

Oracle® SOA Suite

Installation Guide for WebLogic Server

10g Release 3 (10.1.3.4) for UNIX and Microsoft Windows

E13058-03

January 2009

Oracle SOA Suite Installation Guide for WebLogic Server, 10g Release 3 (10.1.3.4) for UNIX and Microsoft Windows

E13058-03

Copyright © 2008, 2009, Oracle. All rights reserved.

Primary Author: Vimmika Dinesh

Contributing Author: Rajesh Polavarapu, Ravisha Chikkegowda, Shreenidhi Raghuram, Shiva Prasad, Vikas Anand, Swaminathan Ramaswamy

Contributors: Guru Balse, Aruna Kasturi, Nirguna Kota, Umesh Kulkarni, Valarie Moore, Bhagat Nainani, Probal Nandy, Jim Peng, Edsel delos Reyes, Ganga Sah, David Shaffer, Navneet Singh, Bo Stern, Clemens Utschig-Utschig, Raj Venkatesan, Sunil Wadhwa, John Wang

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Contents

Preface	v
Audience.....	v
Documentation Accessibility	v
Related Documentation.....	vi
Conventions	vi
 1 Installing Oracle SOA Suite with WebLogic Server	
Overview.....	1-1
System and Database Requirements.....	1-3
Installation and Configuration	1-3
Step 1: Configure the Oracle Database.....	1-4
Step 2: Install Oracle SOA Suite Basic 10.1.3.1.0 for OC4J.....	1-4
Step 3: Create the Oracle SOA Suite Schemas in the Oracle Database.....	1-4
Step 4: Upgrade SOA Schemas to 10.1.3.4.....	1-5
Step 5: Apply SOA Suite Patchset 10.1.3.4.....	1-5
Step 6: Apply Patch for Oracle SOA Suite 10.1.3.4 on WebLogic 9.2	1-5
Step 7: Install and Configure WebLogic Server Version 9.2	1-6
Design-time Deployment Support Oracle SOA Suite 10.1.3.4 on WebLogic 9.2	1-12
From the BPELPM Developer Prompt Using Ant.....	1-12
Prerequisite Checks	1-12
Steps to Deploy Using the BPELPM Prompt	1-13
From JDeveloper	1-13
Prerequisite Checks	1-13
Steps to Deploy Using JDeveloper	1-14
Deploying Human Task and Decision Services EAR Files.....	1-15
Deploying Human Task Form EAR	1-15
Deploying Decision Services EAR.....	1-16
Postinstallation Verification Tasks	1-17
Verifying Installation from the WebLogic Server Console	1-17
Verifying BPEL, ESB, OWSM Consoles	1-21
Verifying the SelectAllByTitle Sample for the Database Adapter	1-25
Verifying the OrderBooking Tutorial Sample.....	1-26
Running Adapter Samples.....	1-28
Configuring Outbound Connection Pool for Adapters in Weblogic	1-28
Deploying Samples Using Ant.....	1-29

Auto Loan Demo	1-30
Prerequisites on JDeveloper Studio 10.1.3.3.....	1-31
Auto Loan Demo Sample.....	1-31
Modelling Auto Loan Flow Process Using JDeveloper Studio	1-32
Known Issues on non-Oracle Platforms	1-32
Deploying J2EE Applications on WebLogic.....	1-34
Running the Sample.....	1-34
Limitations, Known Issues, Troubleshooting Tips	1-34
Limitations	1-34
Known Issues.....	1-35
Troubleshooting Tips.....	1-37
Using Server Start Up Options.....	1-37

A Upgrade Path for SOA Suite 10.1.3.3 on WebLogic 9.2 Installation

B SOA Suite 10.1.3.4 HA Setup Using WebLogic Cluster

Sample HA Topology.....	B-1
Steps for Installation and Configuration of the Sample HA Topology.....	B-2
Prerequisites.....	B-2
Installing Database.....	B-3
SOA Suite 10.1.3.4 (for OC4J) Installation on both Machines.....	B-3
Steps to Configure HA Setup for SOA 10.1.3.3 on WebLogic 9.2	B-3
Post Installation Steps	B-4
NodeManager Setting.....	B-4
Configure Collaxa-config for BPEL	B-4
Configure JGroups for BPEL	B-5
ESB Design Time (ESB-DT).....	B-5
Load Balancer	B-5
Addresses	B-5

C Customizing Role-based Security for SOA Applications

D Additional Configuration Steps for WebLogic Server

Configuring External LDAP	D-1
ESB System Configurations	D-3
Changing to File-based Slide Repository	D-4
Human WorkFlow API Clients.....	D-4
Identify Service Configuration.....	D-4

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 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.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
<code>monospace</code>	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Installing Oracle SOA Suite with WebLogic Server

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

This chapter contains these topics:

- [Overview](#)
- [System and Database Requirements](#)
- [Installation and Configuration](#)
- [Design-time Deployment Support Oracle SOA Suite 10.1.3.4 on WebLogic 9.2](#)
- [Deploying Human Task and Decision Services EAR Files](#)
- [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 WebLogic Server.

The WebLogic Server enables you to set up, operate, and integrate e-business applications across multiple computing platforms using Web technologies. The 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 BPEL Console is the monitoring environment for Oracle BPEL Process Manager. You can run, manage, and test your deployed BPEL process using the Oracle BPEL Console. Oracle BPEL 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 WebLogic Server consists broadly of the following steps:

- Create the Oracle SOA Suite Schema in the Oracle Database
This step involves installation of Oracle Database and creation of the required DB schemas for the Oracle SOA Suite 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 WebLogic Server.
- Apply Oracle SOA Suite Patchset 10.1.3.4 on Oracle SOA Suite 10.1.3.1
This patchset upgrades the existing 10.1.3.1.0 installation to 10.1.3.4.0.
- Apply Patchset and Upgrade
This patchset upgrades to Oracle SOA Suite 10.1.3.4 on WebLogic.
- Configure Oracle SOA Suite on 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 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 Administration, 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 WebLogic 9.2 platform.

System and Database Requirements

[Table 1–1](#) describes the system requirements for using Oracle SOA Suite with the WebLogic Server.

Table 1–1 Oracle SOA Suite System Requirements

Element	Requirement
WebLogic Server	Version 9.2 MP1 or newer
Design Time	Oracle JDeveloper version 10.1.3.3
Oracle SOA Suite for OC4J	Apply SOA Suite patchset 10.1.3.4 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 Oracle Web site for additional details about using these operating systems with the WebLogic Server.
Dehydration store database	Oracle Database 10g (10.1.0.5) 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 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 Schemas in the Oracle Database](#)
- [Step 4: Upgrade SOA Schemas to 10.1.3.4](#)
- [Step 5: Apply SOA Suite Patchset 10.1.3.4](#)

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

- [Step 6: Apply Patch for Oracle SOA Suite 10.1.3.4 on WebLogic 9.2](#)
- [Step 7: Install and Configure WebLogic Server Version 9.2](#)

Step 1: Configure the Oracle Database

Follow these instructions to install Oracle Database 10g.

Note: These instructions assume that you have obtained Oracle Database 10g version 10.1.0.2 and Oracle Database 10g 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 10g 10.1.0.2. Refer to <http://www.oracle.com/pls/db102/homepage> for install instructions.
2. Install the Oracle Database 10g 10.1.0.5 patch in the same Oracle home in which you installed Oracle Database 10g. Refer to http://download.oracle.com/docs/cd/B25553_01/install.1012/b25463/infrainexistingdb.htm#sthref415 for installing the patch.

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 WebLogic 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 Schemas in the Oracle Database

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

1. Navigate to the install\soa_schemas\irca folder in the Oracle SOA Suite Installer 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: Upgrade SOA Schemas to 10.1.3.4

In this step, you will apply the SOA Suite Patchset 10.1.3.4 and also upgrade ORABPEL and ORAESB schemas to 10.1.3.4.

Step 4-1: Upgrade ORABPEL and ORAESB Schemas to 10.1.3.4

To upgrade ORABPEL and ORAESB schemas to 10.1.3.4, perform the following steps:

1. Run the 10.1.3.4 SOA Schema upgrade scripts for `orabpel`/`oraesb` schemas that are available in the SOA Suite 10.1.3.4 Patchset Installation pack.

2. Execute the following script to upgrade the ORABPEL schema:

```
install\soa_schema_upgrade\bpel\scripts\upgrade_10133_10134_
oracle.sql
```

3. Execute the following script to upgrade the ORAESB schema:

```
install\soa_schema_upgrade\esb\sql\oracle\upgrade_10133_
10134_oracle.sql
```

Step 5: Apply SOA Suite Patchset 10.1.3.4

To apply SOA Suite Patchset 10.1.3.4, perform the following steps:

1. You can download the SOA Suite Patchset 10.1.3.4 from metalink.oracle.com with the following patch number - 7272722 Oracle Fusion Middleware Family: Patchset 10.1.3.4 PATCHSET UPLOAD.
2. 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.

3. 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 6: Apply Patch for Oracle SOA Suite 10.1.3.4 on WebLogic 9.2

You must download the Patch for Bug 7337034 (HOTPLUG: SOASUITE 10.1.3.4 ON WEBLOGIC 9.2 - CHANGES FOR HOTPLUGGABILITY) from MetaLink and then apply the patch on Oracle SOA Suite 10.1.3.4.

1. Download the Patch `p7337034_101340_GENERIC.zip` for Bug 7337034.
2. Follow the instructions given in the `Readme.txt` file of the patch and apply the patch on Oracle SOA Suite 10.1.3.4.

3. Shutdown the Oracle SOA Suite post patch upgrade as follows:

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

Step 7: Install and Configure WebLogic Server Version 9.2

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

1. Install 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.

The contents of this zip file are extracted to `Weblogic_SOA10134` folder. This zip file contains both the base (`Weblogic_SOA10134_Base`) and the HA scripts (`WeblogicSOA10134HA`). The directory structure when you unzip the `Weblogic_SOA10134.zip` file is as follows:

```
Weblogic_SOA10134
  Weblogic_SOA10134_Base (Non-HA)
  WeblogicSOA10134HA (HA)
  Readme.txt
```

Note:

- The directory to which you download the Oracle SOA Suite should be the same host on which the WebLogic Server is installed.
 - Unzip the `Weblogic_SOA10134_Base` 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 `Weblogic_SOA10134_Base` folder using the `chmod -R 755 Weblogic_SOA10134_Base` command.
-

3. Modify the following mandatory installation properties in the `Weblogic_SOA10134_Base\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. Ensure that you use forward slash for all the paths in the property files.

However, the proxy settings properties, such as `PROXY_HOST` is non-mandatory.

Property	Description
BEA_HOME	The directory path in which WebLogic Server is installed. For example, <code>BEA_HOME=C:/bea</code> .
WL_HOME	The directory path WebLogic Server is installed, which is usually under <code>BEA_HOME</code> . For example, <code>WL_HOME=C:/bea/weblogic92</code>
JAVA_HOME	JAVA path of WebLogic. For example, <code>JAVA_HOME=C:/bea/jdk150_06</code>
CREATE_NEW_DOMAIN	Set this property to 'NO' if you wish to create SOA Server as a managed server in any existing domain. Set this property to 'YES' if you wish to create SOA Server in the new domain "SOADomain". For example, <code>CREATE_NEW_DOMAIN=YES</code> (or) <code>CREATE_NEW_DOMAIN=NO</code>
DOMAIN_HOME	The path for a new WebLogic domain called SOA Domain. For example, <code>DOMAIN_HOME=C:/bea/user_projects/domains</code>
EXISTING_DOMAIN_NAME	This is the name of the existing domain where you wish to create SOA Server as a managed server. This property value should be set only when you set the <code>CREATE_NEW_DOMAIN</code> as NO. For example, <code>EXISTING_DOMAIN_NAME=MySOADomain</code>
ADMIN_SERVER_PORT	This is the port where the existing domain's Admin Server is running. This property value should be set only when you set the <code>CREATE_NEW_DOMAIN</code> as NO. For example, <code>ADMIN_SERVER_PORT=7001</code>
NEW_ADMIN_SERVER_PORT	This is the port where you wish to run the Admin Server. This property value should be set only when you set the <code>CREATE_NEW_DOMAIN</code> as YES. For example, <code>NEW_ADMIN_SERVER_PORT=8001</code>
SOA_SERVER_NAME	The server, which is created, under SOA Domain. For example, <code>SOA_SERVER_NAME=oracleSOAServer</code>
SOA_SERVER_PORT	The port where <code>SOA_SERVER_NAME</code> is running. For example, <code>SOA_SERVER_PORT = 9700</code>
APPS_HOME	The path where applications and adapters will be deployed from. For example, <code>APPS_HOME=C:/bea/user_projects/apps</code>

Property	Description
SOA_HOME	The directory path in which Oracle SOA Suite is installed. For example, SOA_HOME=C:/product/10.1.3.1/OracleAS_1/
DB_URL	This is the URL to connect to ORABPEL schema. For example, DB_URL=jdbc:oracle:thin:@stbbsn10.us.oracle.com:1521:orcl
DB_BPEL_PASSWORD	The password for ORABPEL schema in database. For example, DB_BPEL_PASSWORD=ORABPEL
DB_ESB_PASSWORD	The password for ORAESB schema in database. For example, DB_ESB_PASSWORD=ORAESB
DB_OWSM_PASSWORD	The password for ORAWSM schema in database. For example, DB_OWSM_PASSWORD=ORAWSM
PROXY_SET	The valid values are true or false. Indicates whether proxy settings need to be configured for the server.
PROXY_HOST	The host name of the proxy server. A valid value should be provided, if PROXY_SET is set to true. For example, PROXY_HOST=www-proxy.us.oracle.com
PROXY_PORT	The port where the proxy server is running. A valid value should be provided, if PROXY_SET is set to true. For example, PROXY_PORT=80
NON_PROXY_HOST	The list of non-proxy hosts that are divided by a symbol. A valid value should be provided, if PROXY_SET is set to true. For example, NON_PROXY_HOST=*.oracle.com *.oraclecorp.com localhost
SECURITY_MODEL	This is to specify the security model to be used for the SOA Apps deployments. SECURITY_MODEL Possible Values: [DDOnly CustomRoles CustomRolesAndPolicy Advanced] For example, SECURITY_MODEL=DDOnly
SECURITY_GROUP_NAME	This value is to be changed, if you desire to map the role to an existing group in your realm. Otherwise, SECURITY_GROUP_NAME defaults to SoaGroup.

4. Run the following script from Weblogic_SOA10134_Base folder at the operating system command prompt:

Note:

- Before running the `setup.bat` or `setup.sh` file, set the environment 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.
- Some of the above properties may have been set in the `SOADomain.properties` file, but they should be set again as environment values, since they are needed even before the `SOADomain.properties` can be read by the script.
- Refer to [Upgrade Path for SOA Suite 10.1.3.3 on WebLogic 9.2 Installation](#) for existing SOA Suite 10.1.3.4 on WebLogic 9.2 customers.
- This script will prompt to enter a password for the default 'weblogic' user at the command prompt. The password entered would be set as password for the 'weblogic' user. By default, the 'weblogic' user would have access to the Admin console and all the SOA Application pages.

For...	Run...
Windows XP	<code>setup.bat</code>
Linux	<code>setup.sh</code>

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.

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 domain, 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 WebLogic Server as follows:

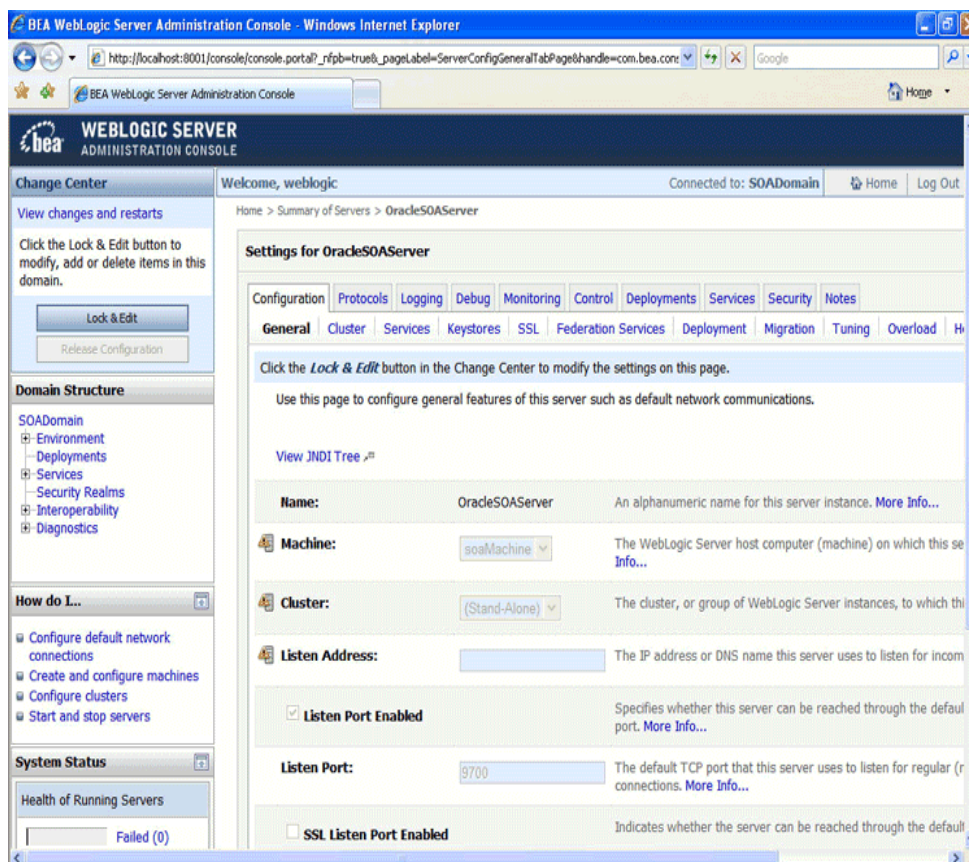
For...	Run...
Windows XP	BEA_HOME\user_projects\SOADomain\bin\startWebLogic.cmd
Linux	BEA_HOME/user_projects/SOADomain/bin/startWebLogic.sh &

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

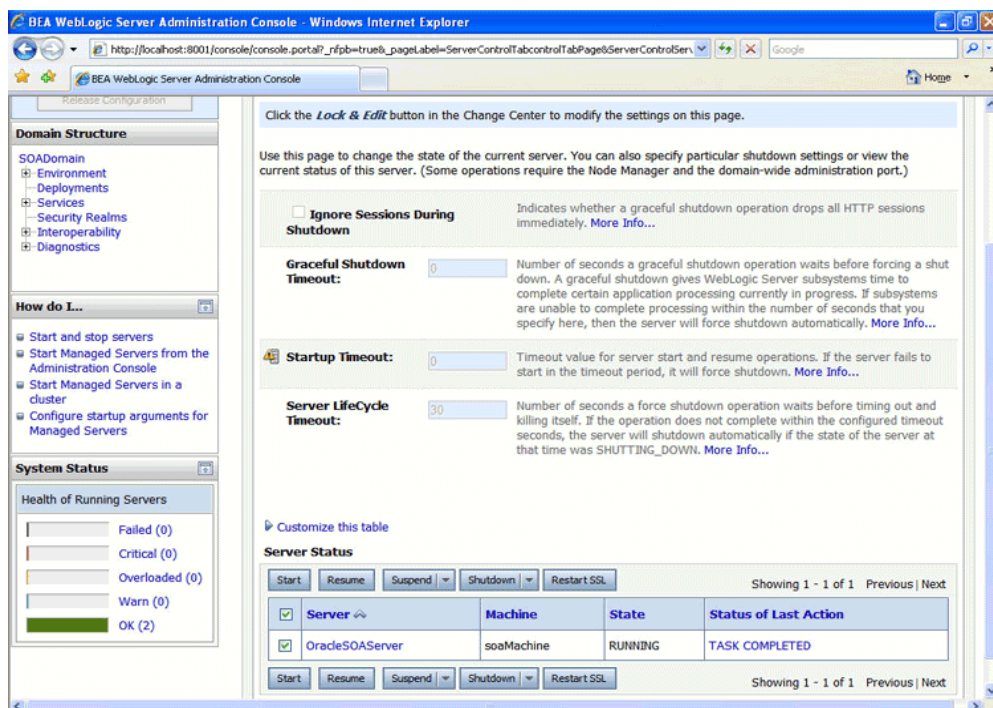
`http://localhost:8001/console`

Installation progress is logged to the `Weblogic_SOA10134_Base\logs\output.log` file.

7. Start the **oracleSOAServer** managed server by following the startup instructions as follows:
- Log in to `http://localhost:8001/console`, using `weblogic` as the username and password. The WebLogic Server Administrative Console window is displayed.
 - 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.

The managed server in SOADomain (OracleSOAServer) should not be started using the `startManagedWeblogic.cmd/sh` commands.

The managed server for SOA uses custom classpath, security policy, and server properties, which will not be set when the managed server is started using the commands. Instead, always use the Admin Server console to start the managed servers.

In order to use the `startManagedWeblogic` commands, include the values for the following attributes in the `setDomainEnv.cmd/sh` files from the Server Start page for OracleSOAServer (OracleSOAServer -> Configuration -> Server Start):

- Classpath
 - Arguments
 - Security Policy File
-

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 10.1.3.4 on WebLogic 9.2

This section describes the various design-time support functions available on WebLogic Server, for the deployment of J2EE applications in JDeveloper. You can deploy BPELPM components on 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.encrypted.password to admin.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 10.1.3.4 (jdevstudio10134.zip) from For Windows - <http://www.oracle.com/technology/software/products/jdev/htdocs/soft10134.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` and `orabpel.jar` files 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.encrypted.password` to `admin.password` of the above user
 - `jndi.url` to **t3://<hostname>:9700**
 - `jndi.InitialContextFactory` to **weblogic.jndi.WLInitialContextFactory**

Creating Connections to Oracle SOA Server

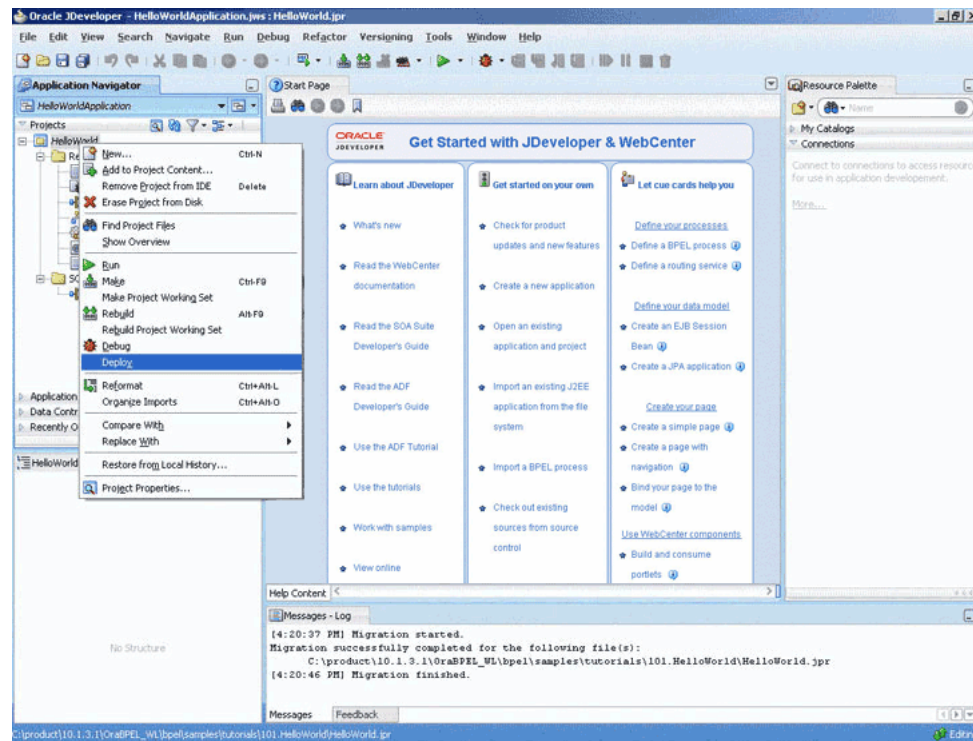
Follow the steps below to create an application server connection and an integration server connection:

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.

```
<application version="1.4" xmlns="http://java.sun.com/xml/ns/j2ee"
(http://java.sun.com/xml/ns/j2ee)

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
(http://www.w3.org/2001/XMLSchema-instance)

xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/application_1_4.xsd"
(http://java.sun.com/xml/ns/j2eehttp://java.sun.com/xml/ns/j2ee/application_1_
4.xsd)>
```

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 file 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 `..\decisionsservices\..` directory of the sample.
2. Note the ear file created in the directory.
3. Create an exploded directory version of the Decisionsservices 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](#)".

Postinstallation Verification Tasks

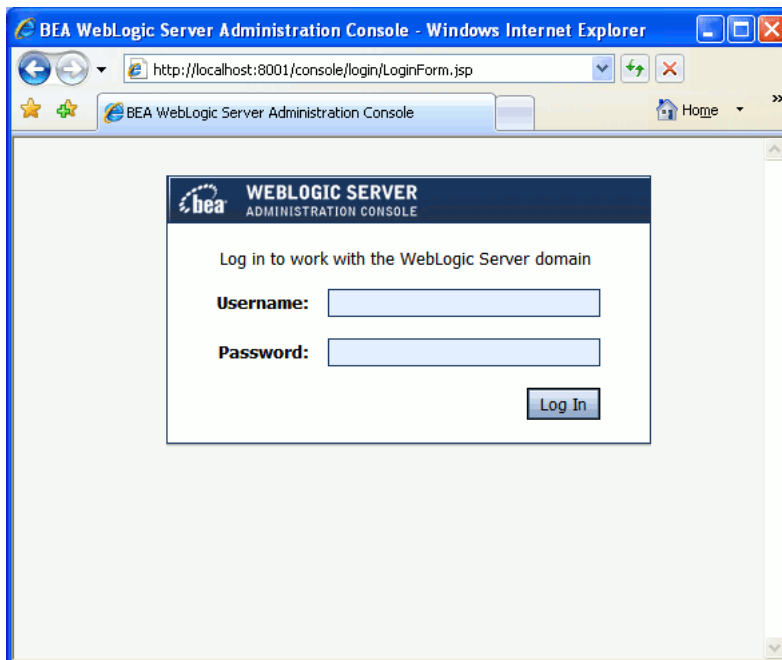
This section describes the postinstallation verification tasks to be performed, and it contains the following topics:

- [Verifying Installation from the 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 WebLogic Server Console

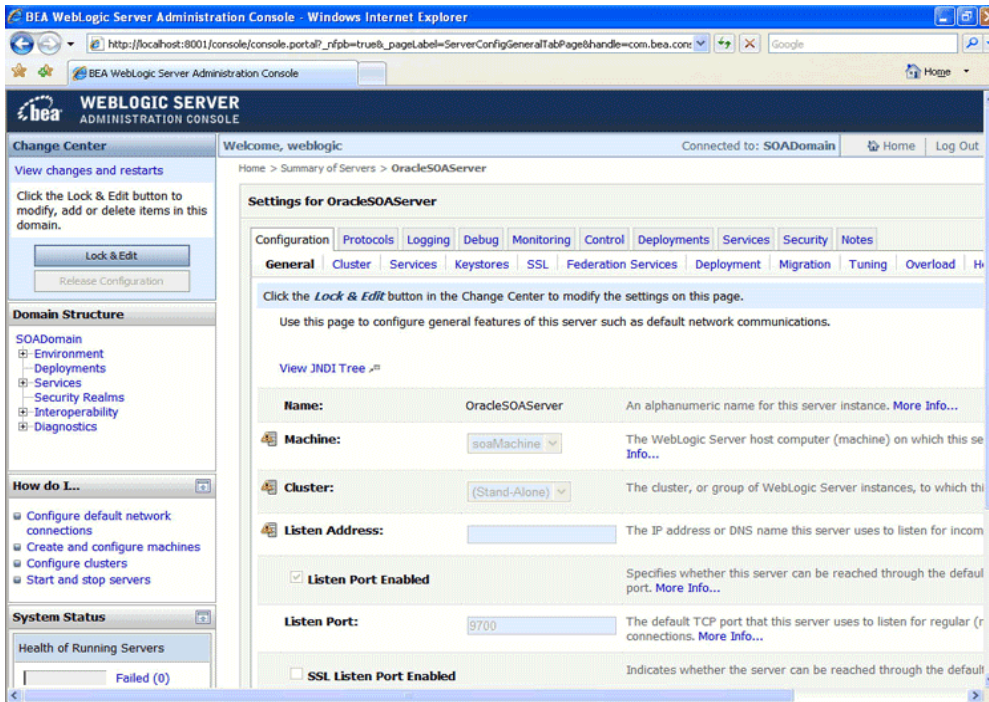
Perform the following steps to check if the Admin console has started:

1. Navigate to <http://localhost:8001/Console>. The Oracle WebLogic Server Admin Console window is displayed.

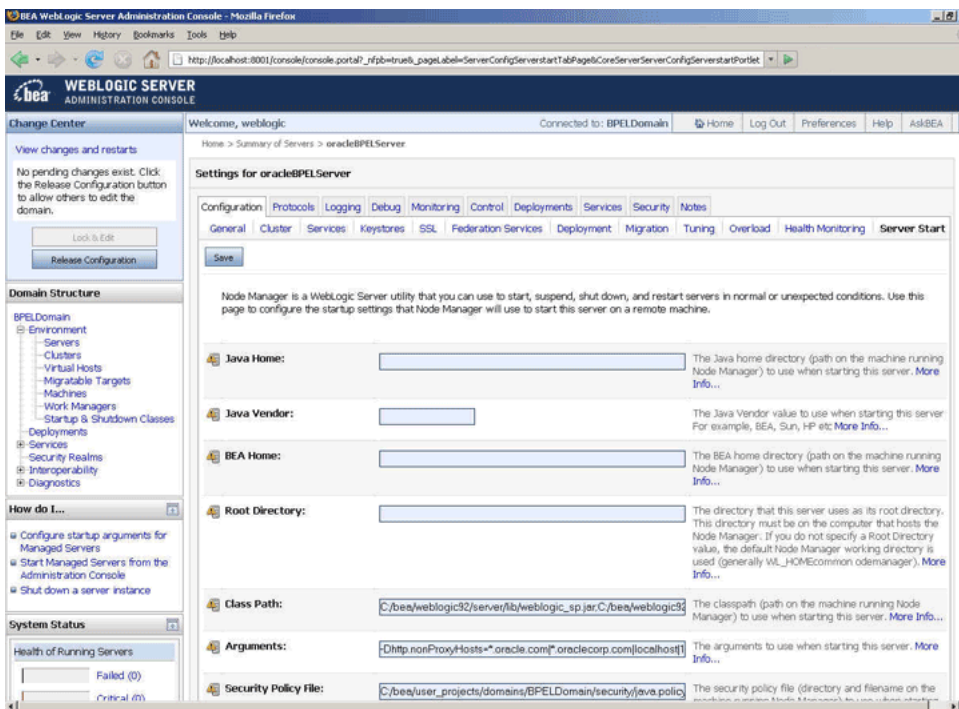


Log in using `weblogic` as the username and password.

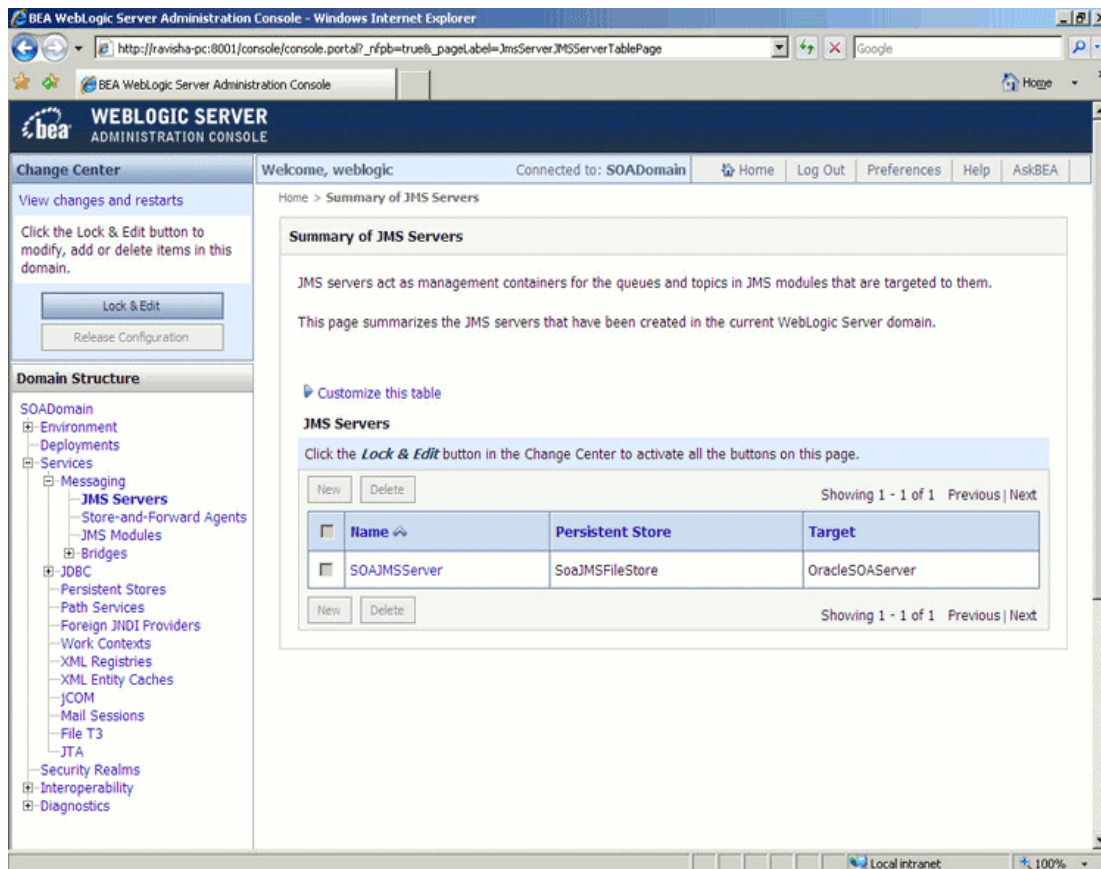
2. Verify that you can view the oracleSOAServer Home page by selecting **Environment -> Servers -> oracleSOAServer -> Configuration**.



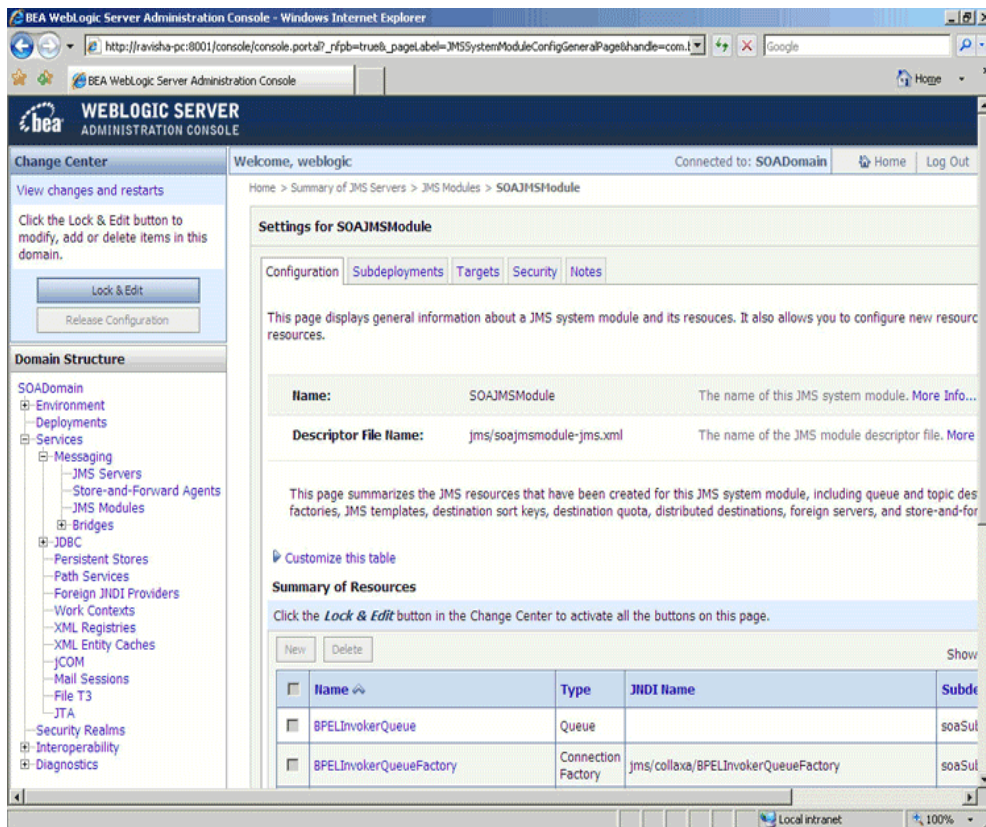
3. Verify that you can view the oracleSOAServer startup properties page by selecting **Environment -> Servers -> oracleSOAServer -> Configuration -> Server Start**. You can also add or modify the server startup properties such as Class Path, Arguments as a BPEL Domain admin.



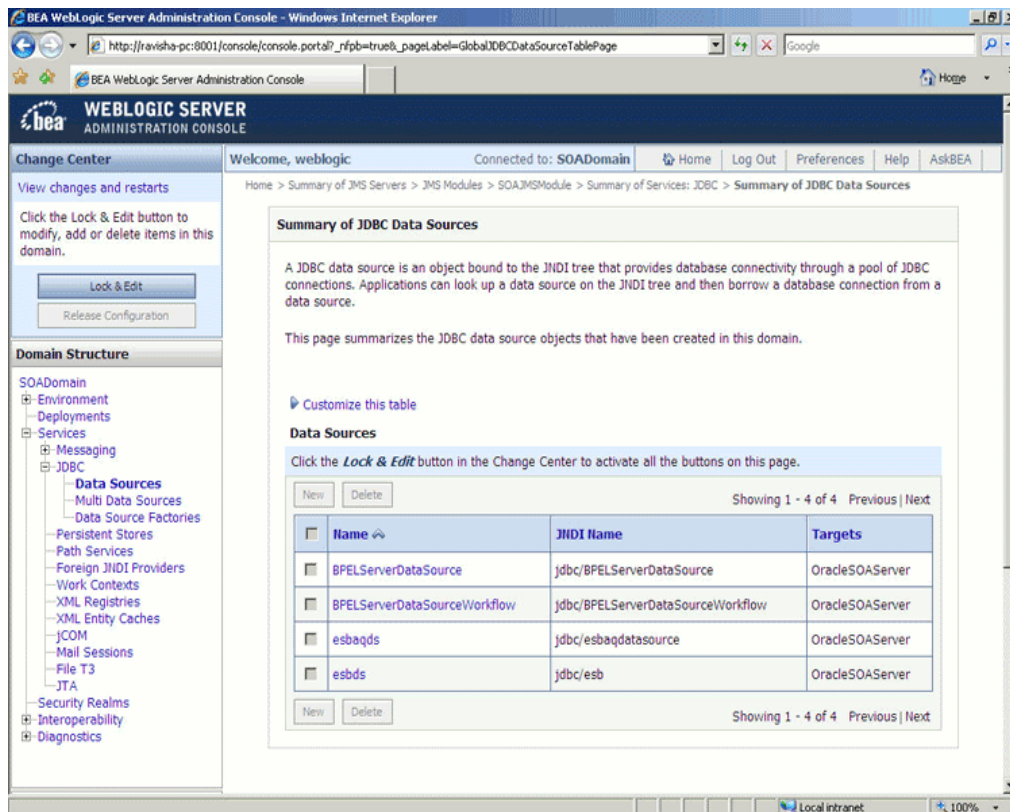
4. Verify that the SOAJMSServer is installed under **Services -> Messaging -> JMS Servers**.



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>/BPEServerConsole/` (Or to the location where the software is installed, for example, `http://<machine-name>:<default_port>/BPEServerConsole/`). 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/Login.jsp`. The OWSM Console window is displayed, as shown in Figure 1-3.

Figure 1–1 BPEL Console Window

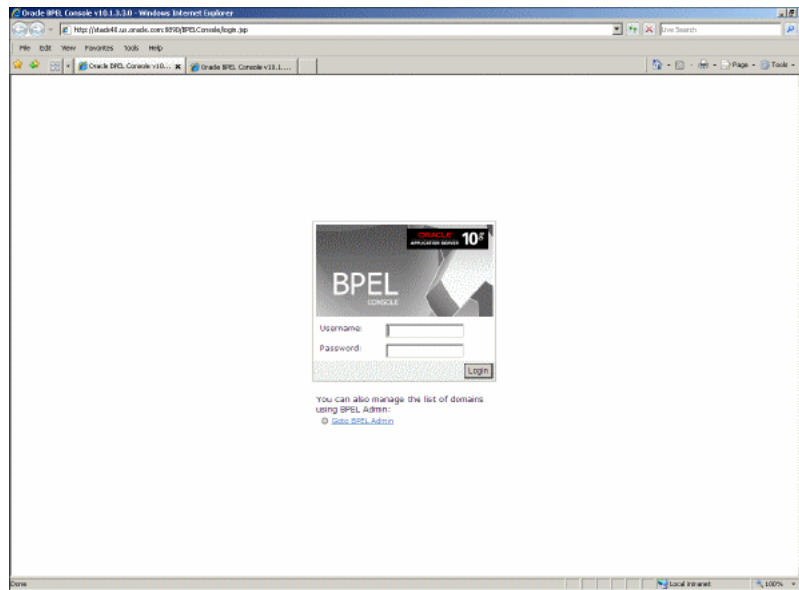
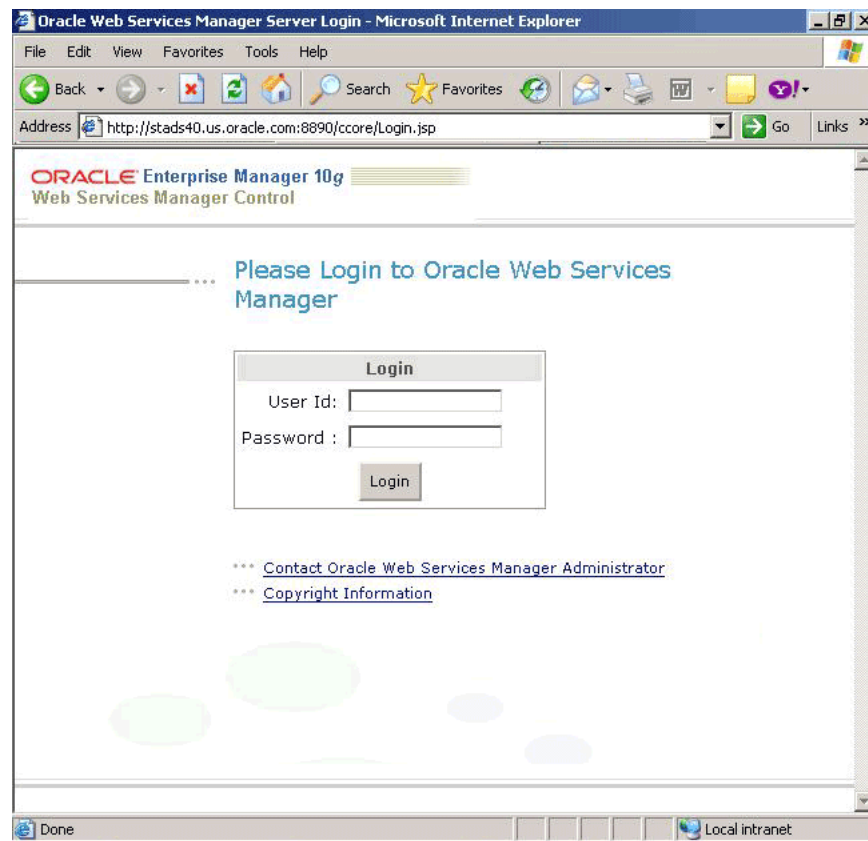
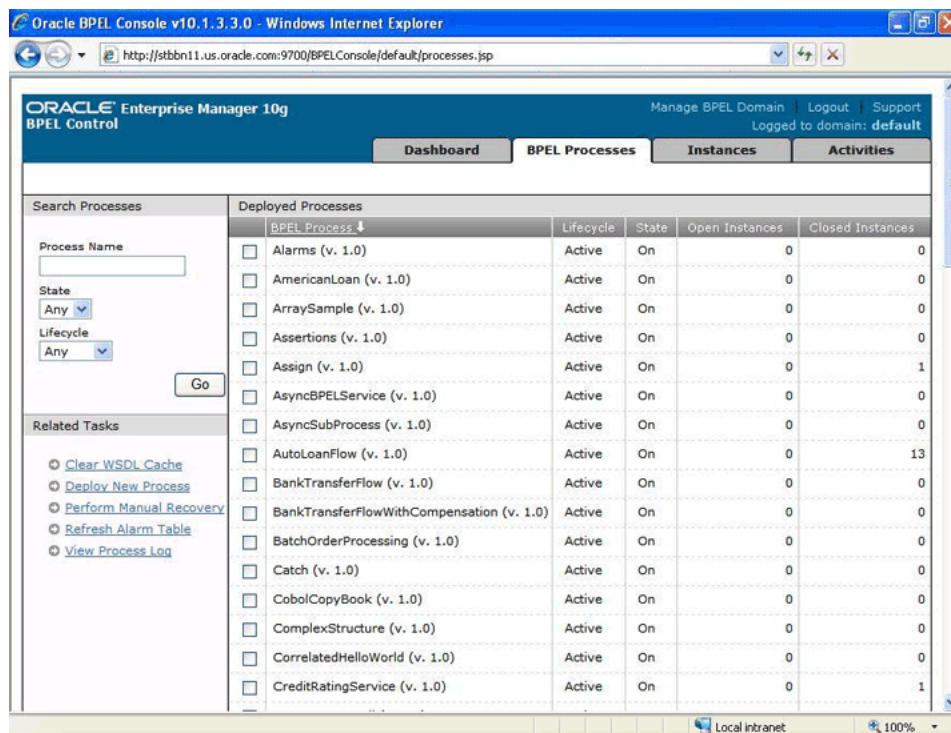


Figure 1–2 ESB Console Window

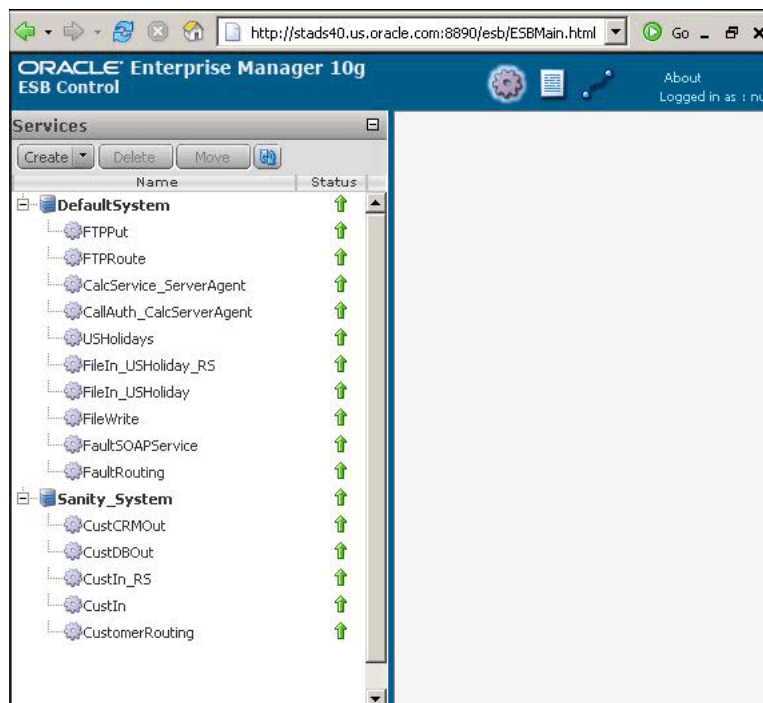


Figure 1–3 OWSM Console Window

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](#).

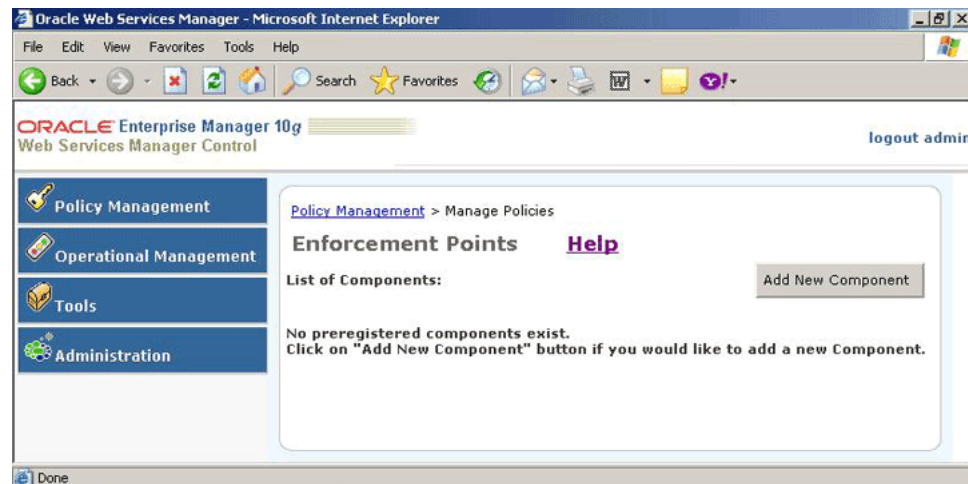
Figure 1–4 Oracle Enterprise Manager BPEL Control

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 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.TaskMetadataServiceLocal
Home</local-home>
<local>oracle.bpel.services.workflow.metadata.ejb.TaskMetadataServiceLocal</loc
al>
<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 `SOADomain.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 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 Administrative Console to deploy the war files to the WebLogic Server.

Access the 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 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 `SOADomain.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 `SOADomain.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 displayed.
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, refer to [Deploying Decision Services EAR](#) under [Deploying Human Task and Decision Services EAR Files](#). Start the application.
3. For each workflow form application, generate the war or ear file, and deploy into **oracleSOAServer**. Start the application.

Note: When deploying BPEL processes into **oracleSOAServer** 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 property 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 10.1.3.4 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** and **orabpel.jar** files 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\AutoLoanDemo\AutoLoanFlow\bpel\decision
services.decs
```

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

```
SOA_
HOME\bpel\samples\demos\AutoLoanDemo\AutoLoanFlow\AutoLoanFlo
w.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>`

Also, the following deployment descriptor changes are needed to successfully deploy the J2EE applications on Weblogic

- Modify the `AutoLoanFlow\public_html\LoanApproval\form\ear\META-INF\application.xml` file.
 Change `<?xml version="1.0" encoding="UTF-8"?>` to `<?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE application PUBLIC "-//Sun Microsystems, Inc.//DTD J2EE Application 1.3//EN" "http://java.sun.com/dtd/application_1_3.dtd">`
- Modify the `AutoLoanFlow\decisionervices\CreditRatingAgent\ear\META-INF\application.xml` file and `AutoLoanFlow\decisionervices\LoanAdvisorAgent\ear\META-INF\application.xml` file.
 Change `<application version="1.4" xmlns="http://java.sun.com/xmlns/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/application_1_4.xsd">` to `<!DOCTYPE application PUBLIC "-//Sun Microsystems, Inc.//DTD J2EE Application 1.3//EN" "http://java.sun.com/j2ee/dtds/application_1_3.dtd">`
- 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. Perform the following steps:

1. Navigate to `Oracle_HOME\bpel\samples\demos\AutoLoanDemo`.
2. Execute `Oracle_HOME\bpel\bin\devprompt.sh` and then run `ant` from `Oracle_HOME\bpel\samples\demos\AutoLoanDemo`. This will generate the directory structure and ear files.
3. Deploy from WebLogic Admin console pointing to `Oracle_HOME\bpel\samples\demos\AutoLoanDemo\AutoLoanFlow\decisionervices\CreditRatingAgent\ear` as an exploded directory.
4. Deploy from WebLogic Admin console pointing to `Oracle_HOME\bpel\samples\demos\AutoLoanDemo\AutoLoanFlow\decisionervices\LoanAdvisorAgent\ear` as an exploded directory.
5. Deploy from WebLogic Admin console pointing to `Oracle_HOME\bpel\samples\demos\AutoLoanDemo\AutoLoanFlowUI\ear` as an exploded directory.
6. Deploy from WebLogic Admin console pointing to `Oracle_HOME\bpel\samples\demos\AutoLoanDemo\AutoLoanFlow\public_html\LoanApproval\form\<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 WebLogic Server version 9.2.

Limitations

Note the following limitations:

- 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 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](#)
- [Decision Services](#)
- [JDK Version Mismatch Causes Admin and Managed Servers Not to Start](#)

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}decisionServiceError',
wsdlMessagePartElement='{http://xmlns.oracle.com/AutoLoanFlow/CreditRatingAgent}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}decisionServiceError requires 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
```

JDK Version Mismatch Causes Admin and Managed Servers Not to Start

The Admin Server and Managed Server fail to startup with the following exception due to the JDK version mismatch:

```
weblogic.management.ManagementException: [Management:141266]Parsing Failure in
config.xml:
javax.xml.namespace.QName; local class incompatible: stream classdesc
serialVersionUID = 4418622981026545151, local class serialVersionUID =
-9120448754896609940
at
weblogic.management.provider.internal.RuntimeAccessImpl.<init>(RuntimeAccessImpl.java:122)
at
weblogic.management.provider.internal.RuntimeAccessService.start(RuntimeAccessService.java:38)
at
weblogic.t3.srvr.ServerServicesManager.startService(ServerServicesManager.java:374)
at
weblogic.t3.srvr.ServerServicesManager.startInStandbyState(ServerServicesManager.java:125)
at weblogic.t3.srvr.T3Srvr.initializeStandby(T3Srvr.java:630)
at weblogic.t3.srvr.T3Srvr.startup(T3Srvr.java:402)
```

If the above error is seen during server startup, then refer to the following links:

http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6267224

<http://forums.bea.com/thread.jspa?threadID=400001009>

Workaround

Set the system property

`-Dcom.sun.xml.namespace.QName.useCompatibleSerialVersionUID=1.0`
for both the Admin Server and the Managed Servers. This issue is fixed in WebLogic 9.2 MP1.

Troubleshooting Tips

The following list explains the troubleshooting tips encountered while installing Oracle BPEL Process Manager with the 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 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";
```

Upgrade Path for SOA Suite 10.1.3.3 on WebLogic 9.2 Installation

The steps to be followed to upgrade a SOA Suite 10.1.3.3 on WebLogic 9.2 installation to 10.1.3.4 are as follows:

1. Stop the managed servers and admin servers in the SOA domain.
2. Apply SOA Suite Patchset 10.1.3.4 on the existing SOA Suite 10.1.3.3 installation (SOA_HOME).

Refer to [Step 5: Apply SOA Suite Patchset 10.1.3.4](#) of the [Installation and Configuration](#) section.

3. Upgrade SOA Schemas to 10.1.3.4.

Run the 10.1.3.4 SOA Schema upgrade scripts for orabpel/oraesb schemas that are available in the SOA Suite 10.1.3.3 Patchset Installation pack.

Execute the following SQL script against ORBPEL schema to upgrade the ORABPEL schema:

```
install\soa_schema_upgrade\bpel\scripts\upgrade_10133_10134_oracle.sql
```

Execute the following SQL script against ORAESB schema to upgrade the ORAESB schema:

```
install\soa_schema_upgrade\esb\sql\oracle\upgrade_10133_10134_oracle.sql
```

4. Apply Patch for Oracle SOA Suite 10.1.3.4 on WebLogic 9.2.

Refer to [Step 6: Apply Patch for Oracle SOA Suite 10.1.3.4 on WebLogic 9.2](#) for more information on applying the patch.

5. Modify the installation properties in the Weblogic_SOA10134_Base\SOADomain.properties file.

Refer to Step 3 of [Step 7: Install and Configure WebLogic Server Version 9.2](#) section.

Choose the following values to retain the existing domain which contains SOA Suite 10.1.3.3 on WebLogic 9.2

- CREATE_NEW_DOMAIN = NO
- DOMAIN_HOME: path of the existing domain's parent directory
- EXISTING_DOMAIN_NAME: name of the existing domain used for SOA Suite 10.1.3.3

- ADMIN_SERVER_PORT: port where the existing domain's admin server is running

Enter appropriate values for other properties in the SOADomain.properties file.

6. Start the Admin server in the SOA Domain.
7. Run the following script from Weblogic_SOA10134_Base folder at the operating system command prompt:

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

This script upgrades the existing domain, which contains the admin server (AdminServer) and Oracle SOA Server managed server (oracleSOAServer). The SOA applications are redeployed and adapter binaries are upgraded to 10.1.3.4. Also, the old SOA applications and adapters will be backed up under <APPS_HOME>/soaApps_backup/<date>.

Note:

- This script will prompt for the password of the default 'weblogic' user at the command prompt.
 - While running the upgrade.bat or upgrade.sh file, set the environment variable BEA_HOME to the 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 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 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 the WL_HOME\server\lib folder.
-
-

SOA Suite 10.1.3.4 HA Setup Using WebLogic Cluster

This appendix describes the configuration steps for installing and configuring clustered Oracle SOA Suite 10.1.3.4 on WebLogic Server 9.2. This appendix contains the following sections:

- [Sample HA Topology](#)
- [Steps for Installation and Configuration of the Sample HA Topology](#)
- [Post Installation Steps](#)

Sample HA Topology

This is a sample topology with the following chosen as its members.

- DOMAIN - Single domain (SOADomain)
- CLUSTER - SOA cluster will contain two managed servers (OracleSOAServer1 and OracleSOAServer2)
- MACHINES - Two machines each hosting one managed server (Machine1 and Machine2)
- ADMIN SERVER - Single admin server will run on Machine1

You can extend the same procedure to configure a cluster with more members as per your requirement. [Figure B-1](#) describes the sample topology diagram.

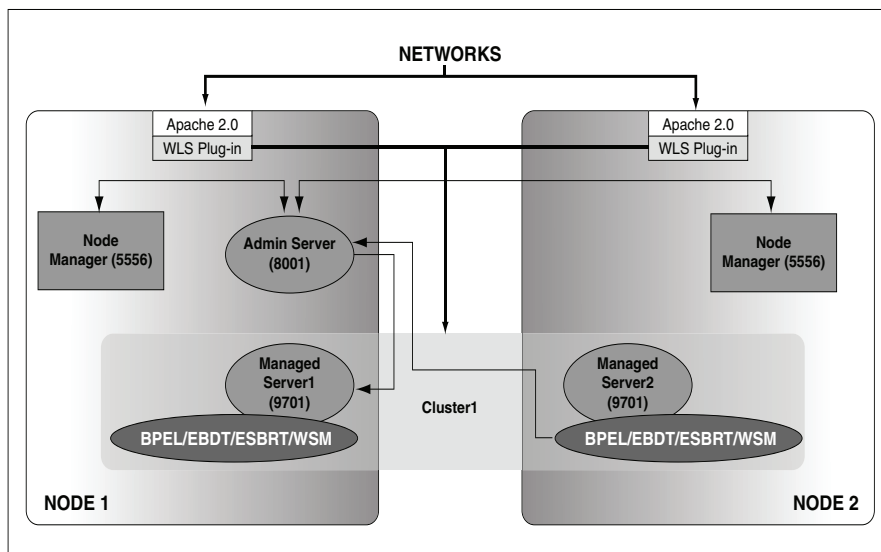
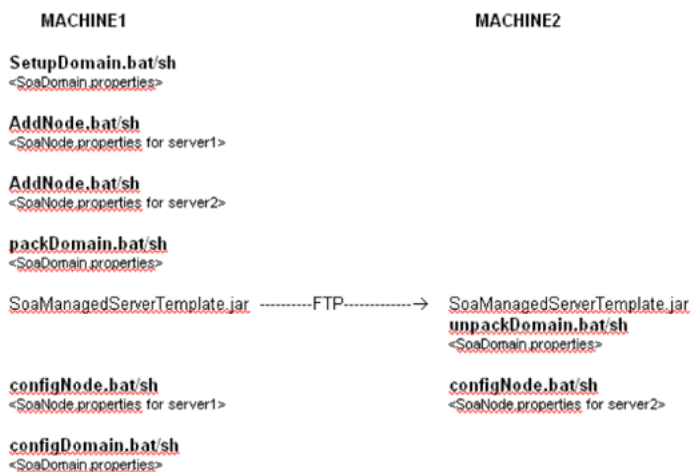
Figure B–1 Topology Diagram

Figure B–2 describes the flowchart of the script to be executed for the sample topology.

Figure B–2 Scripts Execution Flowchart

Steps for Installation and Configuration of the Sample HA Topology

This section consists the detailed steps for installation and configuration of the sample HA topology mentioned in [Sample HA Topology](#).

Prerequisites

The following prerequisites should be met for the installation and configuration of the sample HA topology:

1. Install WebLogic Server 9.2 on both machines (Machine1 and Machine2).
2. Download the `Weblogic_SOA10134_HA.zip` file and extract on both the machines.

Installing Database

Common Dehydration Store:

1. Install Oracle Database 10.2.x on a common machine (either Machine1 or Machine2).
2. Run IRCA from SOASuite 10.1.3.1 Install Pack/Disk1/Soa_schemas/irca to create ORABPEL/ORAESB/ORAWSM schemas on the above database.
3. Run the 10.1.3.4 schema upgrade scripts for ORABPEL/ORAESB schemas from SOASuite 10.1.3.4 Install Pack/Disk1/Soa_schemas_upgrade.

SOA Suite 10.1.3.4 (for OC4J) Installation on both Machines

Perform the following steps to install Oracle SOA Suite 10.1.3.1:

1. Install Oracle SOA Suite 10.1.3.1 for OC4J on both the machines (Machine1 and Machine2).
2. Choose Basic Installation of SOA Suite 10.1.3.4. For Linux, choose the above-created database during installation. Choose the same path for installation on both machines.
3. After installation, ensure that the ORACLE_HOME on both machines is pointing to the same location.
4. Apply PatchSet 4 (Oracle SOASuite 10.1.3.4 for OC4J) on both the installations.
5. Apply Patch for Oracle SOA Suite 10.1.3.4 on WebLogic 9.2.

Refer to [Step 6: Apply Patch for Oracle SOA Suite 10.1.3.4 on WebLogic 9.2](#) for more information on applying the patch.

6. Install Oracle SOA Suite 10.1.3.4 for OC4J on both the physical machines (Machine1 and Machine2).

Note: It is *necessary* to have Oracle SOA Suite for OC4J installed on both the physical machines in the above scenario.

Steps to Configure HA Setup for SOA 10.1.3.3 on WebLogic 9.2

Perform the following steps to configure HA setup for SOA 10.1.3.3 on WebLogic 9.2.

Note: All paths in this section are relative to weblogic_SOA10134_HA unzip location.

1. Make appropriate entries in the SOADomain.properties file and run setupDomain.bat/sh on Machine1.

The domain wide details such as Domain properties, Admin Server properties, DataSource properties will need to be filled in SOADomain.properties.

This script will prompt for the password to be set for the default 'weblogic' user.

2. Make appropriate entries in the SOANode.properties file for OracleSOAServer1 and run addNode.bat/sh on Machine1.

Modify the SOANode.properties file entries for OracleSOAServer2 and run the addNode.bat/sh script again on same Machine (Machine1).

This can be repeated further for more managed servers in the cluster. Ensure that for every run of this script the `SOANode.properties` values are changed for the appropriate managed server.

This script will prompt for the user id and password for connecting to the admin server.

3. After adding all the required managed servers, run the `packDomain.bat/sh` script on Machine1.

Ensure that this step is performed only after all the managed servers are added to the domain using the `addNode.bat/sh` script.

4. Transfer the `SOAManagedServerTemplate.jar` from `WL_HA_SOA_Installables/template` directory of Machine1 to `WL_HA_SOA_Installables/template` directory of Machine2.

This step can be repeated for more machines if required.

5. Run the `unpackDomain.bat/sh` script on Machine2. This step can be repeated for more machines, if required.
6. Edit the `SOANode.properties` file and run `configNode.bat/sh` on Machine1. and Machine2. Ensure that the `SOANode.properties` file entries are updated for each machine.

This script will prompt for the user id and password for connecting to the admin server.

7. Run `configDomain.bat/sh` on Machine1. This needs to be run only on Machine1. The SOA Applications will be deployed to the cluster in this step.

This script will prompt for the user id and password for connecting to the admin server.

Post Installation Steps

Perform the following post installation steps:

NodeManager Setting

Navigate to the `<BEA_HOME>/weblogic92/server/bin` directory and edit the `startNodemanager` script to add following property as part of the `JAVA_OPTIONS` arguments.

On Windows:

```
set JAVA_OPTIONS=%JAVA_OPTIONS%
-Dweblogic.security.SSL.ignoreHostnameVerification=true
```

On Unix:

```
set JAVA_OPTIONS="$ {JAVA_OPTIONS}
-Dweblogic.security.SSL.ignoreHostnameVerification=true"
```

Configure Collaxa-config for BPEL

Update the following properties in `SOA_HOME/bpel/system/config/collaxa-config.xml` on both the machines to configure HA for BPEL runtime:

- `bpelPlatform = weblogic_8`

- `soapServerUrl = LoadBalancerHost:port`
- `soapCallbackUrl = LoadBalancerHost:port`
- `enableCluster = true`
- `clusterName = LoadBalancerHost:port`

Configure JGroups for BPEL

Update the `<SOA_HOME>/bpel/system/config/jgroups-protocol.xml` file on both the machines to setup JGroups between the BPEL instances.

This is used for BPEL process and adapter state synchronization between BPEL instances.

ESB Design Time (ESB-DT)

It is not recommended to have ESB-DT running in the same HA cluster as ESB Runtime (ESB-RT). Select the option mentioned below:

1. Run ESB-DT in a separate non-clustered managed server outside the domain of the cluster.

Download and install Oracle SOA Suite 10.1.3.4 for Weblogic 9.2 and complete the setup.

After running the post installation, disable or delete all the deployments except esbservices deployment on the managed server.

Modify the `DT_OC4J_HOST` and `DT_OC4J_PORT` parameter in `ESB_PARAMETER` (`oraesb`) table with `hostName/port` of the new managed server.

For ESB-DT HA, an Active-Passive cluster needs to be setup separately for ESB-DT.

ESB-DT needs to run as a singleton instance and the `host/port` of this instance should be updated in `DT_OC4J_HOST` and `DT_OC4J_PORT` fields of `ESB_PARAMETER` table in the `ORAESB` schema.

Load Balancer

Perform the following steps to configure external load balancer Apache for WebLogic Cluster setup:

1. Refer to <http://e-docs.bea.com/wls/docs92/plugins/apache.html> to install and configure Apache plugin on WebLogic server.
2. Mention the load balancer hostname and port name in `<SOA_HOME>/bpel/utilities/ant-orabpel.properties` and `<SOA_HOME>/bpel/system/config/collaxa-config.xml` files on all machines.

Addresses

The SOA Apps will be available at the following addresses post installation.

- **Admin Server:** `http://Machine1: 8001/console`
- **BPEL Console:** `http://LoadBalancerhost: port/BPELConsole`
- **OWSM:** `http://LoadBalancer: port/ccore/Login.jsp`
- **WorkList App:** `http://LoadBalancerhost: port/`

- **RuleAuthor**: `http://LoadBalancerhost: port/ruleauthor`
- **ESB-DT**: `http://Machine1: 9700/esb`

Customizing Role-based Security for SOA Applications

A default realm (MyRealm) is created for the SOA domain with one default user - 'weblogic'. The password for the weblogic user can be chosen when prompted by the `setup.bat/sh` script (in case on non-HA installation) `setupDomain.bat/sh` script (in case on HA installation).

The other scripts will prompt for this password for authentication to connect to the admin server when they are executed.

The SOA applications (BPEL/ESB/OWSM/Rule Author) use role-based authentication. The roles used by these applications are mapped to a group called 'SoaGroup' in the default realm. SoaGroup, by default, contains only one user - 'weblogic'. Thus, the 'weblogic' user can access the Admin console and all the SOA applications.

To map to an existing group of users to the SOA applications for authentication/authorization use the overriding property `SECURITY_GROUP_NAME` in `SOADomain.properties` prior to running setup scripts.

Also, by default, the security role to user/group mapping for the SOA applications is packaged as part of the deployment descriptors. The security model of 'DDOnly' is configured by default.

In order to configure the security roles, users/groups from the WebLogic console, use the `SECURITY_MODEL` property of `SOADomain.properties`. The permitted values for `SECURITY_MODEL` are:

- DDOnly
- CustomRoles
- CustomRolesAndPolicy
- Advanced

Refer to WebLogic Documentation for details on the above security models.

Additional Configuration Steps for WebLogic Server

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

- [Configuring External LDAP](#)
- [ESB System Configurations](#)
- [Changing to File-based Slide Repository](#)
- [Human WorkFlow API Clients](#)
- [Identify Service Configuration](#)

Configuring External LDAP

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

- [Step 1: Create an Authentication Provider](#)
- [Step 2: Configuring LDAP in 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: Configuring LDAP in WebLogic Server

WebLogic Server does not support or certify any particular LDAP server. Any LDAP v2 or v3 compliant LDAP server should work with 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 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>

ESB System Configurations

Ensure that the system information for the ESB services deployed are as follows:

- `Virtual Host`: The hostname of ESB design-time instance
- `Port`: The port number of ESB design-time instance
- `Topic Location`: `ESB_JAVA_DEFERRED`

The value of the 'Connection Factory Location' parameter does not matter for ESB on WebLogic 9.2, since ESB, by default, uses AQ messaging and uses the AQ JMS API to connect to the AQ Messaging topics.

Changing to File-based Slide Repository

By default, ESB on WebLogic 9.2 is configured to use database-based slide as the metadata repository. Perform the following steps configure ESB on WebLogic 9.2 to use file-based slide repository:

1. Rename `Domain_file.xml` to `Domain.xml` in the `<SOA_HOME>/integration/esb/config` directory.
2. Rerun IRCA for ORAESB schema from the `<SOASuite 10.1.3.1 Installation pack>\install\soa_schema\irca\irca oraesb` directory.
3. Execute the following script against ORAESB schema to upgrade it to 10.1.3.4:
`<SOASuite 10.1.3.4 Patchset Installation pack>\install\soa_schema_upgrade\esb\sql\oracle\upgrade_10131_10134_oracle.sql`
4. Edit the `Weblogic_SOA10134_Base/ESB_data.aq.sql` file and update the values for the following:
 - `ESB_PARAMETER` properties
 - `DT_OC4J_HOST`
 - `DT_OC4J_PORT`
5. Execute the following script against ORAESB schema to modify it for WebLogic Server:

```
Weblogic_SOA10134_Base/ESB_data.aq.sql
```

Human WorkFlow API Clients

Clients for invoking the Human WorkFlow APIs should include the following additional system properties on the Client side:

```
-Djavax.xml.parsers.DocumentBuilderFactory=oracle.xml.jaxp.JXDocumentBuilderFactory
-Djavax.xml.parsers.SAXParserFactory=oracle.xml.jaxp.JXSAXParserFactory
-Djavax.xml.transform.TransformerFactory=oracle.xml.jaxp.JXSAXTransformerFactory
-Djavax.xml.soap.MessageFactory=oracle.j2ee.ws.saa.j.soap.MessageFactoryImpl
-Djavax.xml.soap.SOAPFactory=oracle.j2ee.ws.saa.j.soap.SOAPFactoryImpl
-Djavax.xml.soap.SOAPConnectionFactory=oracle.j2ee.ws.saa.j.client.p2p.HttpSOAPConnectionFactory
-Djavax.xml.soap.SOAPElementFactory=oracle.j2ee.ws.saa.j.soap.SOAPFactoryImpl
```

Identify Service Configuration

The identity service in the default installation leverages the same model as the SOA Suite deployment on OC4J to obtain users from the `Jazn.com` realm.

This can be changed by configuring the SOA_HOME/bpel/system/services/config/is_config.xml file. Refer to http://download.oracle.com/docs/cd/B31017_01/integrate.1013/b28982/service_config.htm for more details. Also, refer to the is_config.xml example files provided at SOA_HOME/bpel/system/services/config/ldap to connect to the external LDAP providers.

Index

A

auto loan demo, 1-30

C

components
Oracle SOA Suite, 1-2

I

installation
WebLogic Application Server, 1-1, 1-3
Integration Repository Creation Assistant
usage, 1-30

K

known issues
WebLogic Application Server
decision services, 1-35
JMS adapter, 1-35
WebLogic Application Server with Oracle SOA
Suite, 1-35

L

limitations
WebLogic Application Server with Oracle SOA
Suite, 1-34

O

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

P

postinstallation verification tasks
WebLogic Application Server with Oracle SOA
Suite, 1-17

S

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

T

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

W

WebLogic Application Server
auto loan demo, 1-30
installing, 1-1, 1-3
known issues, 1-35
limitations, 1-34
postinstallation verification tasks, 1-17
system requirements, 1-3
troubleshooting tips, 1-37
workarounds, limitations, and unsupported
features, 1-34
WebLogic Application Server with Oracle SOA
Suite, 1-30
WebLogic Server, 1-1
workarounds, limitations, and unsupported features
WebLogic Application Server with Oracle SOA
Suite, 1-34

