables.sql....

Run create_pmt_due_notification_tables.sql ([Y/N]-default
option Y)--> Y
Running create_pmt_due_notification_tables.sql....

Run create_unbilled_tables.sql ([Y/N]-default option Y)--> Y
Running create_unbilled_tables.sql....

Run create_unbilled_initdata.sql ([Y/N]-default option Y)--> Y
Running create_unbilled_initdata.sql....

Run create_messenger_queue_tables.sql ([Y/N]-default option
Y)--> Y
Running create_messenger_queue_tables.sql....

Task creation started....

Run move_indexes_to_tablespaces.sql ([Y/N]-default option Y)--
> Y
Running move_indexes_to_tablespaces.sql...

Task creation completed....Task creation completed....
```

Press **ENTER** to return to menu

9. On the main menu choose Option 3, “Initial data population” and Enter. The script displays the following message and begins populating the data:

   ```
   Starting initial data population...
   ```

10. Press ENTER to return to the menu.

11. When the script finishes, enter **[Q] Quit** to end the script.

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.

6 Configuring WebSphere Application Server for UNIX

Overview

This chapter assumes in-depth understanding of and practical experience with WebSphere Enterprise Server. Consult your WebSphere documentation as necessary.

You must start your WebSphere server instance and bring up the Administration Console before you begin this chapter.

Caution: If you cannot bring up the WebSphere 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 WebSphere Server and any other client machines of your database server.

This chapter provides instructions for configuring WebSphere Server to support Service Manager. It includes:

- UNIX permissions for WebSphere Server
- Starting and Stopping WebSphere
- Capturing your UNIX environment for Service Manager

Caution: The installation and configuration examples 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.

UNIX Permissions for WebSphere Server

Application servers running Service Manager will not function correctly without access to Service Manager configuration files, storage directories, and related resources. When installing Service Manager on WebSphere 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 Service Manager.

DEFAULT	EXAMPLE
Specified during installation.	**edxadmin:edxadmin**

For details on owner and group permissions for WebSphere, please consult the documentation for your platform.

To reset user and group permissions:

1. Switch user to the default owner of your application server home directory.
2. Recursively change the user and group permissions of the application server installation directory and all subdirectories to the application server owner.

   ```
   chown –R edxadmin:edxadmin /opt/WebSphere
   ```

3. Switch user to your application server owner and configure your application server with your new owner.

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.

Starting and Stopping WebSphere

Developers and system administrators will need to be familiar with how to stop and start your WebSphere server and any active Web applications for your platform.

For details on starting and stopping your WebSphere server and for verifying startup, consult the server documentation for your platform.

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, and to schedule and run jobs in the Siebel Command Center.

Caution: 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. This can damage your configuration file. Siebel recommends using the Web console and/or the SHUTDOWN command to ensure a graceful shutdown.

Capturing Your UNIX Environment Variables for Service Manager

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

`$EDX_HOME/bin/edx_config`	Executable shell script prompts you to define environment variables required by your application server
`$EDX_HOME/config/edx_env`	Configuration file stores the environment variables you specify in edx_config (DO NOT EDIT).
`$EDX_HOME/config/edx.config`	Shell script passes the environment data in edx_env to your application server when sourced in your startup script

This section describes how to run edx_config to capture your environment variables and store them in edx_env. For more information on edx.config, see “Passing UNIX Environment Data to WebSphere” on page 36.

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.

Tip: Run edx_config any time you need to modify your Service Manager environment. Do not modify edx_env directly.

Caution: Do not confuse edx_config (underscore) with edx.config (dot), which sources your configuration at server startup.

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

If you have not already done so, verify that the ownership of the **`$EDX_HOME`** directory is set to the user and group of the application server owner. If not, change it before running **`edx_config`**.

Use the values shown in "WebSphere Environment Variables,' below to enter values for each of the specified parameters. You may want to print the *Environment Variables* sections for easy reference.

To set environment data with edx_config for WebSphere:

1. Switch user to the application server owner, in this example **edxadmin**.

   ```
   su - edxadmin
   ```

2. Navigate to the **bin** directory for Service 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, using the variables shown in "WebSphere Environment Variables,' below.

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.

Passing UNIX Environment Data to WebSphere

To pass your Service Manager environment to your application server, you must edit your application server startup script to:

1. Set values for your Siebel home, application server home, and Java home directories.
2. Call and process the configuration script edx.config. This procedure is called "sourcing your configuration."

To pass your Service Manager environment to WebSphere:

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

   ```
   su - edxadmin
   ```

2. Stop your application server.

3. Change directory to the location of your application server startup script and open the file for editing. For example,

```
cd $WAS_HOME/bin
vi startServer.sh
```

4. Declare and initialize the variable **$EDX_HOME** near the beginning of the file with other variable declarations. For example:

```
binDir=`dirname $0`
. $binDir/setupCmdLine.sh

EDX_HOME=/export/home/Siebel/TSM/estatement
export EDX_HOME
```

5. In the same file, source **edx.config** just before the command to start the JVM. The dot and space preceding the pathname are a required part of the syntax. For example:

```
. $EDX_HOME/config/edx.config
```

6. (Recommended for use on the presentment server only.) Add the following commands to startServer.sh to enable you to make use of the Solaris multi-threaded MT-MALLOC library:

```
LD_PRELOAD_32=/usr/lib/libmtmalloc.so
LD_PRELOAD_64=/usr/lib/sparcv9/libmtmalloc.so
export LD_PRELOAD_32
export LD_PRELOAD_64
```

7. Save and close the file.

Start the Server:

When you have finished the steps to pass your Service Manager environment to WebSphere, restart the server. For example:

```
./startServer.sh server1
```

WebSphere Environment Variables

VARIABLE	DESCRIPTION	SOLARIS
WAS_HOME	Application server home	**/opt/WebSphere/AppServer**
APP_OWNER	Application server owner	**edxadmin**
APP_GROUP	Application server group	**edxadmin**
APP_PORT	Application server port	**9080**
ADMIN_PORT	Application server admin port	**9090**

VARIABLE	DESCRIPTION	SOLARIS
JAVA_HOME	Java home directory	**$WAS_HOME/java**
JMS_HOME	MQSeries java client directory	**/opt/mqm/java**

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

Configuring Payment

If you are going to use a payment processor credit card gateway, then you must add the classpaths for the processor in your application server definition, and configure your java security file.

To edit the classpath in the application server JVM definition:

1. Under Servers, click on **Application Servers**.
2. Select the application server that supports the Service Manager application, and under Additional Properties click on **Java Virtual Machine**. For the Classpath property, add the following JAR files:

   ```
   <JSSE_PATH>\jsse.jar;<JSSE_PATH>\jcert.jar;<JSSE_PATH>\jnet.jar;%CLASSPATH%
   ```

3. Substitute the explicit path for <JSSE_PATH>.

Configuring java security

1. Modify the java.security file to configure Java to use the JSSE, which is located in *JAVAHOME/jre/lib/security/java.security* 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

Preparing for a System Failure

The database for Payment should be backed-up regularly, for example nightly, to maximize the recovery effort in the event of a system failure. For example, if the Payment Database Server failed for some reason, the database administrator could restore from the latest backed-up version. A failure to the Payment Database Server also requires that the database administrator restart the Payment services in order to reestablish the database connection pool.

Payment automatically handles single transaction failures by rolling back to the transaction's original state. This happens because each Payment operation is handled within its own transaction context.

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

Preparing WebSphere Server for UNIX

Creating an Application Server Instance for WebSphere

As a convenience, WebSphere provides a default application server instance you can use immediately to deploy and install a J2EE application.

You can choose to create a separate application server instance for a new J2EE application (EAR). For example, you can choose to deploy ear-eStatement.ear on the default instance, and the ear-TBM.ear Web application on a new instance. See the WebSphere documentation for more information on deployment.

To create an application server, follow these steps:

1. Open the WebSphere Administrative Console.
2. Expand the Servers entry by clicking on the plus sign **+**, then click the **`Application Servers`** hyperlink.

3. Click **New**. The Create New Application Server screen appears.
4. Select the Cell/Node combination where you want to create the new server.
5. Choose a name for the new server.
6. Pick a template for the new server, using either server1 as the template or another server you have created. Click the **Next** button. When prompted, click **Finish** to confirm the creation of your new server. Save the Master Configuration to preserve the changes.
7. Place your server on a virtual host and open ports so users can access the server. WebSphere recommends you leave the admin_host for server1 and place new servers on another virtual host.
8. From the Application Servers window, click on the new server you created. The Presentment Server page appears.
9. Find out which ports your new server is running on. Click **Web Container** and then the **Http Transports** link. The HTTP Transport screen appears.
10. Open the same http and https ports on the virtual hosts where you are placing this server. Expand **Environment** in the left pane and choose **Virtual Hosts**. WebSphere displays the Virtual Hosts screen.
11. Click **default_host** to add the new ports. Click **Host Aliases** to see what ports are open for the virtual host. On that page, you can click **New** to add an alias.
12. You can enter an asterisk (*) to allow all traffic on the port you are creating, or you can restrict to an IP address or DNS name. If you are restricting to a particular Web server, it is a good idea to explicitly state via IP, DNS, or FQDN. To accept requests from myserver on port 9081 you could have one of the following three entries:
    - 192.168.1.100:9081 (for IP)
    - myserver:9081 (for DNS)
    - myserver.mydomain.com:9081 (for Fully Qualified Domain Name)
13. After you enter the server and port, click **Apply** then **OK**. After you have finished configuring the virtual host, be sure to click **Save** to keep your changes.

Java Virtual Machine (JVM) Settings for WebSphere

To configure JVM settings for WebSphere:

1. In the WebSphere Administrative console, click **Servers**, then **Application Servers**, then the server for Service Manager, then **Process Definition**, then **Java Virtual Machine**.

   On the General Properties page, specify:

Property Name	Value
Classpath	<TSM_HOME >/lib/log4j-1.2.8.jar <TSM_HOME >/lib/commons-logging-1.0.3.jar <TSM_HOME >/lib/ldeprotocol.jar <TSM_HOME >/lib/Configuration.jar <TSM_HOME >/lib/dom4j-1.4.jar <TSM_HOME >/lib/javachart.jar <TSM_HOME >/config <EDX_HOME>/lib/jakarta-oro-2.0.7.jar <EDX_HOME>/lib/xpp3_1_1_2.jar <EDX_HOME>/lib/edx_common.jar <EDX_HOME>/lib/edx_system.jar <EDX_HOME>/lib/edx_client.jar Specify the following jar files only if you are going to implement credit card functionality using Verisign: <PAYMENT_HOME>/lib/Verisign.jar <PAYMENT_HOME>/lib/jcert.jar <PAYMENT_HOME>/lib/jnet.jar <PAYMENT_HOME>/lib/jsse.jar Where TSM_HOME is the directory where you installed Service Manager; EDX_HOME is TSM_HOME/estatement, and PAYMENT_HOME is TSM_HOME/payment.

2. Set JVM settings as follows. Change the values according to your environment:

Property Name	value
Initial Heap Size	**384**
Maximum Heap Size	**512**
Generic JVM Arguments	**-XX:MaxPermSize=256m**

3. Under Additional Properties, click on **Custom Properties** and create the following properties using the values described here:

Property Name	Value
com.edocs.tasks.loader.alias	The name of the database alias. For example: **edx0**

Property Name	Value
com.edocs.tasks.loader.password	The password of the database user. For example: edx
com.edocs.tasks.loader.user	The name of the database user. For example: **edx_dba**
com.edocs.tasks.statementscanner.skipResolver	**true**
edx.home	Value of EDX_HOME, for example: **/opt/Siebel/TSM/estatement**
java.protocol.handler.pkgs	**com.edocs.protocol|com.edocs.domain.telco.lde.protocol** NOTE: The '|' sign should be used instead of ';'

4. After applying the new settings, **make sure you**:
   - Save the master configuration.
   - Restart the server.

Editing Property Files

Update the following files found in $TSM_HOME/config:

a. hibernate.properties and hierarchy.hibernate.properties
b. hierarchy.cfg.xml
c. persistence.xma.xml
d. app-config.properties
e. log4j.xml
f. tsm.hibernate.properties
g. tsm.xma.xml
h. sm.xma.xml
i. edocs.tsm.properties

a. hibernate.properties and hierarchy.hibernate.properties

In both files, comment out WebLogic-specific settings and enable WebSphere specific settings (under Websphere AppServer environment). See the directions inside the properties files for details.

```
# Weblogic AppServer environment
#
#hibernate.jndi.class=weblogic.jndi.WLInitialContextFactory
#hibernate.transaction.manager_lookup_class=net.sf.hibernate.transact
ion.WeblogicTransactionManagerLookup

#
```

```
# Websphere AppServer environment
#
hibernate.jndi.class=com.ibm.websphere.naming.WsnInitialContextFactor
y
hibernate.transaction.manager_lookup_class=net.sf.hibernate.transacti
on.WebSphereTransactionManagerLookup

# Weblogic AppServer environment
#
#hibernate.jta.UserTransaction=javax.transaction.UserTransaction

#
# Websphere AppServer environment
#
hibernate.jta.UserTransaction=java:comp/UserTransaction
```

Also, 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 for Siebel Billing Manager.

NOTE: Restart your application server after updating hibernate.properties and hierarchy.hibernate.properties files.

b. hierarchy.cfg.xml

Comment out WebLogic-specific settings and enable WebSphere specific settings:

```
<!-- for WebLogic -->

<!--

        <bean id="transactionManager"
class="org.springframework.transaction.jta.JtaTransactionManager">
                <property name="userTransactionName">
                        <value>javax.transaction.UserTransaction</value>
                </property>
                <property name="transactionManagerName">
                        <value>javax.transaction.TransactionManager</value>
                </property>
        </bean>
-->

        <!-- for WebSphere -->
                <bean id="websphereTransactionFactory"
class="org.springframework.transaction.jta.WebSphereTransactionManagerFactoryBean"/>
        <bean id="transactionManager"
class="org.springframework.transaction.jta.JtaTransactionManager">
                <property name="userTransactionName">
                        <value>java:comp/UserTransaction</value>
                </property>
                <property name="transactionManager">
                        <ref local="websphereTransactionFactory"/>
                </property>
        </bean>
```

c. persistence.xma.xml

Comment out WebLogic-specific settings and enable WebSphere specific settings (under JTA Transaction manager for WebSphere):

```
<!-- JTA Transaction manager for WebSphere  -->
    <bean id="websphereTransactionFactory"
class="org.springframework.transaction.jta.WebSphereTransactionManagerFactoryBean"
/>
    <bean id="transactionManager"
class="org.springframework.transaction.jta.JtaTransactionManager">
            <property
name="userTransactionName"><value>java:comp/UserTransaction</value></property>
            <property name="transactionManager">
          <ref local="websphereTransactionFactory"/>
        </property>
    </bean>

<!-- JTA Transaction manager for WebLogic -->
    <!--
    <bean id="transactionManager"
class="org.springframework.transaction.jta.JtaTransactionManager">
            <property
name="userTransactionName"><value>javax.transaction.UserTransaction</value></prope
rty>
            <property
name="transactionManagerName"><value>javax.transaction.TransactionManager</value><
/property>
    </bean>
    -->
..
..
..

<!-- JTA TransactionManager for WebSphere -->
              <prop
key="hibernate.transaction.manager_lookup_class">net.sf.hibernate.transaction.WebS
phereTransactionManagerLookup</prop>

<!-- JTA TransactionManager for WebLogic -->
              <!-- prop
key="hibernate.transaction.manager_lookup_class">net.sf.hibernate.transaction.Webl
ogicTransactionManagerLookup</prop -->
```

d. app-config.properties

Edit app-config.properties and change the parameter in the following line to 6 as follows:

```
tbmb2b.onetimepay.statement_search_months=6
```

If you are installing a system that 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
```

Also edit the SMTP server statement to set your SMTP host for email notification for enrollment.

For more information, please refer to the Developer's Guide for Siebel Billing Manager.

Based on the database server, you must modify the db settings (threshold values) in file too.

e. log4j.xml

If you installed Service Manager in a directory other than the default TSM_HOME, you must edit log4j.xml in TSM_HOME/config:

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

```
<param name="File"
value="/opt/Siebel/TSM/estatement/logs/log4j.log"/>
```

f. 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.

g. tsm.xma.xml

Comment out WebLogic-specific settings and enable WebSphere specific settings.

Here, localhost is your server name and 2809 is the WebSphere iiop port. Modify these according to your environment.

```
                <!-- Initial ContextFactory for WebLogic -->
                <!-- prop
key="hibernate.jndi.class">weblogic.jndi.WLInitialContextFactory</prop -->

                <!-- Initial ContextFactory for Websphere -->
                <prop
key="hibernate.jndi.class">com.ibm.websphere.naming.WsnInitialContextFactory</prop
>

                <!-- Hibernate jndi url for WebLogic -->
                <!-- prop key="hibernate.jndi.url">t3://localhost:7001</prop -->

                <!-- Hibernate jndi url for Websphere -->
                <prop key="hibernate.jndi.url">iiop://localhost:2809</prop>

                <prop key="hibernate.use_outer_join">true</prop>
                <prop key="hibernate.cache.use_query_cache">true</prop>

                <!-- JTA TransactionManager for WebLogic -->
                <!-- prop
key="hibernate.transaction.manager_lookup_class">net.sf.hibernate.transaction.Webl
ogicTransactionManagerLookup</prop -->

                <!-- JTA TransactionManager for WebSphere -->
                <prop
key="hibernate.transaction.manager_lookup_class">net.sf.hibernate.transaction.WebS
phereTransactionManagerLookup</prop>

            </props>
        </property>
```

```
..
..
..


<!--  Websphere application server -->

    <bean id="websphereTransactionFactory"
class="org.springframework.transaction.jta.WebSphereTransactionManagerFactoryBean"
/>
    <bean id="TSMTransactionManager"
class="org.springframework.transaction.jta.JtaTransactionManager">
        <property
name="userTransactionName"><value>java:comp/UserTransaction</value></property>
  <property name="transactionManager">
    <ref local="websphereTransactionFactory"/>
  </property>
    </bean>


<!--  WebLogic application server -->
    <!--
    <bean id="TSMTransactionManager"
class="org.springframework.transaction.jta.JtaTransactionManager">
  <property
name="userTransactionName"><value>javax.transaction.UserTransaction</value></prope
rty>
  <property
name="transactionManagerName"><value>javax.transaction.TransactionManager</value><
/property>
    </bean>
    -->
```

h. sm.xma.xml

Comment out WebLogic-specific settings and enable WebSphere-specific settings:

```
<bean id="jms.context" class="com.edocs.common.eai.jms.EIInitialContextProvider">
    <!-- for WebLogic -->
    <!--property name="contextFactoryClass">
      <value>weblogic.jndi.WLInitialContextFactory</value>
    </property>
    <property name="contextProviderUrl">
      <value>t3://localhost:7001</value>
    </property-->
    <!-- for WebSphere -->
    <property
name="contextFactoryClass"><value>com.ibm.websphere.naming.WsnInitialContextFactor
y</value></property>
    <property
name="contextProviderUrl"><value>iiop://localhost:2809</value></property>
  </bean>
  <bean id="jms.queue.sm" class="com.edocs.common.eai.jms.EIMessageQueue">
    <!--for WebLogic -->
    <!--property name="name">
      <value>edx/jms/serviceManagerQ</value>
    </property>
    <property name="connectionFactoryJndiName">
      <value>edx/tcf/serviceManager</value>
    </property-->
    <!--for WebSphere -->
    <property name="name">
      <value>edx.jms.serviceManagerQ</value>
    </property>
    <property name="connectionFactoryJndiName">
      <value>edx.qcf.serviceManager</value>
    </property>
    <property name="contextProvider">
      <ref bean="jms.context"/>
    </property>
    <property name="listener">
      <ref bean="SMDispatcher"/>
    </property>
  </bean>
```

i. edocs.tsm.properties

Rate plan attributes – Service Manager handles the minutes available with a particular rate plan specially, and so it needs to separate out those values from the other features associated with that rate plan.

The following attributes represent the keys in the database where the value is the number of peak minutes, weekend minutes, and night minutes available with that rate plan, respectively. If a system deploys other keys than these for these rate plan attributes, then you must edit these settings accordingly. These are correct for our out-of-the-box sample Foundation Application.

```
# rate plan attributes

rate_plan_attribute1=PEAKMINUTES
rate_plan_attribute2=WEEKENDMINUTES
rate_plan_attribute3=NIGHTMINUTES
```

Reason Code attributes are used in association with activating and deactivating service, and are defined using this format: value,key:value,key. For example, code 1 for activating service means "Received phone from service provider." If you want to use different codes and meanings, define them here.

```
# activate service reason codes

activate_service_reason_codes=Received phone from service
provider,1:Received phone from another source,2
deactivate_service_reason_codes=Original DSN incorrect,1:Phone
being repaired,2:Phone was stolen,3:Account closed,4
```

The **Overview Transactions Lookback** Attribute determines how far the system looks back for transactions to appear on the "Recent Service Requests" section of the CSM Overview page. The default displays service requests from the last 30 days. You can configure this as needed.

```
# Overview use case days to search back

edocs.tsm.overview.transactions.days_in_past=30
```

The following **UI Mode attributes** are used ONLY in a test environment without a database in place; you can ignore these.

```
edocs.tsm.bizaction.ui_data_folder=/opt/app/edocs/TSM/ui_data_
folder

edocs.tsm.bizaction.ui_mode=false
```

Configuring Charting for WebSphere 1.0

1. Install and configure a virtual display (xfvb – X Virtual Frame Buffer). xvfb might already be installed on your UNIX system, in /usr/X11R6. If not, you must obtain and install a copy. You can obtain a copy at:

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

2. Set the display awareness. Enter:

   ```
   DISPLAY=<yourserver>:1;export DISPLAY
   ```

   ```
   where <yourserver> is the UNIX server IP address.
   ```

   Or, advanced users can modify the startServer.sh file as follows:

   ```
   DISPLAY=<yourserver>:1
   export DISPLAY
   /usr/X11R6/bin/xhost +
   ```

3. Set display permissions using xhost. Once you have installed xvfb in the UNIX machine, go to **`/usr/X11R6/bin`** and type **`xhost +`** to enable X display permission on your Web server.

   Note: It is recommended to add the xhost directory to the PATH environment variable to enable the user to execute it from anywhere in the directory structure.

4. Run xvfb as a background process. Enter:

   ```
   /usr/X11R6/bin/Xvfb :1 -screen 0 800x600x24 &
   ```

5. Restart the WebSphere server. Once the message “Access control disabled client can connect from any host” message appears at the top of the console screen, you should be able to view charts in your application.

7 Configuring Java Resources for WebSphere

Overview

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

See WebSphere Server documentation 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 WebSphere Server and bring up the Administrative Console before you begin this chapter.

Caution: If you cannot bring up the WebSphere Console/Client, 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.

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 your application server. Your application 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 transaction 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 Transaction (Tx) 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 details on configuring JDBC Connections, please see the JDBC documentation for your application and database servers.

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

For details on how to create and configure JDBC Connections, please see your application server documentation.

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 database server.

Configuring JDBC Connections for WebSphere

To configure JDBC Connections for WebSphere:

1. Create the required JDBC Connection Pools according to the following tables. Use the appropriate JDBC values for your database server.
2. Create required JDBC Data Sources according to the following tables. Use the appropriate JDBC values for your database server.
3. Review your connections. Each data source should target the connection pool of the same name.
4. When you are finished, proceed to the next section to configure Java Messaging Service (JMS) for **Error! Unknown document property name.**.

JDBC Connection Pools for Service Manager

Configure a new JDBC Provider. Choose **`Oracle JDBC Driver`** from the JDBC Providers drop-down list.

Also configure the following properties:

Name	Value
Classpath	**`$TSM_HOME/lib/ojdbc14_g.jar`** Enter the explicit path; do not use the variable.

JDBC Data Sources for Service Manager

Create the following **Data Sources (Version 5)** for the new JDBC Provider:

Data Source	**`edxDatabasePool`**	**`edxUserDatabasePool`**	**`edxLoggerDatabasePool`**
JNDI Name	**`edx.databasePool`**	**`edx.user.databasePool`**	**`edx.logger.databasePool`**

ALSO create the following **Data Sources (Version 4)** for the new JDBC Provider:

Data Source	**`ecs`**	**`ecs_security`**	**`ecs_serial_trans`**
JNDI Name	**`jdbc.ecs`**	**`jdbc.ecs_security`**	**`jdbc.ecs_serial_trans`**

Repeat the following procedure for each data source.

To create a data source

1. Go to **`JDBC Providers > Oracle JDBC Driver > Data Sources`** and click **`New.`**
2. Specify the General Property configuration details and click **`Apply`**:

General Property	Value
Name	The name of the data source you're creating, such as **`edxDatabasePool`**
JNDI Name	The JNDI name required for the data source you're creating, such as **`edx.databasePool`**
Container Managed Persistence	(Do not select this option)
Description	**`New JDBC Datasource`**
Database Name	The database name that is used For example **`edx.db`**

General Property	Value
Default User ID	Database user name For example **edx_dba**
Default Password	Database password. For example **edx**

3. Go to **JDBC Providers > Oracle JDBC Driver > Data Sources > edxDatabasePool > Custom Properties** (where **edxDatabasePool** is the name of the data source you're creating). Specify the URL, for example, then click **Apply.**

Custom Property	Value
URL	**jdbc:oracle:thin:@DB_host:DB_port:TSM_DB.** For example, **jdbc:oracle:thin:@localhost:1521:edx0**

4. Go to **JDBC Providers > Oracle JDBC Driver > Data Sources > edxDatabasePool > J2C Authentication Data Entries** (where **edxDatabasePool** is the name of the data source you're creating and click **New.** Specify the user configuration and click **Apply**:

General Property	Value
Alias	Database alias, such as **edx.db**
User ID	Database User ID, such as **edx_dba**
Password	Enter a password for use by Java 2 Connector security such as edx0
Description	None

5. Go to **JDBC Providers > Oracle JDBC Driver > Data Sources > edxDatabasePool** (where **edxDatabasePool** is the name of the data source you're creating) and apply following settings:

Setting	Value
Statement Cache Size	**100**
Datasource Helper Classname	**com.ibm.websphere.rsadapter.OracleDataStoreHelper**
Component-managed Authentication Alias	Server name/alias, such as **mppc/edx.db**
Container-managed Authentication Alias	Server name/alias, such as **mppc/edx.db**
Mapping-Configuration Alias	**DefaultPrincipalMapping**

Additional JDBC Data Sources for Service Manager (CSM Only)

Create the following **Data Sources (Version 4)** for the new JDBC Provider:

Data Source	**ecs**	**ecs_security**	**ecs_serial_trans**
JNDI Name	**jdbc.ecs**	**jdbc.ecs_security**	**jdbc.ecs_serial_trans**

Repeat the following procedure for each data source.

To create a data source:

1. Go to **JDBC Providers > Oracle JDBC Driver > Data Sources (Version 4)** and click **New.**
2. Specify the General Property configuration details and click **Apply**:

General Property	Value
Name	The name of the data source you're creating, such as **ecs**
JNDI Name	The JNDI name required for the data source you're creating, such as **jdbc.ecs**
Description	**An Oracle Datasource**
Database Name	The database name that is used, for example **edx.db**
Default User ID	Database user name, for example **edx_dba**
Default Password	Database password, for example **edx**

3. Go to **JDBC Providers > Oracle JDBC Driver > Data Sources (Version 4) > ecs > Custom Properties** (where ecs is the name of the data source you're creating). Specify the URL, for example, then click **Apply.**

Custom Property	Value
URL	**jdbc:oracle:thin:@DB_host:DB_port:TSM_DB.** For example: **jdbc:oracle:thin:@localhost:1521:edxdb**

Configuring Java Messaging Services (JMS) for Service Manager

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

- **edxAnnotation** supports Line Item Dispute and Annotation features.
- **edxDispute** supports Line Item Dispute and Annotation features.
- **edxLogger** supports Service Manager logging through the Service Manager Web application. edxLogger requires **five** JMS consumers and session pools.
- **edxServiceManager** supports Service Manager.

Tip: If your Web application does not implement Line Item Dispute and Annotation, you need only configure JMS resources for edxLogger and edxServiceManager.

About JMS Resources

JMS enable 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. Your application 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 Topics** can be one of two destinations that you can configure for a JMS server. The other destination is a JMS queue.
- **JMS Session Pools** allow a JMS listener to have multiple threads that improve performance under heavy load.
- **JMS Listener Ports** specify a JMS Connection Factory and JMS Destination that an MDB, employed against that port, will listen upon.

Configuring JMS Resources for Platform Services

Click on **Resources**, then **WebSphere JMS Provider**, and create these three new WebSphere Topic Connection Factories:

Name	**edxLoggerTCF**
JNDI name	**edx.tcf.log**
Port	**Queued**

Name	**edxAnnotationTCF**
JNDI name	**edx.tcf.annotate**
Port	**Queued**

Name	**edxDisputeTCF**
JNDI name	**edx.tcf.dispute**
Port	**Queued**

Go to **Resources**, then **WebSphere JMS Provider**, and for each **WebSphere Topic Connection Factory** apply the following settings:

Setting	Value
Component-managed Authentication Alias	Server name/alias, such as **mppc/edx.db**
Container-managed Authentication Alias	Server name/alias, such as **mppc/edx.db**
Mapping-Configuration Alias	**DefaultPrincipalMapping**
XA Enabled	**Check the check box**

Create this new WebSphere Queue Connection Factory:

Name	**edxServiceManagerQCF**
JNDI name	**edx.qcf.serviceManager**

WebSphere Topic Destinations (WebSphere, Oracle, UNIX)

Click on **Resources**, then **WebSphere JMS Provider**, and create three new WebSphere Topic Destinations:

Name	**edxLoggerTopic**
JNDI Name	**edx.jms.log**
Topic	**edxLoggerTopic**
Persistence	**Application Defined**
Priority	**Application Defined**

Name	**edxAnnotationTopic**
JNDI Name	**edx.jms.annotate**
Topic	**edxAnnotationTopic**
Persistence	**Application Defined**
Priority	**Application Defined**

Name	**edxDisputeTopic**
JNDI Name	**edx.jms.dispute**
Topic	**edxDisputeTopic**
Persistence	**Application Defined**
Priority	**Application Defined**

Create this new WebSphere Queue Destination:

Name	**edxServiceManagerQ**
JNDI Name	**edx.jms.serviceManagerQ**
Persistence	**Application Defined**
Priority	**Application Defined**

WebSphere MDB Listener Port (WebSphere, Oracle, UNIX)

Click on **Application Servers**, the server name, then **Message Listener Service**:

Name	**SMQueueListenerPort**
Connection Factory JNDI Name	**edx.qcf.serviceManager**
Destination JNDI Name	**edx.jms.serviceManagerQ**

Click **SMQueueListenerPort** and specify the following general properties on the **Configuration** tab:

Name	**SMQueueListenerPort**
Initial State	**Started**
Description	**SMQueueListenerPort**
Connection factory JNDI name	**edx.qcf.serviceManager**
Destination JNDI name	**edx.jms.serviceManagerQ**
Maximum sessions	5
Maximum retries	10

Name	**SMQueueListenerPort**
Maximum messages	1

Click on **Application Servers**, the server name, then **Server Component**, click on the **JMS Servers** link and add the edxServiceManagerQ queue destination:

Name	**Internal JMS Server**
Description	**Internal WebSphere JMS Server**
Number of threads	**1**
Queue names	**PlantsByWebSphereQ** **Sample.JMS.Q1** **Sample.JMS.Q2** **edxServiceManagerQ**

After configuring JMS for WebSphere, be sure to s**ave to the master configuration**.

Restarting the Application Server

When restarting the application server you should follow this order:

1. Stop the scheduler and logger.
2. Stop the Service Manager application server.

When you are ready to start the Service Manager application server again, reverse the steps: start the application server, then start the logger and scheduler.

After configuring JMS for WebSphere, the next step is to deploy the application.

Deploying Service Manager

After configuring your WebSphere domain server, you can deploy the EAR files to the appropriate servers.

Caution: When deploying, be sure to select the correct Database Type at Step 2.

- **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
Siebel Platform Services	$TSM_HOME/J2EEApps/websphere	ear-eStatement.ear
Service Manager	$TSM_HOME/J2EEApps/websphere	ear-tbm-b2b.ear
Service Manager CSR	$TSM_HOME/J2EEApps/websphere	ear-tbm-csr.ear

Consult your IBM WebSphere documentation on how to deploy applications.

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

1. In your browser, point to http://localhost:9080/edocs (where localhost:9080 is your server name if you are on a different machine). 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 passwords.

You must set up the necessary applications and jobs in the Command Center before you can run the Service Manager and CSR applications successfully. See the *Administration Guide for Siebel Service Manager* for details.

To run Service Manager (after setting up the system using Command Center):

1. Use the following link, substituting the name of the application server in place of **`localhost`** and the correct port number in place of **`9080`**:

   http://localhost:9080/tbmb/

2. Log on to the application.

3. Select the Self-Service tab.

To run the CSR application (after setting up the system using Command Center):

1. Use the following link, substituting the name of the application server in place of **`localhost`** and the correct port number in place of **`7001`**:

   http://localhost:7001/tbmcsr

2. Log on to the application.

Starting the Scheduler

To start the Service Manager 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 Service Manager installation, **$EDX_HOME/bin.**

3. Run the Scheduler command for WebSphere Server (ws_scheduler), host, and port. This example shows the Scheduler command for WebSphere:

   ```
   ./ws_scheduler –start –url iiop://localhost:2809
   ```

   Tip: This 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.

Starting the Logger

You must edit the WebSphere logger start file, ws_logger, to set it up for use with Billing Manager.

Before editing the logger start file, you must first download and install a patch from the IBM Web site.

To install the WebSphere patch:

1. Go to:

   ```
   https://www6.software.ibm.com/dl/wsmqcsd/wsmqcsd-p
   ```

2. Use your IBM.com user name and password to access the Web site.

3. Select the “WebSphere Embedded Messaging interim fixes for WebSphere Application Server V5.1.1.” option.

4. Click **`Continue.`**

5. Verify your personal information, then click **`Submit.`**

6. Download the appropriate interim fix for **`APAR IY59675`**.

7. Unzip the file and place the **`bipbroker.jar`** file in the **`WEMPS_HOME/classes`** directory. The readme.txt file has additional details.

To edit ws_logger:

1. Edit the ws_logger file in the $EDX_HOME/bin folder and add the property `CCDWAS_AuthDataFile=$EDX_HOME/bin/client.txt` as shown:

```
CMD="$JAVA_HOME/bin/java \
  -Xbootclasspath/p:$WAS_BOOTCLASSPATH \
  $CLIENTSAS \
  $CLIENTSOAP \
  $ORB_RAS_MGR \
  $USER_INSTALL_PROP \
  $EDX_OPTS \
  -Dwas.install.root=$WAS_HOME \
  -Dws.ext.dirs=$WAS_EXT_DIRS \
  -
Djava.security.auth.login.config=$WAS_HOME/properties/wsjaas_c
lient.conf \
  -Dcom.ibm.CORBA.BootstrapHost=$DEFAULTSERVERNAME \
  -Dcom.ibm.CORBA.BootstrapPort=$SERVERPORTNUMBER \
  -Djava.naming.factory.initial=$NAMING_FACTORY \
  -classpath $WAS_CLASSPATH com.ibm.ws.bootstrap.WSLauncher \
com.ibm.websphere.client.applicationclient.launchClient
$EDX_HOME/J2EEApps/websphere/Deployed_ear-eStatement.ear
-CCDWAS_AuthDataFile=$EDX_HOME/bin/client.txt -
CCclasspath=$CLASSPATH
-CCjar=app-logger.jar -CCBootstrapHost=$HOST -
CCBootstrapPort=$PORT -CCverbose=true $@"
```

2. Save the edits to ws_logger.
3. Create a new file named "client.txt" in $EDX_HOME/bin folder.
4. Edit client.txt and add the following line:

   `<J2C Authentication Alias>,<DB user name>,<DB password>`

   For example:
   `demeter/userPool,tbmws,tbmws`

   Where:

   `demeter/userPool is the alias defined in “J2C Authentication Data” which edx.logger.databasePool is using.`

   `tbmws is the database user name defined in “J2C Authentication Data” which edx.logger.databasePool is using.`

   `tbmws is the database password defined in “J2C Authentication Data” which edx.logger.databasePool is using.`

8 Installing Service Manager Synchronizer

Overview

The Service Manager Synchronizer loads data into the database using the Service Manager process.

You must install Synchronizer after installing and setting up Service Manager.

System Requirements

Install and configure the following software before installing Service Manager Synchronizer:

- JDK 1.4 or greater (use the JDK that comes with your application server. 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.
- Siebel Service Manager
- The Siebel Service Manager database
- If you intend to use the sample Foundation Application provided with Service Manager, you must set up the Foundation Application as described in the *Administration Guide for Siebel* ***Error! Unknown document property name.*** before installing Synchronizer.

Create the Case Management Schema

1. Log in as the oracle user, and change your working directory to the location of the case management SQL file. For example::

   ```
   cd /opt/Siebel/TSM/db/tsm/oracle
   ```

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

   ```
   sqlplus edx_dba/edx@edx.db
   ```

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

   Connected to: Oracle9i Enterprise Edition Release 9.2.0.0.0

4. At the SQL prompt, execute the SQL file to create the case management schema. For example:

```
SQL> @casemgmt_create.sql
SQL> @csm_index.sql
SQL> exit
```

Installing Synchronizer

To install Service Manager Synchronizer:

1. Unzip $TSM_HOME/TSM-zips/ETL.zip. This creates the following directory structure:

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

Directory Name	Description
config	This directory contains the following two sub-directories: castor - 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. etl - 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	Contains sample data files used by Service Manager Synchronizer as defined in the standard tsmproperties.xml file.
lib	Contains all required JAR files needed to run Service Manager Synchronizer. The default configuration is designed to support Oracle.
buildrun.xml	This file contains the commands used to run Service Manager Synchronizer.

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

   hibernate.properties

Change the following parameters (in bold) as necessary to match your environment:

```
############################################################
## Data source information for Oracle jdbc driver
############################################################
hibernate.connection.driver_class=oracle.jdbc.OracleDriver
hibernate.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

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

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

## Hibernate related settings for SQLServer

############################################################
#hibernate.dialect=net.sf.hibernate.dialect.SQLServerDialect
```

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

   ```
   ANT_HOME= ant home
   JAVA_HOME= java home
   PATH = $JAVA_HOME/bin:$ANT_HOME/bin:$PATH
   export ANT_HOME JAVA_HOME PATH
   ```

4. Run ant to update the configuration:

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

   Note, ant -f buildrun.xml displays the available ant tasks.

5. Run the following script to validate installation/configuration:

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

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

6. Run the following script on a clean database to execute Service Manager Synchronizer:

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

   Again, check the output for any errors or failures, and correct the configuration to resolve these problems, if any. 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 whether 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 view 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 is 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

To configure and tune the system for the best Synchronizer performance

1. Service agreement cache in config/ehcache.xml file – The default cache size value is 300000. This consumes approximately 1.7 – 1.8 GB memory. The application uses no more than 500MB memory excluding the cache. If you can't allocate over 2GB memory for the application, 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 WebSphere 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 WebSphere Server documentation and *Deploying and Customizing J2EE Applications.*

To set up and run the Foundation Application for Service Manager, see Chapter 2, "Setting up the Foundation Application" in the *Administration Guide for Siebel Service Manager*.

9 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.
- UNIX users should also switch user to **`root`**, which is the default owner of the Uninstall directory.

To uninstall Service Manager:

1. Start the database server and application server.
2. Navigate to the application server console. Stop the EARs and uninstall them.
3. Stop the database server and delete the current database instances in the ORACLE_HOME.
4. Stop the application server.
5. Navigate to the **`Uninstall`** folder of your Service Manager home directory, **`$TSM_HOME.`**
6. Launch the Service Manager Uninstaller with the command `./Uninstall_TSM`**.** The dot and slash are required, and there is no space after the slash.

   `./Uninstall_TSM`

   The Uninstall screen appears.
7. 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.

8. Change the directory to your Service Manager home directory and manually remove any remaining files and directories as necessary.

9. Click **Done** to close the uninstaller.

10. Repeat this procedure on your application server and any other installations.

Removing the Databases

Follow the steps below to remove the Service Manager, Case Management, and Payment database tables and indexes. The process involves running SQL scripts as the owner of the database.

Caution: You should be aware that this procedure **completely** removes the database elements and should be used with care.

To remove the Service Manager database for Oracle:

1. Change your working directory to the $TSM_HOME/db/oracle directory.

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

   ```
   sqlplus edx_dba/edx@edx0
   ```

   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, execute the SQL file to drop the schema. For example:

   SQL> @drop_bsl_schema.sql
   SQL> @drop_hierarchy_schema.sql

To remove the Case Management database for Oracle:

1. Change your working directory to the *$TSM_HOME/db/tsm/oracle* directory.

2. At the SQL prompt, execute the SQL file to drop the case management schema. For example:

   SQL> @casemgmt_drop.sql

To remove the Payment database for Oracle:

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

1. Log in as the Oracle user.
2. Change your working directory to **$PAYMENT_HOME/db/oracle**.
3. Start a sqlplus session on the Payment database server.
4. Remove payment tables and stored procedures by running:
   ```
   @drop_payment_db.sql
   ```
5. You should see the following message after running each stored procedure:
   ```
   PL/SQL procedure successfully completed.
   ```

Index