Oracle® SOA Suite

Installation Guide for BEA WebLogic Server

10*g* Release 3 (10.1.3.3) for UNIX and Microsoft Windows

E10871-01

June 2008

ORACLE®

Oracle SOA Suite Installation Guide for BEA WebLogic Server, 10*g* Release 3 (10.1.3.3) for UNIX and Microsoft Windows

E10871-01

Contents

A   Appendix - Notes on Solaris Installation

Index

Preface

This guide is the primary source of installation information for Oracle SOA Suite.

This preface contains these topics:

- Audience
- Documentation Accessibility
- Related Documentation
- Conventions

Audience

Oracle SOA Suite Installation Guide for BEA WebLogic Server is intended for customers who want to install Oracle SOA Suite.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at

http://www.oracle.com/accessibility/

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, 7 days a week. For TTY support, call 800.446.2398. Outside the United States, call +1.407.458.2479.

Related Documentation

For more information, see these Oracle resources:

- *Oracle Application Server Installation Guide* for your platform
- *Oracle Database Administrator's Guide*

In North America, printed documentation is available for sale in the Oracle Store at

```
http://oraclestore.oracle.com/
```

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

```
http://www.oracle.com/technology/membership
```

If you already have a username and password for OTN, then you can go directly to the documentation section of the OTN Web site at

```
http://www.oracle.com/technology/documentation
```

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Installing Oracle SOA Suite with the BEA WebLogic Server

This chapter provides the requirements and procedures for installing Oracle SOA Suite with BEA WebLogic Server.

This chapter contains these topics:

- Overview
- System and Database Requirements
- Installation and Configuration
- Design-time Deployment Support Oracle SOA Suite 10.1.3.1 on WebLogic 9.2
- Deploying Human Task and Decision Services EAR Files
- Additional Configuration Steps of the BEA WebLogic Server
- Postinstallation Verification Tasks
- Auto Loan Demo
- Limitations, Known Issues, Troubleshooting Tips

> **See Also:** The following documentation after completing installation:
>
> - *Oracle BPEL Process Manager Quick Start Guide*
> - *Oracle BPEL Process Manager Order Booking Tutorial*
> - *Oracle BPEL Process Manager Developer's Guide*
> - *Oracle Application Server Adapter for Files, FTP, Databases, and Enterprise Messaging User's Guide*
> - *Oracle Application Server Adapter Concepts*

Overview

You can install and use Oracle SOA Suite with the BEA WebLogic Server.

The BEA WebLogic Server enables you to set up, operate, and integrate e-business applications across multiple computing platforms using Web technologies. The BEA WebLogic Server includes both the run-time components and the tools to develop and design applications.

Oracle SOA Suite provides a complete set of service infrastructure components for designing, deploying, and managing composite applications. Oracle SOA Suite

enables services to be created, managed, and orchestrated into composite applications and business processes. Composites enable you to easily assemble multiple technology components into one SOA composite application. Oracle SOA Suite plugs into heterogeneous IT infrastructures and enables enterprises to incrementally adopt SOA.

Oracle Business Rules (Business Rules) and Oracle Adapters plug into the Service Infrastructure, a normalized transport infrastructure, make up the Enterprise Service Bus (ESB). With the addition of the Oracle BPEL Process Manager (BPEL) and Human Task service components, the suite forms a complete Business Process Management (BPM) platform.

The following components comprise Oracle SOA Suite:

- Oracle Enterprise Service Bus (ESB)
- Oracle BPEL Process Manager (BPEL)
- Human Task
- Oracle Web Services Manager (OWSM)
- Oracle Business Rules

Oracle SOA Console is the monitoring environment for Oracle SOA Suite. You can run, manage, and test your deployed SOA process using the Oracle SOA Console. Oracle SOA Console provides a Web-based interface for management, administration, and debugging of processes deployed to Oracle SOA Server.

The Installation of Oracle SOA Suite for BEA WebLogic Server consists broadly of the following steps:

- Create the Oracle BPEL Process Manager Schema in the Oracle Database

  This step involves installation of Oracle Database and creation of the required DB schemas for the Dehydration store for BPELPM on Oracle Database.

- Installation of the Oracle SOA Suite 10.1.3.1 for OC4J

  This comes with an embedded OC4J J2EE container. Further steps will configure this Oracle SOA Suite to work on top of the BEA WebLogic Server.

- Apply Oracle SOA Suite Patchset 10.1.3.3 on Oracle SOA Suite 10.1.3.1

  This patchset upgrades the existing 10.1.3.1.0 installation to 10.1.3.3.0.

- Apply Opatch on Oracle SOA Suite 10.1.3.3

  This opatch upgrades 10.1.3.3 Oracle SOA Suite to10.1.3.3 Oracle SOA Suite on WebLogic.

- Configure Oracle SOA Suite on BEA WebLogic Server Version 9.2

  This step involves running a command-based script, which will configure the Oracle SOA Suite installed earlier to run on BEA WebLogic Server. The script performs the following:

  - Creates application server - oracleSOAServer
  - Configures the oracleSOAServer shared libraries with Oracle SOA Suite Binaries
  - Creates and configures the required dataSources/JMS resources.
  - Deploys the required J2EE applications for BPEL Console, BPEL Admininstration, Human WorkFlow, ESB, OWSM, and Business Rules.

The above steps, which are further detailed in Installation and Configuration, summarize the installation and configuration of Oracle SOA Suite on BEA WebLogic 9.2 platform.

System and Database Requirements

Table 1–1 describes the system requirements for using Oracle SOA Suite with the BEA WebLogic Server.

Table 1–1   Oracle SOA Suite System Requirements

Element	Requirement
BEA WebLogic Server	Version 9.2
Oracle SOA Suite for OC4J	Apply SOA Suite patchset 10.1.3.3 on Oracle SOA Suite 10.1.3.1
	Note: Refer to Step 2: Install Oracle SOA Suite Basic 10.1.3.1.0 for OC4J for installing Oracle SOA Suite for OC4J.
Web browsers	Internet Explorer 6.0 or Mozilla Firefox 2.0
Operation systems	Microsoft Windows XP, Microsoft Windows 2003, Red Hat Enterprise Linux release 3, and Red Hat Enterprise Linux release 4
	Note: See the BEA Web site for additional details about using these operating systems with the BEA WebLogic Server.
Dehydration store database	Oracle Database 10g (10.2.0.2) or higher
	Note: This certification matrix reflects the Oracle SOA Suite certification on Oracle Application Server, and may vary with the application server being used. Confirm the certification matrix of the application server with Oracle Database version.

Installation and Configuration

This section describes the steps involved in installing and configuring the Oracle Database, creating a schema in the Database, and installing and configuring BEA WebLogic Server.

This section contains the following topics:

- Step 1: Configure the Oracle Database

- Step 2: Install Oracle SOA Suite Basic 10.1.3.1.0 for OC4J

- Step 3: Create the Oracle SOA Suite Schema in the Oracle Database

- Step 4: Apply SOA Suite Patchset 10.1.3.3

> **Note:**   Oracle Database Lite is automatically installed with the Oracle BPEL Process Manager for Developers install type described in this chapter. However, you *cannot* use Oracle Database Lite as the dehydration store.

- Step 5: Apply Opatch on Oracle SOA Suite 10.1.3.3

- Step 6: Install and Configure BEA WebLogic Server Version 9.2

Step 1: Configure the Oracle Database

Follow these instructions to install Oracle Database 10*g*.

> **Note:** These instructions assume that you have obtained Oracle Database 10*g* version 10.1.0.2 and Oracle Database 10*g* Patch version 10.1.0.5.
>
> For all other Database versions, refer to *http://www.oracle.com/technology/documentation/index.html*.

1. Install Oracle Database 10*g* 10.1.0.2.

2. Open SQL*Plus and log in as a user with the SYSDBA privilege.

3. Shut down the database:

   ```
   SQL> SHUTDOWN IMMEDIATE
   ```

4. Install the Oracle Database 10*g* 10.1.0.5 patch in the same Oracle home in which you installed Oracle Database 10*g*.

5. If using Linux only, then log in as the root user and run the following command from the operating system command prompt:

   ```
   /etc/init.d/init.cssd stop
   ```

6. Start the database in upgrade mode in SQL*Plus:

   ```
   SQL> STARTUP UPGRADE
   ```

7. Run the following script:

   ```
   SQL> @ORACLE_HOME/rdbms/admin/catpatch.sql;
   ```

8. Shut down the database:

   ```
   SQL> SHUTDOWN IMMEDIATE
   ```

9. Restart the database:

   ```
   SQL> STARTUP
   ```

10. Run the following script:

    ```
    SQL> @ORACLE_HOME/rdbms/admin/utlrp.sql;
    ```

Step 2: Install Oracle SOA Suite Basic 10.1.3.1.0 for OC4J

The install instructions to install basic Oracle SOA Suite 10.1.3.1 for OC4J is available at

http://www.oracle.com/technology/software/tech/soa/index.html

You must install Oracle SOA Suite into it's own directory outside of WebLogic. The WebLogic installation will refer to binaries and property files from this installation. This external installation will must be there permanently, it's not a temporary staging area. Even though it also contains OC4J, you will not be starting and stopping it. This is an important prerequisite prior to the WAS install.

> **Note:** In this step, you are required to install only the basic installation of Oracle SOA Suite 10.1.3.1, and not the advanced.

> **WARNING:** **Do not start Oracle SOA Server from the Windows Start Menu or by running the Oracle_Home\bpel\bin\startorabpel script. These actions are not supported.**

Step 3: Create the Oracle SOA Suite Schema in the Oracle Database

> **Note:** The scripts to configure Oracle SOA Suite on the BEA WebLogic Server require that the JAVA_HOME environment parameter be set prior to running the script.

1. Navigate to the Disk1\install\soa_schemas\irca folder in the Oracle SOA Suite Installation Setup files directory.

2. Set ORACLE_HOME to point to the Oracle Database Installation location. For example,

   `set ORACLE_HOME=c:\Oracle10g`

3. Enter `irca.bat` on Windows and `./irca.sh` in Linux.

   This runs the irca script to create the schemas required for BPEL, ESB, and OWSM.

4. Enter `sys` password when prompted.

   The `orabpel`, `oraesb`, and `orawsm` schemas are loaded on the Oracle Database.

Step 4: Apply SOA Suite Patchset 10.1.3.3

You must download the SOA Suite patchset 10.1.3.3 from MetaLink and then apply the patchset on Oracle SOA Suite 10.1.3.1. Perform the following steps:

1. Log in to OracleMetaLink at *http://metalink.oracle.com*. The OracleMetaLink home page is displayed.

2. Click **Patches & Updates**. The Patches & Updates page is displayed.

3. Click **Simple Search**.

4. In the **Search By** field, enter `6148874`. The details of the patchset 6148874 are displayed.

5. Follow the instructions in the patchset to install the patchset on the Oracle SOA Suite 10.1.3.1.

> **Caution:** You should not start/restart the Oracle SOA Suite instance of OC4J server after applying the patch.

6. Shutdown the SOA Suite Post patch upgrade as follows:

For...	Run...
Windows XP	SOA_HOME\opmn\bin> opmnctl stopall
Linux	SOA_HOME\opmn\bin> ./opmnctl stopall

Step 5: Apply Opatch on Oracle SOA Suite 10.1.3.3

You must download the 6511843 patchset from MetaLink and then apply the patchset on Oracle SOA Suite 10.1.3.3.

1. Log in to OracleMetaLink at *http://metalink.oracle.com/*. The OracleMetaLink home page is displayed.

2. Click **Patches & Updates**. The Patches & Updates page is displayed.

3. Click **Simple Search**.

4. In the **Search By** field, enter **6511843**. The details of the patch 6511843 are displayed.

5. Download the `p6511843_101330_Generic.zip` for bug 6511843.

6. Follow the instructions given in the `Readme.txt` file of patch 6511843 and apply the patch on Oracle SOA Suite 10.1.3.3.

Step 6: Install and Configure BEA WebLogic Server Version 9.2

> **Note:** These instructions assume that you have obtained BEA WebLogic Server version 9.2.

1. Install BEA WebLogic Server version 9.2.

    > **Note:** If installing on Linux, then change the permissions using the command `chmod a+x platform921_linux32.bin`. Then, run the `./platform921_linux32.bin` command.

2. Download the Oracle SOA Suite 10.1.3.3 BEA WebLogic Server 9.2 at

    http://www.oracle.com/technology/software/products/ias/htdocs/101310.html and unzip to your local machine. The contents of this zip file are extracted to `WL_SOA_ Installables` folder.

    > **Note:**
    >
    > - The directory to which you download the Oracle SOA Suite should be the same host on which the BEA WebLogic Server is installed.
    >
    > - Unzip the `WL_SOA_Installables` folder as a non-root user (same user as used to install Oracle SOA Suite 10.1.3.1 for OC4J). For example, `Oracle`.
    >
    > - If installing on Linux, then change the permissions to the `WL_ SOA_Installables` folder using the `chmod -R 755 WL_SOA_ Installables` command.

3. Modify the following mandatory installation properties in the `WL_SOA_ Installables\SOADomain.properties` file:

> **Note:** Mandatory properties cannot have a comment tag or contain blank values. Failure to follow this requirement results in errors during installation. Also, ensure that you enter the appropriate information for each of the fields. Any typo will cause errors during installation.
>
> However, the proxy settings properties, such as `PROXY_HOST` is non-mandatory.

Property	Description
`BEA_HOME`	The directory path in which BEA WebLogic Server is installed.
	For example, BEA_HOME=C:\bea.
`WL_HOME`	The directory path WebLogic Server is installed, which is usually under `BEA_HOME`.
	For example, WL_HOME=C:\bea\weblogic92
`JAVA_HOME`	JAVA path of WebLogic.
	For example, JAVA_HOME=C:\bea\jdk150_06
`DOMAIN_HOME`	The path for a new WebLogic domain called SOADomain.
	For example, DOMAIN_HOME=C:\bea\user_projects\domains
`APPS_HOME`	The path where applications and adapters will be deployed from.
	For example, APPS_HOME=C:\bea\user_projects\apps
`SOA_HOME`	The directory path in which Oracle SOA Suite is installed.
	For example, SOA_HOME=C:\product\10.1.3.1\ OracleAS_1\
`DRIVER_TYPE`	The datasource class that the installable utilizes to create datasources for the oracleSOAServer manager server.
	For example, DRIVER_TYPE=oracle.jdbc.xa.client.OracleXADataSource
`DB_URL`	The is the URL to connect to ORABPEL schema.
	For example, DB_URL=jdbc:oracle:thin:@stbbn10.us.oracle.com:1521:orcl
`DB_BPEL_ PASSWORD`	The password for ORABPEL schema in database.
	For example, DB_BPEL_PASSWORD=ORABPEL
`DB_OWSM_ PASSWORD`	The password for ORAWSM schema in database.
	For example, DB_OWSM_PASSWORD=ORAWSM
`DB_ESB_ PASSWORD`	The password for ORAESB schema in database.
	For example, DB_ESB_PASSWORD=ORAESB
`SOA_SERVER_ NAME`	The server, which is created, under SOADomain.
	For example, SOA_SERVER_NAME=oracleSOAServer
`SOA_SERVER_ PORT`	The port where SOA_SERVER_NAME is running.
	For example, SOA_SERVER_PORT = 9700
`PROXY_HOST`	The host name of the proxy server.
	For example, PROXY_HOST=www-proxy.us.oracle.com

Property	Description
PROXY_PORT	The port where the proxy server is running. For example, PROXY_PORT=80
NON_PROXY_HOST	The list of non-proxy hosts that are divided by a | symbol. For example, NON_PROXY_ HOST=*.oracle.com|*.oraclecorp.com|localhost|

4. Run the following script from `WL_SOA_Installables` folder at the operating system command prompt:

For...	Run...
Windows XP	setup.bat
Linux	setup.sh

This script creates the domain folder called `SOADomain` in the `BEA_HOME\user_projects\domains\` directory, which contains the Admin Server (AdminServer) and Oracle SOA Server managed server (**oracleSOAServer**). This configures the required applications, database connections, and adapters.

> **Note:**
>
> - While running the `setup.bat` or `setup.sh` file, set the evironment variable `BEA_HOME` to bea folder. For example, `C:\bea` in Microsoft Windows or `\home\userfolder\bea` in Linux.
>
> - Based on the `BEA_HOME` variable value, the setup script assumes the jdk folder name to be `jdk150_06` and appends this value to `BEA_HOME`, sets it to the `JAVA_HOME` variable, and checks for the path existence in the file structure. For example, `JAVA_HOME=BEA_HOME/jdk150_06`.
>
>   If `JAVA_HOME` path does not exist, then setup file throws a message asking to set the `JAVA_HOME` before running the setup file.
>
> - Based on the `BEA_HOME` variable value, the setup file assumes the WebLogic folder name is `weblogic92` and appends this value to `BEA_HOME`, sets it to the variable `WL_HOME`, and checks for the path existence in the file structure. For example, `WL_HOME=BEA_HOME\weblogic92`.
>
>   If `WL_HOME` path does not exist, then setup file throws a message asking to set the `WL_HOME` before running the setup file.
>
> - Setting `WL_HOME` as environment variable sets the variable `WL_JAR_PATH`, which contains the following value: `WL_HOME\server\lib\weblogic.jar`.
>
>   The `WL_JAR_PATH` is used to load the WebLogic ant task class `weblogic.ant.taskdefs.management.WLSTTask`.
>
>   The setup script assumes that `weblogic.jar` is available at `WL_HOME\server\lib` folder.
>
> - Refer to Appendix - Notes on Solaris Installation for Solaris specific installation information.

5. Start `NodeManager` as follows:

For...	Run...
Windows XP	`BEA_HOME\weblogic92\server\bin\startNodeManager.cmd`
Linux	`BEA_HOME/weblogic92/server/bin/startNodeManager.sh &`

When you start the node manager, it creates a mapping to the SOA docmain, which enables you to start and stop **oracleSOAServer** remotely using admin console. You can also start and stop the node manager from the Windows Services by running the `installNodeMgrSvc.cmd` from the `BEA_HOME\weblogic92\server\bin\` directory.

6. Start BEA WebLogic Server as follows:

For...	Run...
Windows XP	`BEA_HOME\user_projects\domains\BPEL_Domain\bin\startWebLogic.cmd`
Linux	`BEA_HOME/user_projects/domains/BPEL_Domain/bin/startWebLogic.sh &`

This server has to be started before the user can access the SOADomain Administrative Console at the following URL:

`http://`*localhost*`:8001/console`

Installation progress is logged to the *WL_SOA_Installables*`\logs\output.log` file.

7. Start the **oracleSOAServer** managed server by following the startup instructions as follows:

   a. Log in to *http://localhost:8001/console*, using `weblogic` as the username and password. The BEA WebLogic Server Administrative Console window is displayed.

   b. Select **Environment** -> **Servers** -> **oracleSOAServer**. The Settings of oracleSOAServer General page is displayed.
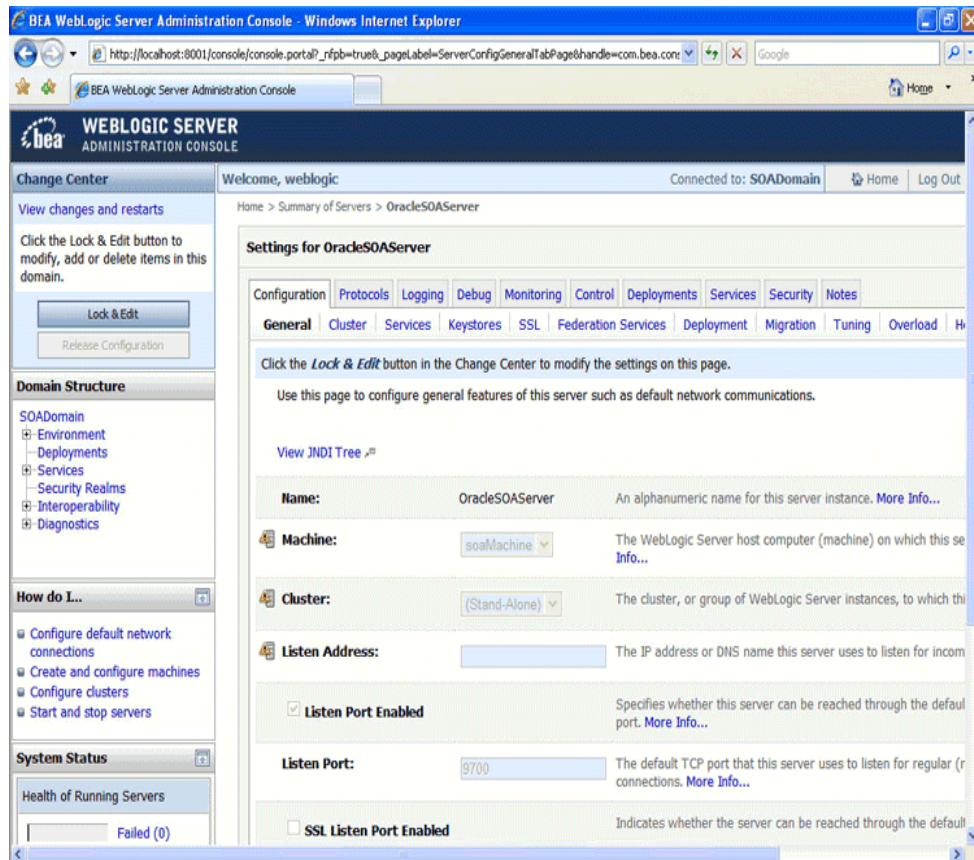
c. Click the **Control** tab. The Settings of OracleSOAServer Control page is displayed.
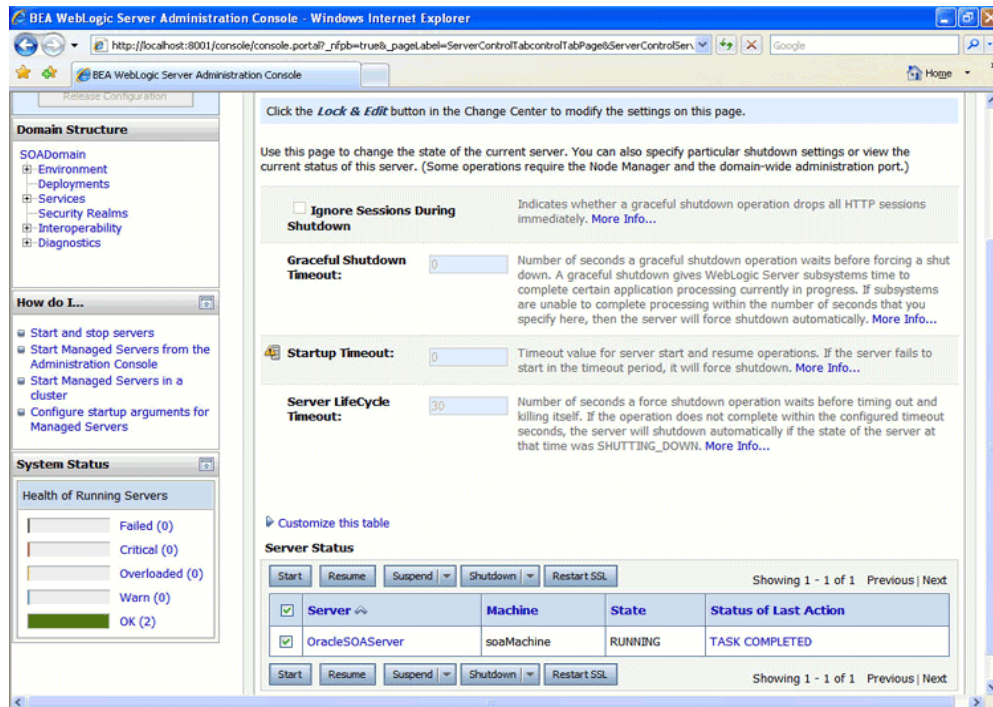
    d. In the Server Status pane, select **oracleSOAServer** and click **Start**. The Server Life Cycle Assistant page is displayed.

    e. Click **Yes**. The oracleSOAServer status shows RUNNING in the Server Status pane.

> **Note:** Do *not* start Oracle BPEL Server from the Windows Start Menu or by running the *SOA_HOME*\bpel\bin\startorabpel script. These actions are not supported.

8. Log in to the BPEL Console at the following URL, using weblogic as the username and password:

http://localhost:9700/BPELConsole

Design-time Deployment Support Oracle SOA Suite 10.1.3.1 on WebLogic 9.2

This section describes the various design-time support functions available on BEA WebLogic Server, for the deployment of J2EE applications in JDeveloper. You can deploy BPELPM components on BEA WebLogic Server by using the following two methods:

- From the BPELPM Developer Prompt Using Ant
- From JDeveloper

From the BPELPM Developer Prompt Using Ant

You can use ant in the BPELPM developer prompt to deploy J2EE applications. This section contains the following topics:

- Prerequisite Checks
- Steps to Deploy Using the BPELPM Prompt

Prerequisite Checks

1. Ensure that bpelPlatform is set to **weblogic_8** in the BPEL_ HOME\bpel\system\config\collaxa-config.xml file.

2. Ensure that the following properties are set in BPEL_ HOME\bpel\utilities\ant-orabpel.properties file:

- platform to **weblogic_8**
- admin.user to valid user in WebLogic realm
- admin.password to the password of the above user
- jndi.url to **t3://<hostname>:9700**
- jndi.InitialContextFactory to **weblogic.jndi.WLInitialContextFactory**

> **Note:** If the admin.user property is not set correctly, then the deployment may throw authentication errors.

Steps to Deploy Using the BPELPM Prompt

Follow these instructions to deploy BPELPM from the developer prompt using ant:

1. Open a BPELPM Developer prompt.

2. Run `ant.sh/bat` from the `BPEL_HOME\bpel\system\appserver\oc4j\ant\bin` directory of the BPEL application.

    > **Note:** For more information, refer to `C:\product\10.1.3.1\OraBPEL_OC4J\bpel\GETTING_STARTED.html`.

The only exceptions to be noted are as follows:

- If the BPEL Process contains any Decision Service applications, UI applications, or Work Flow applications, then these applications will not be automatically deployed in WebLogic Server by the `ant` script.

- The corresponding EAR/WAR files is custom built for WebLogic platform but must be manually deployed on the target server **oracleSOAServer**.

- Use Weblogic Admin console (*http://<hostname>:8001/console*) to deploy the EAR/WAR files to **oracleSOAServer**.

    > **Note:** Refer to Auto Loan Demo for more details.

From JDeveloper

You can also deploy J2EE applications from JDeveloper. This section contains the following topics:

- Prerequisite Checks

- Steps to Deploy Using JDeveloper

Prerequisite Checks

1. Download JDeveloper Studio 10.1.3.3 (`jdevstudio10133.zip`) from

    For Windows - *http://www.oracle.com/technology/software/products/jdev/htdocs/soft10133.html*.

2. Copy the `bpm-services.jar` file from the `SOA_HOME\bpel\system\services\lib` directory to `JDEV_HOME\integration\lib` directory.

3. Copy the `orabpel-ant.jar` file from the `SOA_HOME\bpel\lib` directory to the `JDEV_HOME\integration\lib` directory.

4. Ensure that the following properties are set in in *SOA_HOME*`\bpel\utilities\ant-orabpel.properties` file: Ensure that `bpelPlatform` is set to **weblogic_8** in the *SOA_HOME*`\bpel\system\config\collaxa-config.xml` file.

    - `platform` to **weblogic_8**

    - `admin.user` to valid user in WebLogic realm

    - `admin.password` to the password of the above user

- `jndi.url` to **t3://<hostname>:9700**

- `jndi.InitialContextFactory` to **weblogic.jndi.WLInitialContextFactory**

Creating Connections to Oracle SOA Server

Follow the steps below to create an application server connection and an integration server connection:

1. Create an application server connection of the Standalone OC4J 10.1.3 type.

   - Choose OC4J standalone as server type as there is no plugin available for WebLogic

   - Ignore errors when testing this connection. This is due to OPMN absent on WebLogic

2. Create an Integration Server connection to **hostname:<default_port>**. The default port is as mentioned in the `SOADOMAIN.properties` file.

   - Choose the above-created AppServer connection

   - BPEL and ESB should pass when this connection is tested

Steps to Deploy Using JDeveloper

Follow these instructions to deploy BPELPM from the developer prompt using JDeveloper:

1. From JDeveloper, right-click and deploy the BPEL application into the required domain.
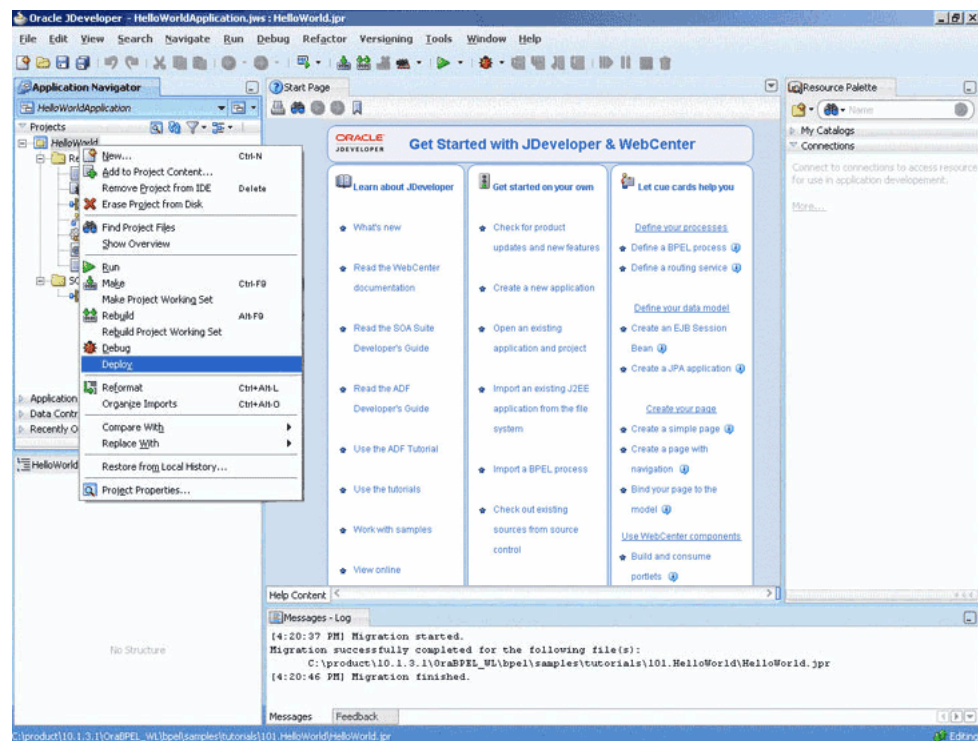
The only exceptions to be noted are as follows:

- If the BPEL Process contains any Decision Service applications, UI applications, or Work Flow applications, then these applications will not be automatically deployed in WebLogic Server by JDeveloper.

- The corresponding EAR/WAR files is custom built for WebLogic platform but must be manually deployed on the target server **oracleSOAServer** in WebLogic.

- Use Weblogic Admin console (*http://<hostname>:8001/console*) to deploy the EAR/WAR files to **oracleSOAServer**.

> **Note:** Refer to Auto Loan Demo for more details.

Deploying Human Task and Decision Services EAR Files

This section describes steps to deploying the human task form ear and decision services ear:

- Deploying Human Task Form EAR
- Deploying Decision Services EAR

Deploying Human Task Form EAR

To deploy human task form ear:

1. Change to the …\public_html\…\form in the directory of the sample.

2. Note the ear file created in the directory.

3. Extract the application.xml from the ear into the META-INF folder. Open application.xml and add the following DTD to it.

```
<!DOCTYPE application PUBLIC
    "-//Sun Microsystems, Inc.//DTD J2EE Application 1.3//EN"
    "http://java.sun.com/dtd/application_1_3.dtd">
```

4. Add the modified application.xml back into the ear in the same folder structure in the META-INF folder.

5. Extract the war file from inside the ear file. Extract the web.xml from the war into the WEB-INF folder. Open web.xml and add the following DTD to it.

```
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application
2.3//EN" "http://java.sun.com/dtd/web-app_2_3.dtd">
```

6. Put the modified web.xml back into the war in the same folder structure under WEB-INF. Then, add the modified war file back in to the ear.

> **See Also:** For deploying the ear please look up the section under "Deploying J2EE Applications on WebLogic".

Deploying Decision Services EAR

To deploy decision services ear:

1. Change to the ..\decisionservices\.. directory of the sample.

2. Note the ear file created in the directory.

3. Create an exploded directory version of the Decisionservices ear file. For example, if the ear file in your machine is DecisionService.ear, then the steps to create an exploded directory are as follows:

a. Rename the `DecisionService.ear` to `DecisionService.zip`.

b. Create a new folder with the name of the ear called **DecisionService**.

c. Extract the `DecisionService.zip` to **DecisionService** folder that you created in Step b. The `META-INF` folder and `DecisionService-web.war` file is created in the **DecisionService** folder.

d. Navigate to **DecisionService** folder and rename the `DecisionService-web.war` to `DecisionService-web.zip`.

e. Create a new folder with the name of the war file called **DecisionService-web** in the **DecisionService** folder.

f. Extract the `DecisionService-web.zip` to **DecisionService-web** folder that you created in Step e.

g. Delete the `DecisionService-web.zip` from the **DecisionService** folder.

h. Navigate to **DecisionService-web** folder, and you will notice the following three folders: `META-INF`, `public_html`, and `WEB-INF`.

> **Note:** Here, WebLogic needs the decision services ear to be deployed as an exploded directory version and not as an .ear file.
>
> Repeat steps a to h to create an exploded directory version of the ear file that you would want to deploy.

> **See Also:** For deploying the ear please look up the section under "Deploying J2EE Applications on WebLogic".

Additional Configuration Steps of the BEA WebLogic Server

The configuration steps mentioned in this section are optional and you can perform these only if there is a need:

- Using Application Security
- Using High Availability Setup

Using Application Security

This section describes the following steps to set up application security by using external LDAP store for BEA WebLogic Server 9.2:

- Step 1: Create an Authentication Provider
- Step 2: Confirguring LDAP in BEA WebLogic Server

Step 1: Create an Authentication Provider

1. Log in to *http://localhost:8001/console*, using `weblogic` as the username and password.

2. Select **Security Realms** -> **myrealm** -> **Providers** -> **Authentication**.

3. Click the **Lock & Edit** button in the Change Centre pane to activate all the buttons on this page.

4. Click **New** to create a new authentication provider, for example, LDAP Authenticator. The Create a New Authentication Provider page is displayed.

5. Enter a name of the authentication provider in the **Name** field (for example, LDAP_1)and select **LDAPAuthenticator** in the Type drop-down.

6. Click **OK**. The Authentication Providers table displays the name of the LDAP provider that you created.

Step 2: Confirguring LDAP in BEA WebLogic Server

BEA WebLogic Server does not support or certify any particular LDAP server. Any LDAP v2 or v3 compliant LDAP server should work with BEA WebLogic Server. The LDAP authentication providers, in this release of WebLogic Server (v9.2), are configured to work with the SunONE (iPlanet), Active Directory, Open LDAP, and Novell NDS LDAP servers.

You can use an LDAP authentication provider to access other types of LDAP servers. Choose either the LDAP Authentication provider (LDAPAuthenticator) or the existing LDAP provider that most closely matches the new LDAP server and customize the existing configuration to match the directory schema and other attributes for your LDAP server. The server comes with the following authentication providers, which help to configure different LDAP servers:

- iPlanet authentication provider

- Active Directory authentication provider

- Open LDAP authentication provider

- Novell authentication provider

- Generic LDAP authentication provider

If you select the LDAP authentication provider, then every LDAP authentication provider has the following attributes:

- Enable communication between the LDAP server and the LDAP Authentication provider. For a more secure deployment, BEA recommends using the SSL protocol to protect communications between the LDAP server and WebLogic Server. Enable SSL with the SSLEnabled attribute only if the SSL is enabled for LDAP server. This is referenced by the Hostname and Port (default: 389) attributes.

- Configure options that control how the LDAP Authentication provider searches the LDAP directory. This is referenced by User name attribute and the Static Group User name attribute.

- Specify where in the LDAP directory structure users are located. This is referenced by the User Base DN (Distinguished Name) attribute.

- Specify where in the LDAP directory structure groups are located. This is referenced by the Group Base DN attribute.

- Define how members of a group are located.

Perform the following steps to configure LDAP in BEA WebLogic Server:

1. Edit the provider-specific attributes of the LDAP authentication provider through the Administration Console.

    a. Log in to *http://localhost:8001/console*, using `weblogic` as the username and password.

    b. Select **Security Realms** -> **myrealm** -> **Providers** -> **LDAP_1**. The Settings of LDAP_1 page is displayed.

    c. Click **Provider Specific**.

d. Click the **Lock & Edit** button in the Change Centre pane to activate all the buttons on this page.

e. Edit the required attributes in the Provider Specific page.

f. Click **Save**.

2. Edit performance options that control the cache for the LDAP server.

a. Click the **Performance** tab.

b. Edit Max Group Hierarchies in Cache. The maximum size of the LRU cache for holding group membership hierarchies if caching is enabled. The default is 100.

c. Edit Group Hierarchy Cache TTL.The maximum number of seconds a group membership hierarchy entry is valid in the LRU cache. The default is 60.

d. Click **Save**.

Failover

You can configure an LDAP provider to work with multiple LDAP servers and enable failover, if one LDAP server is not available. To enable failover, change the Host attribute in the *security_realm* > **Providers** > *provider_specific* page, to contain a list of hostnames and ports, for example, hostname1:389, hostname2:389. When using failover, the Parallel Connect Delay and Connect Timeout attributes have to be set for the LDAP authentication provider:

- **Parallel Connect Delay**: Specifies the number of seconds to delay when making concurrent attempts to connect to multiple servers. An attempt is made to connect to the first server in the list. The next entry in the list is tried only if the attempt to connect to the current host fails. This setting might cause your application to block for an unacceptably long time, if a host is down. If the value is greater than 0, then another connection setup thread is started after the specified number of delay seconds has passed. If the value is 0, then connection attempts are serialized.

- **Connection Timeout**: Specifies the maximum number of seconds to wait for the connection to the LDAP server to be established. If the value is 0, there is no maximum time limit and WebLogic Server waits until the TCP/IP layer times out to return a connection failure. Set to a value over 60 seconds depending upon the configuration of TCP/IP.

> **Note:** After you create the LDAP authentication provider, perform the following changes and restart the servers that are running under SOADomain:
>
> - Select **Security Realms** > **myrealm** > **Providers** > **DefaultAuthenticator** and change the Control Flag to SUFFICIENT.
>
> - Select **Security Realms** > **myrealm** > **Providers** > *yourLDAPAuthenticator* and change the Control Flag to SUFFICIENT.
>
> Users in LDAP server must be inside a **SoaGroup** group in the LDAP directory. (You should create a **SoaGroup** group in the LDAP directory and add the desired users to that group, otherwise the LDAP users cannot access applications inside the SOADomain).
>
> Ensure that admin.user and admin.password in `SOA_ HOME\bpel\utilities\ant-orabpel.properties` are updated with the credentials of a valid user from the LDAP Authenticator.
>
> For more information, refer to *http://e-docs.bea.com/wls/docs92/secmanage/atn.html#wp1198953*

Using High Availability Setup

This section describes the High Availability (HA) support available for Oracle SOA Suite 10.1.3.3 on BEA WebLogic Server 9.2. This section contains the following topics:

- Prerequisite Checks

- Steps to Configure HA for Oracle SOA Suite

Prerequisite Checks

Ensure that HA setup of Oracle SOA Suite is configured on two nodes on two machines. Let's assume the hostnames of the two nodes as hostname01 and hostname02. Also, the load balancing URL as *http://<loadbalancer>:9800*.

Steps to Configure HA for Oracle SOA Suite

Follow these instructions to configure HA for Oracle SOA Suite on BEA WebLogic Server:

1. Configure Oracle SOA Suite on BEA WebLogic Server on hostname01 and hostname02 separately.

    > **Note:**
    >
    > - To configure Oracle SOA Suite on BEA WebLogic Server on a hostname, refer to "Installation and Configuration".
    >
    > - While configuring ensure that `DB_URL` propertypoints to the same database in `WL_SOA_ Installables\SOADomain.properties` for both the nodes.

2. Install any load balancing software on one of the hosts (hostname01 or hostname02) or some other host, and point *http://hostname01:9700* and

http://hostname02:9700 using the common load balancing URL
(*http://<loadbalancer>:9800*).

3. Modify the `SOA_HOME\bpel\system\config\collaxa-config.xml` on both
   hostname01 and hostname02.

   ■ Update **soapCallbackUrl** property in the `collaxa-config.xml` file to
     http://<loadbalancer>:9800 so that the **soapCallbackUrl** property points to the
     load balancer URL.

4. Undeploy `esbservices` application in any one of these hosts, either hostname01
   or hostname02, before running oracleSOAServer.

   ■ This application should not be running on both the hosts. Failure to undeploy
     this application (esbservices) might result in producing unwanted errors.

5. Start **oracleSOAServer** on both the hostname01 and hostname02 hosts.

6. Log in to the BPEL Process Manager Console at
   http://<loadbalancer>:9800/BPELConsole.

7. Log in to the ESB Console at *http://<hostname01>:9700/esb* (assuming that you have
   retained `esbservices` application in hostname01 machine).

8. Log in to the OWSM at *http://<loadbalancer>:9800/ccore/Login.jsp*.

Postinstallation Verification Tasks

This section describes the postinstallation verification tasks to be performed, and it
contains the following topics:

■ Verifying Installation from the BEA WebLogic Server Console

■ Verifying BPEL, ESB, OWSM Consoles

■ Verifying the SelectAllByTitle Sample for the Database Adapter

■ Verifying the OrderBooking Tutorial Sample

■ Running Adapter Samples

■ Deploying Samples Using Ant

Verifying Installation from the BEA WebLogic Server Console

Perform the following steps to check if the BEA Admin Console has started:

1. Navigate to *http://localhost:8001/Console*. The Oracle BEA WebLogic Server Admin
   Console window is displayed.
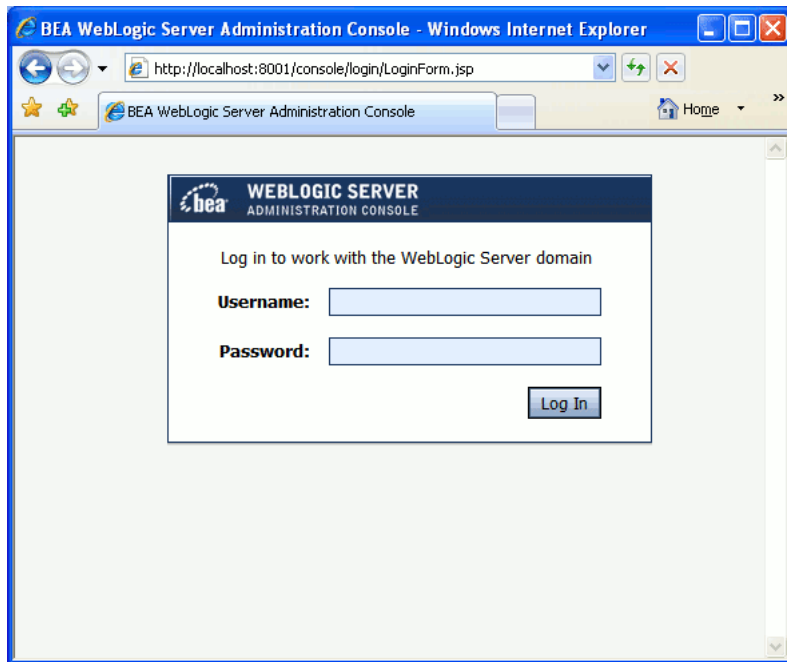
Log in using `weblogic` as the username and password.

2. Verify that you can view the oracleSOAServer Home page by selecting **Environment** -> **Servers** -> **oracleSOAServer** -> **Configuration**.
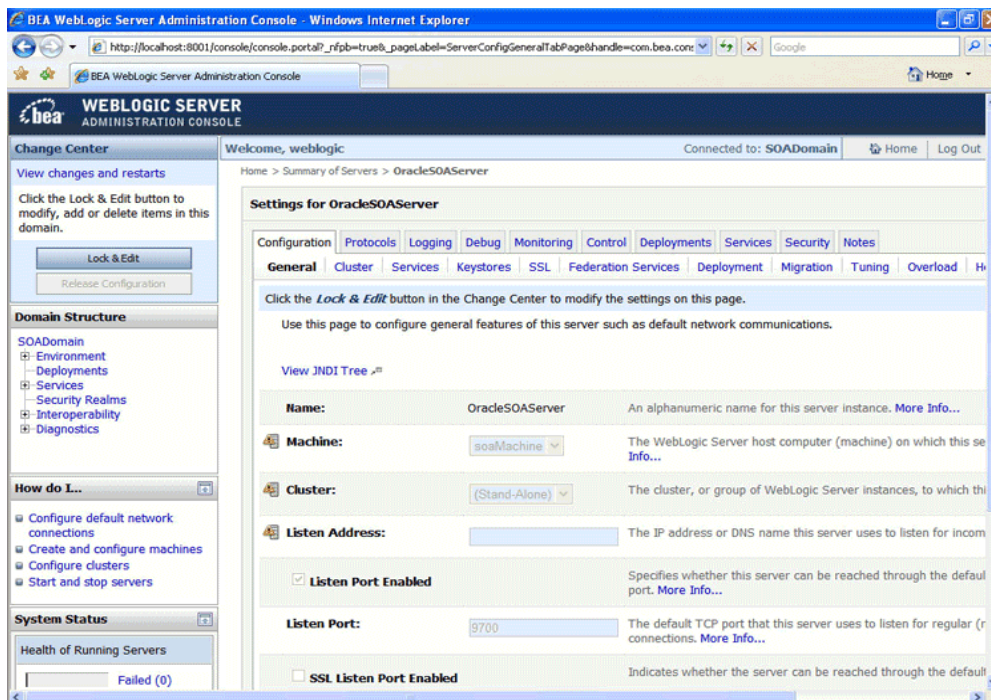
3. Verify that you can view the oracleSOAServer startup properties page by selecting **Environment** -> **Servers** -> **oracleSOAServer** -> **Configuration** -> **Server Start**. You can also add or modify the server startup properties such as Class Path, Arguments as a BPEL Domain admin.
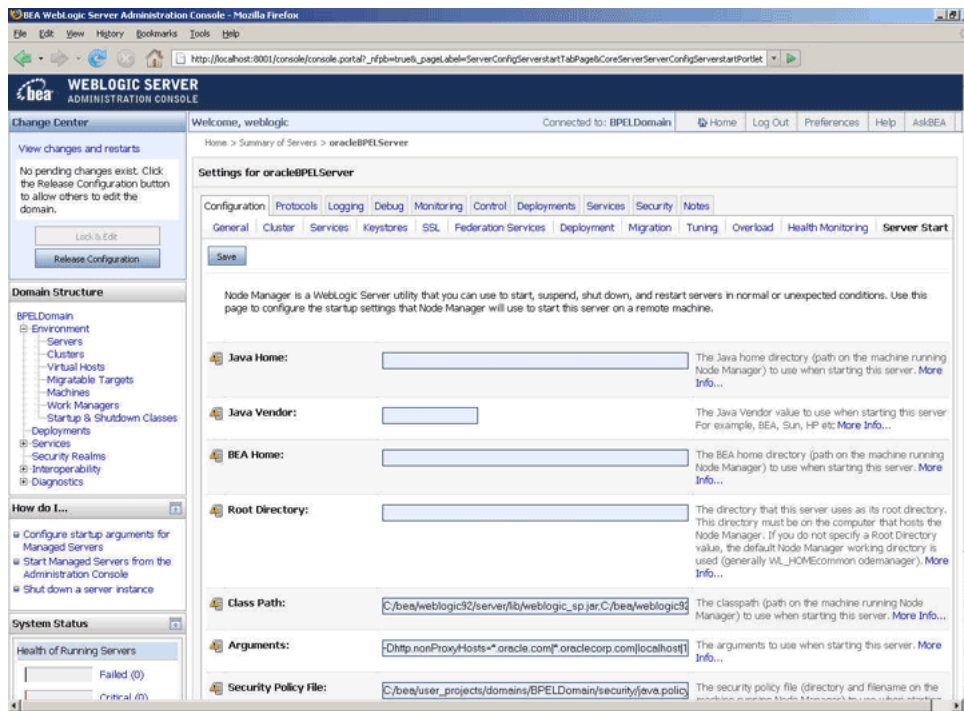
4. Verify that the SOAJMSServer is installed under **Services** -> **Messaging** -> **JMS Servers**.
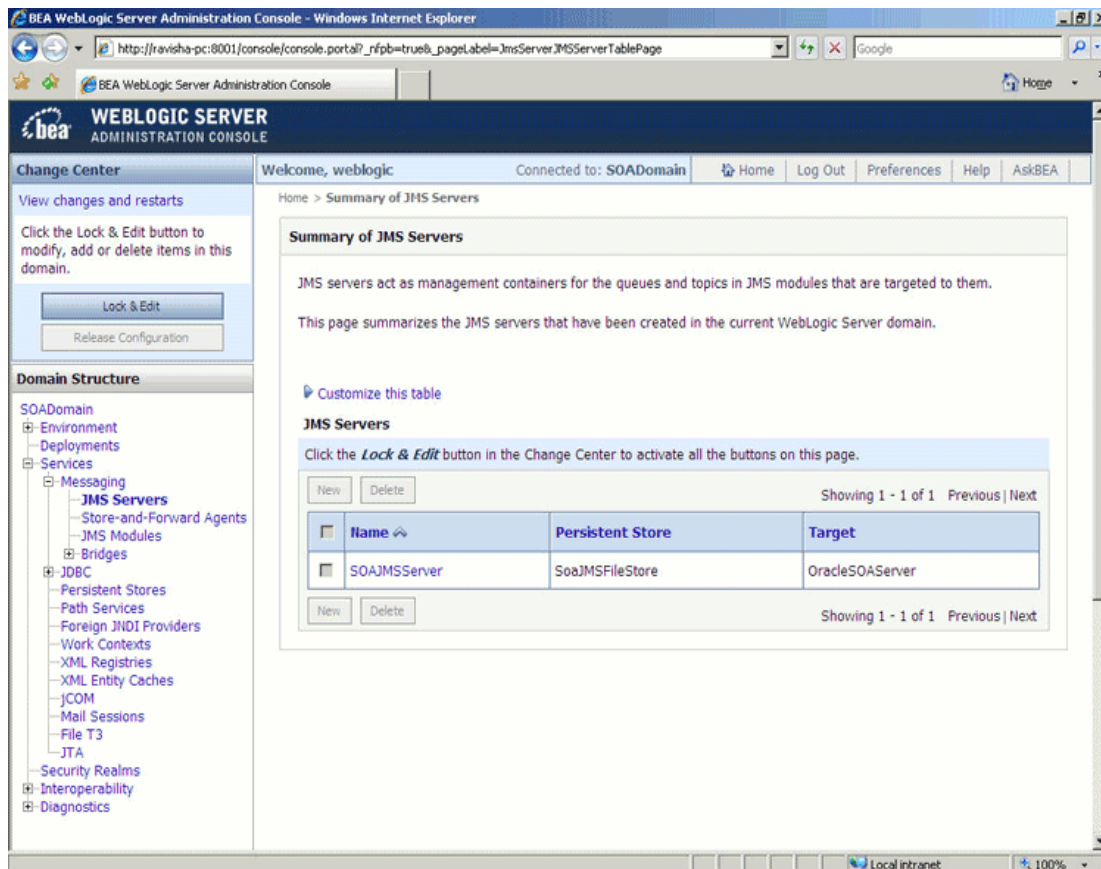
5. Verify that the summaries of the JMS resources that have been created for the JMS System module are displayed under **Services** -> **Messaging** -> **JMS Modules** -> **SOAJMSModule** -> **Configuration**.
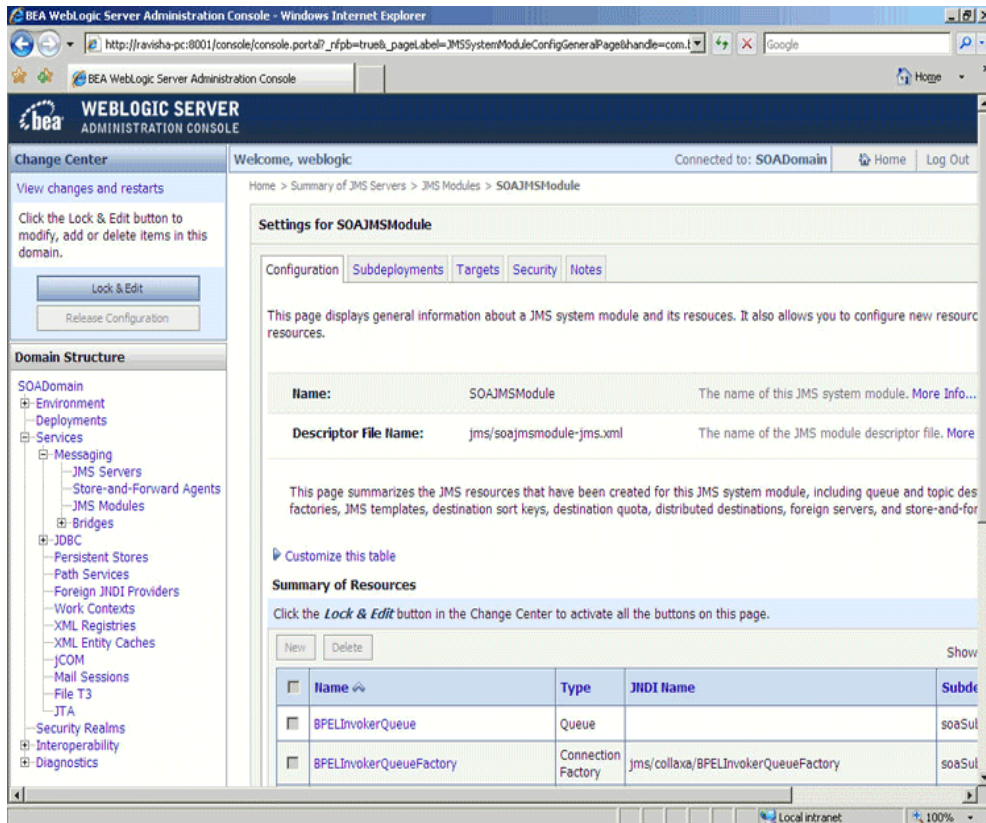
6. Verify that the BPELServerDataSource and BPELServerDataSourceWorkflow are the two JDBC data sources that are created under **Services** -> **JDBC** -> **Data Sources**.
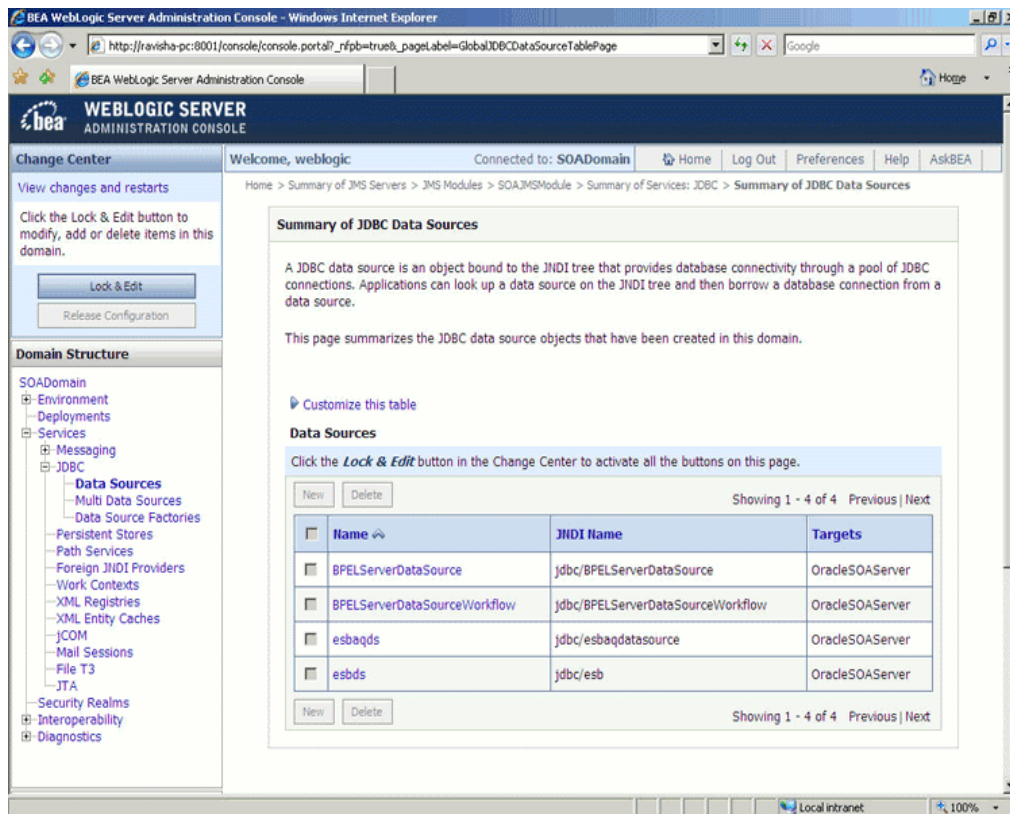
Verifying BPEL, ESB, OWSM Consoles

Perform the following steps to check if the BPEL, ESB, OWSM Consoles have started:

1. Navigate to *http://localhost:<default_port>/BPELConsole/* (Or to the location where the software is installed, for example, *http://<machine-name>:<default_port>/BPELConsole/*. The BPEL Console window is displayed as shown in Figure 1–1.

    Navigate to the *http://localhost:<default_port>/esb/*. The ESB Console window is displayed, as shown in Figure 1–2.

    Navigate to the *http://localhost:<default_port>/ccore/*. The OWSM Console window is displayed, as shown in Figure 1–3.

    Note: You can use the BPEL, ESB, OWSM log in windows only if the admin security is configured in WebLogic.

    The *<default_port>* is defined in the constants.properties file.

Figure 1–1   BPEL Console Window
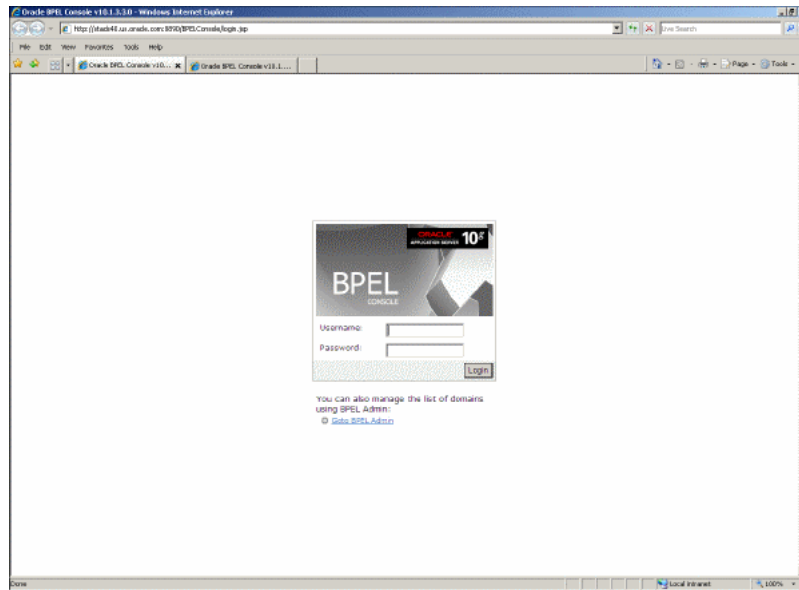
Figure 1–2   ESB Console Window

Figure 1–3 OWSM Console Window
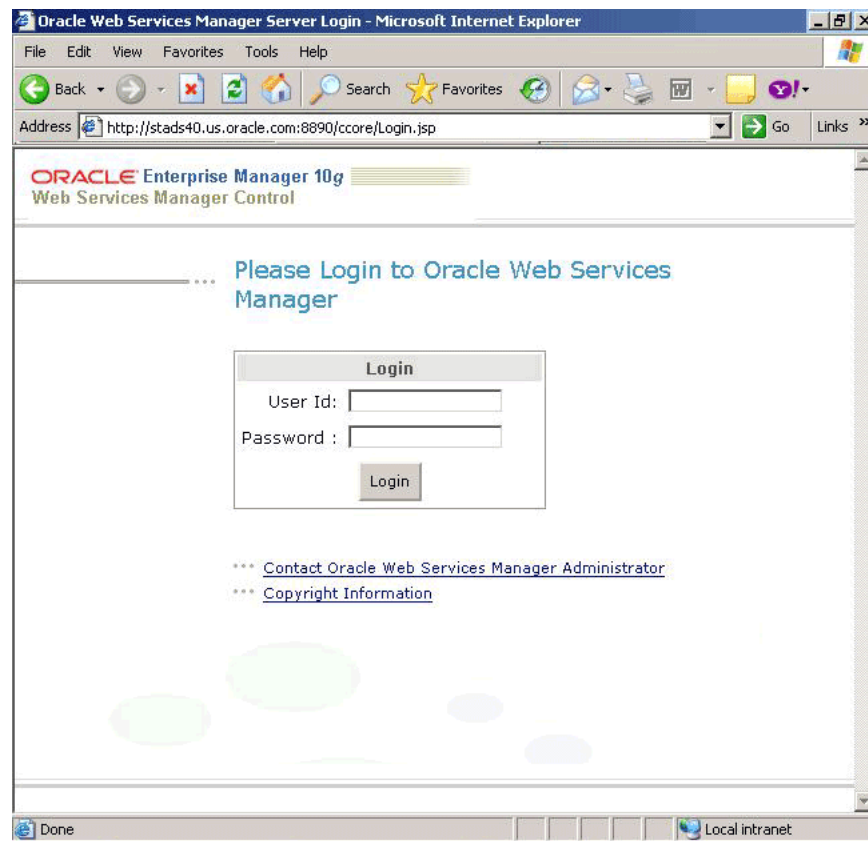
2. Log in to the BPEL Console using the username and password, the Oracle Enterprise Manager BPEL Control page is displayed, as shown in Figure 1–4.

Figure 1–4   Oracle Enterprise Manager BPEL Control
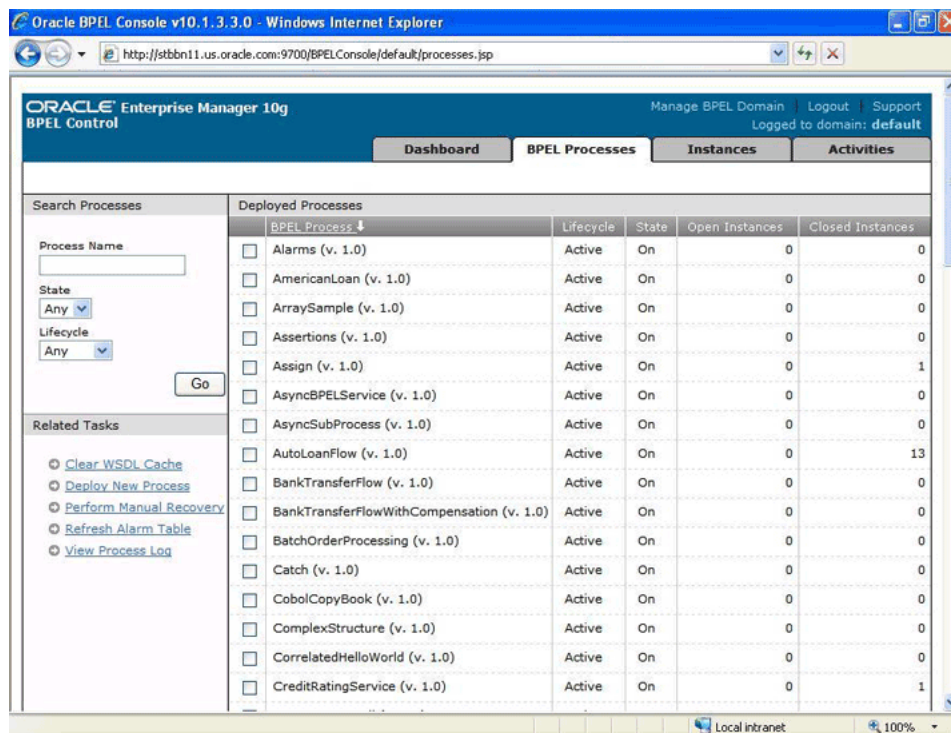
Log in to the ESB Console using the username and password, the Oracle Enterprise Manager ESB Control page is displayed, as shown in Figure 1–5.

Figure 1–5   Oracle Enterprise Manager ESB Control
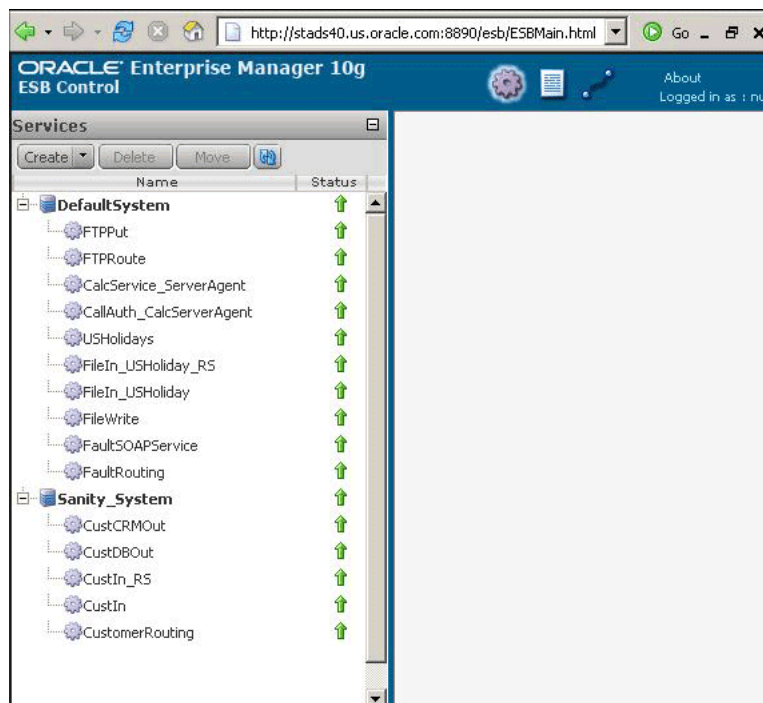
Log in to the OWSM Console using the username and password, the Oracle Enterprise Manager Web Services Manager Control page is displayed, as shown in Figure 1–6.

Figure 1–6   Oracle Enterprise Manager OWSM Control
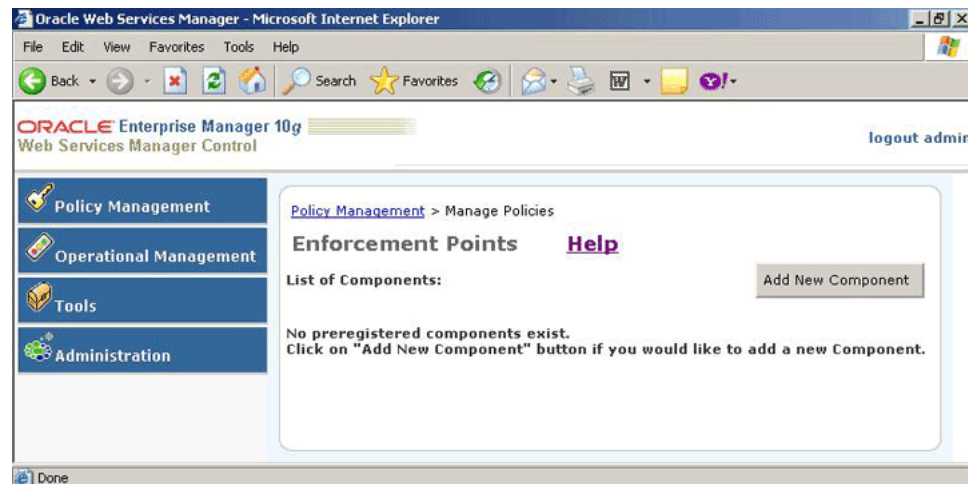
3. Verify that the `esbprotocol.jar` file is copied to `<BEA_HOME>\jrockitXX\jre\lib\ext` directory.

   If not this may be due to a different `JAVA_HOME` specified in the `SOADomain.properties` file. The WebLogic managed servers uses jrockit JRE, by default, to start the servers (on Windows and Linux) and ESB expects the `esbprotocol.jar` file to be present in the `jre/lib/ext` directory of the JRE. Copy the `esbprotocol.jar` file found in `SOA_HOME/integration/esb/lib` before starting the managed server on WebLogic. If `esbprotocol.jar` is not found in the server's `jre/lib/ext`, ESB console will throw the following error post server start:

   "Designtime cache has not been initialized".

Verifying the SelectAllByTitle Sample for the Database Adapter

1. Log in to the database and start SQL*Plus.

2. Run the `setup.sql` script:

   ```
   SQL> @Oracle_Home/samples/tutorials/122.DBAdapter/sql/setup.sql;
   ```

   This script creates and populates the `movies` table in the database.

3. Point the database adapter to your database in the WebLogic Console under **Resources**, **Resource Adapters**, **DB Adapter**, **J2C Connection Factories**, **BPEL Samples**, **Custom Properties**, **Connection String**. Also, set the username and password.

4. Restart **oracleSOAServer**.

5. Select **Start**, **All Programs**, **Oracle - *Oracle_Home***, **Oracle SOA Suite**, **Developer Prompt**.

6. Change to the following directory:

   ```
   tutorials\122.DBAdapter\SelectAllByTitle
   ```

7. Run the following command:

```
ant
```

This compiles and deploys all projects dependent on this tutorial. Projects are deployed into *Oracle_Home*\bpel\domains*domain_name*\deploy.

8. Select **Start**, **All Programs**, **Oracle - *Oracle_Home***, **Oracle SOA Suite**, **BPEL Console**.

9. Click **SelectAllByTitle** in the **Deployed BPEL Processes** list.

10. Refer to the MOVIES table, and enter the movie title on the Initiate page. For example, 'The Aviator'.

11. Click **Post XML Message**.

12. View the results and inspect the instance.

> **Note:** Refer to for
> *http://www.oracle.com/technology/products/integration/esb/index.html* ESB
> Samples. You can try and deploy the samples following the
> instructions in the samples.

Verifying the OrderBooking Tutorial Sample

The Web application DTD link in the web.xml files included with Oracle SOA Suite must be modified before deployment to the BEA WebLogic Server.

1. Search for the web.xml file in the *Oracle_Home*\bpel\samples\tutorials\127.OrderBookingTutorial\PriceQuote\SelectManufacturingUI\WEB-INF directory.

2. Make the following change in the above mentioned web.xml file:

Change:

```
<ejb-local-ref id="EjbRef_TaskServiceBean_Message">
<ejb-ref-name>ejb/local/TaskServiceBean</ejb-ref-name>
<ejb-ref-type>Session</ejb-ref-type>
<local-home>oracle.bpel.services.workflow.task.ejb.TaskServiceLocalHome</local-home>
<local>oracle.bpel.services.workflow.task.ejb.TaskServiceLocal</local>
<ejb-link>TaskServiceBean</ejb-link>
</ejb-local-ref>
<ejb-local-ref id="EjbRef_TaskMetadataServiceBean_Message">
<ejb-ref-name>ejb/local/TaskMetadataServiceBean</ejb-ref-name>
<ejb-ref-type>Session</ejb-ref-type>
<local-home>oracle.bpel.services.workflow.metadata.ejb.TaskMetadataServiceLocalHome</local-home>
<local>oracle.bpel.services.workflow.metadata.ejb.TaskMetadataServiceLocal</local>
<ejb-link>TaskMetadataServiceBean</ejb-link>
</ejb-local-ref>
```

To:

```
<ejb-local-ref id="EjbRef_TaskServiceBean_Message">
<ejb-ref-name>ejb/remote/TaskServiceBean</ejb-ref-name>
<ejb-ref-type>Session</ejb-ref-type>
<local-home>oracle.bpel.services.workflow.task.ejb.TaskServiceRemoteHome</local-home>
```

```
<local>oracle.bpel.services.workflow.task.ejb.TaskServiceRemote</local>
<!--ejb-link>TaskServiceBean</ejb-link-->
</ejb-local-ref>
<!--    <ejb-local-ref id="EjbRef_TaskMetadataServiceBean_Message">
<ejb-ref-name>ejb/local/TaskMetadataServiceBean</ejb-ref-name>
<ejb-ref-type>Session</ejb-ref-type>
<local-home>oracle.bpel.services.workflow.metadata.ejb.TaskMetadataServiceLocal
Home</local-home>
<local>oracle.bpel.services.workflow.metadata.ejb.TaskMetadataServiceLocal</loc
al>
<ejb-link>TaskMetadataServiceBean</ejb-link>
</ejb-local-ref>
-->
```

3. Create an xml file by the name weblogic.xml in the *Oracle_Home*\bpel\samples\tutorials\127.OrderBookingTutorial\PriceQuote\SelectManufacturingUI\WEB-INF directory.

   Add the following to the weblogic.xml file:

```
<!DOCTYPE weblogic-web-app PUBLIC "-//BEA Systems, Inc.//DTD Web Application
6.1//EN" "http://www.bea.com/servers/wls610/dtd/weblogic-web-jar.dtd">
<weblogic-web-app>
   <reference-descriptor>
      <ejb-reference-description>
        <ejb-ref-name>ejb/remote/TaskServiceBean</ejb-ref-name>
        <jndi-name>ejb/bpel/services/workflow/TaskServiceBean</jndi-name>
      </ejb-reference-description>
   </reference-descriptor>
</weblogic-web-app>
```

   Save the weblogic.xml file.

4. Rename CompleteTask.jsp to CompleteTask_oc4j.jsp in *SOA_HOME*\bpel\samples\tutorials\127.OrderBookingTutorial\PriceQuote\SelectManufacturingUI\ directory.

5. Copy CompleteTask.jsp from *SOA_HOME*\bpel\samples\tutorials\127.OrderBookingTutorial directory to *SOA_HOME*\bpel\samples\tutorials\127.OrderBookingTutorial\PriceQuote\SelectManufacturingUI\ directory.

6. Select **Start**, **All Programs**, **Oracle - Oracle_Home**, **Oracle SOA Suite**, **DeveloperPrompt**.

7. Change directories to the following:

```
tutorials\127.OrderBookingTutorial
```

8. Start SQL*Plus and run the following script:

```
SQL> @PracticeFiles\insertTable.sql;
```

   This creates the required sample tables in the database.

9. Change all the BPEL partner links in the bpel.xml files to update to the default port, as defined in the constants.properties file.

10. Run the following command:

```
ant
```

This compiles and deploys all projects dependent upon this tutorial. However, WAR files for CreateOrderBookingUI and SelectManufacturingUI must be manually deployed into the BEA WebLogic Server.

11. Change to the *<ORACLE_HOME>\j2ee\home\applications* directory.

12. Note the `CreateOrderBookingUI.war` file that was created when you ran `ant` in Step 10.

13. Change to the `OrderApproval\public_html\OrderApproval\form` directory.

14. Note the `default_OrderApproval_1_0_OrderApproval.ear` file that was created when you ran ant in Step 10.

15. Select **Install Application** in the BEA Administrative Console to deploy the war files to the BEA WebLogic Server.

    Access the BEA Administrative Console at the following URL:

    `http://`*hostname*`:8001/console`

16. Select **oracleSOAServer** as the deployment target.

17. Repeat Steps 12 through 16 for the `war` or `ear` file.

18. Restart **oracleSOAServer** from the BEA console.

19. Run the following OrderBooking Tutorial steps:

    a. Initiate the process using *http://localhost:<default_port>/CreateOrderBookingUI* where default_port is as defined in the constants.properties file.

    b. Open the console in audit or flow mode. Follow the steps that appear on the console and click task links to complete the task.

    c. After the process moves beyond supplier selection, the human workflow is added, for manual user approval (or rejection). This process has a timeout of 5 minutes and defaults to order status is rejected. Follow this step by opening the worklist URL at

    http://localhost:default_port/integration/worklistapp/Login

    where *default_port* is as defined in the constants.properties file.

Running Adapter Samples

Ensure that the outbound connection pool properties shown in Table 1–2 are modified.

Configuring Outbound Connection Pool for Adapters in Weblogic

You should create the required outbound connection pools that are used by BPEL Process Partnerlinks before deploying BPEL Processes using Adapters. Perform the following steps to create the required outbound connection pools:

1. Log in to *http://localhost:8001/console*, using `weblogic` as the username and password.

2. Select **Deployments**, *<adapter_name>*, **Configuration**, and **Outbound Connection Pools**. The Outbound Connection Pool Configuration Table is dispalyed.

3. Click **Lock & Edit**.

4. Click **New**. The Create a New Outbound Connection page is displayed.

5. Select the outbound connection displayed in the Outbound Connection Group.

6. Click **Next**. The JNDI Name for Outbound Connection Instance page is displayed.

7. Enter the required JNDI name as referenced by the partnerlink WSDL of the BPEL process under jca:address location.

8. Click **Finish**. The Save Deployment Plan Assistant page is displayed.

9. Select a deployment plan location in the Location field, and click **Finish**. The Settings for *<adapter_name>* page is displayed.

10. Return to the Outbound Connection Pools page and select the outbound connection pool that you created under the Groups and Instances column. The Outbound Connection Properties page is displayed.

11. Click the respective property value column to update the properties.

12. Click **Save**.

13. Click the **Activate Changes** button to activate the changes you have made.

Table 1–2   Outbound Connection Pool Properties

Adapter Type	Properties
Database	■ `driverClassName`
	■ `connectionString`
FTP	■ `host`
	■ `port`
	Note: A new authentication alias must be created for connecting to the FTP server.
Applications	■ `connectionString`
	■ `userName`
	■ `password`
AQ	■ `connectionString`
	■ `userName`
	■ `password`
JMS	■ `connectionFactoryLocation`
	■ `isTopic`
	■ `isTransacted`
	Note: The `istopic` property must be set to `false` for queues. The `isTransacted` property must be set to `false` for the JMS samples to run.
MQ	■ `channelName`
	■ `portNumber`
	■ `queueManagerName`
	■ `hostName`

Deploying Samples Using Ant

Ensure that admin.user and admin.password in *SOA_ HOME\bpel\utilities\ant-orabpel.properties* are updated with the credentials of a valid user from the authentication store setup for authentication.

Samples can be deployed from the developer prompt using the `ant` script following the above step.

The samples containing only BPEL processes can be fully deployed using the `ant` script.

Samples containing additional components such as Decision Service applications, workflow forms, and UI applications must be deployed in the following manner.

1. Use the `ant` script to deploy the BPEL process of the sample.

2. For each Decision Service application, manually edit the jsps and the decisionservice.xml file to replace the variables for domain, host the port as required. Generate the `war` or `ear` file, and deploy into `DecisionServer`. Start the application.

3. For each workflow form application, generate the `war` or `ear` file, and deploy into **oracleSOAServer**. Start the Application.

4. For each UI Application, manually edit the `doApply.jsp` to replace the variables for domain, host the port as required. Generate the `war` or `ear` file, and deploy into **oracleSOAServer**. Start the application.

> **Note:** When deploying BPEL processes into BPEL Server on WebLogic 9.2, you only must specify the following two properties in build property files:
>
> ■ http.hostname = <SOA_hostname>
>
> ■ http.port = 9700
>
> These properties can be either defined in the build.properties file in your project or in the ant-orabpel.properties file. You can also create a customized build property file, which will overwrite the other two build propery files when properties get loaded by ant.
>
> Once properties are loaded by ant, the order in which the properties are loaded are as follows:
>
> 1. Customized build property file
>
>   To use this file when deploying a BPEL project, use the following command:
>
>    ```
>    ant -propertyfile <name>, where <name> is the build property
>    filename created by users.
>    ```
>
> 2. build.properties file in your BPEL project
>
> 3. If `BPEL_HOME` environment variable is specified, then `BPEL_HOME/utilities/ant-orabpel.properties` will be used, otherwise, `JDEV_HOME/integration/bpel/utilities/ant-orabpel.properties` is loaded by ant, where `JDEV_HOME` is the JDeveloper installation directory.
>
> It is recommended using build.properties file or customized build property files when deploying BPEL processes using ant.

Auto Loan Demo

This appendix describes how to run Auto Loan Demo on Oracle SOA Suite 10.1.3.1 on WebLogic 9.2 application server. It contains these sections:

■ Prerequisites on JDeveloper Studio 10.1.3.3

■ Auto Loan Demo Sample

■ Modelling Auto Loan Flow Process Using JDeveloper Studio

- Known Issues on non-Oracle Platforms

- Deploying J2EE Applications on WebLogic

- Running the Sample

Prerequisites on JDeveloper Studio 10.1.3.3

The following one-time changes should be performed on JDeveloper:

1. Replace the **bpm-services.jar** within JDeveloper at `jdev\integration\lib` with the updated jar from *SOA_HOME*`\bpel\system\services\lib`

2. Replace the **orabpel-ant.jar** within JDeveloper at `jdev\integration\lib` with the updated jar from SOA_HOME`\bpel\lib`.

3. Modify the following properties in `jdev\integration\bpel\utilities\ant-orabpel.properties` file:

   - Platform to **weblogic_8**

   - admin.user to a valid user in weblogic realm

   - admin.password to the password of the above user

   - jndi.url to **t3://<hostname>:9700**

   - jndi.InitialContextFactory to **weblogic.jndi.WLInitialContextFactory**

4. On JDeveloper, create an Application Server connection of type "Standalone OC4J 10.1.3".

5. On JDeveloper, create an Integration Server connection to "<hostname>:9700"

Note: Ignore errors during test connection regarding Mediator at this stage.

Auto Loan Demo Sample

The Auto Loan Flow sample has the following components:

- BPEL Process: AutoLoanFlow BPEL Process <bpel jar>

- Decision Service Applications (Business Rules Applications)

  – CreditRatingAgent <ear>

  – LoanAdvisorAgent <ear>

- UI Application: AutoLoanFlowUI <ear>

- HWF Tform application: AutoLoanflow LoanApproval <ear>

Since the AutoLoanFlow sample that is bundled with Oracle SOA Suite standalone is written for OC4J Application Server, it cannot be run as is on WebLogic 9.2 Application Server. Specifically, the Decision Service applications must be regenerated for WebLogic platform, using JDeveloper. Weblogic requires that the following mandatory deployment descriptor files be present in the application that serves Webservices:

- weblogic.xml

- weblogic-webservices.xml

- weblogic-webservices-policy.xml

Also the java-wsdl-mapping file needs WebLogic specific modifications.

The next section describes the steps to regenerate the Decision Services Applications in Auto Loan Flow for WebLogic.

Modelling Auto Loan Flow Process Using JDeveloper Studio

Perform the following steps to modify the AutoLoanFlow sample for WebLogic:

1. Delete the following file from the filesystem:

   SOA_HOME\bpel\samples\demos\AutoLoadDemo\AutoLoanFlow\bpel\decisionservices.decs

2. Open the AutoLoanFlow sample from JDeveloper Studio as a JDeveloper project using the following file:

   SOA_HOME\bpel\samples\demos\AutoLoanDemo\AutoLoanFlow\AutoLoanFlow.jpr

3. Open the **AutoLoanFlow.bpel** file from the Applications Navigator (found within the AutoLoanFlow project).

4. From the Services swim lane of **AutoLoanFlow.bpel,** delete the following decision service partnerlinks:

   - CreditRatingAgent

   - LoanAdvisorAgent

5. Follow the steps II, III, IV and V of "Modelling Auto Loan Broker Process" from *SOA_HOME*\bpel\samples\demos\AutoLoanDemo\AutoLoanBroker.pdf to re-create the two Decision Service applications.

Known Issues on non-Oracle Platforms

The AutoLoanFlow BPEL process has two Decision Service applications as partnerlinks (CreditRatingAgent and LoanAdvisorAgent). By default, the context-root generated for both these J2EE applications are same with the value - /rules/${domain_id}/${process_id}/${process_revision}

The ${} attributes are replaced by actual values during the build and deploy of the Auto Loan Flow. However, as the context-root is not unique for these two applications, these cannot be deployed on WebLogic. When the second application is deployed/started on WebLogic it would complain that the context-root is already in use.

This is an issue on non-Oracle application servers when a BPEL pProcess references more than one Decision Service partnerlinks generated from JDeveloper Studio. As a workaround, after generating the Decision Service applications on JDeveloper and before doing a build and deploy, perform the following:

- Modify the AutoLoanFlow\decisionservices\CreditRatingAgent\ear\META-INF\application.xml file.

  Change <context-root>/rules/${domain_id}/${process_id}/${process_revision}</context-root> to <context-root>/rules/${domain_id}/${process_id}/${process_revision}/CreditRatingAgent</context-root>

■ Modify the
`AutoLoanFlow\decisionservices\CreditRatingAgent\war\WEB-INF\web.xml` file.

Change `<url-pattern>CreditRatingAgent</url-pattern>` to
`<url-pattern>/</url-pattern>`

■ Finally, build and deploy the Auto Loan Flow using the Integration Server Connection. In the application navigator, right-click the BPEL project and select Deploy.

This would automatically deploy the BPEL process into BPEL engine running at the Integration Server connection.

The following J2EE applications should be manually deployed into WebLogic using the WebLogic Admin console:

■ CreditRatingAgent.ear

■ LoanAdvisorAgent.ear

■ AutoLoanFlowUI.ear

■ <domain>_AutoLoanFlow_<version>_LoanApproval.ear

Deploying J2EE Applications on WebLogic

Perform the following steps to deploy the applications to WebLogic:

1. Start the Weblogic server using the `startWeblogic.cmd/sh` command.

2. Log in to WebLogic Admin console using *http://<hostname>:8001/console*.

3. Select **Deployments**.

4. Click **Lock & Edit**.

5. Navigate to the directory where the target ear file is located on the file system.

6. Select the ear file and choose **Deploy**.

7. Choose **oracleSOAServer** as the target server, and select "I will provide the deployment in this directory" option.

8. Click **Finish Deployment**.

9. Click **Activate Changes**.

10. Start the deployed application from list of deployments.

Running the Sample

When the process is deployed, perform the following steps to test the sample:

1. Open the AutoLoanFlow UI at *http://<hostname>:9700/AutoLoanFlowUI*.

2. Click the **Initiate New BPEL Loan Flow** link.

3. Accept the default payload and click **Submit Loan Application**.

4. Log in to the worklist at *http://<hostname>:9700/integration/worklistapp* using jstein/welcome1 as the username and password.

5. Click the Task title (Loan Approval for Irving Stone).

6. Examine the task payload, the credit rating for that loan should be 500 with "Medium" risk and a Credit Max Amount of 50000.0.

The provider for the Loan Offer should be "Premium Bank" with an APR of 4.0

7. Approve the task.

8. Verify the AutoLoanFlow instance.

Limitations, Known Issues, Troubleshooting Tips

This section describes the limitations, known issues, and troubleshooting tips for Oracle SOA Suite 10.1.3.1 on BEA WebLogic Server version 9.2.

Limitations

Note the following limitations:

■ BEA WebLogic Server 9.2 and Oracle SOA Suite 10.1.3.1 should be installed as the same user on Linux and the user should not be a root user.

> **Note:** If you install BEA WebLogic Server 9.2 and Oracle SOA Suite 10.1.3.1 as different users, then the file permissions and ownership for files under the following directories should be verified and changed to Oracle SOA Suite install user:
>
> ■ *SOA_HOME*\bpel\lib
>
> ■ *SOA_HOME*\bpel\lib\rules
>
> ■ *SOA_HOME*\bpel\system\appserver\oc4j\webservices\lib
>
> ■ *SOA_HOME*\bpel\system\services\lib
>
> ■ *SOA_HOME*\bpel\system\config
>
> ■ *SOA_HOME*\bpel\utilities\

Known Issues

Note the following known issues:

JMS Adapter

■ JMS Adapter throws the following NullPointerException during initialization on non-Oracle platforms:

```
JmsConnectionFactory: Unable to set connectionparameters for
OracleConnectionManager
java.lang.NullPointerException
at
oracle.tip.adapter.jms.JmsConnectionFactory.<init>(JmsConnectionFactory.java:91
)
at oracle.tip.adapter.jms.JmsManagedConnectionFactory.createConnectionFactory
(JmsManagedConnectionFactory.java:80)
```

This is a benign error and does not stop the JMS connection factory from initializing.

Decision Services

■ The following data type binding warnings and errors are displayed during deployment and start of Decision Service (Business Rules) Applications. These errors and warnings can be ignored.

```
<WS data binding error>could not find schema type
'{http://www.w3.org/2001/XMLSchema}NCName
<WS data binding error>could not find schema type
'{http://websphere.ibm.com/webservices/}SOAPElement
java.lang.IllegalStateException
at weblogic.wsee.bind.runtime.internal.AnonymousTypeFinder$GlobalElementNode.
getSchemaProperty(AnonymousTypeFinder.java:253)
at
weblogic.wsee.bind.runtime.internal.AnonymousTypeFinder.getHiddenArrayElement
ComponentTypeNamed(AnonymousTypeFinder.java:104)
<WS data binding error>could not find schema type
'{http://www.w3.org/2001/XMLSchema}long
<WS data binding error>could not find schema type
'{http://xml.apache.org/xml-soap}Element
<WS data binding error>could not find schema type
'{http://www.w3.org/2001/XMLSchema}anyType
<WS data binding error>could not find schema type
'{http://www.w3.org/2001/XMLSchema}string
could not identify anonymous schema type named
'http://xmlns.oracle.com/AutoLoanFlow/CreditRatingAgent:tProperty[0,unbounded]'
, ignoring
<WS data binding error>While processing <exception-mapping> for
wsdlMessageName='{http://xmlns.oracle.com/AutoLoanFlow/CreditRatingAgent}decis
ionServiceError',
wsdlMessagePartElement='{http://xmlns.oracle.com/AutoLoanFlow/CreditRatingAgen
t}errorInfo'.  Unable to find a BindingType in the binding file  for
javaTypeName ='oracle.bpel.services.rules.DecisionServiceError',
xmlTypeName='e=errorInfo@http://xmlns.oracle.com/AutoLoanFlow/CreditRatingAgent
'.  The cause of this error is likely because an <exception-mapping> specified
for
{http://xmlns.oracle.com/AutoLoanFlow/CreditRatingAgent}decisionServiceErrorreq
uires that a <java-xml-type-mapping> exist for java
type='oracle.bpel.services.rules.DecisionServiceError',
xmlTypeName='e=errorInfo@http://xmlns.oracle.com/AutoLoanFlow/CreditRatingAgent
', with a <root-type-qname> of
{http://xmlns.oracle.com/AutoLoanFlow/CreditRatingAgent}errorInfo
<WS data binding error>oracle.bpel.services.rules.DecisionServiceError is not
understood because there is no type mapping for exception class
```

Troubleshooting Tips

The following list explains the troubleshooting tips encountered while installing Oracle BPEL Process Manager with the BEA WebLogic Server, and their resolutions:

Using Server Start Up Options

For any class path and security permission errors that you encounter while configuring Oracle SOA Suite on the BEA WebLogic Server, perform the following steps to correct the class path and security policy file options:

1. Log in to *http://localhost:8001/console,*using `weblogic` as the username and password.

2. Select **Environment** -> **Servers** -> **oracleSOAServer**. The Settings of oracleSOAServer page is displayed.

3. Click the **Server Start** tab.

4. Edit the following properties:

   - Class Path: Contains the path on the machine running Node Manager, which is used to start the oracleSOAServer. You can append a class path value to the class path mentioned in this field.

   - Arguments: Contains the arguments, which is used to start the oracleSOAServer. You can add arguments to the existing argument list that are required to start the server.

   - Security Policy File: Contains the security policy file, which is used to start the oracleSOAServer. You can also add your own policy file in this location. To do, you must add the following line inside the grant scope of the policy file:

     ```
     permission com.collaxa.security.ServerPermission "server",
     "read";
     ```

A

Appendix - Notes on Solaris Installation

This appendix describes the steps specific to Solaris installation.

Run the `setup_solaris.sh` script from `WL_SOA_Installables` folder at the operating system command prompt for Solaris.

Index

A

auto loan demo, 1-32

B

BEA WebLogic Application Server
    auto loan demo, 1-32
    installing, 1-1, 1-3
    known issues, 1-36, 1-37
    limitations, 1-36
    postinstallation tasks, 1-15
    postinstallation verification tasks, 1-19
    system requirements, 1-3
    troubleshooting tips, 1-37
    workarounds, limitations, and unsupported
        features, 1-36
BEA WebLogic Application Server with Oracle SOA
    Suite, 1-32
BEA WebLogic Server, 1-1

C

components
    Oracle SOA Suite, 1-2

I

installation
    BEA WebLogic Application Server, 1-1, 1-3
Integration Repository Creation Assistant
    usage, 1-32

K

known issues
    BEA WebLogic Application Server
        decision services, 1-37
        JMS adapter, 1-36
    BEA WebLogic Application Server with Oracle
        SOA Suite, 1-36

L

limitations
    BEA WebLogic Application Server with Oracle
        SOA Suite, 1-36

O

Oracle Business Rules, 1-2
Oracle SOA Suite, 1-1

P

postinstallation tasks
    BEA WebLogic Application Server with Oracle
        SOA Suite, 1-15
postinstallation verification tasks
    BEA WebLogic Application Server with Oracle
        SOA Suite, 1-19

S

system requirements
    BEA WebLogic Application Server with Oracle
        SOA Suite, 1-3

T

troubleshooting tips
    BEA WebLogic Application Server
        using server start up options, 1-37
    BEA WebLogic Application Server with Oracle
        SOA Suite, 1-37

W

workarounds, limitations, and unsupported features
    BEA WebLogic Application Server with Oracle
        SOA Suite, 1-36