Installation Guide
Self Service for Communications

Sun Solaris Operating Environment™ Software
and the BEA WebLogic® Server

Version 5.0
Data Published:4/21/2005

Siebel Systems, Inc., 2207 Bridgepointe Parkway, San Mateo, CA 94404

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

Proprietary Information

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

Contents

1 Preface

About Customer Self-Service and Siebel Tools

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

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

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

Siebel's Self-Service for Communications includes:

Communications Billing Manager

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

Communications Self-Service Manager

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

Communication Analytics Manager

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

Rate Plan Advisor

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

About This Guide

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

A PDF version of this guide is also available on your product CD-ROM.

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

Self Service Manager Developers Guide	How to extend, develop and otherwise work with the SELF SERVICE MANAGER product.
Billing Manager Data Definition Guide	How to create Data Definition Files (DDFs) for use in indexing your application and extracting data for live presentment.
Billing Manager Presentation Design Guide	How to create Application Logic Files (ALFs) to present statement data for dynamic online display.

Obtaining Siebel Software and Documentation

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

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

If You Need Help

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

Information to provide

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

Please be prepared to provide Technical Support the following information:

Contact information:

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

Product and platform:

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

Specific details about your problem:

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

Contacting Siebel Technical Support

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

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

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

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

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

Customer Central
https://support.edocs.com

Email Support
mailto:support@edocs.com

Escalation process

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

Siebel escalation process has two tiers:

1. **Technical Escalation -** Siebel technical escalation chain ensures access to the right technical resources to determine the best course of action.

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

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

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

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

2 Getting Started

Before You Install; Preparing Your Platform

Before installing Self Service Manager, verify that your platform is ready:

- Install and test required hardware and software for your platform.
- Install and configure Communications Billing Manager.
- 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, and application servers to support the InstallAnywhere GUI.

Overview of the Installation Process

The process of installing and setting up Self Service Manager includes the following steps:

1. Installing Self Service Manager on your database and application servers using InstallAnywhere.
2. Configuring the database server.
3. Configuring the application server.

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

Once you successfully install Self Service Manager and configure your database and application servers, you can customize and deploy your J2EE application.

Configuring Your Database Server

Configuring your database server requires you to:

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

Configuring Your Application Server

Configuring your application server requires you to:

1. Define application server environment variables.
2. Configure JDBC resources for Self Service Manager on your application server.
3. Configure JMS resources for Self Service Manager on your application server.

Customizing and Deploying J2EE Applications

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

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

Self Service Manager System Requirements

Platform Services and Self Service Manager (Solaris/Oracle/WebLogic)

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

The following table lists the specific combinations supported for Self 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 8 with patches 108434-09 and 108435-09 **OR** Sun Solaris 9

HARDWARE

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

JAVA/C++

- Sun Java 2 SDK Standard Edition 1.4.1 (or the version shipped with WebLogic 8.1)
- Sun C++ runtime packages **`SUNWesu`** and **`SUNWlibC`**

SUPPORTED DATABASE SERVERS

New installation of Self Service Manager

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

SUPPORTED APPLICATION SERVERS

- BEA WebLogic Server 8.1 SP3

SUPPORTED BROWSERS

- Netscape Navigator 6.2 or higher
- Microsoft Internet Explorer 5.5 SP2, or 6.0 or higher (on networked PC)

3 Installing Self Service Manager

This chapter provides a step-by-step guide to installing Self 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 Self Service Manager components.

DEFAULT	EXAMPLE
`root:other`	`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 21.

Caution We do NOT recommend administering Self Service Manager with the user and group nobody:nobody.

Installing Self Service Manager

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

We recommend that you install and configure Self 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 Self 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 Self Service Manager in Console* Mode (UNIX)" on page 19 for details.

Platform Services, Billing Manager and Self Service Manager

You must install Self Service Manager using InstallAnywhere. Billing Manager must already be installed.

To install Self 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 as described in "*Obtaining Siebel Software and Documentation*" on page 8.
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 Self Service Manager. If lost, contact Siebel Technical Support at http://support.edocs.com/
8. CHOOSE INSTALL FOLDER: Click **`Next`** to accept the default or specify another directory. Siebel recommends that you install and configure Self Service Manager in the same top-level directory structure, first on the database server, then on the application server.
9. CHOOSE PRODUCT FEATURES: Choose which feature you want to install first, or click **`Custom`** and choose to any combination at the same time.

   Self Service for Communications - Installs all components for Billing Manager and Self Service Manager.

   Custom - Installs any combination of products.
10. CHOOSE SHORTCUT FOLDER: Click **`Next`** to install Self Service Manager in the Siebel program group.
11. PRE-INSTALLATION SUMMARY: Review the screen to confirm your product and version, install folder, product components, and disk space required and available, then click **`Install.`**

    Note that the Self Service Manager install folder is also referred to as TSM_HOME (default is *siebel/TBM*). 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.

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

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

Installing Self Service Manager in Console Mode (UNIX)

The installation procedures in this guide show how to install Self Service Manager using the InstallAnywhere GUI (the default).

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

To install Self Service Manager in Console Mode for UNIX:

1. Navigate to the InstallAnywhere directory for your platform and run the command to invoke InstallAnywhere, using the **`–i console`** flag. For example:

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

   InstallAnywhere displays the banner:

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

2. Respond to each prompt to proceed to the next step as directed. If you want to change something on a previous step, type **`back`**.

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

The Self Service Manager Directory Structure

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

The default Self Service Manager installation directories are:

- Self Service Manager (TSM_HOME): *siebel/TBM*

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

Where to Find Database Components

siebel/TBM/estatement/db contains platform-specific subdirectories for database creation and configuration.

Where to Find Application Server Components

siebel/TBM/estatement/`J2EEApps` contains platform-specific subdirectories for Siebel J2EE and web applications to be deployed to your application server. Be sure to deploy the correct version for your platform.

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.

The Self Service Manager database installation also assumes that you have install Billing Manager.

Siebel recommends that you install and configure Self 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 Self 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 Self Service Manager

Caution The installation and configuration examples shown in this guide use default Self 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 Self Service Manager.

UNIX Permissions for Your Database Server

Before creating the Self Service Manager database with **`edx_admin.sh`**, you should verify that the owner and group permissions (**`userid:groupid`**) of the Self 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 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 the database components with the default owner and group for your platform. After installation, change the user and group ownership of Self Service Manager database server components to that of the **DB Admin user**.

DB ADMIN USER	DB USER
`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 Self Service Manager **instance owner**.

   ```
   chown -R edxadmin:edxadmin /opt/siebel/TBM/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/TBM/estatement/db
   ```

4. Switch user to your database instance owner and run **`edx_admin.sh`** to create the Self 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 Self 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 Self 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	Self Service Manager home path	**opt/siebel/TBM/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**

Using Database Partitioning with Self 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, Self 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. Siebel recommends choosing the range partition on the `Z_DOC_DATE` column.

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

5 Configuring Self 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 Self Service Manager database, you run the database configuration shell script **`edx_admin.sh.`** This topic describes each step in detail.

Tip

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

About edx_admin.sh for Oracle

To create and configure the Self Service Manager production database, you run the *edx_admin.sh* script for database configuration. This script adds database components to the Billing Manager database.

Before running *edx_admin.sh*, you should:

- Make a full backup of your current Billing Manager database.
- Start the database instance that accesses the database you are upgrading.
- Check the status of all user objects. If any of them indicate an INVALID status, contact the database administrator to correct this problem.
- Confirm that all login sessions using the Billing 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 Self 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. Change directory to your Self Service Manager database home directory. For example:

   ```
   cd /opt/siebel/TBM/estatement/db/oracle
   ```

3. Run the following script:

   casemgmt_create.sql (this script is not required for Billing Manager)

If you encounter errors during database creation and configuration, ask your DBA for help to remove the items created by *casemgmt_create.sql* in the partially configured database before configuring the database again.

Upgrading an Existing Database

Run the following data migration scripts, which are located in $TBM_HOME*/db/oracle/migration/452to453*:

- **AddTsmSchema.sql -** Run this script first. If you get one or two messages "name used by an existing object", please ignore them.
- **MigrateData.sql** - This script should not report **any** error messages.

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 Self Service Manager database installed, you can proceed to configure your application server.

6 Configuring the WebLogic Application Server

Overview

This chapter assumes in-depth understanding of and practical experience with application server administration. Consult WebLogic Server documentation at http://edocs.bea.com as necessary.

You must start your WebLogic Server instance and bring up the Administrative Console before you begin this chapter.

Caution If you cannot bring up the WebLogic Console, you will be unable to proceed with configuring your application server for Self Service Manager.

Siebel recommends that you install and configure Self 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 Self Service Manager, do so now.

For distributed environments, ensure that you have any required **database client software** installed on WebLogic Server and any other client machines of your database server.

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

- UNIX permissions for WebLogic Server
- Starting and Stopping WebLogic Server
- Capturing your Unix environment for Self Service Manager

Caution The installation and configuration examples shown in this guide use default Self 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 Self Service Manager.

About the Sample UNIX Domain Used in this Guide

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

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

Caution

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

UNIX Permissions for WebLogic Server

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

This guide uses the example username and password **`edxadmin:edxadmin`** as the application server owner and group for Self Service Manager.

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

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

To reset user and group permissions:

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

   ```
   su - root
   ```

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

   ```
   chown –R edxadmin:edxadmin /export/home/bea
   ```

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

   ```
   su - edxadmin
   ```

Tip

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

Starting and Stopping WebLogic Server

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

About Sourcing Your Configuration

Before you start your server instance, you must edit its WebLogic Server startup script to **source** your customized version of the configuration file **`edx.config`**, which passes your Self Service Manager environment to WebLogic Server at startup. For details, see "*Passing UNIX Environment Data to WebLogic*" on page 32.

Starting and Stopping an Active Application Server

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

The default command-line startup shell scripts are fine for an inactive production environment where there are no running jobs. However, the startup process stops immediately if you enter a **`Ctrl+C`** (often used to force a hard shutdown of the server) in the startup directory, or if you close the terminal session. This can damage your configuration file. Siebel recommends using the web console and/or the SHUTDOWN command to ensure a graceful shutdown.

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

Capturing Your UNIX Environment

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

`$EDX_HOME/bin/edx_config`	Executable shell script prompts you to define environment variables required by your application server
`$EDX_HOME/config/edx_env`	**non-editable** configuration file stores the environment variables you specify in edx_config

$EDX_HOME/bin/edx_config	Executable shell script prompts you to define environment variables required by your application server
$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 WebLogic*" on page 32.

Using *edx_config* to Capture Environment Data

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

You should run *edx_config* any time you need to modify your Self 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. For details, see "Using *edx_config* to Capture Environment Data" on page 30.

Caution

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

You can accept the default values, if appropriate, or enter your own.

To capture environment data with *edx_config* for WebLogic:

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

   ```
   su - edxadmin
   ```

2. Navigate to the **bin** directory for Self 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 for your database home, database username and password, application server, Java home, and application server home. The WebLogic home directory pathname is where its application server files were installed, usually someplace like */opt/bea/weblogicXX* where XX is your WebLogic version number. For example:

```
Enter the database server to use, Oracle (oracle) or DB2 (db2)
: [oracle] [q]
Enter Oracle home directory : [/export/home/oracle] [q]
Oracle DB username : [edx_dba] [q]
Oracle password : [edx] [q]
Oracle DB alias (tnsname) : [edx.db] [q]
Enter the application server to use, weblogic (wl) or
websphere (ws) : [wl] [q]
Enter Java root directory : [/export/home/bea/jdk142_04] [q]
Enter Weblogic Application Server root directory :
[/export/home/bea/weblogic81] [q]
```

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

Updating the Payment and Billing Manager configuration files:

1. Log on as the WebLogic server owner, and change your working directory to *%PAYMENT_HOME%/config*, for example:

   ```
   cd /siebel/TBM/payment/config
   ```

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

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

WebLogic Environment Variables

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

VARIABLE	DESCRIPTION	SOLARIS
APP_OWNER	app server owner	**`edxadmin`**
APP_GROUP	app server group	**`edxadmin`**
APP_PORT	app server port	**`7001`**
ADMIN_PORT	app server admin port	**`7002`**
JAVA_HOME	Java home directory	**`$WLHOME/jdk142_04`**

WebLogic Domain Start Scripts

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

```
mydomain/startWebLogic.sh
```

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

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

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

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

Passing UNIX Environment Data to WebLogic

This section describes how to use **edx.config** to pass your environment data to WebLogic at server startup. For information on **edx_config**, see "*Capturing Your UNIX Environment*" on page 29.

Configuring WebLogic for Self Service Manager

You will edit the WebLogic start scripts for your domain to set values for some domain-specific variables.

Users with clustered installations or with custom domain names need to study the new features of domain configuration in your WebLogic Server documentation at http://edocs.bea.com.

To edit WebLogic start scripts:

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

   ```
   su - edxadmin
   ```

2. Stop WebLogic Server and all application server instances. For details, see "*Starting and Stopping WebLogic Server*" on page 29.

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

   ```
   cd $WL_HOME/user_projects/domains/mydomain/startWebLogic.sh
   vi startWebLogic.sh
   ```

4. Define and export the environment variable for your Self Service Manager home directory. For example:

   ```
   TSM_HOME=/opt/siebel/TBM
   export TSM_HOME
   ```

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

   ```
   set MEM_ARGS="–Xss1m –noclassgc -server -Xms512m -Xmx1024m -
   XX:MaxPermSize=384m -XX:NewSize=128m -XX:MaxNewSize=128m -
   XX:+UseLWPSynchronization -XX:+UseThreadPriorities -
   Xconcurrentio -Xrs"
   ```

Tip Be careful not to over allocate memory to the JVM. System processes like the garbage collector consume available memory, and performance can degrade when the application server is not able to respond to other requests.

6. **Password**: (optional) Set **WLS_USER** equal to your system username and **WLS_PW** equal to your system password for no username and password prompt during server startup. Both are required to bypass the startup prompt.

Caution Instead of using a cleartext password variable, BEA recommends that you use a boot identity file. See WebLogic Server documentation at http://bea.com.

7. Source **edx.config** just after the **STARTMODE** command to set production mode (and immediately before the $**JAVACMD** to start the JVM). The dot and space preceding the pathname are a required part of the syntax. For example:

   ```
   STARTMODE=false
   ```

   ```
   . /opt/siebel/TBM/estatement/config/edx.config
   ```

8. Update the CLASSPATH for Self Service Manager by making sure that the following lines are in the CLASSPATH:

   ```
   CLASSPATH=$CLASSPATH:$EDX_HOME/lib/edx_system.jar
   CLASSPATH=$CLASSPATH:$EDX_HOME/lib/edx_common.jar
   CLASSPATH=$CLASSPATH:$EDX_HOME/lib/xpp3_1_1_2.jar
   CLASSPATH=$CLASSPATH:$INS_HOME/lib/Configuration.jar
   CLASSPATH=$CLASSPATH:$INS_HOME/lib/commons-logging.jar
   CLASSPATH=$CLASSPATH:$INS_HOME/lib/javachart.jar
   CLASSPATH=$CLASSPATH:$INS_HOME/lib/ldeprotocol.jar
   CLASSPATH=$CLASSPATH:$INS_HOME/lib/log4j.jar
   CLASSPATH=$CLASSPATH:$INS_HOME/config
   CLASSPATH=$CLASSPATH:/opt/siebel/TBM/payment/lib/Verisign.jar
   CLASSPATH=$CLASSPATH:/opt/siebel/TBM/payment/lib/jnet.jar
   CLASSPATH=$CLASSPATH:/opt/siebel/TBM/payment/lib/jsse.jar
   CLASSPATH=$CLASSPATH:/opt/siebel/TBM/payment/lib/jcert.jar
   ```

   Insert these lines before the existing CLASSPATH statement.

9. Update the Java options by inserting the following lines:

   ```
   JAVA_OPTIONS="$JAVA_OPTIONS -XX:+UseParallelGC"
   JAVA_OPTIONS="$JAVA_OPTIONS -XX:+UseAdaptiveSizePolicy"
   JAVA_OPTIONS="$JAVA_OPTIONS -
   Djava.protocol.handler.pkgs=com.edocs.protocol|com.edocs.domai
   n.telco.lde.protocol"
   JAVA_OPTIONS="$JAVA_OPTIONS -
   Dcom.edocs.tasks.statementscanner.skipResolver=true"
   ```

10. Update the PATH to provide access to the jikes compiler. Insert this statement before the startWebLogic script echoes the $PATH, and provide the correct path to jikes:

    ```
    PATH=$PATH:/opt/jikes
    ```

11. Save and close **startweblogic.sh**.

Editing XML Properties

Self Service Manager property files

edocs.tsm.properties

Check that the following property is set for your needs:

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

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

tsm.xma.xml

Check, and if necessary, replace the following properties:

- **"url"** with the server name, port number and SID of the deployment database

```
<bean id="TSMDataSource"
class="org.apache.commons.dbcp.BasicDataSource" destroy-
method="close">
    <property name="driverClassName">
      <value>${hibernate.connection.driver_class}</value>
      <!--value>oracle.jdbc.driver.OracleDriver</value-->
    </property>
    <property name="url">
      <value>${hibernate.connection.url}</value>
      <!--value>jdbc:oracle:thin:@localhost:1521:edx1</value-->
    </property>
    <property name="username">
      <value>${hibernate.connection.username}</value>
      <!--value>edx_dba</value-->
    </property>
    <property name="password">
      <value>${hibernate.connection.password}</value>
      <!--value>edx</value-->
    </property>
  </bean>

 <bean id="TSMDataSource"
class="org.springframework.jndi.JndiObjectFactoryBean">
      <property name="jndiName">
        <value>edx.user.databasePool</value>
      </property>
    </bean>
```

7 Configuring Java Resources for WebLogic

Overview

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

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

You must start your WebLogic Server instance and bring up the Administrative Console before you begin this chapter.

Caution If you cannot bring up the WebLogic Console, you will be unable to proceed with configuring your application server for Self Service Manager.

Java Compiler

Specify the path to the jikes compiler for your WebLogic server. Information about jikes can be obtained at http://jikes.sourceforge.net/.

1. Download version 1.20, dated 18 April 2004, for your platform following the directions on the jikes web site.

2. Define jikes as java compiler in the WebLogic admin console: In your WebLogic server configuration specify the Java Compiler: Mydomain → Severs → myserver → General → Java Compiler (bottom of page) = jikes.

3. Done. In a previous section, you edited the WebLogic startup script so that WebLogic can find jikes. **If jikes is not available to WebLogic, Weblogic reports no errors**.

Configuring Java Database Connectivity (JDBC) for Self Service Manager

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

About JDBC Connections for Self Service Manager

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

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

For more 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 Self Service Manager email performance.

Configuring JDBC Connections

Create the following JDBC resources.

JDBC Connection Pools for Self Service Manager

No new connection pools are required for Self Service Manager.

JDBC Datasources for Self Service Manager

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

- **Database type** = Oracle
- **Database Drive**r = BEA Oracle Driver (Type 4) – Not XA

For each connection pool, use the following names and properties:

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

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

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

Configuring Java Messaging Services (JMS) for Self Service Manager

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

- **edxServiceManager** supports asynchronous messaging

About JMS Resources for Self Service Manager

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

Configuring JMS Resources for WebLogic

Use the JMS settings in this chapter to configure JMS settings for WebLogic Server. For general information about configuring Java resources for WebLogic, see WebLogic Server documentation at http://edocs.bea.com.

If your web application does not use Line Item Dispute and Annotation, you need only configure JMS for edxLogger.

Make sure you have chosen the correct properties for the resource you are creating, and that each resource maps to others of the **same name**.

Tip

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

JMS Connection Factories

Create three JMS connection factories, using WebLogic Server documentation at http://bea.com. You can accept the default **Properties** for all three connection factories, or consult your application server administrator to tune these values.

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

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

JMS (JDBC) Stores

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

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

JMS Servers

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

`Name`	`(Persistent) Store`
`edxServiceManagerServer`	`edxServiceManagerStore`

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

JMS Queues

Create the following JMS Queue:

`Name`	`JNDI Name`
`edxServiceManagerQ`	`edx/jms/serviceManagerQ`

Deploying Self Service Manager

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

The EAR files are located at:

Feature	**Location**	**File Name**
Self Service CSR **(optional)**	%TSM_HOME%/*J2EEApps/tbm-csr/*	*ear-tbm-csr.ear*

Consult your BEA WebLogic documentation on how to deploy applications.

To check Self Service Manager, use the following links, using the name of the application server in place of **`localhost`**:

http://localhost:7001/tbmb/login/perform.do and log on to the application

http://localhost:7001/tbmb//manage_services/change_voicemail_password/search.do which is the first page of the wizard.

Where to Go From Here

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

Restart the application server and the Billing Manager scheduler when you are done deploying the EAR files.

Installing the Sample Data

After running the Billing Manager utility to create the sample data environment for Billing Manager, follow these steps to set up the sample data environment for Self Service Manager.

1. Copy the sample data set into the $EDX_HOME*/Input/TBM* directory and run the Indexer job first, then the Hierarchy Synchronizer, Report, and report Indexer jobs, in order. Check to make sure all jobs complete successfully. If the jobs did not run successfully, then check your job's configuration.

2. Initialize the CSR application by running http://{your_server_name}{:your_server_port}/tbmcsr/initialize.jsp. Click submit when the submit button appears.

3. Run http://{your_server_name}{:your_server_port}/tbmcsr, and log in user the username/password combination of edocsadmin/111111 (six ones).

4. Click on Manage CSR and create a new CSR Administrator with username/password combination of importer/111111. This user is used by ETL as well as by service manager to process bulk and asynchronous transactions.

5. Check *sm.xma.xml* configuration to make sure, the following sections are set correctly:

```
 <bean id="SMDispatcher"
class="com.edocs.application.tsm.sm.core.TSMDispatcher">

                <property
name="name"><value>SMDispatcher</value></property>

                <property name="serviceManager"><ref
bean="ServiceManager"></ref></property>

                <property
name="userName"><value>importer</value></property>

                <property
name="userPass"><value>0E4CE6E30D8F03B3C5F258CD2DF8BB0C</value
></property>

                <property name="theMessageQueue"><ref
bean="jms.queue.sm"/></property>

        </bean>


   <bean id="jms.context"
class="com.edocs.common.eai.jms.EIInitialContextProvider">

                <property
name="contextFactoryClass"><value>weblogic.jndi.WLInitialConte
xtFactory</va

lue></property>

                <property
name="contextProviderUrl"><value>t3://{your_server_name}{:your
_server_port_if_any}</value></property>

        </bean>
```

Port number in "t3://{your_server_name}{:your_server_port_if_any}" should be the port on which WebLogic server is listening on.

If the settings in this file are not correct, you will not be able to test bulk transactions.

8 Installing ETL

ETL Installation

This section describes the ETL installation instructions.

System Software Requirements

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

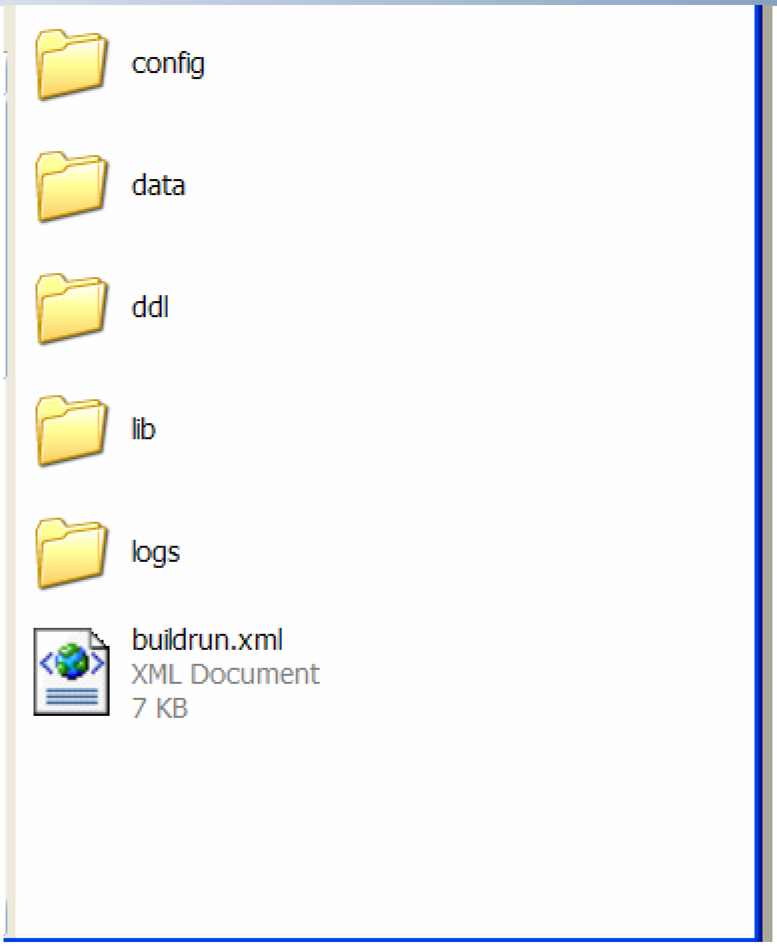
- JDK 1.4 or greater (use the JDK that comes with WebLogic). If you have multiple versions of the JDK installed, make sure that JDK 1.4 is first in the PATH or configure the ETL run script so that it is first in the PATH.
- ANT 1.6 or higher. The default ETL script is an ANT script (*buildrun.xml*). Ant is provided with the Self Service Manager installation, in the TSM_HOME*/views/sample/apache-ant-1.6.1* directory. After installing ant, $ANT_HOME/bin must be in the PATH.

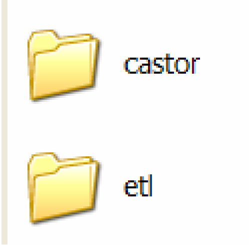
ETL Installation Instructions

Installation Checklist

This section provides a checklist of installation items.

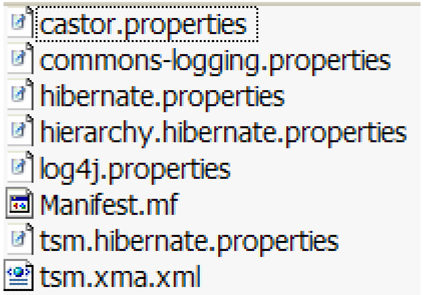
1. Install required Self Service Manager software
   - Billing Manager
   - Self Service Manager
   - Billing Manager and Self Service Manager database.
2. Unzip *ETL.zip*. This will create the directory structure shown below.



Directory Name	Description
config	This directory contains the two sub-directories shown below.  The castor directory contains files are required to map data from the ETL load files to Self Service Manager. The ETL load file DTD descriptions are also contained in the DTD directory. The *etl* directory contains the default *tsmproperties.xm*l file that provides configuration information for ETL. This file should be examined and changed to meet the needs of the installation.
data	The data directory contains sample data files used by ETL as defined in the standard tsmproperties.xml file.

Directory Name	Description
lib	The lib directory contains all required JAR files needed to run ETL. The default configuration is designed to support Oracle.
logs	The logs directory is initially empty. It will contain error file output.
buildrun.xml	This file contains the commands used to run ETL.

ETL must be configured to access the Self Service Manager data source. There are three files archived in TSM_HOME/*lib/tsmetl-config.jar*. The contents are shown below.



castor.properties
commons-logging.properties
hibernate.properties
hierarchy.hibernate.properties
log4j.properties
Manifest.mf
tsm.hibernate.properties
tsm.xma.xml

3. Update the files in the following table to meet the Self Service Manager database requirements:

File Name	Description

hibernate.properties	Change the following parameters to match your environment:
	hibernate.connection.driver_class=oracle.jdbc.driver.OracleDriver
	hibernate.connection.url=jdbc:oracle:thin:@localhost:1521:edx0
	hibernate.dialect=net.sf.hibernate.dialect.OracleDialect
	#hibernate.default_schema=edx0
	hibernate.connection.username=edx_dba
	hibernate.connection.password=edx
	hibernate.show_sql=false
	hibernate.use_outer_join=true
	hibernate.max_fetch_depth=3

hierarchy.hibernate.properties	Change the following parameters to match your environment:
	##
	## Data source information for Oracle jdbc driver
	##
	hibernate.connection.driver_class=oracle.jdbc.driver.OracleDriver
	hibernate.connection.url=jdbc:oracle:thin:@localhost:1521:edx0
	hibernate.dialect=net.sf.hibernate.dialect.OracleDialect
	
	hibernate.connection.pool_size=5
	hibernate.show_sql=false
	hibernate.cglib.use_reflection_optimizer=true
	hibernate.cache.provider_class=net.sf.hibernate.cache.HashtableCacheProvider
	hibernate.connection.username=edx_dba
	hibernate.connection.password=edx

File Name	Description
tsm-xma.xml	Change the TSMDataSource property element to match your installation. The default values are shown below: <bean id="TSMDataSource" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close"> <property name="driverClassName"> <value>oracle.jdbc.driver.OracleDriver</value> </property> <property name="url"> <value>jdbc:oracle:thin:@localhost:1521:edx0</value> </property> <property name="username"> <value>edx_dba</value> </property> <property name="password"> <value>edx</value> </property> </bean>
buildrun.xml	This file provides predefined targets to run the ETL program. Some of the targets require a database connection in order to run SQL initialization. Set the **db_url**, **db_password** and **db_user properties** as shown below. <property name="db_user" value="edx_dba" /> <property name="db_password" value="edx"/> <property name="db_url" value="jdbc:oracle:thin:@localhost:1521:edx0"/>

4. Run **config-update ANT <target>**

5. Run ant to build the configuration. For example:

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

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

6. Check *tsmproprties.xml* to make sure the configuration is correct.

7. Run **etl-ivp ANT etl-ivp** to validate installation/configuration.

8. Run **etl-run ANT etl-run** on a clean database to execute ETL.

9. You can verify that ETL has run successfully by checking that there is data in the following tables:

   EDX_TSM_DEVICE_TYPE: 20 rows

   EDX_TSM_RATEPLAN: 23 rows

   EDX_TSM_RATEPLAN_GROUP: 4 rows

   EDX_TSM_RP_FEATURE: 243 rows

   EDX_TSM_RP_FEATURE_INSTANCE: 1324 rows

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.

10. After ETL runs successfully, log back on to the CSR application and set up your company and admin user(s). The sample data will be under the **`__default__`** company.
11. Log into the TBM/TSM application and verify that the appropriate rate plans, rate plan groups, etc. are available.

9 Uninstalling Self Service Manager

Uninstalling Self Service Manager

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

Uninstall Self 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 Self 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 Self Service Manager:

1. Navigate to the **`Uninstall`** folder of your Self Service Manager home directory, **`$TSM_HOME.`**
2. Launch the Self 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.
3. Click **`Uninstall`**. A second uninstall screen appears showing Self Service Manager components being removed from your machine.

   When the uninstaller is finished, a screen appears listing any items that could not be removed.
4. Change the directory to your Self Service Manager home directory and manually remove any remaining files and directories as necessary.
5. Click **`Done`** to close the uninstaller.
6. Repeat this procedure on your application server and any other installations.

Index