Installation Guide

Siebel Self-Service for Cards

Sun Solaris Operating Environment™ Software
Microsoft Windows 2000® Operating System
and the BEA WebLogic® Server

V5.0.1.1
Date Published: 4.18.2005

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

Printed in the United States of America

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

Table of Contents

Preface

About This Guide

This guide is intended for system administrators and other IT professionals and describes how to install Siebel Self-Service for Cards, configure the third-party platforms that support the Siebel Self-Service for Cards production environment, and deploy Self-Service for Cards 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 Siebel Self-Service for Cards documentation set. For more information about using Siebel Self-Service for Cards, see the following guides:

Siebel Self-Service for Cards Administration Guide	How to set up and run a live Siebel Self-Service for Cards application in a J2EE environment.

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
- 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.

1 Getting Started

Before You Install; Preparing Your Platform

Before installing Siebel Self-Service for Cards, 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.

Overview of the Installation Process

The process of installing and setting up Siebel Self-Service for Cards includes the following steps:

1. Installing Siebel Self-Service for Cards on your 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 Siebel Self-Service for Cards 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:

Define database server environment variables.

1. Create and configure the Siebel Self-Service for Cards database.
2. Connect to your Siebel Self-Service for Cards database before configuring your application server.

Configuring Your Application Server

Configuring your application server requires you to:

Define application server environment variables.

1. Configure JDBC resources for Siebel Self-Service for Cards on your application server.
2. Configure JMS resources for Siebel Self-Service for Cards on your application server.

Customizing and Deploying J2EE Applications

After installing Siebel Self-Service for Cards and configuring your database and application servers, you can:

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

Siebel Self-Service for Cards System Requirements

This guide assumes you are installing Siebel Self-Service for Cards on a Solaris or Windows 2000 operating system, Oracle database, and WebLogic application server.

Solaris/Oracle/WebLogic

The following table lists the specific combinations supported for Siebel Self-Service for Cards. **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

- Disk space (database) 10 GB
- Disk space (application) 60 MB
- Sun SPARC platform
- Swap space 512 MB per CPU (1 GB recommended)
- RAM 512 MB per CPU (1 GB recommended)

JAVA

- Sun Java 2 SDK Standard Edition 1.4.1 (version shipped with WebLogic 8.1)

ANT

- Ant version 1.6.1

JIKES

- jikes-1.20-1

XVFB

- X Virtual Frame Buffer

SUPPORTED DATABASE SERVERS

New installation of Siebel Self-Service for Cards

- 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)

Windows 2000/Oracle/WebLogic

The following table lists the specific combinations supported for Siebel Self-Service for Cards. **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

- Windows 2000 Server with Service Pack 4

HARDWARE

- Disk space (database) 10 GB
- Disk space (application) 60 MB
- Pentium (or compatible class processor) running at 800Mhz or faster
- Swap space 512 MB per CPU (1 GB recommended)
- RAM 512 MB per CPU (1 GB recommended)

JAVA

- Sun Java 2 SDK Standard Edition 1.4.1 (version shipped with WebLogic 8.1)

ANT

- Ant version 1.6.1

JIKES

- jikes-1.20-1

SUPPORTED DATABASE SERVERS

New installation of Siebel Self-Service for Cards

- 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)

2

Installing Siebel Self-Service for Cards with InstallAnywhere

This chapter provides a step-by-step guide to installing Siebel Self-Service for Cards 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

You must have **root** privilege on each server to install and uninstall Siebel Self-Service for Cards 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 23.

Caution Siebel does NOT recommend administering Siebel Self-Service for Cards with the user and group nobody:nobody.

Installing Siebel Self-Service for Cards

You must install Siebel Self-Service for Cards on all your application servers using InstallAnywhere. If you operate in a distributed environment, after installing Siebel Self-Service for Cards on an application server, copy the database scripts (the content of the DB folder) over to your database server.

You have the option to install Siebel Self-Service for Cards 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 Siebel Self-Service for Cards in Console *Mode (UNIX)*" on page 21 for details.

Installing Siebel Self-Service for Cards

To install Siebel Self-Service for Cards 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. Solaris: Launch InstallAnywhere by running **`cardmain.bin.`**
   Windows: Launch InstallAnywhere by running **`cardmain.exe.`**

5. INTRODUCTION: InstallAnywhere recommends that you quit all programs before installing.

6. ENTER SERIAL NUMBER provided when you purchased Siebel Self-Service for Cards. If lost, contact Siebel Technical Support at http://support.edocs.com/

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

8. CHOOSE PRODUCT FEATURES: Choose which feature you want to install first, or click **`Custom`** and choose to any combination at the same time:

   Card Manager - Installs all components for Siebel Self-Service for Cards.

   Custom - Installs any combination of products.

9. CHOOSE SHORTCUT FOLDER: Click **`Next`** to install Siebel Self-Service for Cards in the Card Manager program group (Windows only).

10. 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 Siebel Self-Service for Cards install folder is also referred to as EDX_HOME.

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

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

12. Repeat the installation for other Siebel Self-Service for Cards servers on your network as necessary.

Installing Siebel Self-Service for Cards in Console Mode (UNIX)

The installation procedures in this guide show how to install Siebel Self-Service for Cards 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 Siebel Self-Service for Cards 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:

   ```
   ./Card Managerins.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 Siebel Self-Service for Cards home directory (**`$EDX_HOME)`** in response to the installation prompts.

The Siebel Self-Service for Cards Directory Structure

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

The default Siebel Self-Service for Cards installation directories are:

- Platform Services */Siebel/CardManager*

Where to Find Database Components

/Siebel/CardManager/db contains specific subdirectories for database creation and configuration.

Where to Find Application Server Components

Siebel/Card Manager/J2EE contains specific subdirectories for SiebelJ2EE and web applications to be deployed to your application servers.

3

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 Siebel Self-Service for Cards in the same top-level directory structure, first on the database server, then the application server.

This chapter provides instructions for configuring your database server to support a **new** Siebel Self-Service for Cards 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 Siebel Self-Service for Cards

Caution — The installation and configuration examples shown in this guide use default Siebel Self-Service for Cards pathnames, privileges, and permissions. If you choose not to accept the default values, make sure your values are consistent on all servers across your installation of Siebel Self-Service for Cards.

UNIX Permissions for Your Database Server

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

   ```
   chown -R edxadmin:edxadmin /opt/Siebel/CardManager
   ```

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/CardManager/db
   ```

4. Switch user to your database instance owner and run **edx_admin.sh** to create the Siebel Self-Service for Cards 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 Siebel Self-Service for Cards 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 Siebel Self-Service for Cards, 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	Siebel Self-Service for Cards home path	**opt/Siebel/CardManager**
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**

4

Configuring the Databases for Siebel Self-Service for Cards

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

Tip

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

About creating the Siebel Self-Service for Cards database

To create and configure the Siebel Self-Service for Cards production database, you run several scripts for database configuration. If you have to abort database setup before it completes successfully, see “Uninstalling Siebel Self-Service for Cards” on page 83.

Before creating the Siebel Self-Service for Cards database, you should:

- Upgrade your database server software **as necessary**.
- 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.

Install the Platform Services (Command Center) Database

For Oracle and Solaris

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 Siebel Self-Service for Cards database home directory. For example:

   ```
   cd /opt/Siebel/CardManager/db/CommandCenterDB
   ```

3. Start the **`edx_admin`** shell script with the command:

   ```
   ./edx_admin.sh
   ```

   Command Center Server Administration Main Menu Version 1.0 appears.

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

4. Select Option 1, **`Sign in Menu`**. A second sign-in screen appears.

5. Enter the username, password, and database name for the OLTP 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. See “Oracle Database Server Environment Variables” on page 25 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 ...> oltp
    [4] Enter the password for SYS user...> change_on_install
   ```

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

6. 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
```

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 Server Administration Main Menu.

7. Select Option 3, **Install Command Center.** The Install menu appears.

```
Install Siebel’s CommandCenter
 ------------------------------------
 [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 Siebel Self-Service for Cards, 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.

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

9. Select Option 4, **`Install Application Database I`**.

   This option creates new Siebel Self-Service for Cards 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.

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

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

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

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

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

12. (Optional) If you want to view log files for your database configuration, select Option 7, **`View Status Log Directory`**. Log files are copied to the database subdirectory of your Siebel Self-Service for Cards home directory, for example

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

13. Select **`Return to Previous Menu`**. The Main menu appears.

14. Select Option 4, **`Initial Data Population`**.

    ```
    ------------------------------------------------
    [1] Import initial data set
    [2] Export Command Center database data
    [R] Return to previous menu
    --------------------------------------------------
    Enter Your Selection: 4
    ```

    The Initial Data Population Menu appears.

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

15. Select Option 1, **`Import Initial Data Set`**.

    This option populates the newly defined database with an administrative dataset that supports the Siebel Self-Service for Cards 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 Command Center database data`, is used for migration; it is not necessary for new installations because the database is still empty.)

16. Select Option **`R`** (Return to Previous Menu), and then press **`Enter`**. The Main menu appears.

17. Select Option **`Q`** (Quit), and then press **`Enter.`**

For Windows 2000 and Oracle

The steps described here assume you have a clean database with no previous instance running. If this is not the case, you must follow the Uninstall steps described later in this section before installing.

Please note the following:

- Either the Administrator or user assigned to the Oracle group (ORA_DBA) should install the database.
- You must execute all command scripts from the Windows Command prompt.
- You must adhere to the Optimal Flexible Architecture for Oracle data files.

Installing

1. Replace the following parameter values in *set_ora_env.bat*:

```
${ORACLE_BASE}
${ORACLE_HOME}
${ORACLE_SID}
${SYS_PASSWORD}
${DB_USERNAME}
${DB_PASSWORD}
```

2. Configure New Database Settings on Windows. Execute the following command from the command prompt;

```
createOracleService.bat
```

   This creates *%ORACLE_SID%* directory under *%ORACLE_BASE/admin* which contains *arch*, *bdump*, *cdump*, *create*, *pfile* and *udump* subdirectories.

3. Configure the Initialization file as follows:

   a. Copy *init.ora* from *%EDX_HOME%\db\CommandCenterDB* into *%ORACLE_BASE%\admin\%ORACLE_SID%\pfile\init%ORACLE_SID%.ora*
   b. Replace the slashes from "/" to "\" and the values for the following parameters:

```
background_dump_dest: Replace the "/" to "\", $ORACLE_BASE,
and $ORACLE_SID
control_files: Replace the  $db_control_file_locn1,
db_control_file_locn2, and $db_control_file_locn3
db_name: replace the $ORACLE_SID
log_archive_dest: Replace the "/" to "\", $ORACLE_BASE, and
$ORACLE_SID
user_dump_dest: Replace the "/" to "\", $ORACLE_BASE, and
$ORACLE_SID
utl_file_dir: Replace the "/" to "\" and $db_install_dir
```

c. Create *init%ORACLE_SID%.ora* under *%ORACLE_HOME%\Database* folder and add the following content:

```
IFILE={location_of_actual_init%ORACLE_SID%.ora}
```

For example, if ORACLE_BASE=c:\oracle and ORACLE_SID=edx0, then:

```
IFILE=c:\oracle\admin\edx0\pfile\initedx0.ora
```

4. Create the Database Service (Note: please specify absolute paths. For example, *c:\oracle\oradata\${REDO_LOG1_LOCATION}*.):

a) Replace following parameter values in *create_db.sql*:

```
${REDO_LOG1_LOCATION}
${REDO_LOG2_LOCATION}
${ORACLE_SID}
${SYSTEM_FILE_NAME}
${TEMP_FILE_NAME}
${UNDO_FILE_NAME}
```

b. Execute *create_db.bat* from the command prompt.

5. Initialize the Database:

a) Replace appropriate parameter values in *init_db.sql*:

```
${ORACLE_HOME}
```

b) Execute *init_db.bat* from the command prompt.

6. Configure the Tablespaces. Specify absolute paths or the file name. For example, *c:\oracle\oradata\${EDX_DATA_FILE_NAME}*.):

a) Replace appropriate parameter values in *configure_ts_w2k.sql*:

```
${EDX_DATA_FILE_NAME}
${EDX_DATA_IDX_FILE_NAME}
${EDX_PWC_DATA_FILE_NAME}
${EDX_PWC_DATA_IDX_FILE_NAME}
${EDX_APP_DATA_FILE_NAME}
${EDX_APP_DATA_IDX_FILE_NAME}
${EDX_CDA_DATA_FILE_NAME}
${EDX_CDA_DATA_IDX_FILE_NAME}
${EDX_LOAD_DATA_FILE_NAME}
${EDX_LOAD_DATA_IDX_FILE_NAME}
${EDX_FS_DATA_FILE_NAME}
${EDX_FS_DATA_IDX_FILE_NAME}
${EDX_STAGE_DATA_FILE_NAME}
${EDX_STAGE_DATA_IDX_FILE_NAME}
```

b) Execute *configure_ts.bat* from the command prompt.

7. Create the User:
   a) Replace appropriate parameter values in *setup_user_w2k.sql*:

   ```
   ${DB_USERNAME}
   ${DB_PASSWORD}
   ```

   b) Execute *setup_user.bat* from the command prompt.

8. Populate the Siebel data by running *populate_db.bat* from the command prompt.

Uninstall Steps:

1. Set the ORACLE_SID variable to the correct one (i.e. the instance which has to be uninstalled). For example, in the command prompt:

   ```
   set ORACLE_SID=<actual %ORACLE_SID% value>
   ```

2. Shut down the database instance. The command is:

   ```
   oradim -shutdown -sid <%ORACLE_SID% value>
   ```

3. Delete the database instance. The command is:

   ```
   oradim -delete -sid <%ORACLE_SID% value>
   ```

4. Remove the %ORACLE_SID% folder under *%ORACLE_BASE%\admin* directory.

5. Remove all the database file locations under *oradata* or other locations.

6. Remove the *init%ORACLE_SID%.ora* and the *PWD%ORACLE_SID%.ora* files under *%ORACLE_HOME%\database*.

Create a New Payment Database

Configuring the Payment database consists of running a script that creates tables and indexes in the Siebel Self-Service for Cards database that Payment uses.

Make sure Solaris Command center database has been created.

Running the Payment database creation script for Oracle in Unix:

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

   ```
   su - oracle
   ```

2. You may have to set the environment variable for the Oracle SID to the Siebel Self-Service for Cards Oracle database instance before starting *sqlplus*. For example:

   ```
   export ORACLE_SID=oltp
   ```

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

   ```
   cd /opt/Siebel/CardManager/db/PaymentDB
   ```

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

5. The Payment database installation menu displays:

   ```
   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 :
   ```

6. 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 Command Center ORACLE_SID ...>edx0
   ```

7. After you enter the ORACLE_SID value, the main menu reappears. Select **`[2] Install Application Database`**.

8. When the script finishes, enter **`[3] Install Application Database II`**.

9. When the script finishes, enter **`[4] Initial Data Population`**. This step populates the Payment database with data required to run Payment.

10. 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.

Running the Payment database creation script for Oracle in Windows:

Make sure Windows Command center db has been created.

1. Replace the following parameters in create_payment_db.bat under %EDX_HOME%\db\paymentdb

   ```
   ${ORACLE_BASE}
   ${ORACLE_HOME}
   ${ORACLE_SID}
   ${DB_USERNAME}
   ${DB_PASSWORD}
   ```

2. Execute *create_payment_db.bat* from the command prompt

3. Stop and restart the listener.

Install the Siebel Self-Service for Cards Transactional Schema (OLTP Instance)

Prerequisites

You should install the Command Center/OLTP Database before installing Transaction schema.

1. Set JAVA_HOME, and ANT_HOME in the system CLASSPATH.

2. Update the system path with ANT_HOME*/bin*.

Installing

1. Extract all the installation files to a single directory.

2. Replace the following parameter values in ***tranusr.properties***:

   ```
   ${ORACLE_SID}  - SID of the Command Center database
   ${SYS_PASSWORD} - SYS password of the database
   ${EDX_USRNM}   - Command Center user name
   ${EDX_PW}      - Command Center user password
   ${TRAN_USRNM}  - Transactional user name
   ${TRAN_PW}     - Transactional user password
   ${COMMON_USRNM} - Common user name
   ${COMMON_PW}   - Common user password
   ```

   CM OLTP Tablespace File Locations:

```
${L_DB_EDX_DATA_TB_FILE_LOC_L} - Large data tablespace location

${L_DB_EDX_INDX_TB_FILE_LOC_L} - Large index tablespace location

${L_DB_EDX_DATA_TB_FILE_LOC_M} - Medium data tablespace location

${L_DB_EDX_INDX_TB_FILE_LOC_M} - Medium index tablespace location

${L_DB_EDX_DATA_TB_FILE_LOC_S} - Small data tablespace location

${L_DB_EDX_INDX_TB_FILE_LOC_S} - Small index tablespace location

${L_DB_EDX_DATA_TB_FILE_LOC_ST} – Stage data tablespace location

${L_DB_EDX_INDX_TB_FILE_LOC_ST} - Stage index tablespace location
```

```
${L_DB_EDX_DATA_TB_FILE_LOC_C} - Customer data tablespace location
${L_DB_EDX_INDX_TB_FILE_LOC_C} - Customer index tablespace location
${L_DB_EDX_DATA_TB_FILE_LOC_V} - Transactional data tablespace location
${L_DB_EDX_INDX_TB_FILE_LOC_V} - Transactional index tablespace location
```

For example:

```
L_DB_EDX_INDX_TB_FILE_LOC_V =e:\oradata
```

Installation creates the data file in e:\oradata\ORACLE_SID\data

e:\oradata should be available to continue installation.

For Solaris: Make sure installation directory and the directory paths provided in property file have give read, write permission to Oracle user. (You must login as Oracle user to the system to run the installation)

Make sure there are no white spaces at the end of each value in the property file.

3. Run **`ant traninstall`** from the installation folder and follow the instructions on the screen:

```
[1]. Create Transactional Tablespaces
[2]. Create Transactional and Common Schema
[3]. Install Transactional Modules
[4]. Create Synonyms In Common Schema
[5]. Run all steps at once (Steps 1, 2, 3, 4)
[Q]. Quit
```

Uninstalling

1. Replace ${ORACLE_SID} parameter to the correct one (i.e. the instance which has to be uninstalled) in *tranusr.properties*

2. Run **`ant trandelete`**.

3. Manually remove the folders in this location: `*ORACLE_BASE*\admin\*ORACLE_SID*.` Also remove the reporting tablespaces, system tablespace and redo log locations.

Install the Reporting Database (OLAP Instance)

Prerequisites:

1. Set JAVA_HOME, and ANT_HOME in the system CLASSPATH.

2. Update the system path with ANT_HOME*/bin*.

Installing Reporting Database (OLAP)

1. Extract all the installation files to a single directory.

2. Replace the following parameter values in `rptusr.properties`:

```
${ORACLE_BASE} - ORACLE_BASE of the Oracle Server
${ORACLE_HOME} - ORACLE_HOME of the Oracle Server
${ORACLE_SID}  - SID of the Reporting database instance to be
created.
${SYS_PASSWORD} - SYS password
${DB_USERNAME} - Reporting db username
${DB_PASSWORD}  -Reporting db password
${DB_CONTROL_FILE_LOCN1}   - First control file location
${DB_CONTROL_FILE_LOCN2}   - Second control file location
${DB_CONTROL_FILE_LOCN3}   - Third control file location
```

For example: `DB_CONTROL_FILE_LOCN3=e:\oradata`

Installation creates the third control file in e:\oradata\ORACLE_SID.

e:\oradata should be available to continue installation

`${SYSTEM_FILE_LOCN}` - Location for system, temporary and undo tablespaces. Ex: - SYSTEM_FILE_LOCN=e:\oradata

Installation creates the files in e:\oradata\ORACLE_SID\data.

e:\oradata should be available to continue installation

`${REDO_LOG_FILE_LOCN}` - Redo-Log file location

For example: REDO_LOG_FILE_LOCN=e:\oradata

Installation creates the redo log files in e:\oradata\ORACLE_SID \redo.

e:\oradata should be available to continue installation)

CM OLAP Tablespace File Locations:

```
${EDX_RPT_LARGE_DATA}  - Large data tablespace location
${EDX_RPT_LARGE_IDX}   - Large index tablespace location
${EDX_RPT_MEDIUM_DATA} - Medium data tablespace location
${EDX_RPT_MEDIUM_IDX}  - Medium index tablespace location
${EDX_RPT_SMALL_DATA}  - Small data tablespace location
${EDX_RPT_SMALL_IDX}   - Small index tablespace location
${EDX_RPT_STAGE_DATA}  - Stage data tablespace location
${EDX_RPT_STAGE_IDX}   - Stage index tablespace location
${EDX_RPT_CUST_DATA}   - Customer data tablespace location
${EDX_RPT_CUST_IDX}    - Customer index tablespace location
${EDX_RPT_TRANS_DATA}  - Transactional data tablespace location
${EDX_RPT_TRANS_IDX}   - Transactional index tablespace location
```

For example: EDX_RPT_TRANS_IDX=e:\oradata

(Installation creates the data file in e:\oradata\ORACLE_SID\data e:\oradata should be available to continue installation)

For Solaris: Please make sure installation directory and the directory paths provided in property file have give read, write permission to Oracle user. (You must login as Oracle user to the system to run the installation)

Make user there are no white spaces at the end of each value in the property file.

3. Run **`ant rptinstall`** from the installation folder and follow the instructions on the screen:

```
[1]. Create Oracle Instance
[2]. Create Reporting Tablespaces
[3]. Create Reporting schema
[4]. Install Reporting schema
[5]. Initial Data Population
[6]. Run all steps at once(Steps 1,2,3,4,5)
[Q]. Quit
```

Caution

If the sample data are to be used as described in “Setting up Sample Data” on Page 79, run only steps 1-3 and Q.

Uninstalling

1. Replace `${ORACLE_SID}` parameter to the correct one (i.e. the instance which has to be uninstalled) in *rptusr.properties*
2. Run **`ant rptdelete.`**
3. Manually remove the folders in this location: *`ORACLE_BASE`*`\admin\`*`ORACLE_SID.`* Also remove the reporting tablespaces, system tablespace and redo log locations.

Configuring Oracle Services

The next step in setting up the database server is to edit two Oracle configuration files that control access to the oltp database to support Command Center jobs.

Tip

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

- **`listener.ora`** includes service names and address of all listeners on a computer, the instance names of the databases for which they listen, and listener control parameters. The address for a server in **`listener.ora`** requires the SID (SID_NAME) of a database server in **`tnsnames.ora`**.

  You need to modify **`listener.ora`** on the **database server** machine.
- **`tnsnames.ora`** includes a list of service names of network databases that are mapped to connect descriptors. It is used by clients and distributed database servers to identify potential server destinations. The address of a given database server in **`tnsnames.ora`** matches the address of a listener for that server in **`listener.ora`**.

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

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

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 = oltp)
    (ORACLE_HOME = /export/home/oracle/product/9.2.0)
    )

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

4. Save and close **`listener.ora`**.

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

   ```
   cd /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`**.

   ```
   oltp.db =
    (DESCRIPTION =
    (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))

   )

   (CONNECT_DATA =
    (SID = oltp)
    )
    )

   olap.db =
    (DESCRIPTION =
    (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))

   )
   ```

```
(CONNECT_DATA =
 (SID = olap)
 )
 )
```

7. Save and close **`tnsnames.ora`**.

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

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

   ```
   lsnrctl stop
   ```

   ```
   lsnrctl start
   ```

10. After the Oracle listener has been restarted, you should see a service handler for the oltp and olap instances.

    ```
    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 configurations.

Connecting to Your Oracle Database

Once you have configured Oracle services, you should now be able to connect to your Siebel Self-Service for Cards 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 database, with arguments for your database username, password, and connection string (database alias). For example:

   ```
   sqlplus  edx_dba/edx@oltp.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 oltp
   ```

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

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

5 Installing and Configuring MQ for FDR Simulator

Installing MQ

Install the IBM MQSeries version 5.3 server components.

Setting up queues in Solaris

1. Login with: **`su - mqm`**

   password: `mqm`

2. Change dir to `/otp/mqm/bin/` and copy the scripts from `EDX_HOME/ODSSimulator/script` folder

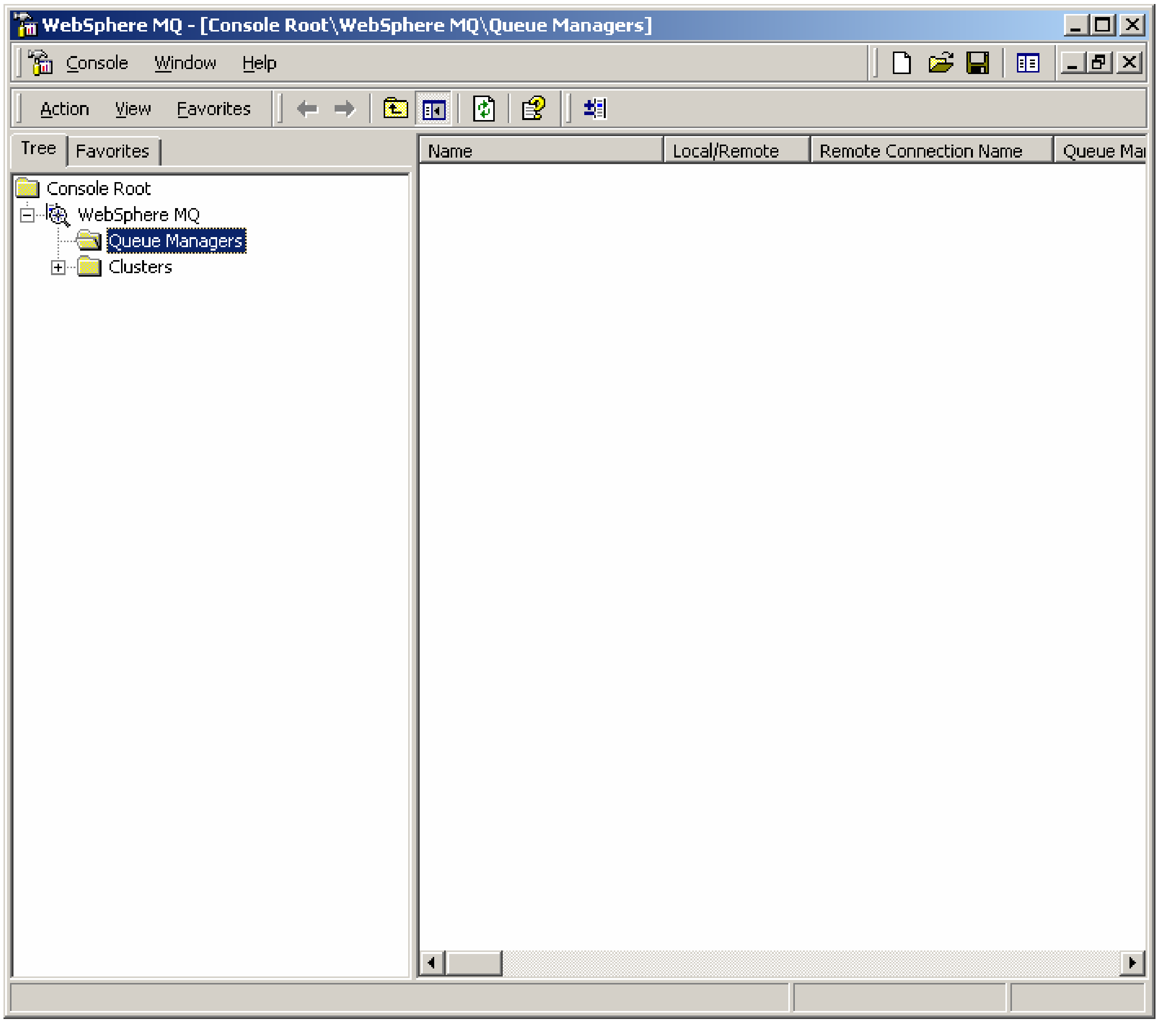
3. Run the following script at command prompt: **`cm_config`**

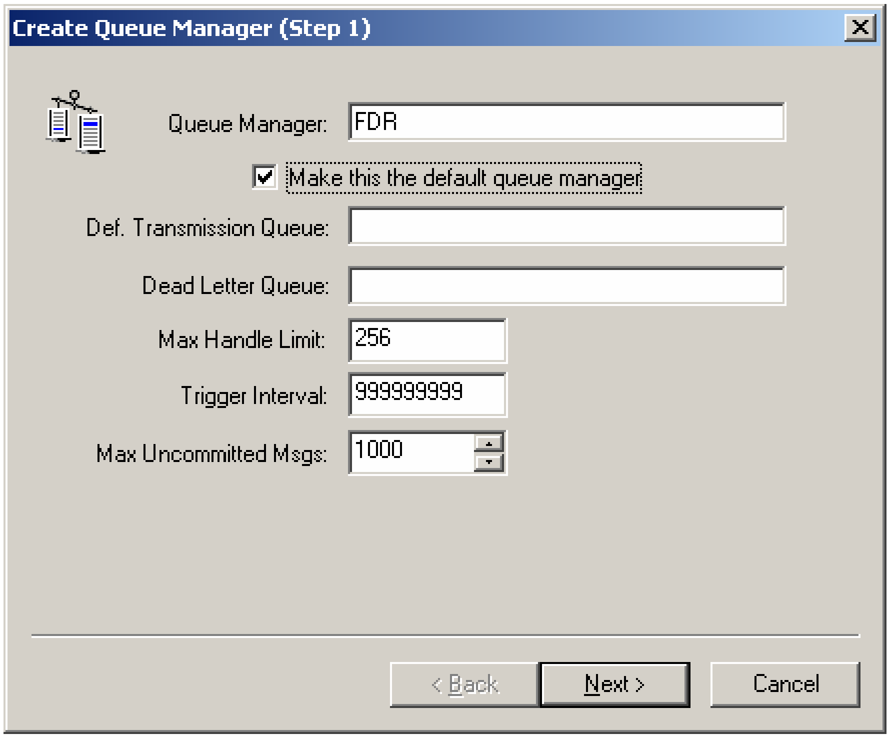
4. Start the Listener for FDR Server channel by giving the following command at the prompt: **`runmqlsr -m FDR -t TCP -p 1414 &`**

Setting up queues in Windows

The following examples show how to create a queue manager and queues in MQ Series to support the FDR simulator. These examples are shown for the Windows system.

Creating the Queue Manager:





Create Queue Manager (Step 1)

Queue Manager: FDR

☑ Make this the default queue manager

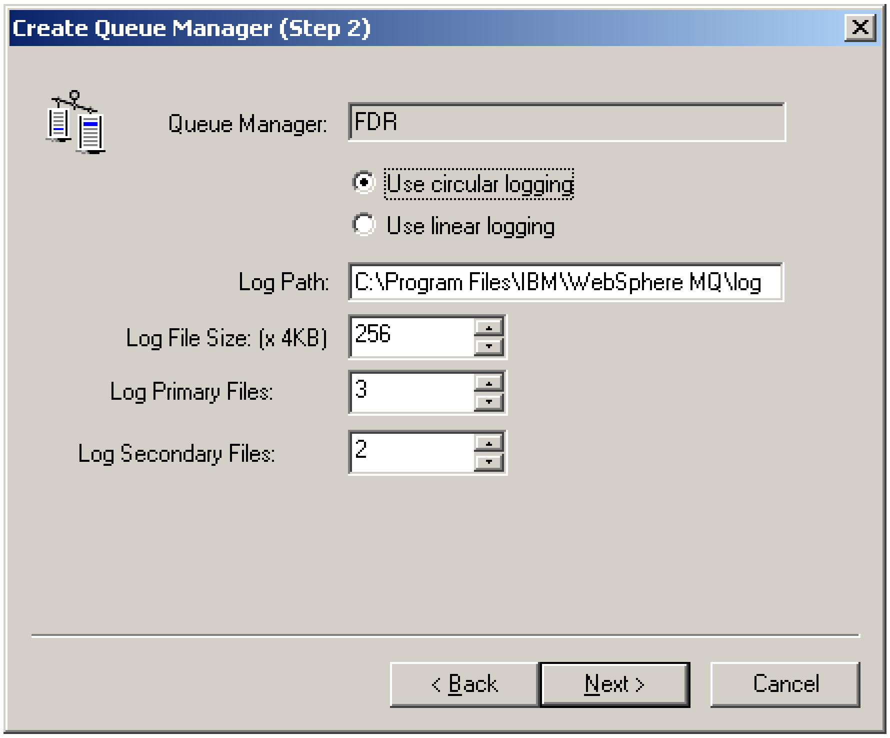
Def. Transmission Queue:

Dead Letter Queue:

Max Handle Limit: 256

Trigger Interval: 999999999

Max Uncommitted Msgs: 1000



Create Queue Manager (Step 2)

Queue Manager: FDR

● Use circular logging

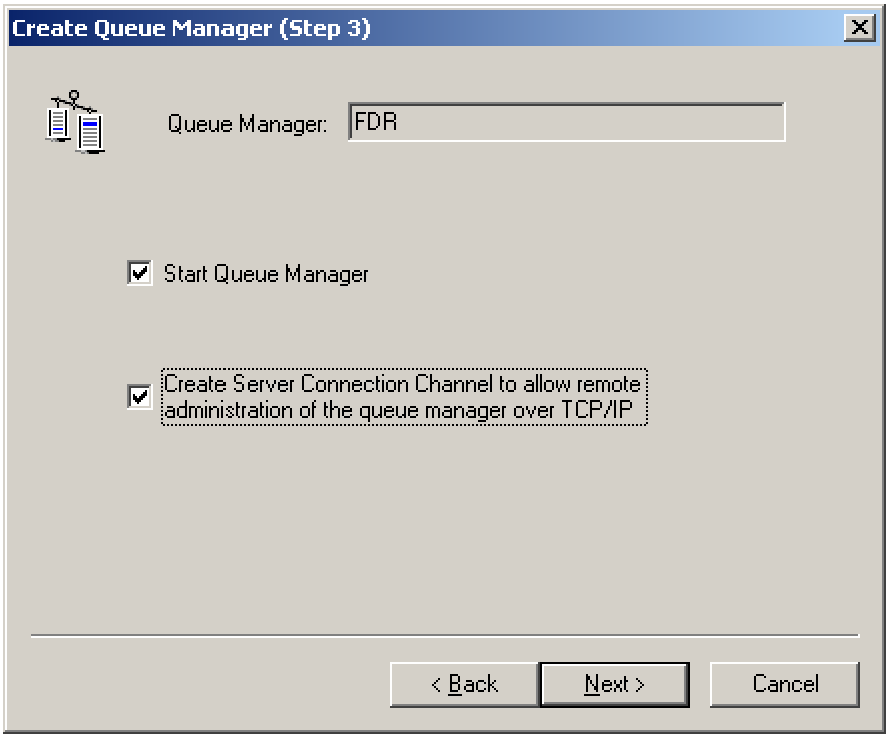
○ Use linear logging

Log Path: C:\Program Files\IBM\WebSphere MQ\log

Log File Size: (x 4KB) 256

Log Primary Files: 3

Log Secondary Files: 2

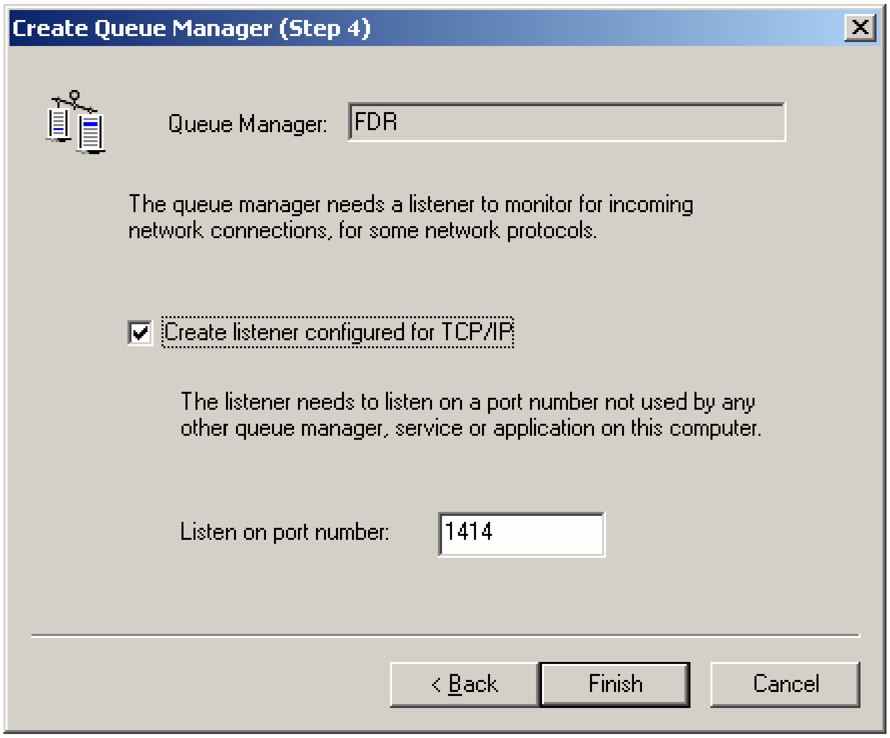


Create Queue Manager (Step 3)

Queue Manager: FDR

☑ Start Queue Manager

☑ Create Server Connection Channel to allow remote administration of the queue manager over TCP/IP



Create Queue Manager (Step 4)

Queue Manager: FDR

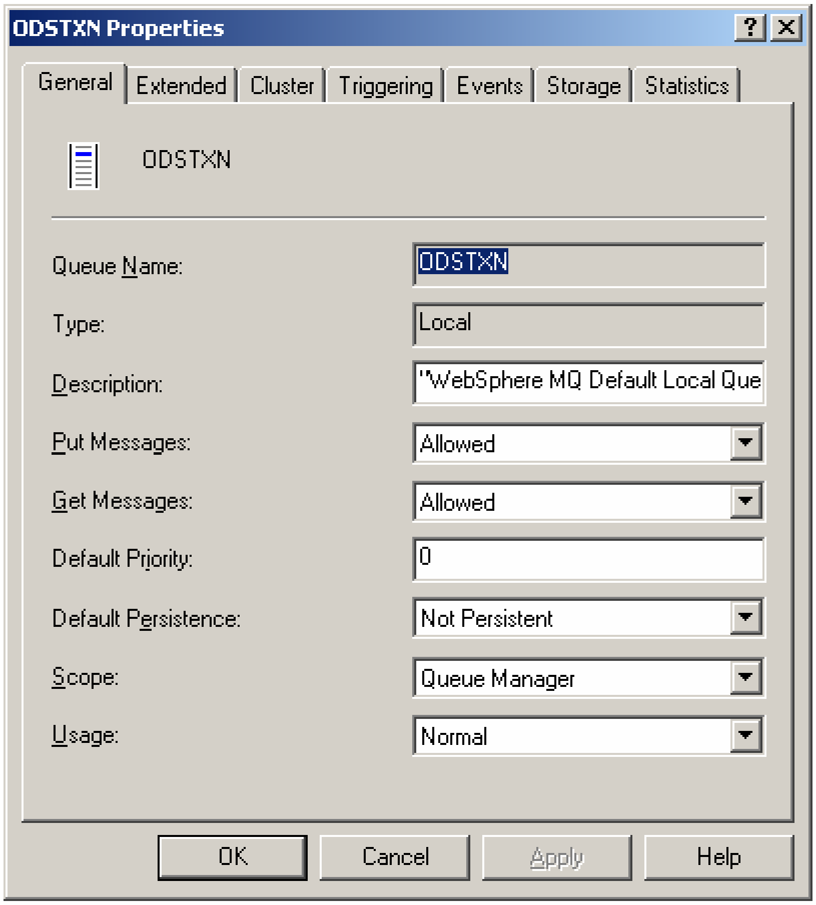
The queue manager needs a listener to monitor for incoming network connections, for some network protocols.

☑ Create listener configured for TCP/IP

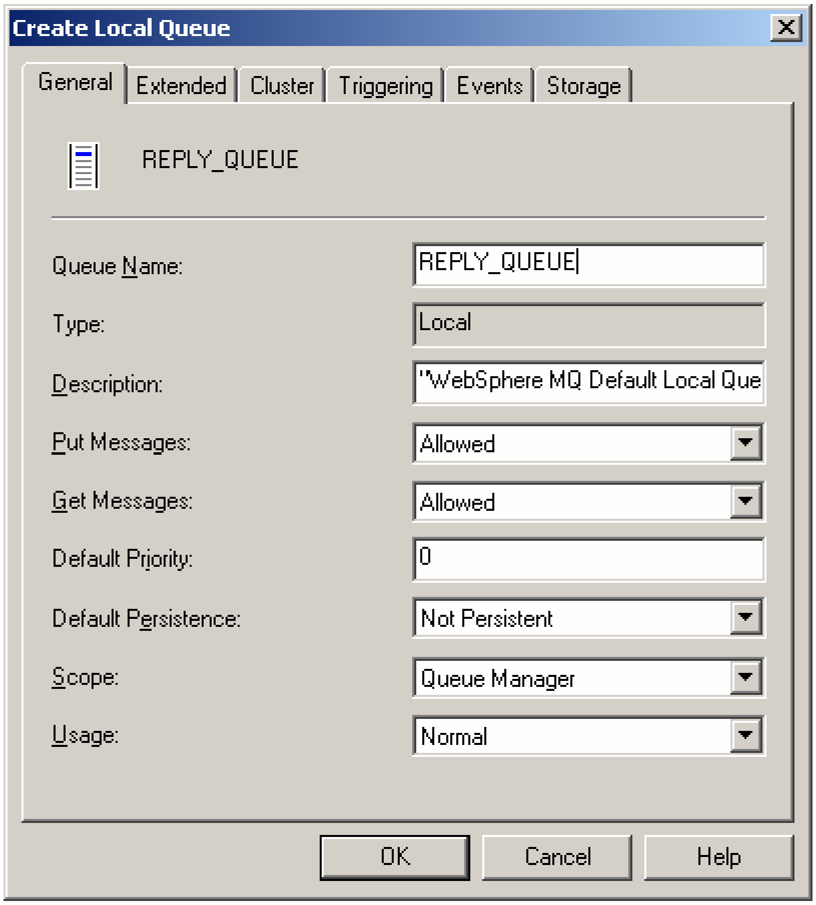
The listener needs to listen on a port number not used by any other queue manager, service or application on this computer.

Listen on port number: 1414

Creating Queues:



Queue Name:	ODSTXN
Type:	Local
Description:	"WebSphere MQ Default Local Que
Put Messages:	Allowed
Get Messages:	Allowed
Default Priority:	0
Default Persistence:	Not Persistent
Scope:	Queue Manager
Usage:	Normal



Queue Name: REPLY_QUEUE

Type: Local

Description: "WebSphere MQ Default Local Que

Put Messages: Allowed

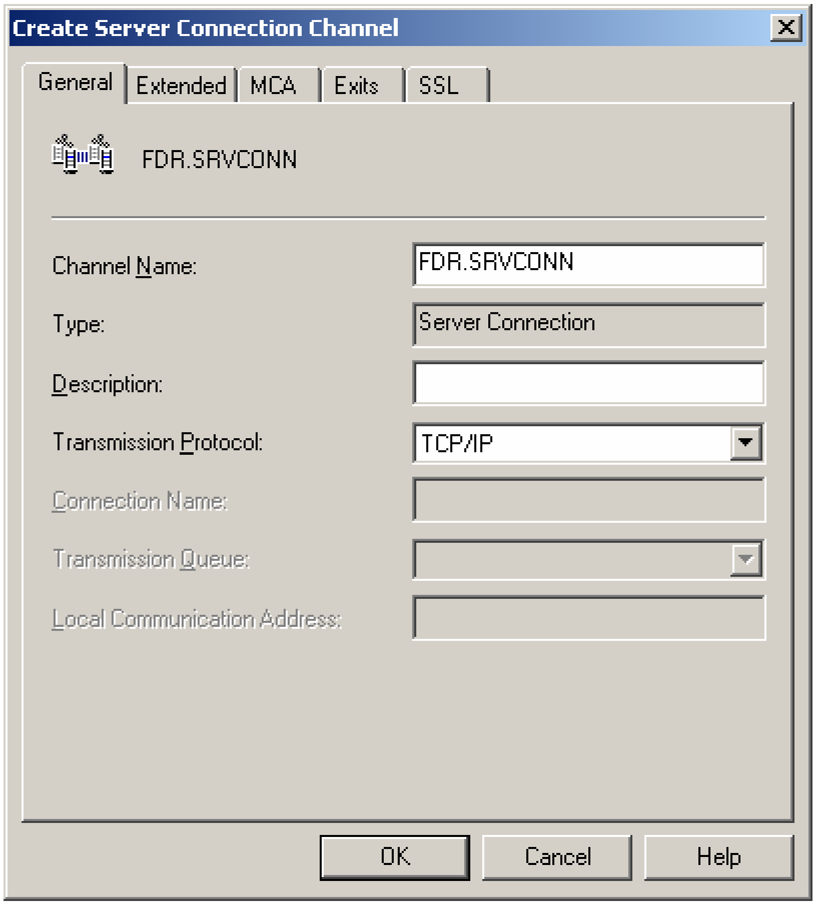
Get Messages: Allowed

Default Priority: 0

Default Persistence: Not Persistent

Scope: Queue Manager

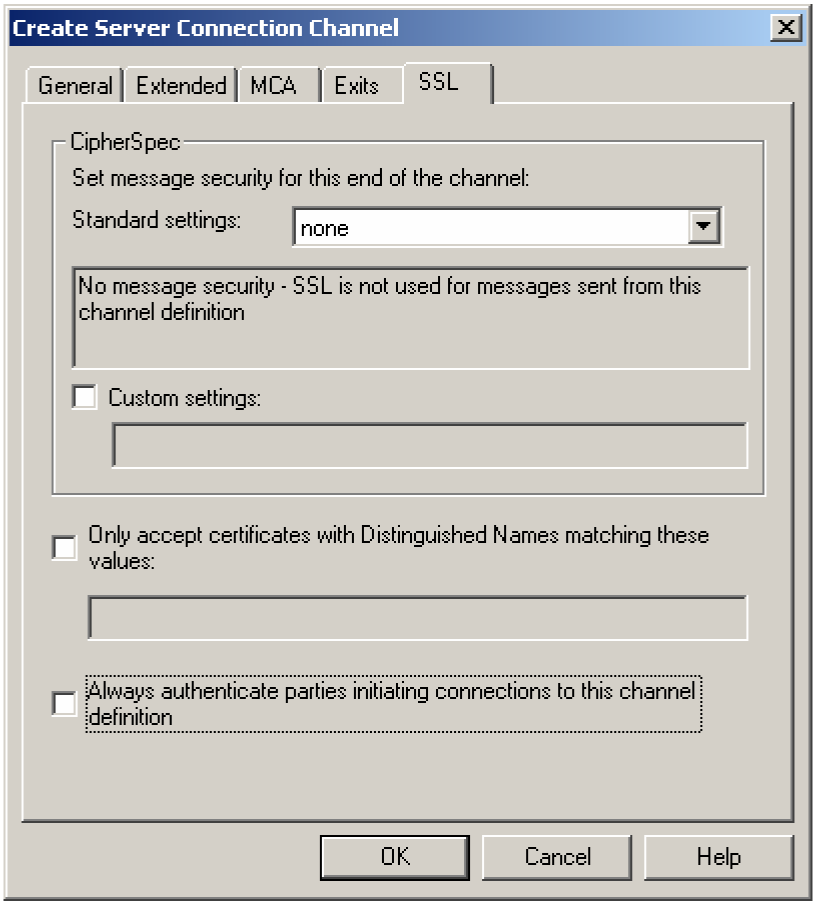
Usage: Normal



Create Server Connection Channel

FDR.SRVCONN

Channel Name:	FDR.SRVCONN
Type:	Server Connection
Description:	
Transmission Protocol:	TCP/IP
Connection Name:	
Transmission Queue:	
Local Communication Address:	



CipherSpec

Set message security for this end of the channel:

Standard settings: none

No message security - SSL is not used for messages sent from this channel definition

□ Custom settings:

□ Only accept certificates with Distinguished Names matching these values:

□ Always authenticate parties initiating connections to this channel definition

Setting up the ODSSimulator (Solaris and Windows 2000):

Prerequisites: Java 1.3.1 or later and Ant 1.6.1

1. Set JAVA_HOME, and ANT_HOME in the system CLASSPATH.
2. Update the system path with ANT_HOME/*bin*.
3. Copy the ODSSimulator directory from the EDX_HOME to the local system.
4. Change your working directory to where you copied the ODSSimulator folder to and type *ant run* at the command prompt.

6

Configuring Java Resources for WebLogic

Creating Servers

You will need to create an administrative server, and three servers for the Card Billing Manager applications. The servers can be on the same or different machines. The server names used in this document are: admin, CommandCenter, cmBPS and cmCSR.

Tip, if all the servers are on one system, you can use the C shell to start all the managed servers from one shell.

Starting and Stopping WebLogic Server

You must be familiar with how to stop and start WebLogic Server and any active web applications for your platform. Please 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`**, thus passing your Siebel Self-Service for Cards environment to WebLogic Server at startup. For details, see Passing UNIX Environment Data to WebLogic on page 53.

Starting and Stopping an Active Application Server

Improperly starting or stopping an application server in an active 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.

Solaris

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.

To start WebLogic in an active production environment, Siebel recommends that you use the **`nohup`** command to ignore hang-ups. This will leave 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.

Windows

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

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

Capturing Your UNIX Environment for Self-Service for Cards

Self-Service for Cards installs several configuration files that you use to define your Self-Service for Cards environment. These configuration scripts are required **only on the Command Center 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/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 53.

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 for Cards 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 51.

Caution

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

To capture environment data with *edx_config* for WebLogic:

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

   ```
   su - edxadmin
   ```

2. Navigate to the **bin** directory for Self-Service for Cards 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.

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.

Capturing Your Windows Environment for Self-Service for Cards

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

%EDX_HOME%\config\edx.config.bat	Script passes the environment data in **edx_env.bat** to your application server when called in your startup script

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

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

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

- edx.config.bat
- edx_env.bat

To edit Windows environment data with edx.config.bat

1. Navigate to **%EDX_HOME%\config** and open **edx_env.bat**.
2. Modify the default settings to reflect your environment. For example:

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

@rem define EDX_HOME
@set EDX_HOME=C:\Siebel\CardManager
@rem define EDX_HOME

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

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

Caution

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

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:

```
startWebLogic.sh
```

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

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, the examples in this guide must be changed accordingly or they may not work.

Passing UNIX Environment Data to WebLogic

All servers

1. Define and export the environment variable for your Card Billing Manager home directory. For example:

   ```
   EDX_HOME=/opt/Siebel/CardManager
   export EDX_HOME
   ```

2. Copy **`com.ibm.mq.jar`** and **`com.ibm.mqjms.jar`** from MQ_HOME/java/lib to the application server and source these files in the classpath.

   Note that if the ODSSimulator directory is on the same machine as the application server, you can source this directory instead:

   ```
   EDX_HOME/ODSSimulator/lib/com.ibm.mq.jar;
   ```

   ```
   EDX_HOME/ODSSimulator/lib/com.ibm.mqjms.jar;
   ```

   See “Configuring ODSSimulator” on page 79 for details.

3. Add some or all of the following to the JAVA_OPTIONS:

-Xbatch -Xcomp -XX:+DisableExplicitGC -XX:+UseParallelGC -XX:+UseAdaptiveSizePolicy-XX:CICompilerCount=1 -XX:CodeCacheMinimumFreeSpace=2M -XX:ReservedCodeCacheSize=64M	You can find information regarding this options here: http://java.sun.com/docs/hotspot/VMOptions.html These parameters only need to be added to startManagedWeblogic.
-XX:+PrintCompilation -XX:+PrintHeapAtGC -XX:+PrintGCDetails	These parameters can be added if you are debugging a core dump or system crash.
-XX:ParallelGCThreads=4	This parameter specifies the number of CPUs in your system. If you have more than one, use the parameter with the correct number.
MEM_ARGS="-Xms1024m -Xmx1024m -XX:MaxPermSize=256m -XX:PermSize=128m"	See *JVM Settings* on page 70.
-Dweblogic.jsp.windows.caseSensitive=true	Note that this parameter contains "windows" even if you are on a Unix system!.

-Dsematree.lib.db.conn_init_sql=	Defect fix and performance improvements
-Dreport.publish=true	This is used if the templates for reporting and emails are to be picked from the published view as opposed to the ear file.

Card Manager and Card Manager Administration (CSR) servers only

1. Add some or all of the following to the JAVA_OPTIONS:

-Djava.awt.headless=true	Fixes bug XPROJ00002393
-DODS_HOST=<MQ Hostname> -DFDR.SRVCONN=FDR.SRVCONN -DPORT=1414	Properties to be added in order for the application to use the FDR or the FDR simulator

Command Center server only

Tip

This section applies to all UNIX platforms supported by 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 for Self-Service for Cards.

Using *edx.config* to source your configuration

`edx.config` is a shell script that you call and process in your application server startup script to pass your Siebel Self-Service for Cards environment (stored in *edx_env*) to WebLogic. This procedure is also called **sourcing your configuration**.

Caution

Do not confuse **`edx.config`** (dot) with **`edx_config`** (underscore), which prompts you to enter the environment data stored in **edx_env**. For details, see Using edx_config to capture environment data

To source your configuration for WebLogic (overview):

Edit `$WL_HOME`*`/user_projects/domains/mydomain/startWebLogic.sh`* to set the server name to be started, set the home directory for Card Manager, set Java options for your JVM, and call and process (source) the configuration script *`edx.config`*.

Sourcing Your Configuration

Users with clustered installations or with custom domain names will need to study the new features of domain configuration in your WebLogic Server documentation at http://e-docs.bea.com/wls/docs70/adminguide/startstop.html.

To edit *startWebLogic.sh* for WebLogic:

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

   ```
   su - edxadmin
   ```

2. Stop WebLogic Server and all application server instances. For details, see Starting and Stopping WebLogic Server.

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

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

4. Set the server name variable for the server instance you wish to start. For example:

   ```
   SERVER_NAME=myserver
   ```

5. Define and export the environment variable for your Card Manager home directory. For example:

   ```
   EDX_HOME=/opt/Siebel/CardManager
   export EDX_HOME
   ```

6. **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=“-ms128m –mx128m –Xss1m –noclassgc”
   ```

Parameter	Description	Example
`-ms`	Sets the initial Java heap size	**`-ms128m`**
`-mx`	Sets the maximum Java heap size	**`–mx128m`**
`-Xss`	Sets the maximum native stack size for any thread	**`–Xss1m`**
`-noclassgc`	Disables the Java class garbage collection	**`–noclassgc`**

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.

7. Add some or all of the following to the JAVA_OPTIONS:

-Dreport.publish=true	This is used if the templates for reporting and emails are to be picked from the published view as opposed to the ear file.

8. **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.

9. Tell WebLogic where to find jikes by editing the WebLogic startup script: Add the path to Jikes to the WebLogic startup script's system path.

```
PATH=/opt/JIKES:$PATH
export PATH
```

   Where opt/JIKES is where you installed Jikes. Note: Be sure to place this path in the script before the line to call the classpath, as in:

```
# Jikes Path
PATH=/opt/FIJI:$PATH
export PATH
CLASSPATH="${WEBLOGIC_CLASSPATH}:${POINTBASE_CLASSPATH}:$
{JAVA_HOME}/jre/lib/rt.jar:${WL_HOME}/server/lib/webservi
ces.jar:${CLASSPATH}"
export CLASSPATH
# Call WebLogic Server
echo "."
echo "CLASSPATH=${CLASSPATH}"
echo "."
echo "PATH=${PATH}"
```

10. Source **edx.config** just after calling the classpath 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
. $EDX_HOME/config/edx.config
```

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

Passing Windows Environment Data to WebLogic

All servers

1. In your **domain** service or startup script, set your Self-Service for Cards home directory, **%EDX_HOME%.**

```
set EDX_HOME=C:\Siebel\CardManager
```

2. Copy **`com.ibm.mq.jar`** and **`com.ibm.mqjms.jar`** from MQ_HOME/java/lib to the application server and source these files in the classpath.

   Note that if the ODSSimulator directory is on the same machine as the application server, you can source this directory instead:

   `EDX_HOME/ODSSimulator/lib/com.ibm.mq.jar;`

   `EDX_HOME/ODSSimulator/lib/com.ibm.mqjms.jar;`

   See “Configuring ODSSimulator” on page 79 for details.

   Add some or all of the following to the JAVA_OPTIONS:

-Xbatch -Xcomp -XX:+DisableExplicitGC -XX:+UseParallelGC -XX:+UseAdaptiveSizePolicy-XX:CICompilerCount=1 -XX:CodeCacheMinimumFreeSpace=2M -XX:ReservedCodeCacheSize=64M	You can find information regarding this options here: http://java.sun.com/docs/hotspot/VMOptions.html These parameters only need to be added to startManagedWeblogic.
-XX:+PrintCompilation -XX:+PrintHeapAtGC -XX:+PrintGCDetails	These parameters can be added if you are debugging a core dump or system crash.
-XX:ParallelGCThreads=4	This parameter specifies the number of CPUs in your system. If you have more than one, use the parameter with the correct number.
MEM_ARGS="-Xms1024m -Xmx1024m -XX:MaxPermSize=256m -XX:PermSize=128m"	See *JVM Settings* on page 70.
-Dweblogic.jsp.windows.caseSensitive=true	Note that this parameter contains "windows" even if you are on a Unix system!.
-Dsematree.lib.db.conn_init_sql=	Defect fix and performance improvements
-Dreport.publish=true	This is used if the templates for reporting and emails are to be picked from the published view as opposed to the ear file.

Card Manager and Card Manager Administration (CSR) servers only

Add some or all of the following to the JAVA_OPTIONS:

-Djava.awt.headless=true	Fixes bug XPROJ00002393
-DODS_HOST=<MQ Hostname> -DFDR.SRVCONN=FDR.SRVCONN -DPORT=1414	Properties to be added in order for the application to use the FDR or the FDR simulator

Command Center server only

`edx.config.bat` is a script that you call and process in your application server startup script to pass your Siebel Self-Service for Cards environment to WebLogic.

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

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

To pass your Siebel Self-Service for Cards environment to WebLogic (overview):

1. Determine whether you wish to start WebLogic as a Windows Service or directly from the startup script. Use the appropriate procedure for your service or startup scripts.

2. In your **domain** service or startup script, set your Self-Service for Cards home directory, **%EDX_HOME%.**

   ```
   set EDX_HOME=C:\Siebel\CardManager
   ```

3. Tell WebLogic where to find jikes by editing the WebLogic startup script: Add the path to Jikes to the WebLogic startup script's system path:

   ```
   set PATH=C:\jikes-1.20\bin;%PATH%
   ```

   Where C:\JIKES is where you installed Jikes. Note: Be sure to place this path in the script before the line to call the classpath, as in:

   ```
   @REM Jikes Path

   set PATH=C:\jikes-1.20\bin;%PATH%

   set
   CLASSPATH=%WEBLOGIC_CLASSPATH%;%POINTBASE_CLASSPATH%;%JAV
   A_HOME%\jre\lib\rt.jar;%WL_HOME%\server\lib\webservices.j
   ar;%CLASSPATH%

   @REM Call WebLogic Server

   set JAVA_OPTIONS=%JAVA_OPTIONS% -Xdebug -Xnoagent -
   Djava.compiler=NONE -
   Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=5
   005

   echo .

   echo CLASSPATH=%CLASSPATH%
   ```

4. After calling the classpath in your **domain** service or startup script, call and process the configuration script **edx.config.bat.** This procedure is called **sourcing** your configuration. Source **edx.config.bat** 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

   call "%EDX_HOME%\config\edx.config.bat"
   ```

Configure JDBC Connection Pools

The following table lists the JDBC connection pools you must deploy. Be sure to restart the server if changes to Connection Pools or Datasources are made.

Pool Name	Server
ecsOraclePool	All three
ecsTransSerial	All three
edxLoggerConnectionPool	All three
edxMessagingPool	All three
edxOlapConnectionPool	All three
edxSmfConnectionPool	All three
edxAdminConnectionPool	Command Center only
edxOtlpConnectionPool	Card Billing Manager and Card Self-Service Manager only
NLSConnectionPool	Card Billing Manager and Card Self-Service Manager only

JDBC Connection Pools for all servers

edxLoggerConnectionPool

Use the **Oracle Thin** driver that comes with WebLogic for this connection pool.

General Tab	
URL	jdbc:oracle:thin:@DB_host:DB_port:oltp_DB:SID. For example, **`jdbc:oracle:thin:@localhost:1521:oltp`**
Driver Classname	**`oracle.jdbc.driver.OracleDriver`**
Database User	**`edx_tran`**
Password	**`edx_tran`**

After the wizard completes, go to the Configuration page to make adjustments using the values shown in the following table (on the Connections tab, click **`Show`** for Advanced Options):

Connections Tab	
Initial Capacity	**`1`**
Maximum Capacity	**`20`**
Capacity Increment	**`5`**
Login Delay	**`1`**
Statement Cache Size	**`300`**
Test Frequency	**`60`**
Allow Shrinking	**`True`** (box checked)
Shrink Frequency	**`15`**
Test Reserved Connections	**`TRUE (checked)`**
Test Released Connections	**`FALSE (unchecked)`**
Test Table Name	**`SQL select 1 from dual where rownum < 1`**

Click **`Apply`** to save these values.

ecsTransSerial and ecsOraclePool

Create two JDBC Connection Pool using the **BEA Oracle Driver (Type 4)** – Not XA:

Use the following parameters:

General Tab	
DB Name	Database SID. For example, **`oltp`**
Host Name	*<database server hostname>*
Database User	**`edx_tran`**
Password	**`edx_tran`**

After the wizard completes, go to the Configuration page to make adjustments using the values shown in the following table (on the Connections tab, click **`Show`** for Advanced Options):

Connections Tab	
Init SQL	**`SQL alter session set nls_sort = 'generic_baseletter'`**
Test Table Name	**`SQL select 1 from dual where rownum < 1`**

edxOlapConnectionPool

Create a JDBC Connection Pool using the **BEA Oracle Driver (Type 4)** – **Not XA** for **all servers**.

Db name:	SID, for example **`olap`**
Hostname:	*<database server hostname>*
Database user name:	**`edx_cc_rpt`**
Password:	**`edx_cc_rpt`**

edxMessagingPool

Use the **BEA's Oracle Driver (Type 4)** (NOT XA) when creating the new Connection Pool.

Db name	**`oltp`**
Hostname	<oltp database host>
Database user name	**`edx_tran`**
Password	**`edx_tran`**

Connections Tab	
Initial Capacity	**`1`**
Maximum Capacity	**`20`**
Capacity Increment	**`5`**
Login Delay	**`1`**
Statement Cache Size	**`300`**
Test Frequency	**`60`**
Allow Shrinking	**`True`** (box checked)
Shrink Frequency	**`15`**
Test Reserved Connections	**`TRUE (checked)`**
Test Released Connections	**`FALSE (unchecked)`**
Test Table Name	**`SQL select 1 from dual where rownum < 1`**

edxSmfConnectionPool

Use the **BEA's Oracle Driver (Type 4)** (NOT XA) when creating the new Connection Pool.

Db name	**`Oltp`**
Hostname	<*database server hostname*>
Database user name	**`edx_tran`**
Password	**`edx_tran`**

Connections Tab	
Initial Capacity	**`1`**
Maximum Capacity	**`20`**

Connections Tab	
Capacity Increment	**5**
Login Delay	**1**
Statement Cache Size	**300**
Test Frequency	**60**
Allow Shrinking	**True** (box checked)
Shrink Frequency	**15**
Test Reserved Connections	**TRUE (checked)**
Test Released Connections	**FALSE (unchecked)**
Test Table Name	**SQL select 1 from dual where rownum < 1**

JDBC Connection Pools for Card Manager and Card Manager Administration (CSR) servers

NLSConnectionPool

Create a JDBC Connection Pool using the **BEA Oracle Driver (Type 4)** – **Not XA** for Card Billing Manager and Card Self-Service Manager servers

Db name	NLS2
Hostname	NLS-DEV01
Database user name	system
Password	osprey

Connections Tab	
Initial Capacity	**1**
Maximum Capacity	**20**

Connections Tab	
Capacity Increment	**`5`**
Login Delay	**`1`**
Statement Cache Size	**`300`**
Test Frequency	**`60`**
Allow Shrinking	**`True`** (box checked)
Shrink Frequency	**`15`**
Test Reserved Connections	**`TRUE (checked)`**
Test Released Connections	**`FALSE (unchecked)`**
Test Table Name	**`SQL select 1 from dual where rownum < 1`**

edxOltpConnectionPool

Use the **Oracle Thin** driver that comes with WebLogic for this connection pool.

Database name	**`oltp`**
Hostname	*<database server hostname>*
Database user name	**`edx_tran`**
Password	**`edx_tran`**

General Tab	
URL	jdbc:oracle:thin:@DB_host:DB_port:oltp_DB:SID. For example, **`jdbc:oracle:thin:@localhost:1521:oltp`**
Driver Classname	**`oracle.jdbc.driver.OracleDriver`**

Connections Tab	
Initial Capacity	**`1`**
Maximum Capacity	**`20`**
Capacity Increment	**`5`**
Login Delay	**`1`**
Statement Cache Size	**`300`**
Test Frequency	**`60`**
Allow Shrinking	**`True`** (box checked)
Shrink Frequency	**`15`**

Connections Tab	
Test Reserved Connections	**`TRUE (checked)`**
Test Released Connections	**`FALSE (unchecked)`**
Test Table Name	**`SQL select 1 from dual where rownum < 1`**

JDBC Connection Pools for CommandCenter

edxAdminConnectionPool

General Tab	
URL	jdbc:oracle:thin:@DB_host:DB_port:CommandCenter_DB:SID. For example, **`jdbc:oracle:thin:@localhost:1521:oltp`**
Driver Classname	**`oracle.jdbc.driver.OracleDriver`**
Database User	**`edx_tran`**
Password	**`edx_tran`**

After the wizard completes, go to the Configuration page to make adjustments using the values shown in the following table (on the Connections tab, click **`Show`** for Advanced Options):

Connections Tab	
Initial Capacity	**`1`**
Maximum Capacity	**`20`**
Capacity Increment	**`5`**
Login Delay	**`1`**
Statement Cache Size	**`300`**
Test Frequency	**`60`**
Allow Shrinking	**`True`** (box checked)
Shrink Frequency	**`15`**
Test Reserved Connections	**`TRUE (checked)`**
Test Released Connections	**`FALSE (unchecked)`**
Test Table Name	**`SQL select 1 from dual where rownum < 1`**

Click **`Apply`** to save these values.

Configure JDBC Datasources

Datasources for all servers

The following Data sources need to be deployed on all servers.

edxLoggerDataSource

JNDI name	`edx.logger.databasePool`
Pool Name	`edxLoggerConnectionPool`

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

edxReportingDataSource

JNDI name:	`edx.report.databasePool`
Pool Name:	`edxOlapConnectionPool`
Honour Global Transaction:	`False`

ecs

JNDI Name	`jdbc.ecs`
Pool Name	`ecsOraclePool`
Row Prefetch Enabled:	`False`
Enable Two Phase Commit	`False`
Stream Chunk Size	`256`
Row Prefetch Size	`48`

ecs_security

JNDI Name:	`jdbc.ecs_security`
Pool Name:	`ecsOraclePool`
Row Prefetch Enabled:	`False`

Enable Two Phase Commit:	`False`
Stream Chunk Size:	`256`
Row Prefetch Size:	`48`

ecs_serial_trans

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

edxMessagingDataSource

JNDI name:	`edx.messaging.databasePool`
Pool Name:	`edxMessagingPool`

edxSmfDataSource

JNDI name	`edx.smf.databasePool`
Pool Name	`edxSmfConnectionPool`
Honour Global Transaction	`false`

Datasources for Card Manager and Card Manager Administration (CSR) servers

The following data source also must be configured on the Card Billing Manager and Card Self-Service Manager app servers.

edxCardManagerDataSource

JNDI name	`edx.databasePool`
Pool Name	`edxOltpConnectionPool`

edxNLSDataSource

JNDI name	`edx.NLS.databasePool`
Pool Name	`NLSConnectionPool`

Datasources for the CommandCenter server

edxAdminDataSource

JNDI name	**`edx.databasePool`**
Pool Name	**`edxAdminConnectionPool`**

Configuring JMS Resources

Connection Factories:

Name:	CMJMSFactory
JNDI name:	edx/qcf

JMS Store:

You must create a directory called MessageFileStore somewhere, and specify the path to that directory in the Directory parameter.

Name:	MessageFileStore
Directory	*/opt/MessageFileStore*

JMS Server:

Name:	CMJMSServer
Persistent Store	MessageFileStore

JMS Queue:

Within the JMS Server you created, select the Destinations folder, then Select JMS Queue.

Name:	CMJMSQueue
JNDI name:	edx/queue/outbound

Configuring JMS for many WLS servers

1. Once a JMS server is configured right click it and clone the existing server.

   Name it CMJMSQueueBPS1, CMJMSQueueCSR1, etc.

2. Target it to your JVM/WLS server (CommandCenter, CMCSR, CMUSER, etc).

   Every WLS server must have it's own JMS server in a non-clustered environment.

3. Create a destination for each of the JMS servers you created and name it CMJMSQueueBPS1, CMJMSQueueCSR1, etc.

4. Apply and restart. This gets rid of JMS errors in the logs.

ECS JVM Settings

ECS requires minimum memory requirements for deployments. An example of these settings are shown in the common command file

NT: *%WL_HOME%\weblogic81\common\bin\commEnv.cmd.*

UNIX: *$WL_HOME/weblogic81/common/bin/commEnv.sh*

A segment of this file is shown below. Set the **MEM_ARGS** to the shown values:

```
    BEA)
       JAVA_VM=-jrockit
       MEM_ARGS="-Xms1024m -Xmx1024m -XX:MaxPermSize=256m -
   XX:PermSize=128m"
   ;;

     Sun)
       JAVA_VM=-server
       MEM_ARGS="-Xms1024m -Xmx1024m -XX:MaxPermSize=256m -
   XX:PermSize=128m"
```

Be sure to edit the shown sections for both **PRODUCTION_MODE=true** and the **else** section.

BEA Server Java Compiler

This section explains how to make the jikes available to WebLogic. This is required for the EJB stub creations when ECS is initially deployed. Information about jikes can be obtained at http://www-124.ibm.com/developerworks/oss/jikes/.

1. To download the jikes compiler: On the IBM jikes page select ‘downloads’ on the left hand side of the page. Select version jikes-1.20-1.

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.

Setting up XVFB for Solaris (X Virtual Frame Buffer)

XVFB is required for charting to function on the application.

Java's graphic packages are based on java.awt which require X libraries and an X display.

The web server rendering the charts must have a real or virtual X display device and necessary X Libraries loaded. Note: It is not necessary for the web server machine to have an X Display device.

Installing XVFB

If you do not already have XVFB installed on your machine, you can obtain a tarred version at ftp://www.ferret.noaa.gov/special_request/xvfb/solaris/, along with further usage information. (Note that if you download this software package, you may need to use gunzip and tar xvf for unzipping and extraction if you don't have the specialized utility.)

To install X libraries on the Application Server:

1. Untar **`X11R6.tar.gz`** into the **`/`** directory (as root)
2. Execute the following commands:

   `# gzip –d X11R6.tar.gz` (if gzip is installed)

   `# tar xvf X11R6.tar`

Starting up XVFB

1. Startup XVFB after installing it and subsequently, this process need to be done EVERY time the physical web server is restarted
2. If xvfb is down (you can check by saying, ps -ef |grep xv), do the following steps.

   ```
   # XVFB_HOME=/opt/temp/xvfb/XVFB
   ```

   ```
   # export XVFB_HOME
   ```

   ```
   # $XVFB_HOME/Xvfb :1 -co $XVFB_HOME/rgb -fp
   $XVFB_HOME/fonts/misc/,$XVFB_HOME/fonts/Speedo/,$XVFB_HOME/fon
   ts/Type1/,$XVFB_HOME/fonts/75dpi/,$XVFB_HOME/fonts/100dpi/ -sp
   $XVFB_HOME/SecurityPolicy &
   ```

3. If you don’t automate the startup of XVFB in the WebLogic startup script, then from the shell where you want to start WebLogic, execute the following lines:

   ```
   # DISPLAY=localhost:1
   ```

   ```
   # export DISPLAY
   ```

   ```
   # ./startWebLogic.sh
   ```

The XVFB package for Solaris is in **`/opt/temp/xvfb/solarisxvfb.tar`** on avalanche (if you need to set it up on another box).

Automating starting of XVFB

If you need to automate starting XVFB (so you won’t have to type the lines every time the server reboots or have to stop WebLogic) follow these steps:

1. Enter the following commands:

   ```
   # cd /etc/init.d
   ```

```
# vi chart
```

2. Add the following to the chart file:

```
#!/bin/sh

XVFB_HOME=/opt/temp/xvfb/XVFB;export XVFB_HOME $XVFB_HOME/Xvfb
:1 -co $XVFB_HOME/rgb -fp
$XVFB_HOME/fonts/misc/,$XVFB_HOME/fonts/Speedo/,$XVFB_HOME/fon
ts/Type1/,$XVFB_HOME/fonts/75dpi/,$XVFB_HOME/fonts/100dpi/ -sp
$XVFB_HOME/SecurityPolicy & DISPLAY=localhost:1;export DISPLAY


# chmod 777 chart

# cd /etc/rc3.d

# ln  –s /etc/init.d/chart S99chart
```

3. Add the following line to **`startWebLogic.sh:`**

```
DISPLAY=localhost:1;export DISPLAY
```

JTA Timeout

On the WebLogic console, click on your domain in the tree (the default domain is mydomain). Click on the JTA tab, and increase the Timeout Seconds (first parameter on the page) to 7200 seconds.

Failure to do this causes the ETLLoader job to fail during the OLTPDataTransfer task.

WebLogic Server Tuning

On the WebLogic console, expand the Servers tree, and click on one of your servers. Select the Tuning tab, and update the following parameters to the new values:

Stuck Thread Max Time = 7200
Stuck Thread Timer Interval=7200

Update these values on all servers.

Execute Queue Settings

You can use the default settings but will need to change to the values given below to obtain better performance. **This only applies to the CMUSER application server**.

Select the WebLogic server that is running the CMUSER application. On the General tab, scroll to the bottom and click on **`Advanced Options`**. Then scroll to the bottom and click on **`Configure Execute Queues`**. Select the existing Execute Queue (**`weblogic.kernel.Default`**), and enter the following parameters:

Parameter	**Value**
Queue Length	65536
Queue Length Threshold Percent	90
Thread Count	75
Threads Increase	5
Threads Maximum	900
Threads Minimum	30
Thread Priority	5

7

Continuing Installation

Configure Case Management Templates

The default location of the case management templates has been set to */opt/Siebel/CardManager/Templates*.

If you wish to change the location of the templates to anything other than the above, update the *basfeature_detail* table using the following command, which must be run against the oltp database with the Card Member db user credentials.

```
update basfeature_detail
set value = '<Some Path>' where NAME='UploadFileRoot'
```

Make sure you run the `commit` command as well after running the above statement.

Deploying the Ears

There are three EAR files for Card member and two EAR files for the Command Center.

Card Member: *cmuser-1.0.ear*

CSR: ear-cmcsr-1.0.ear

Command Center: *wl-cc.ear*

CM Command Center: *cmcc.ear*

Three servers in one domain is required for the deployment of CM and Platform Services (CommandCenter/Payment) systems. These can be on the same machine or different machines. For convenience we will call these servers CommandCenter, CmBPS and CmCSR.

The ears will have to be deployed as given below in each of the domains:

CommandCenter – *wl-cc.ear* and *cmcc.ear*

CmBPS – *cmuser.ear*

CmCSR – *cmcsr.ear*

Running the Siebel Self-Service for Cards and Command Center Applications

URL to access the **Card Member** app:

```
http://<machine name>:<port>/cmuser/login/show.do
```

URL to access the **Siebel Administrator** (cmcsr) app:

```
http://<machine name>:<port>/cmcsr/login/show.do
```

The bootstrap user id for the first logon as the Card Self-Service Manager application is edocsadm.

URL to access the **Command Center** app:

```
http://<machine name>:<port>/eaDirect
```

XMA Configuration Files

Caution Any changes that you make to an XMA file requires an application server restart in order for the changes to take effect.

To configure system messages (Email)

For CMUSER emails

1. Edit the XML configuration file: *EDX_HOME/modules/messaging/messaging.xma.xml*

   Edit the following properties to specify how messages are sent and who appears as the sender (from):

```
 <bean id="config"
class="com.edocs.common.messaging.config.ConfigBean">


  <property name="from"><value>CSR@myco.com</value></property>
  <property name="retry"><value>12</value></property>
  <property name="smtpHost"><value>EXCHANGE</value></property>


        </bean>
```

For Dynamic links in CMUSER

The emails delivered in the application have links to different tabs, such as the AnswerCenter, PaymentCenter etc.

Before you start the cmuser server, you must modify the *messaging.xma.xml* under the XMA tree in messaging folder. *xma/config/modules/messaging*.

1. Specify the URL for the following parameter:

   ```
   <entry key="LoginUrl">
   <value>www.myco.com/cmuser/login/show.do</value></entry>
   ```

   Replace the highlighted text with the complete URL:

   ```
   http://ip:portnumber
   ```

   Where **`ip`** is the machine name.

   For example:

   ```
   <entry
   key="LoginUrl"><value>http://cmapp:9000/cmuser/login/show.do</
   value></entry>
   ```

   or

   ```
   <entry
   key="LoginUrl"><value>http://1.210.110.15:9000/cmuser/login/sh
   ow.do</value></entry>
   ```

Enabling Impersonation capability in the Card Self-Service Manager application

1. Before starting the application server, modify the 'userUrlPath' property in *EDX_HOME/ xma/config/modules/smf/smfconfig.xma.xml* to update the "card member app url" - (http://<hostname>:<port>/contextpath). For example:

   http://localhost:8001/cmuser

8

Setting up Sample Data

Pre-Populating the OLTP and OLAP Databases

Load sample data before performing any online enrollments. Copy the following files from EDX_HOME/SampleApp folder to the given servers before installing the sample data.

- sampledb.xml
- OLAP schema dump (e.g. demo_olap.dmp)
- routing_table.sql
- userenroll.sql

Configuring ODSSimulator

- Copy **`com.ibm.mq.jar, com.ibm.mqjms.jar`**, **`connector.jar`**, and **`jta.jar`** from MQ_HOME/java/lib to EDX_HOME/ODSSimulator/lib, where MQ_HOME is the directory where you installed MQ Series.

Prerequisites

1. Set JAVA_HOME, and ANT_HOME in the system CLASSPATH.
2. Update the system path with ANT_HOME*/bin*.

To install the sample data:

1. Run `ant` **`–buildfile sampledb.xml.`**
2. Run the three required loads individually (1, 2, and 3) or select 4 and run them in sequence automatically:

```
[1]. Load Customer Accounts

[2]. Load Reporting Data

[3]. Load EPICware Data
```

```
  [4]. Run all steps at once (Steps 1,2,3)

  [Q]. Quit
```

Details about each step are described below.

Loading customer accounts

To load customer account data:

1. Enter the requested data:

```
Enter the name of the OLTP instance -oltp
Enter Transaction Username -edx_cc_tran
Enter Transaction User Password-edx_cc_tran
```

Loading reporting data

To load reporting data:

1. Enter the requested data:

```
Enter Backup Filename-demo_olap.dmp
Enter the name of the OLAP instance -olap
Enter SYSTEM Password-manager
Enter New Reporting Schema-edx_cc_rpt
Enter Old Reporting Schema-edx_cc_rpt
Would you like to restore the data as well (Y/N):- y
```

Loading EPICware data

To load reporting data:

1. Enter the requested data:

```
Enter the name of the OLTP instance-oltp
Enter Command Center Username-edx_dba
Enter Command Center User Password-edx_dba
```

Sample Data for Card Manager and Card Manager Administration (CSR) servers

Account Number	SSN	CVC	User ID	Password	Comments
7771 4848 5385 5838	333231111	234	maryloug	edocsedocs1	
7771 4848 5368 8932	111327777	615	adambrown	edocsedocs1	
7871 1727 9365 1834	555231111	321	joeenglish	edocsedocs1	
7871 1728 5352 2610	888231111	400	eugenee	edocsedocs1	
7771 4848 5396 7807	333765555	523	gregjames	edocsedocs1	
7771 4848 5357 7473	111886666	394	(Specify at enrollment)		Account which can be used as the sample account for enrollment demo
-			csradmin	csradmin1	Card Self-Service Manager Admin account

Functionality Available to Each User

Eligible to Login	Eligible for Direct Bill Pay	Order Con-venience Checks	Dispute Trans-actions	Replacement Card		Primary Name	Secon-dary Name	Prod. Type
				Eligible	Card in Process of being Re-issued			
Y	Y	Y	Y	Y	N	Mary Lou Garner	Mike Iacaboni	21
Y	Y	Y	Y	Y	Y	Greg James III	Lisa James	21
Y	N	Y	Y	Y	-	Adam Brown		7
Y	N	Y	Y	Y		Eugene Everett Sr.		50
Y	N	N	Y	N		Lisa Spizo		3
Y	N	N	Y	N		Kirby JoeJoe English		3

Sample Routing Numbers

To enroll a bank account using the Foundation Application, you can use the following sample routing numbers:

Sample Routing Numbers
291070001
091016566
091000022
291973739
291074719

9

Uninstalling Siebel Self-Service for Cards

Uninstalling Siebel Self-Service for Cards

You can uninstall and remove Siebel Self-Service for Cards components, and deployed J2EE applications using the Self-Service for Cards Uninstaller.

Uninstall 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 Siebel Self-Service for Cards 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 for Cards:

1. Navigate to the **`Uninstall`** folder of your Card Manager home directory, **`$EDX_HOME.`**

2. Launch the Uninstaller:

   ```
   Solaris: ./Uninstall Card Manager.exe (the dot and slash are
   required; no space after the slash)
   Windows: Run Uninstall Card Manager.exe
   ```

   The Uninstall screen appears.

3. Click **`Uninstall`**. A second uninstall screen appears showing Self-Service for Cards 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 for Cards 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.

Removing the Databases

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

Uninstalling deletes the entire OLTP instance, the Command Center schema, and the Siebel Self-Service for Cards OLTP schema.

To completely uninstall the Siebel Self-Service for Cards databases, follow the steps in this chapter. Follow the uninstall procedure for each database instance.

Solaris

To remove the databases for Solaris:

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

   ```
   su – oracle
   ```

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

   ```
   echo $ORACLE_SID
    oltp
   ```

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

   ```
   ORACLE_SID=oltp
   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 files. For example:

   ```
   cd ./product/oradata
   ```

   ```
   rm –rf oltp
   ```

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

Windows

To remove the databases for Windows:

1. Log in as the Windows Administrator to remove files.
2. If the database instance name is incorrect, set it as follows:

   ```
   Set ORACLE_SID=oltp
   ```
3. Shut down the database server.
4. Stop the Oracle Windows oltp service.
5. Delete the Oracles Windows oltp service.

   ```
   Oradim –delete –sid oltp
   ```
6. Locate all files associated with the database instance you defined.

   ```
   cd %ORACLE_BASE$
   dir *.dbf
   ```
7. Use Windows Explorer or the command line to remove the folder and files. For example:

   ```
   Del *.dbf
   ```

Removing the Payment Database

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

Follow the steps below to remove the Payment database tables and indexes. The process involves running an SQL script as the as the owner of the database.

Caution: If you want to remove the Payment database tables and indexes, you must do that before removing the Payment database package. Be aware that this procedure **completely** removes the payment database elements.

To remove the Payment database for Oracle:

Log in as the Oracle user.

1. Solaris: Change your working directory to **`EDX_HOME/db/paymentdb`**
   Windows: **`set ORACLE_SID=oltp`**
2. Start a sqlplus session on the Payment database server.
3. Remove payment tables and stored procedures by running:

   ```
   @drop_payment_db.sql
   ```
4. You should see the following message after running each stored procedure:

```
PL/SQL procedure successfully completed.
```