

Oracle[®] Insurance Claims Adjudication for Health

Installation Guide

Version: 2.12.0.0.0

Part number: E27047-01

December 5, 2011

Copyright © 2011, Oracle and/or its affiliates. All rights reserved.

Installation Guide

Version: 2.12.0.0.0

Part number: E27047-01

December 5, 2011

License Restrictions Warranty/Consequential Damages Disclaimer

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

Warranty Disclaimer

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

Restricted Rights Notice

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, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

Hazardous Applications Notice

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Trademark Notice

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

Third Party Content, Products, and Services Disclaimer

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Table of Contents

Preface

ix Audience

ix Typographic Conventions

ix Command Syntax

x Product name

x Related Documentation

Chapter 1 Introduction

1-1 Concepts

1-1 Application Modularity

1-1 Database Users and Roles

1-4 Enabling Replication of Setup Data

1-4 Release directory structure

1-4 Overview of the directory structure

Chapter 2 Initial installation requirements

2-1 Install and configure an OHI Database

2-1 Install Oracle Database software

2-1 Create OHI Claims database

2-3 Install and configure Oracle Fusion Middleware

2-4 Installing Oracle WebLogic Server

2-10 Installing Oracle Application Development Runtime

2-17 Configuring Oracle Fusion Middleware for running ADF Applications

2-25 Domain configuration for OHI Claims

2-31 Setting Up a WebLogic Cluster for running OHI Claims on multiple nodes

2-37 Initial configuration for OHI Claims in Oracle Fusion Middleware

- 2-37 Logging configuration
- 2-38 Set required defaults
- 2-38 Set up a directory for File Exchange
- 2-39 Authentication and User Provisioning
- 2-42 Internal System User

2-44 System Sizing Guide

- 2-44 Overview
- 2-45 JVM Options
- 2-47 Connection Pool Sizing Guidelines

Chapter 3 Release installation

3-1 Install database objects

- 3-1 Change Installation Configuration
- 3-1 Run Installer
- 3-4 Enable Total Recall (optional)

3-7 Install application

- 3-7 Creating WebLogic Work Manager
- 3-7 Configuring OID Authentication Provider
- 3-29 Testing LDAP Configuration Without Deploying OHI Claims Application (Optional)
- 3-29 Set up JDBC Data Sources
- 3-45 Installing the UI Customization Library through WLS Admin Server Console
- 3-49 Installing The OHI Application Through WLS Admin Server Console
- 3-52 Installing OHI Claims Application To Use Custom Work Manager (OR) To Change Context-Root

3-0 Validate Installation

3-0 Configure OHI Claims properties file

3-0 OHI Claims Command Line Properties

Chapter 4 Appendices

4-1 Appendix A - Web Service Locations

4-2 Appendix B - Seed Data

4-2 Types of Seed Data

4-2 Restrictions on using Seed Data

Preface

This document describes the installation information for the Oracle Insurance Claims Adjudication for Health application.

AUDIENCE

This document is intended for database managers and system managers and others responsible for the installation of Oracle products.

TYPOGRAPHIC CONVENTIONS

The following conventions are used in this document

Note Note layout is used to draw the readers attention to additional information on the topic.

Warning Warning layout is used to warn the reader that not handling correctly the instructions that are listed may cause errors in the (installation of) software or data.

Monospace formatted text is used to display OS commands, shell scripts, SQL queries, etc.

COMMAND SYNTAX

Command syntax is represented in monospace font .

The following conventions apply to command syntax:

`monospace formatted`

Monospace type indicates OS commands, directory names, user names, path names, and file names.

brackets []

Words enclosed by brackets indicate keys (e.g., [Return] Key).

Note Brackets have a different meaning when used in command syntax.

italics

Italics indicates a variable, including variable parts of the file names. It is also used for emphasizing.

UPPERCASE

Uppercase letters indicate Structured Query Language (SQL) reserved words, initialization parameters and environment variables.

backslash \

Each backslash indicates a command that is too long to fit on one line:

```
dd if=/dev/rdsk/c0t1d0s6 of=/dev/rst0 bs=10b \  
count=10000
```

braces { }

Braces indicate mandatory items: .DEFINE {macro1 }

brackets []

Brackets indicate optional items: cvtcr *termname* [*outfile*]

Note Brackets have a different meaning when used in ordinary text.

ellipses ...

Ellipses indicate a random number of similar items:

CHKVAL *fieldname value1 value2 ... valueN*

italics

Italics indicates a variable. Replace the variable by value:

library_name

vertical bar |

The vertical bar allows choosing between two or more alternatives enclosed in braces or brackets:

SIZE *filesize* [K|M]

PRODUCT NAME

'OHI Claims' and 'OHI Claims Adjudication' are used in this document as an alias for the product name Oracle Insurance Claims Adjudication for Health.

RELATED DOCUMENTATION

For more documentation on Oracle products, see [Oracle technetwork](#)¹.

1. <http://www.oracle.com/technetwork/indexes/documentation/index.html>

Introduction

Disclaimer:

This document is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

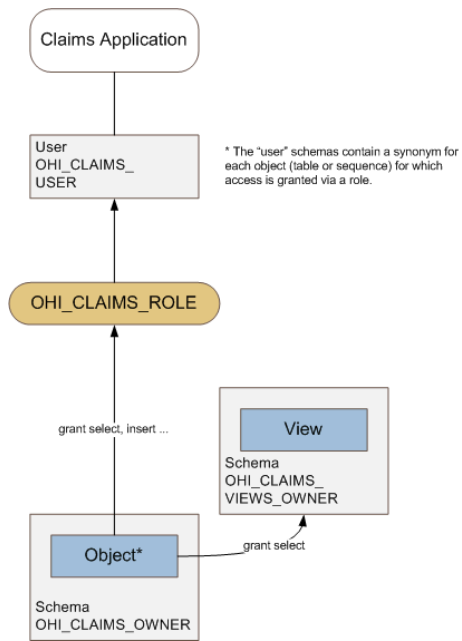
CONCEPTS

APPLICATION MODULARITY

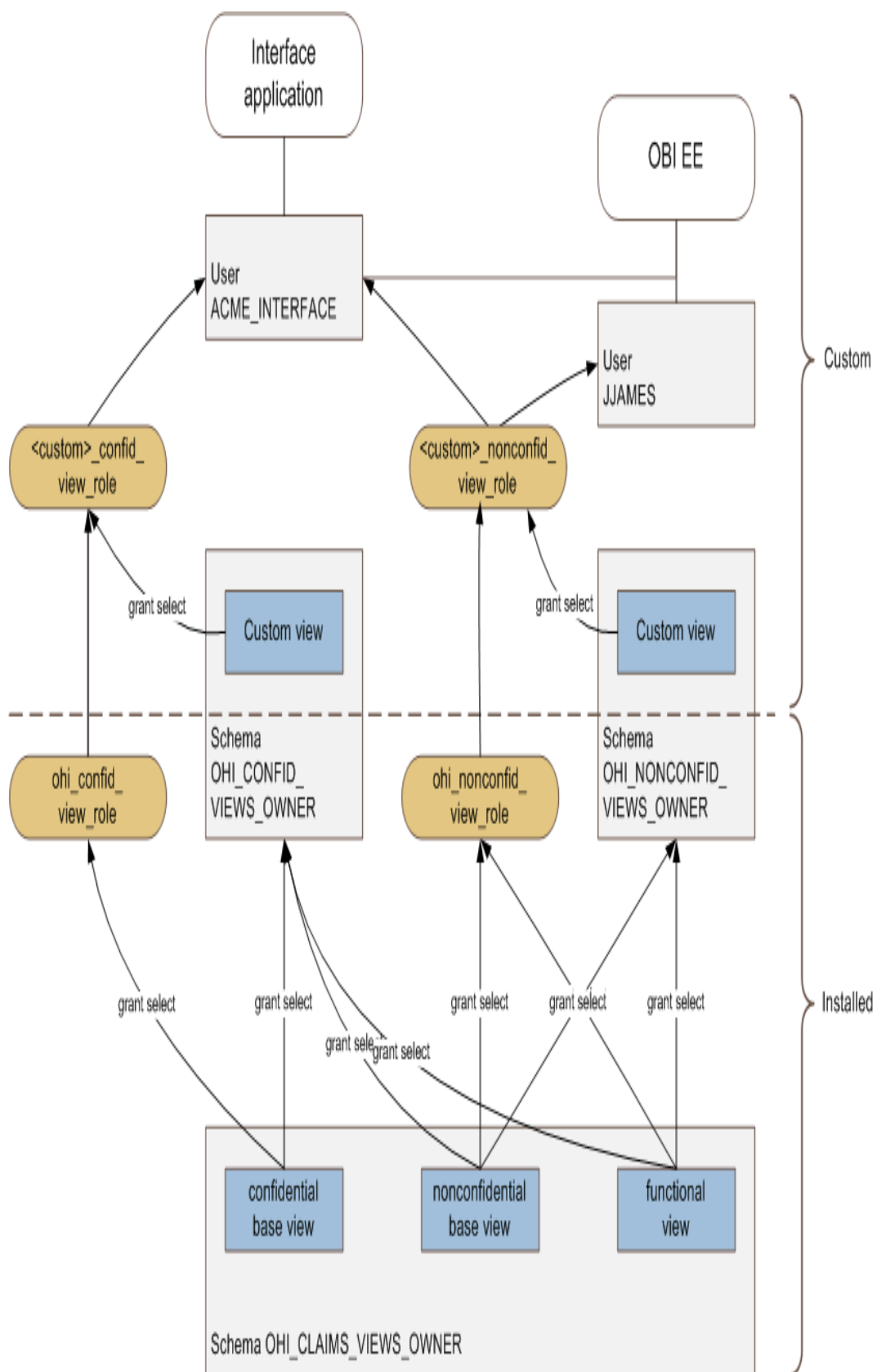
The Oracle Health Insurance suite (OHI) is composed of several applications. In order to ensure full compatibility between these applications, they share a common base. The applications you have licensed are installed and upgraded together; each application has the same version number.

DATABASE USERS AND ROLES

In order to support the modularity, each application has a separate database schema. The following diagram shows the schemas and their relationships:



The diagram below shows how the operational reporting views and the corresponding roles are defined, and how the roles can be assigned for specific purposes:



ENABLING REPLICATION OF SETUP DATA

During the lifecycle of the application, there is a regular need to transport seed data or setup data from one environment to another using the Configuration Migration tool. This data is identified by an ID, a numeric field.

In order to prevent clashes between data created in one environment and existing data in the target environment, it is a prerequisite to ensure that a generated ID is unique across environments. The mechanism used for this purpose is to have the last digit of an ID indicate the source of the row. We recommend the following convention:

Source environment	Description	Discriminator Digit	Examples
OHI Factory	This is the environment at Oracle where the seed data is maintained. The seed data delivered by Oracle always has an ID ending on a zero.	0	17650 17660
Setup	The environment in which you maintain your setup and configuration. This environment should not contain test data.	1	6341 6351
Production	The production environment.	2	3452 3462
Test	The environment in which you test the setup. Setup changes made in the Setup environment are transferred to this environment for testing.	3	165423 165433

Plan the environments and assign unique discriminator values for each environment.

RELEASE DIRECTORY STRUCTURE

OVERVIEW OF THE DIRECTORY STRUCTURE

The distribution contains a number of directories that contain all the necessary information and sources to perform the install. The root directory is the directory where you decide to host the released files. It can be any location or name of your choosing and will be referenced throughout this document as <OHI_ROOT>.

For the database installation it is required to have this directory structure available on a windows based machine running Oracle database client software. For the Application server installation, the required installation / configuration files can be made available by copying them to a location on the server or sharing them from their original location.

Initial installation requirements

This chapter lists specific instructions for installation of the Oracle software components that are required to run the OHI Claims Adjudication application.

See the Certification Guide for specific versions of operating systems and Oracle software that the OHI Claims Adjudication application is certified to work with.

Note All OHI Claims Adjudication releases contain the *complete* application. The only difference between installing OHI Claims Adjudication for the *first time* and *upgrading* it to a new release, is thus the *pre-installation activities* (which only need to be executed when installing OHI Claims Adjudication for the *first time*).

When the initial installation requirements are met, continue with the chapter [Release Installation](#)¹.

INSTALL AND CONFIGURE AN OHI DATABASE

INSTALL ORACLE DATABASE SOFTWARE

First install Oracle Database software required for Oracle Insurance Claims Adjudication for Health (OHI Claims); for specific certification details see OHI Claims Certification Guide.

Set up Real Application Clusters

Set up RAC when required.

CREATE OHI CLAIMS DATABASE

Now create the OHI Claims database. For this activity the following requirements and restrictions apply.

Character Set

The character set of the database must be AL32UTF8.

Block size

For OHI Claims, use an 8K block size.

1. 1742-DSY

Tablespaces

Make sure the following tablespaces exist:

- OHI_CLAIMS_TAB
- OHI_CLAIMS_IDX

All tablespaces must be created 11gR2 default style (locally managed, system/uniform managed extent allocation, Automatic Segment Space Management).

Temporary tablespace

A default temporary tablespace TEMP (this name is mandatory) should be created.

Undo

Automatic undo must be used.

Parameters

- OPTIMIZER_MODE = FIRST_ROWS_10
- NLS_LENGTH_SEMANTICS = CHAR
- STATISTICS_LEVEL=TYPICAL

Note Unless specified otherwise, keep all optimizer parameters (gv\$sys_optimizer_env) default.

Required privileges

OHI Claims uses queues in the Oracle database. The owner of the queue objects, the base owner schema, requires execute privileges on the SYS.DBMS_AQIN package.

For installing OHI Claims database artifacts the SYSTEM account is used. In this process also database grants are given by the system database user. To be able to do that, SYSTEM user needs GRANT ANY OBJECT PRIVILEGE (without grant option).

Functional reporting views are based on access restrictions, that use a context to determine the user. This context needs to be created as user SYS.

Thus, following commands should be executed as SYS:

```
connect sys as sysdba
GRANT EXECUTE ON sys.dbms_aqin TO system WITH GRANT OPTION;
GRANT GRANT ANY OBJECT PRIVILEGE TO system;
create context FUN_USER_CONTEXT using OHI_CLAIMS_OWNER.FUN_CONTEXT_PKG;
```


Creating additional schemas in the database

Oracle recommends that the Oracle database instance that is used by OHI Claims is used solely for the purpose of running the OHI Claims system.

Warning In the case that additional database schemas are created in the Oracle database instance, make sure that these are not prefixed with *OHI*.

Set up Total Recall (optional)

The Total Recall Option of Oracle Server (also known as Flashback Archiving) is used to log changes to setup tables. Configuring which table to archive and how is considered a responsibility of the database administrator.

In order to use Flashback Archiving the following settings need to be made:

- The user that will be used to switch archiving on tables on and off should be granted the "FLASHBACK ARCHIVE ADMINISTER" privilege (grant FLASHBACK ARCHIVE ADMINISTER to <user>).
- This user should be granted "ALTER TABLE" rights on the tables that need to be archived (or stopped being archived).
- It is advisable to create a separate tablespace for the Flashback Archive.
- Create a Flashback Archive, for example:

```
CREATE FLASHBACK ARCHIVE [DEFAULT] fda1 TABLESPACE tbs1 QUOTA 10G  
RETENTION 5 YEAR;
```

INSTALL AND CONFIGURE ORACLE FUSION MIDDLEWARE

OHI Claims runs on an Oracle Fusion Middleware Application Server. This may also be referred to as Oracle WebLogic Server. When running on more than one node, the application servers should be configured as a cluster.

This guide assumes experience with setting up Oracle WebLogic Server. For details regarding the installation process please consult the product documentation.

The Certification Guide specifies the required version of the Oracle WebLogic Server software that must be installed. It also describes how the software can be obtained and how the documentation can be accessed.

This chapter outlines the installation of the Oracle WebLogic Server software. Subsequently, the setup of a domain is explained for the following situations:

- A simple, non-clustered environment that is suitable for development and testing purposes. The description of this configuration also demonstrates how Oracle ADF runtime libraries are added to a WebLogic Server domain.
- An advanced, clustered setup that is typically used in production deployments and that is executed on multiple nodes.

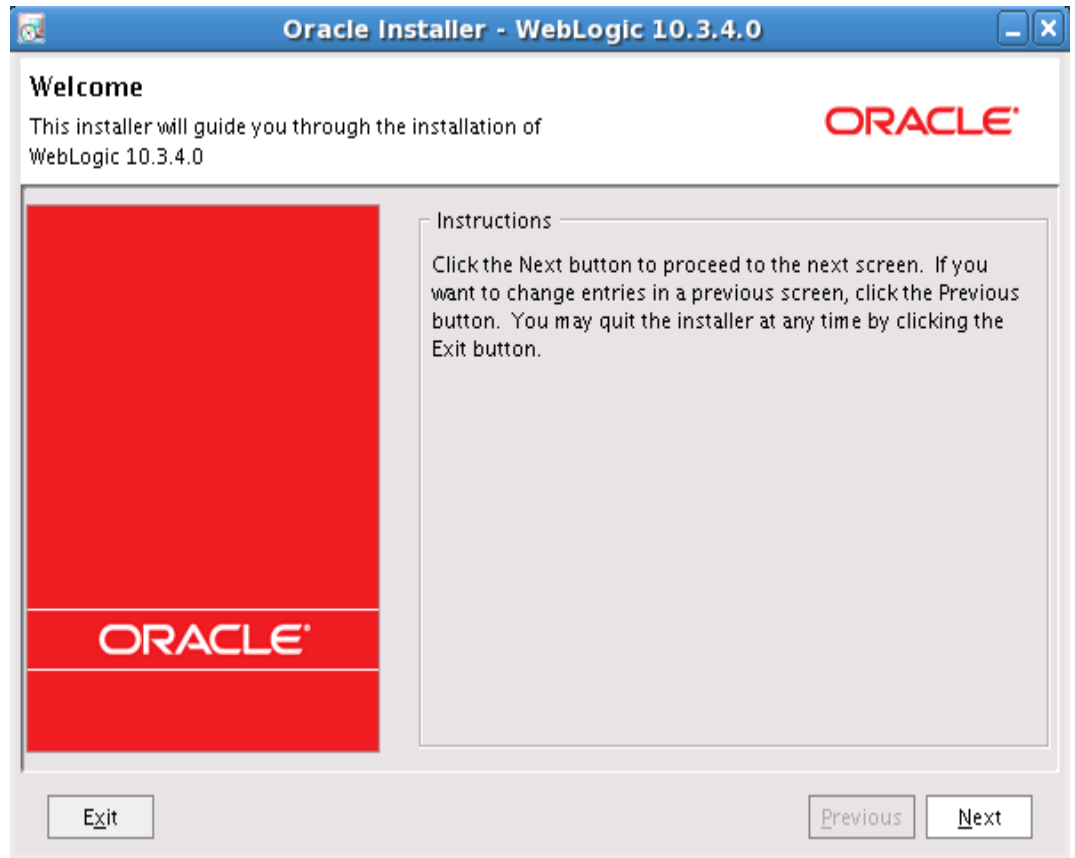
INSTALLING ORACLE WEBLOGIC SERVER

The following steps describe how to install Oracle WebLogic Server. The Certification Guide lists the software that needs to be used as well as the download locations.

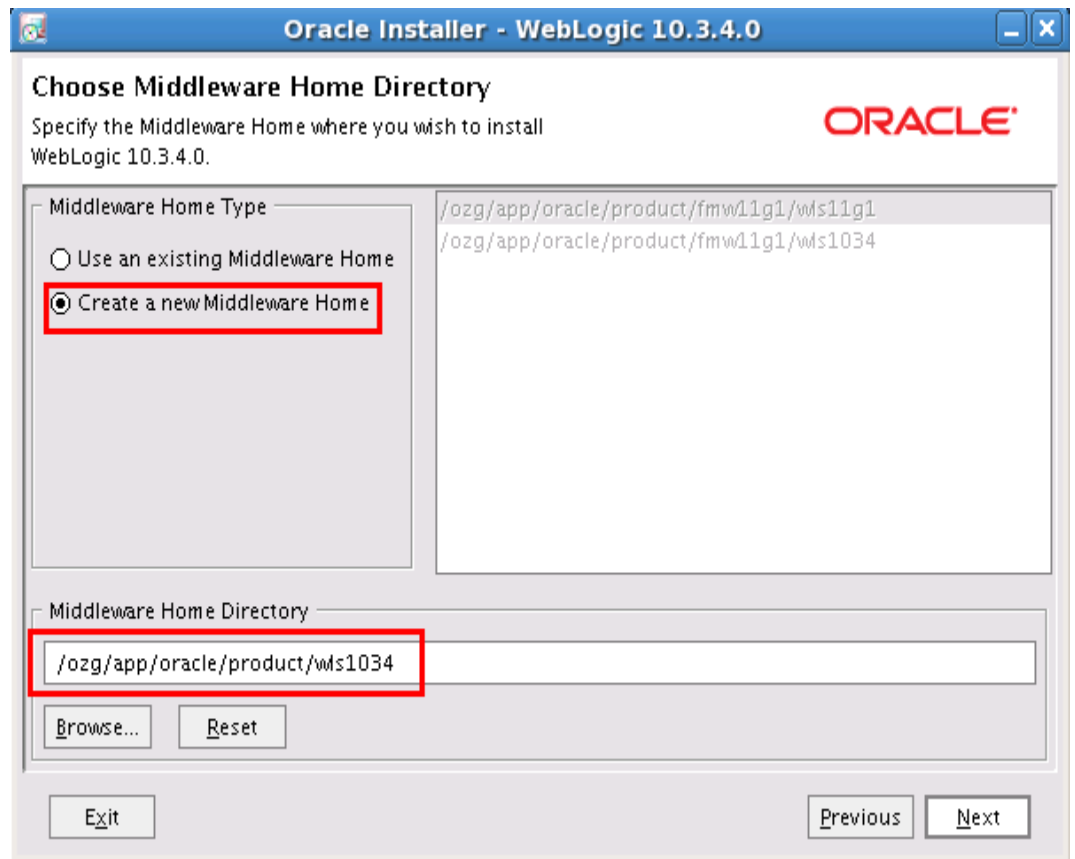
Step 1: Navigate to that folder and run the installer by entering the following command in command line: **java -jar wls1034_generic.jar**

Note JAVA_HOME should be set before running the installer

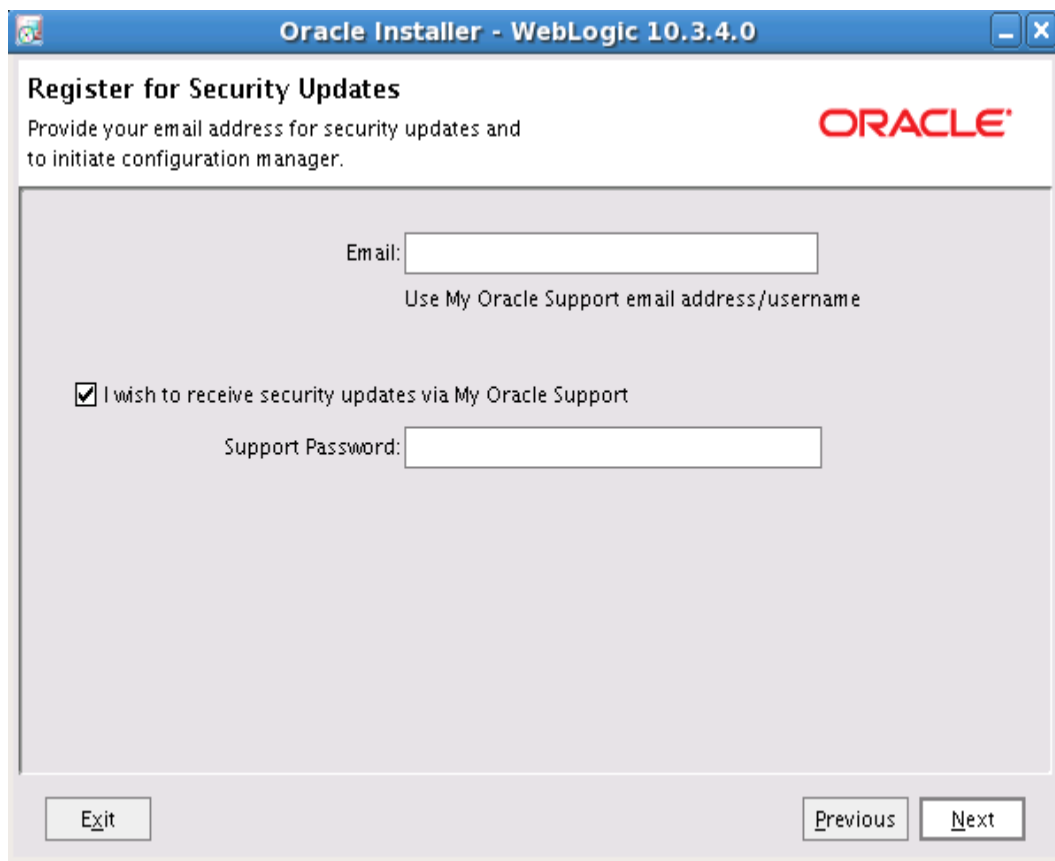
Step 2: In the **Welcome** screen click on **Next** button



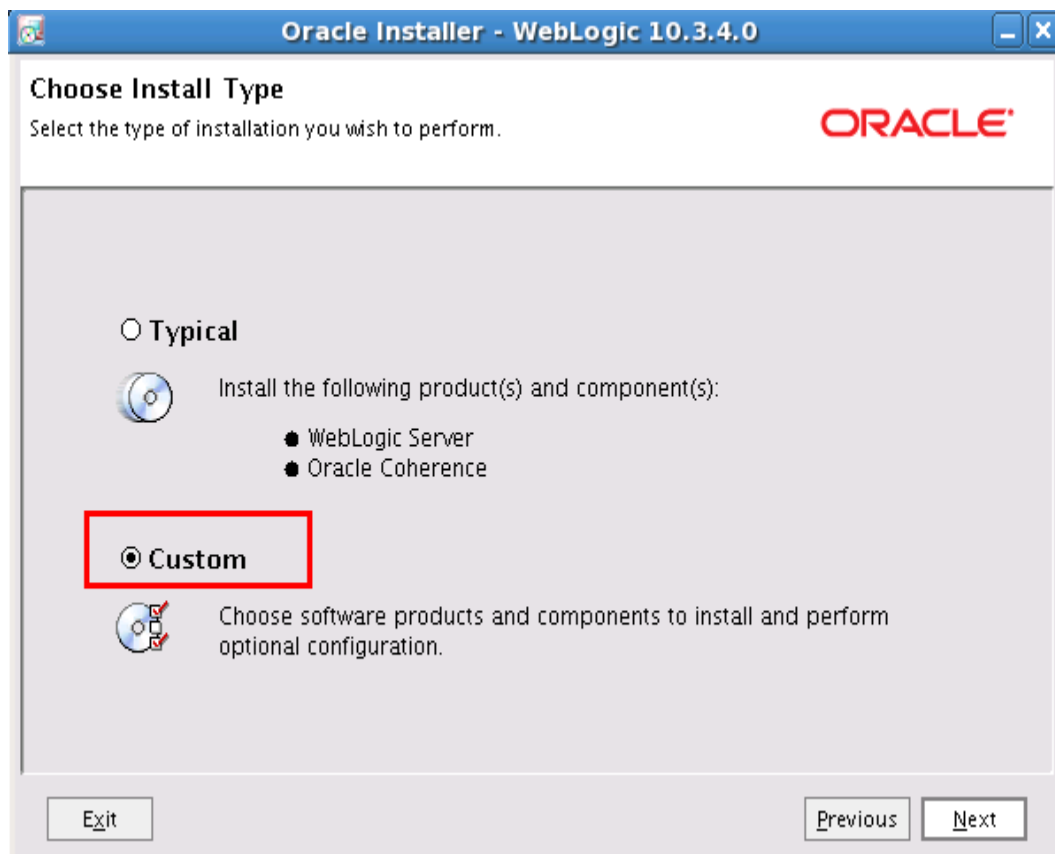
Step 3: In the **Choose Middleware Home Directory** page, select the option **Create a new Middleware Home** and enter the path in **Middleware Home Directory**. Click on **Next** button



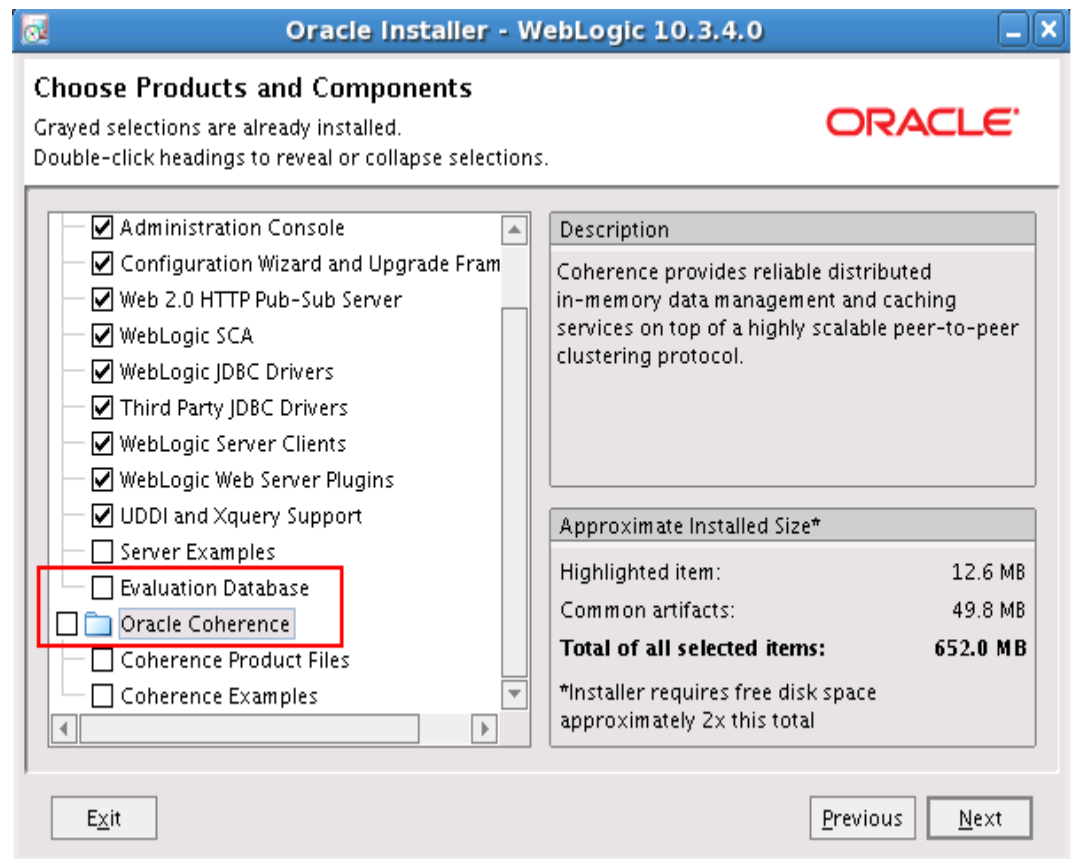
Step 4: In the **Register for Security Updates** page, enter your My Oracle Support Email address and Support Password (optionally, this can be skipped). Click on **Next** button



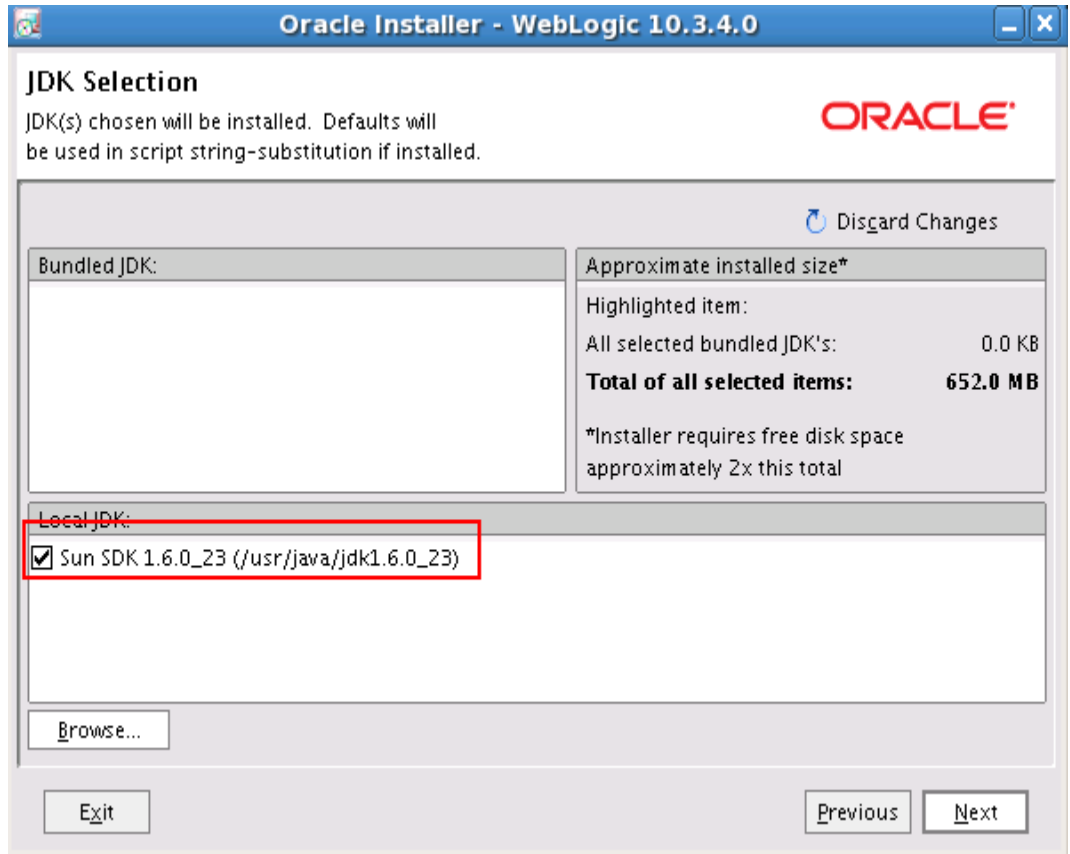
Step 5: In the **Choose Install Type** page, select the option **Custom**. Click on **Next** button



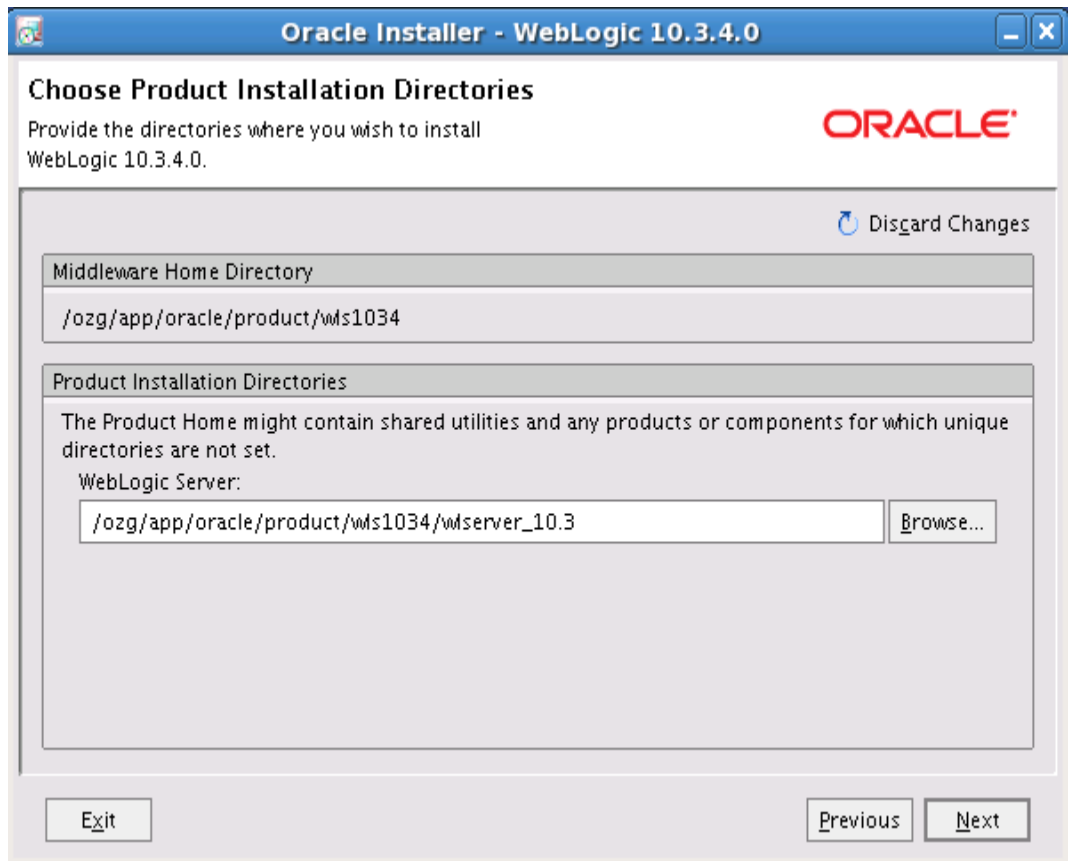
Step 6: In the **Choose Products and Components** page, **deselect** the options **Evaluation Database** and **Oracle Coherence**. Click on **Next** button



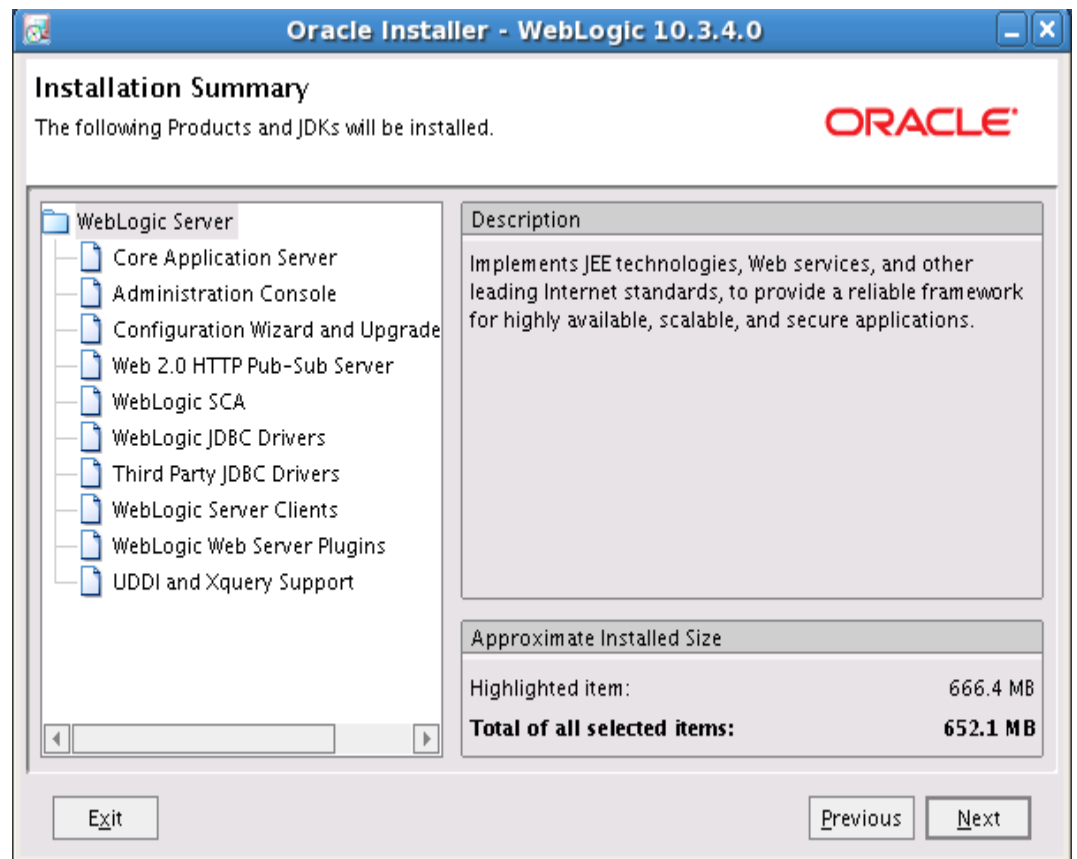
Step 7: The installer for 64-bit machines does not have bundled JDK. So, In **JDK Selection** page, click on **Browse** button to navigate to your JDK installation directory. Click on **Next** button



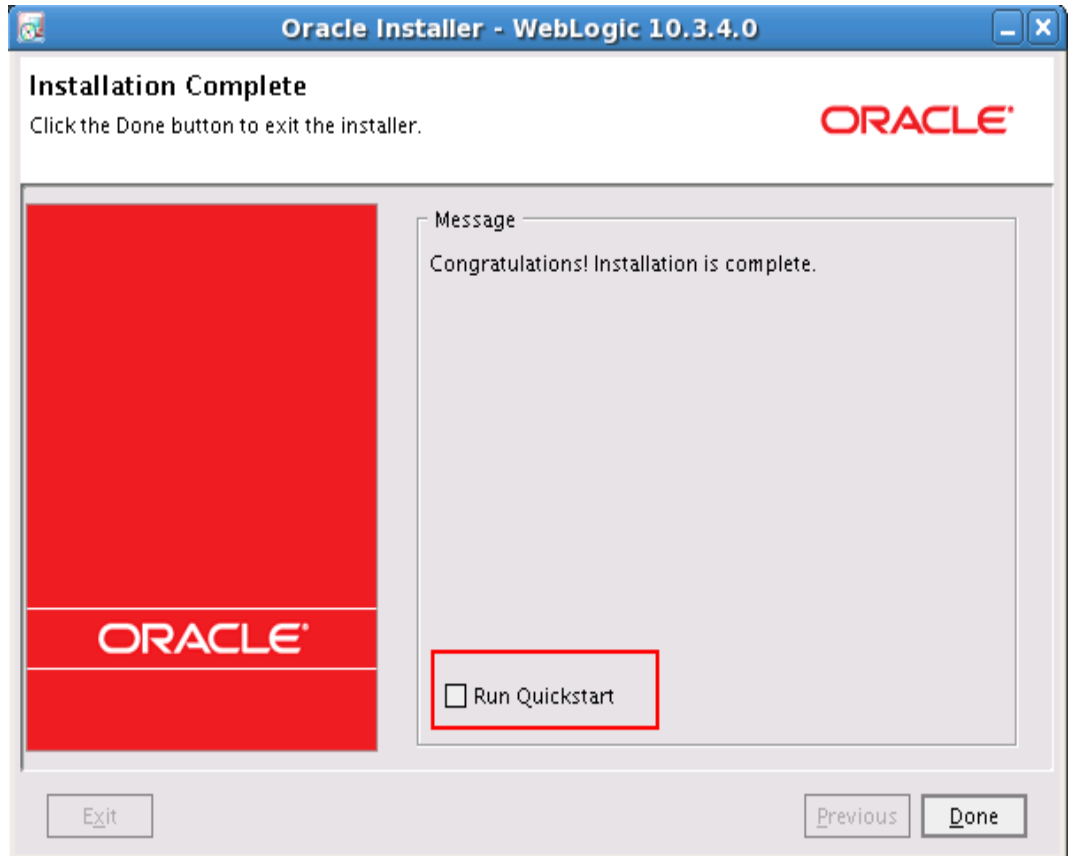
Step 8: In the **Choose Product Installation Directories** page accept the default setting and click on **Next** button



Step 9: In **Installation Summary** page, click on **Next** button



Step 10: When the installation is complete, un-check the check box **Run Quickstart** and click on **Done** button. Oracle WebLogic Server 10.3.4 is now installed.

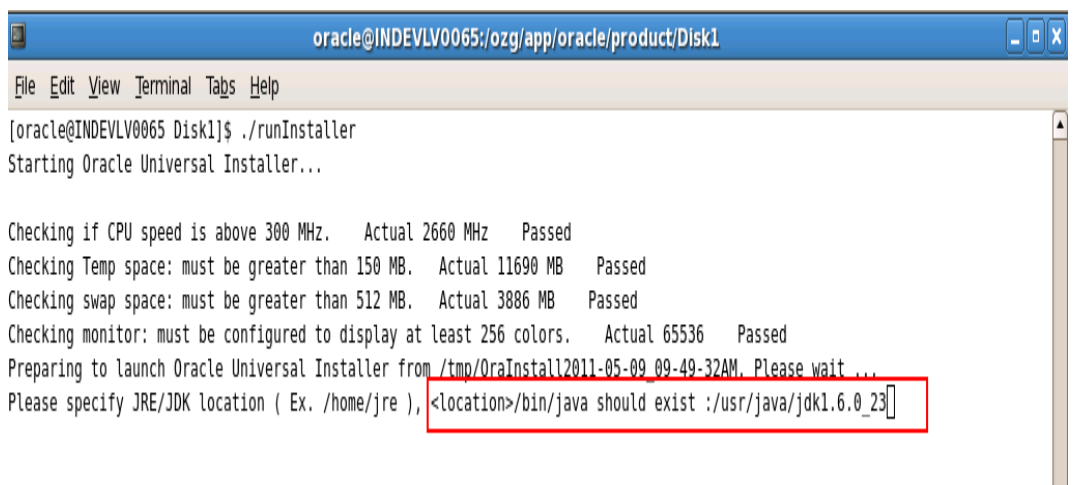


INSTALLING ORACLE APPLICATION DEVELOPMENT RUNTIME

The following steps describe how to install Oracle Application Development Runtime. The Certification Guide lists the software that needs to be used as well as the download locations.

Step 1: Unzip the downloaded archive and navigate to the **Disk1** folder; run the installer by entering the following command in command line: **./runInstaller**

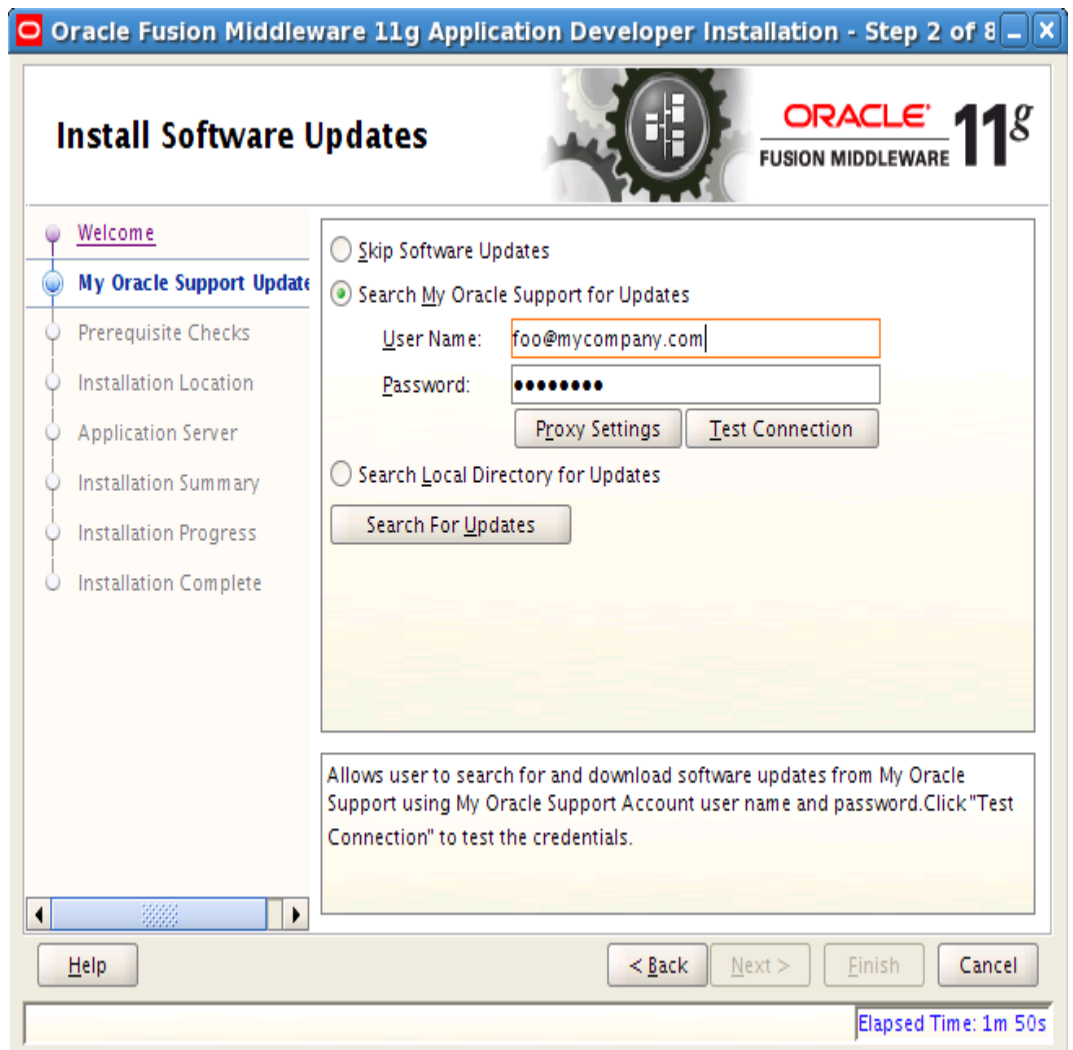
Note The installer will ask you to enter JDK installation directory



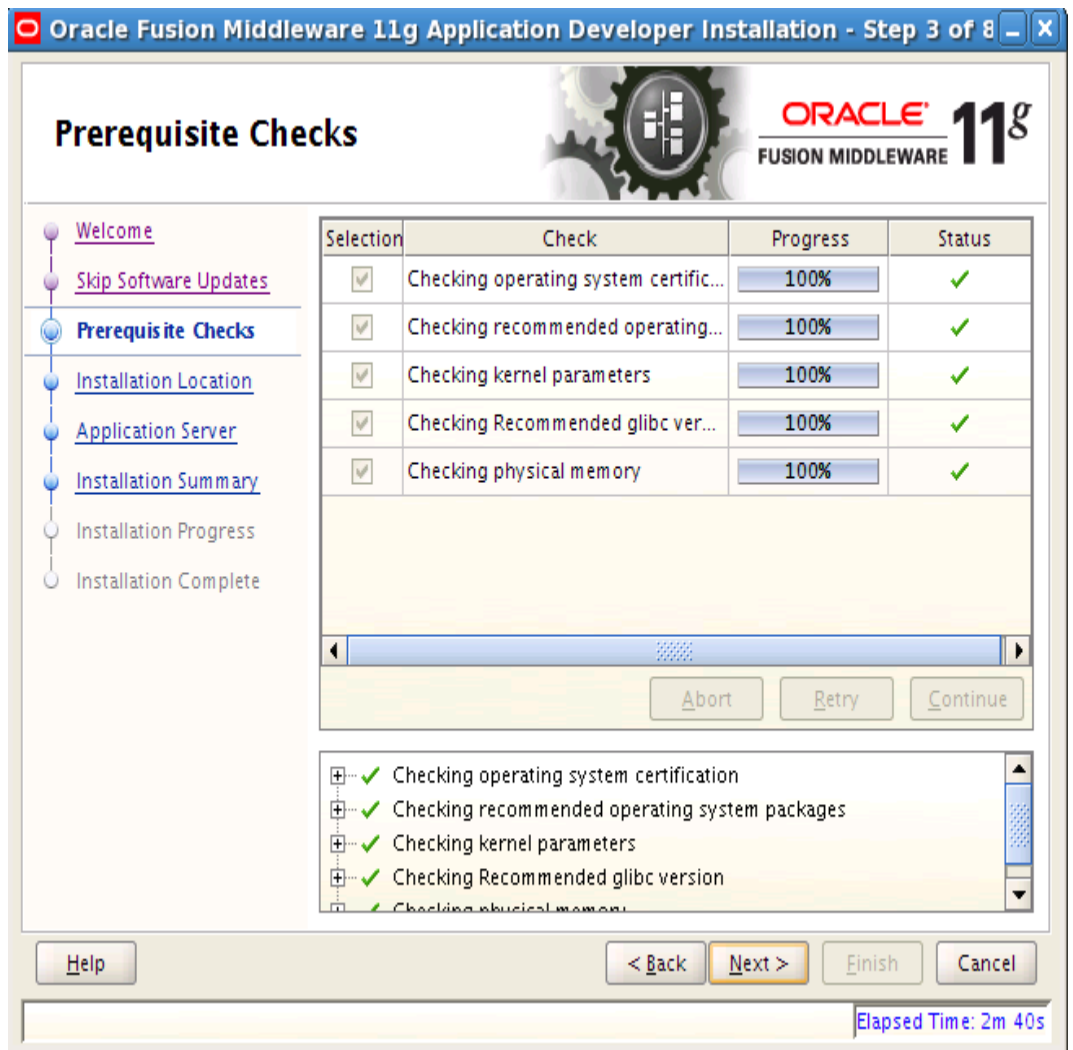
Step 2: In **Welcome** page, click on **Next** button



Step 3: In **Install Software Updates** page, enter your My Oracle Support User Name and Password (optionally, this can be skipped). Click on **Next** button



Step 4: Once **Prerequisite Checks** is completed, click on **Next** button



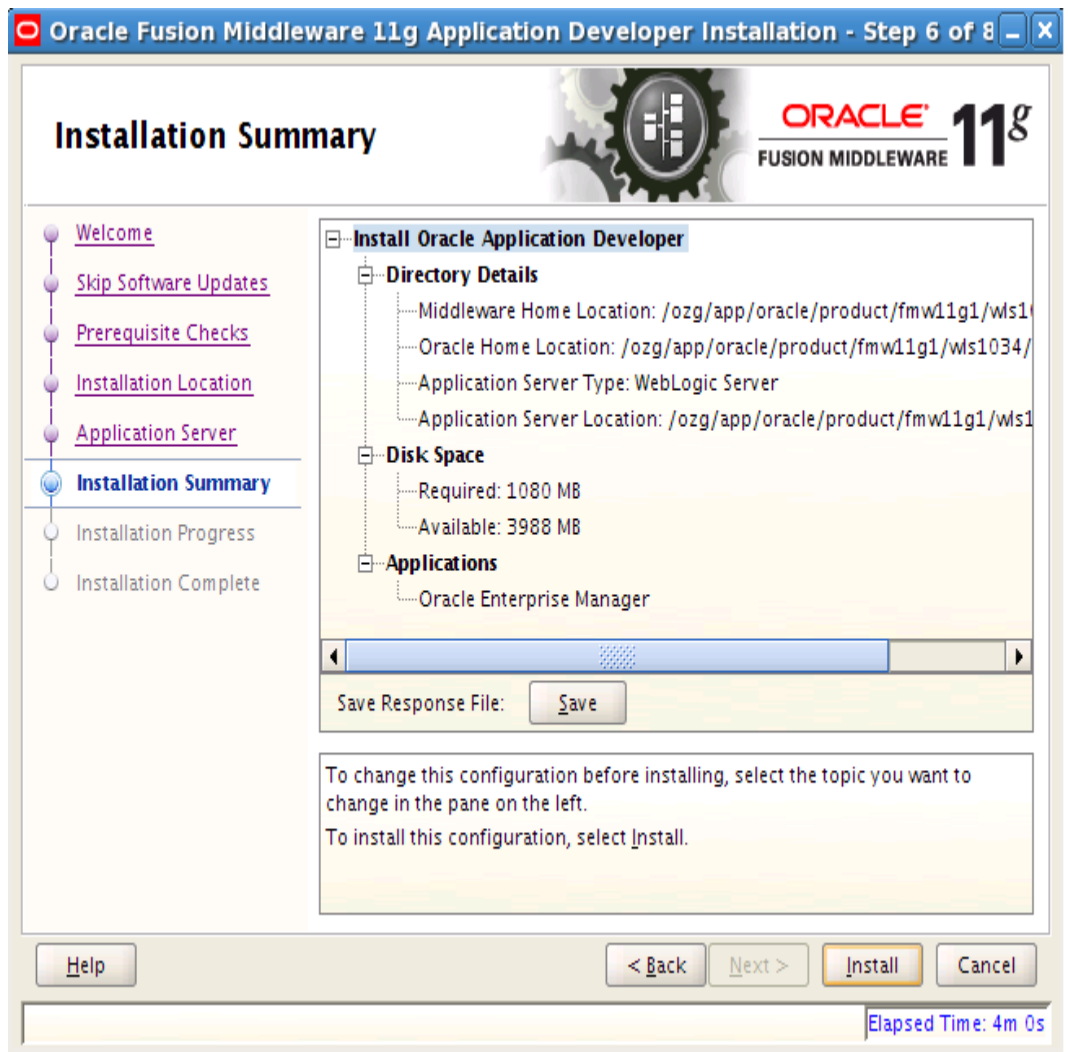
Step 5: In **Specify Installation Location** page, change the value of **Oracle Middleware Home** to suit your WLS installation directory and click on **Next** button



Step 6: In **Application Server** page, accept the default values and click on **Next** button



Step 7: In **Installation Summary** page, click on **Install** button



Step 8: Once the installation is complete, click on **Finish** button in **Installation Complete** page. Oracle Application Development Runtime 11.1.1.4.0 is now installed.



CONFIGURING ORACLE FUSION MIDDLEWARE FOR RUNNING ADF APPLICATIONS

After installing Oracle WebLogic Server and Oracle Application Development Runtime, perform the following steps:

- Create a domain in which the OHI Claims application will be configured and installed.
- Detailed instructions are available in the documentation library (http://download.oracle.com/docs/cd/E12839_01/web.1111/b31974/deployment_topics.htm#ADFFD1831).

Alternatively, follow these steps to create a domain.

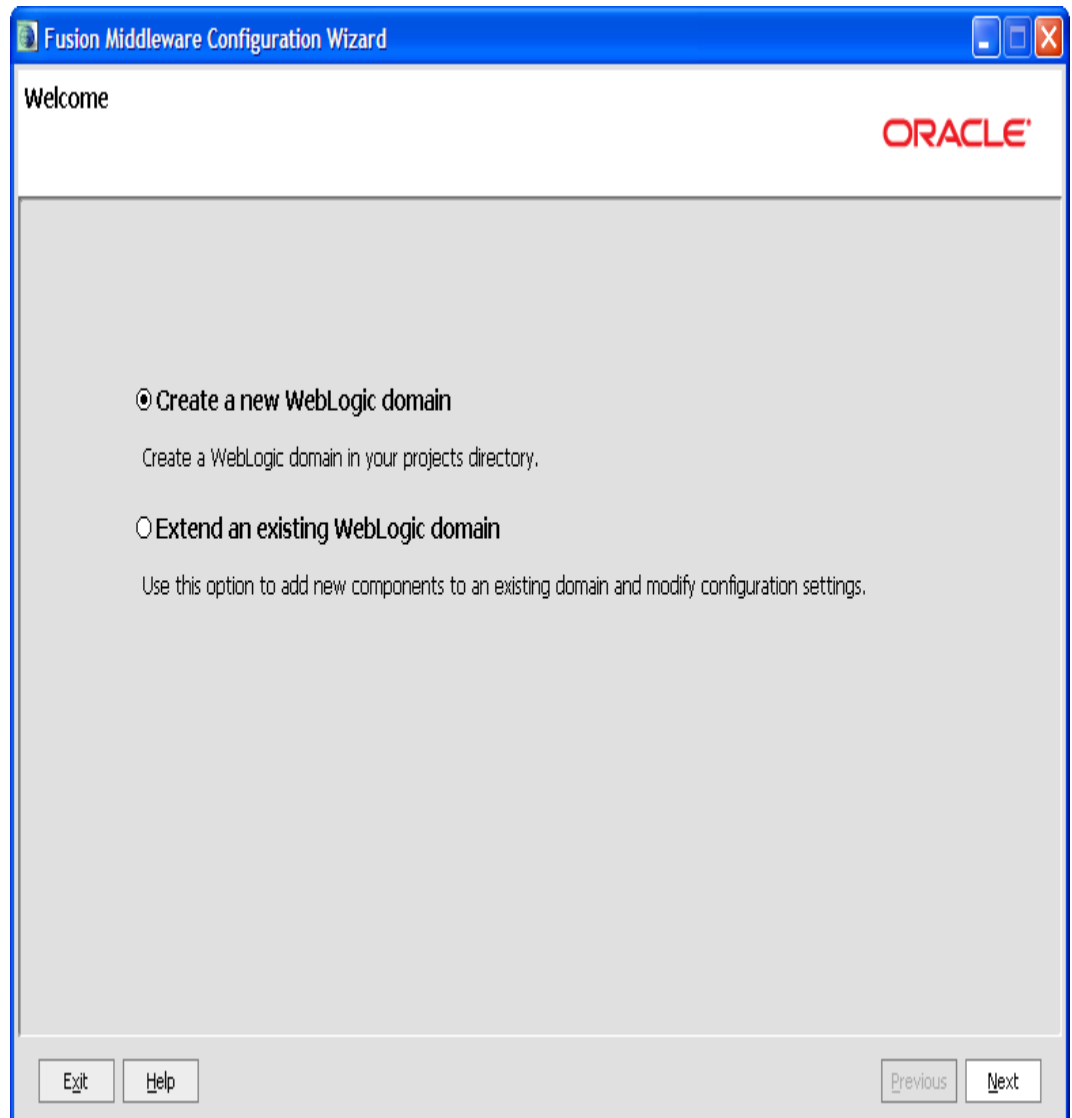
Note The following domain setup is suitable for development and testing purposes but should not be used in production situations.

Step 1: Go to `<MIDDLEWARE_HOME_DIRECTORY>/wlsserver_10.3/common/bin` in command prompt. Here `MIDDLEWARE_HOME_DIRECTORY` is the path where you installed WLS 10.3.4.

Step 2: Issue the following command: **./config.sh**

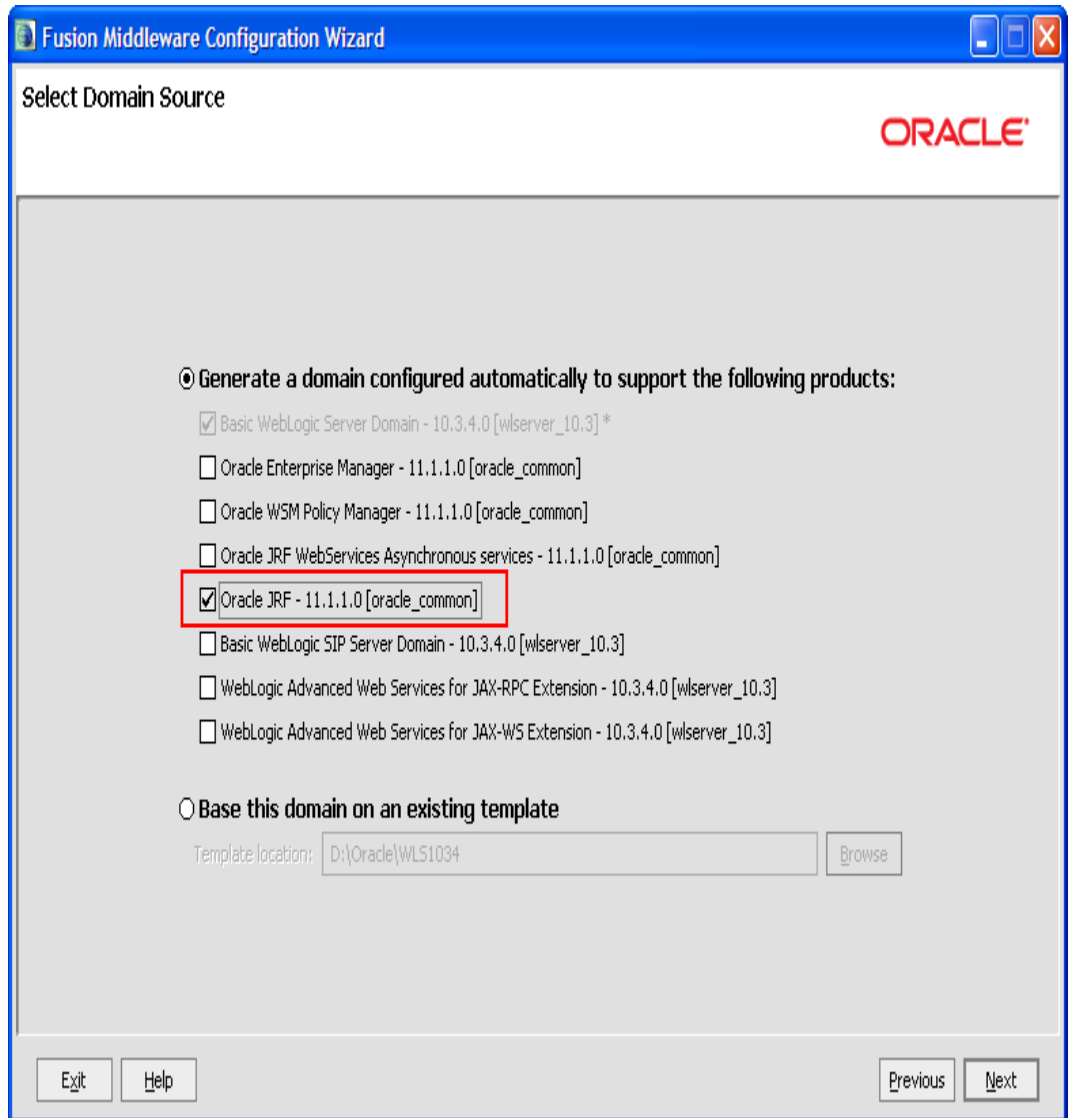
Step 3: **Fusion Middleware Configuration Wizard - Welcome** screen appears.

Step 4: In **Welcome** page, leave the default selection **Create a new WebLogic domain** and click on **Next** button



Step 5: In the **Select Domain Source** page, select the check box **Oracle JRF - 11.1.1.0 [oracle_common]** and click on **Next** button

2. [dsy1742-DSY.html#dsy4990-DSY](#) (Configure OHI Claims properties file)
3. <http://coherence.oracle.com/display/COH34UG/well-known-addresses>
4. <http://coherence.oracle.com/display/COH34UG/Production+Checklist#ProductionChecklist-CoherenceEditionsandModes>
6. <http://download.oracle.com/javase/1.4.2/docs/guide/awt/AWTChanges.html#headless>
7. http://download.oracle.com/docs/cd/E12839_01/web.1111/e13709/load_balancing.htm#CHDGFIBD



Step 6: In the **Specify Domain Name and Location** page, edit the values for **Domain name** and **Domain location** to suit your requirements or leave the default values and click on **Next** button.

Note For consistency, Oracle recommends the value "ohi_domain" as domain name.

2. [dsy1742-DSY.html#dsy4990-DSY](#) (Configure OHI Claims properties file)
3. <http://coherence.oracle.com/display/COH34UG/well-known-addresses>
4. <http://coherence.oracle.com/display/COH34UG/Production+Checklist#ProductionChecklist-CoherenceEditionsandModes>
6. <http://download.oracle.com/javase/1.4.2/docs/guide/awt/AWTChanges.html#headless>
7. http://download.oracle.com/docs/cd/E12839_01/web.1111/e13709/load_balancing.htm#CHDGFIBD



Step 7: In the **Configure Administrator User Name and Password** page, enter the values for **User password** and **Confirm user password** and click on **Next** button.

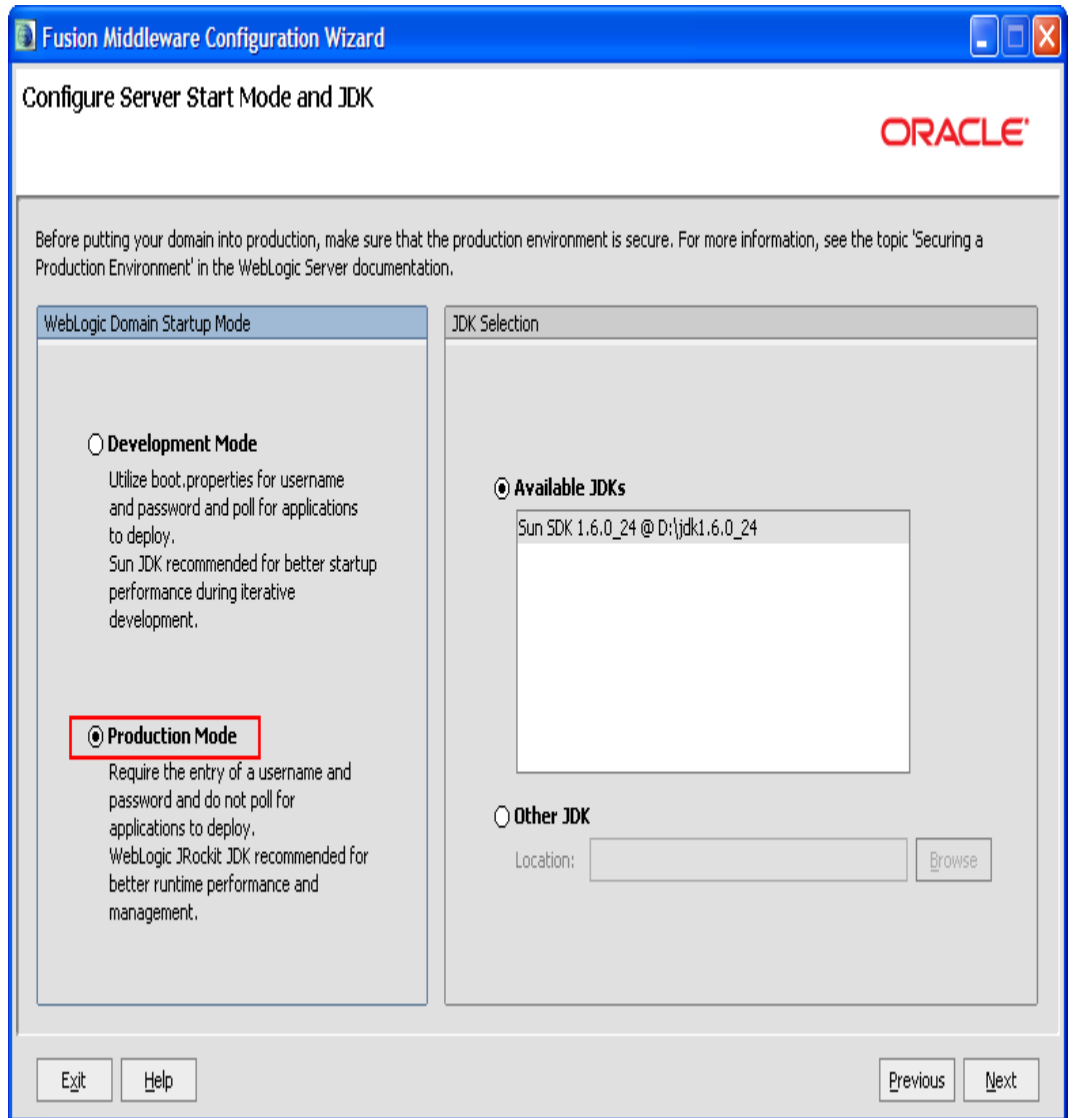
2. [dsy1742-DSY.html#dsy4990-DSY](#) (Configure OHI Claims properties file)
3. <http://coherence.oracle.com/display/COH34UG/well-known-addresses>
4. <http://coherence.oracle.com/display/COH34UG/Production+Checklist#ProductionChecklist-CoherenceEditionsandModes>
6. <http://download.oracle.com/javase/1.4.2/docs/guide/awt/AWTChanges.html#headless>
7. http://download.oracle.com/docs/cd/E12839_01/web.1111/e13709/load_balancing.htm#CHDGFIBD

The screenshot shows a window titled "Fusion Middleware Configuration Wizard" with the subtitle "Configure Administrator User Name and Password". The Oracle logo is in the top right. A "Disgard Changes" link is in the top left. The main area contains four input fields: "*Name:" with "weblogic", "*User password:" with "*****", "*Confirm user password:" with "*****", and "Description:" with "This user is the default administrator.". At the bottom are buttons for "Exit", "Help", "Previous", and "Next".

Note The password must be at least 8 alphanumeric characters with at least one number or special character.

Step 8: In the **Configure Server Start Mode and JDK** page, change the value for **WebLogic Domain Startup Mode** to **Production Mode** and click on **Next** button.

2. [dsy1742-DSY.html#dsy4990-DSY](#) (Configure OHI Claims properties file)
3. <http://coherence.oracle.com/display/COH34UG/well-known-addresses>
4. <http://coherence.oracle.com/display/COH34UG/Production+Checklist#ProductionChecklist-CoherenceEditionsandModes>
6. <http://download.oracle.com/javase/1.4.2/docs/guide/awt/AWTChanges.html#headless>
7. http://download.oracle.com/docs/cd/E12839_01/web.1111/e13709/load_balancing.htm#CHDGFIBD



Step 9: In the **Select Optional Configuration** page, select the options that you want to configure. Else, leave with the default settings and click on **Next** button.

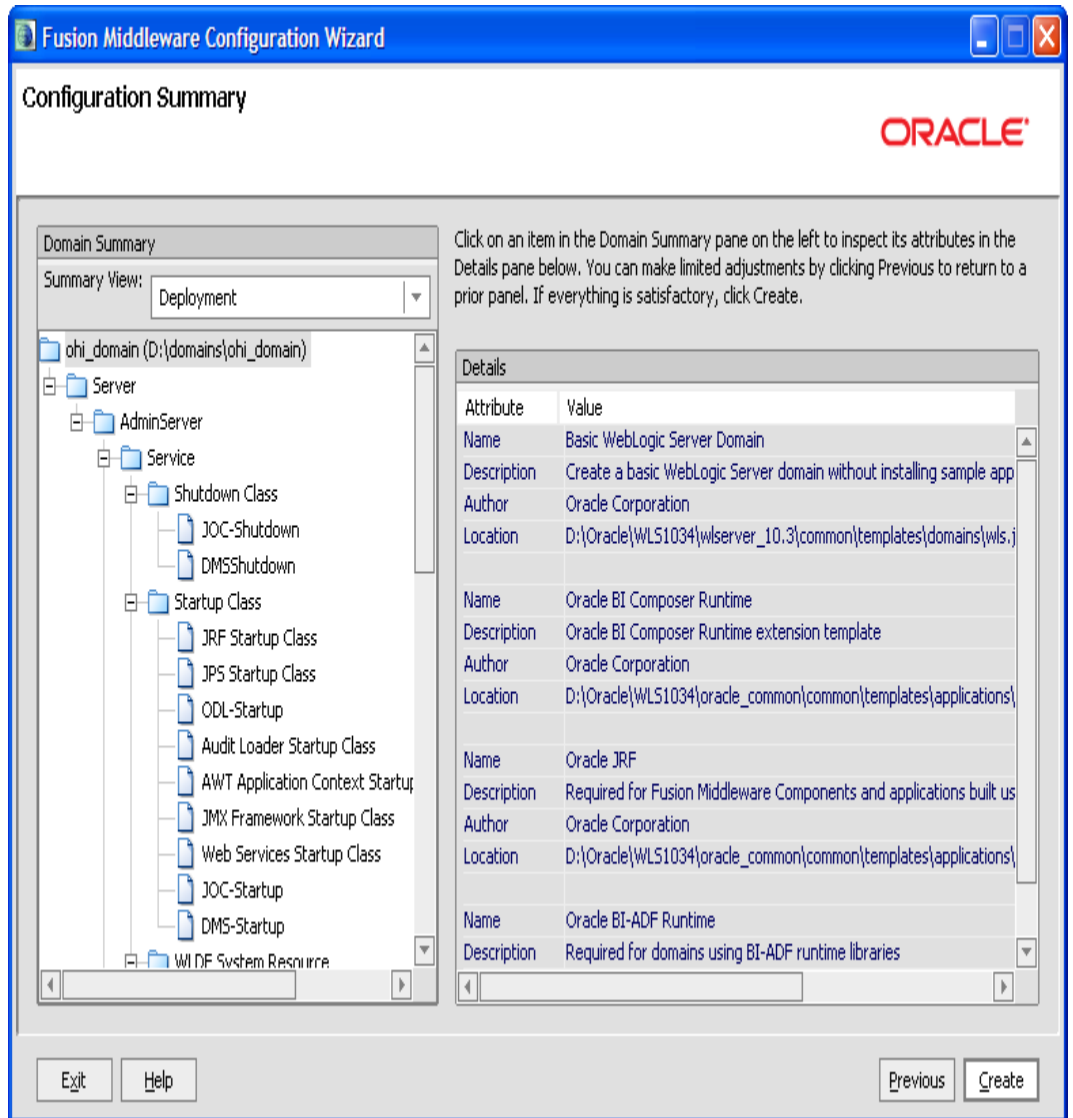
2. [dsy1742-DSY.html#dsy4990-DSY](#) (Configure OHI Claims properties file)
3. <http://coherence.oracle.com/display/COH34UG/well-known-addresses>
4. <http://coherence.oracle.com/display/COH34UG/Production+Checklist#ProductionChecklist-CoherenceEditionsandModes>
6. <http://download.oracle.com/javase/1.4.2/docs/guide/awt/AWTChanges.html#headless>
7. http://download.oracle.com/docs/cd/E12839_01/web.1111/e13709/load_balancing.htm#CHDGFIBD



Note The default AdminServer listening port is 7001.

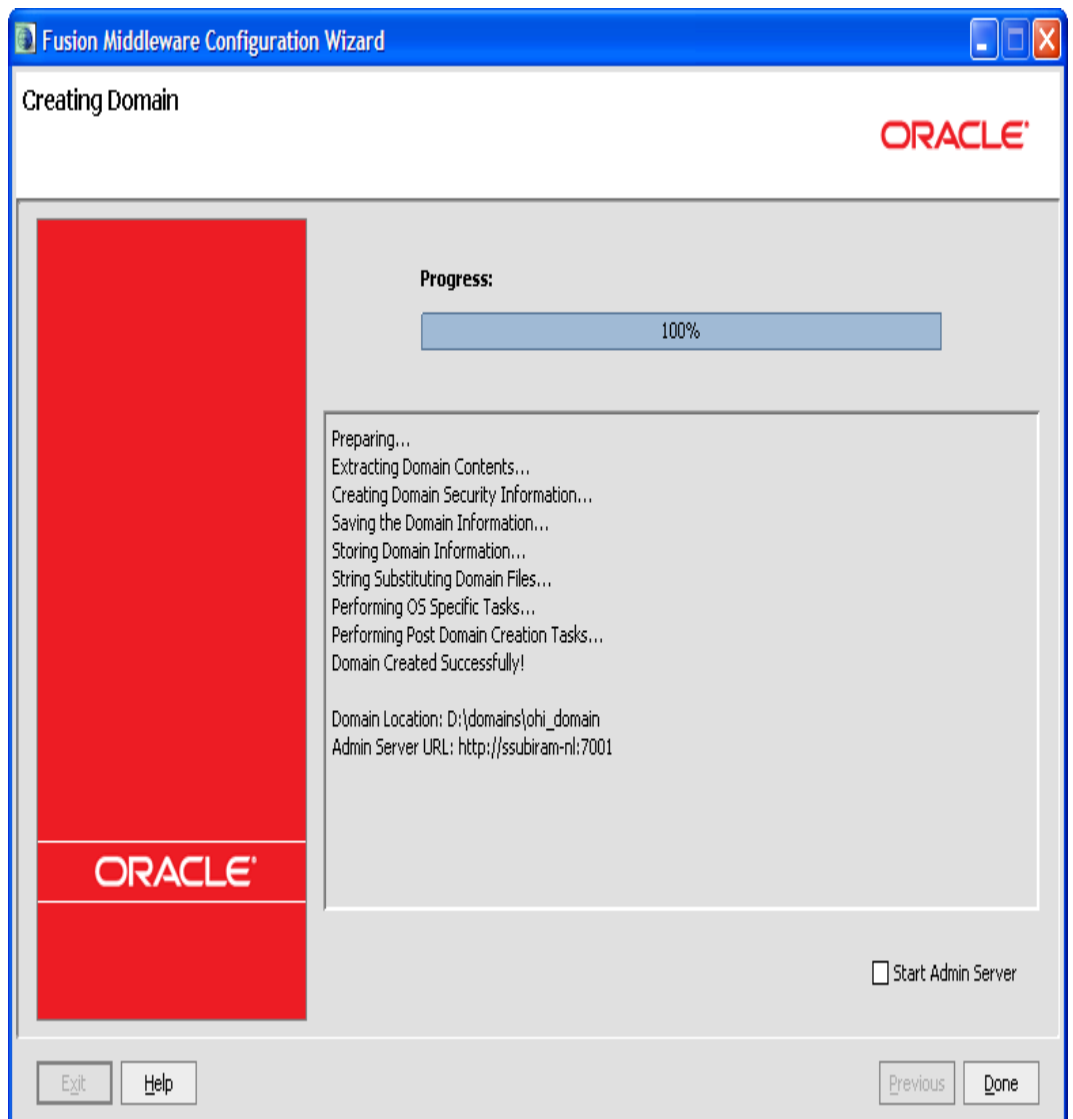
Step 10: In the **Configuration Summary** page, click on **Create** button.

2. [dsy1742-DSY.html#dsy4990-DSY](#) (Configure OHI Claims properties file)
3. <http://coherence.oracle.com/display/COH34UG/well-known-addresses>
4. <http://coherence.oracle.com/display/COH34UG/Production+Checklist#ProductionChecklist-CoherenceEditionsandModes>
6. <http://download.oracle.com/javase/1.4.2/docs/guide/awt/AWTChanges.html#headless>
7. http://download.oracle.com/docs/cd/E12839_01/web.1111/e13709/load_balancing.htm#CHDGFIBD



Step 11: In the **Creating Domain** page, click on **Done** button once the domain is created.

2. [dsy1742-DSY.html#dsy4990-DSY](#) (Configure OHI Claims properties file)
3. <http://coherence.oracle.com/display/COH34UG/well-known-addresses>
4. <http://coherence.oracle.com/display/COH34UG/Production+Checklist#ProductionChecklist-CoherenceEditionsandModes>
6. <http://download.oracle.com/javase/1.4.2/docs/guide/awt/AWTChanges.html#headless>
7. http://download.oracle.com/docs/cd/E12839_01/web.1111/e13709/load_balancing.htm#CHDGFIBD



DOMAIN CONFIGURATION FOR OHI CLAIMS

This chapter contains directions for the following topics:

- Redirecting console log output
- Setting up OHI Claims properties files
- Coherence settings
- Setting OHI Claims Domain environment variables

Redirect JVM Output to a Log File

By default, the JVM output for a WebLogic server is written to the console. It is recommended to redirect the console output to file.

Note that in development mode, the default size of a logfile before it is rotated is only 500Kb. Hence, it is also recommended to change the size of the log files before rollover to 10240 Kb and to specify the number of log files that will be retained. These configuration settings can be changed through the WebLogic Server Console.

Setting up OHI Claims Properties Files

Create a directory that will hold OHI Claims properties and configuration files. This directory will be referenced as <PROPERTIES_ROOT> throughout this document.

Copy the following files that were delivered as part of the specific release from the <OHI_ROOT>/properties directory to the <PROPERTIES_ROOT>:

- logback.xml
- ohi-claims.properties

A description of the properties files is available [here](#)².

Also copy file <OHI_ROOT>/util/security/ohi-claims-security.config to the <PROPERTIES_ROOT>.

Coherence settings

OHI Claims uses Oracle Coherence. The IT infrastructure on which the system is installed determines the configuration for Oracle Coherence. This paragraph describes the following configuration options:

- Restrict a Coherence cluster to one machine
- Control multiple Coherence clusters that are spread across multiple machines
- Control multiple Coherence clusters that are executed on one machine
- Specific settings for running Coherence in a Production environment

Restrict a Coherence cluster to one machine

The <PROPERTIES_ROOT> directory contains a Coherence configuration file (single-server-tangosol-coherence-override.xml) that ensures that a Coherence cluster is restricted to a single machine.

Note These settings will constrain Coherence to run on a single machine. It will not prevent Coherence from clustering with other JVMs on the same machine that also run Coherence. Therefore, it is not suitable for setting up multiple Coherence clusters on a single machine.

Copy the following properties file that was delivered as part of the specific release from the <OHI_ROOT>/properties directory to the <PROPERTIES_ROOT>:

- single-server-tangosol-coherence-override.xml

2. dsy1742-DSY.html#dsy4990-DSY (Configure OHI Claims properties file)

3. <http://coherence.oracle.com/display/COH34UG/well-known-addresses>

4. <http://coherence.oracle.com/display/COH34UG/Production+Checklist#ProductionChecklist-CoherenceEditionsandModes>

6. <http://download.oracle.com/javase/1.4.2/docs/guide/awt/AWTChanges.html#headless>

7. http://download.oracle.com/docs/cd/E12839_01/web.1111/e13709/load_balancing.htm#CHDGFIBD

Run multiple Coherence clusters of multiple JVMs on the same machine of same set of machines

In order to control which JVMs can join in a particular Coherence cluster, the Coherence Well Known Addresses (WKA) feature may be used.

This can be used to:

- Control multiple Coherence clusters that are spread across multiple machines
- Control multiple Coherence clusters that are executed on one machine

A preconfigured `tangosol-coherence-override.xml` file for these situations cannot be provided as required host names or IP addresses must be used. The following sample files show the basic structure.

Example: Building a cluster across multiple machines

The following sample override file controls a Coherence cluster that runs on JVMs on several machines (host1, host2, ..., hostN):

```
<coherence>
  <cluster-config>
    <unicast-listener>
      <well-known-addresses>
        <socket-address id="1">
          <address system-property="tangosol.coherence.wka1">host1<
/          address>
          <port system-property="tangosol.coherence.wka1.port">8088</port>
        </socket-address>
        <socket-address id="2">
          <address system-property="tangosol.coherence.wka2">host2<
/          address>
          <port system-property="tangosol.coherence.wka2.port">8088</port>
        </socket-address>
        ...
        <socket-address id="N">
          <address system-property="tangosol.coherence.wkaN">hostN<
/          address>
          <port system-property="tangosol.coherence.wkaN.port">8088</port>
        </socket-address>
      </well-known-addresses>
    </unicast-listener>
  </cluster-config>
</coherence>
```

Start the JVM on host1 with the following command-line parameters:

Note These options should be specified on one line, it was formatted differently in this guide for readability.

```
-Dtangosol.coherence.wka1=host1
-Dtangosol.coherence.wka1.port=8088
-Dtangosol.coherence.localport=8088
-Dtangosol.coherence.override=tangosol-coherence-override.xml
```

Start the JVM on host2 with the following command-line parameters:

```
-Dtangosol.coherence.wka2=host2
-Dtangosol.coherence.wka2.port=8088
```

```
-Dtangosol.coherence.localport=8088  
-Dtangosol.coherence.override=tangosol-coherence-override.xml
```

Example: Controlling a cluster of multiple JVMs on one machine

The following sample override file controls a Coherence cluster that runs on multiple JVMs on the same machine (host1):

```
<coherence>  
  <cluster-config>  
    <unicast-listener>  
      <well-known-addresses>  
        <socket-address id="1">  
          <address system-property="tangosol.coherence.wka1">host1<  
/address>  
          <port system-property="tangosol.coherence.wka1.port">8088</port>  
        </socket-address>  
        <socket-address id="2">  
          <address system-property="tangosol.coherence.wka2">host1<  
/address>  
          <port system-property="tangosol.coherence.wka2.port">8089</port>  
        </socket-address>  
        ...  
        <socket-address id="N">  
          <address system-property="tangosol.coherence.wkaN">host1<  
/address>  
          <port system-property="tangosol.coherence.wkaN.port">8090</port>  
        </socket-address>  
      </well-known-addresses>  
    </unicast-listener>  
  </cluster-config>  
</coherence>
```

Start the first JVM on host1 with the following command-line parameters:

Note These options should be specified on one line, it was formatted differently in this guide for readability.

```
-Dtangosol.coherence.wka1=host1  
-Dtangosol.coherence.wka1.port=8088  
-Dtangosol.coherence.localport=8088  
-Dtangosol.coherence.override=tangosol-coherence-override.xml
```

Start the second JVM on host1 with the following command-line parameters:

```
-Dtangosol.coherence.wka2=host1  
-Dtangosol.coherence.wka2.port=8089  
-Dtangosol.coherence.localport=8089  
-Dtangosol.coherence.override=tangosol-coherence-override.xml
```

For more information please check the [Coherence documentation on Well Known Addresses](#)³.

3. <http://coherence.oracle.com/display/COH34UG/well-known-addresses>

4. <http://coherence.oracle.com/display/COH34UG/Production+Checklist#ProductionChecklist-CoherenceEditionsandModes>

6. <http://download.oracle.com/javase/1.4.2/docs/guide/awt/AWTChanges.html#headless>

7. http://download.oracle.com/docs/cd/E12839_01/web.1111/e13709/load_balancing.htm#CHDGFIBD

Specific settings for running Coherence in a Production environment

By default, Oracle Coherence runs in Development mode. The [production checklist in the Coherence documentation](#)⁴ states that *it is recommended to use the development mode for all pre-production activities, such as development and testing. This is an important safety feature, because Coherence automatically prevents these nodes from joining a production cluster. The production mode must be explicitly specified when using Coherence in a production environment.*

In the Production environment (and only in the Production environment), the system property `tangosol.coherence.mode` should be set to value `prod` in the script that is used to start Coherence nodes.

Next to that, Oracle advises to use system property `tangosol.coherence.cluster` to name the cluster. To join the cluster, all members must specify the same cluster name. Suggested naming convention:

OHI-`<systemproperty.ohi.environment.identifier>`.

```
-Dtangosol.coherence.mode=prod
-Dtangosol.coherence.cluster=<cluster_name>
```

Set Domain Environment Variables for OHI Claims

Go to `<MIDDLEWARE_HOME_DIRECTORY>/user_projects/domains/<DOMAIN_NAME>/bin` where `<DOMAIN_NAME>` is the name of the domain that was given in Step 6 of the previous section. Edit the file `setDomainEnv.sh` in that directory and add the following lines at the beginning as shown in this sample:

```
USER_MEM_ARGS="-Xms8192m -Xmx8192m -XX:PermSize=1024m
-XX:MaxPermSize=1024m -XX:NewSize=2048m -XX:MaxNewSize=2048m "
USER_MEM_ARGS="$USER_MEM_ARGS -XX:+UseConcMarkSweepGC "
USER_MEM_ARGS="$USER_MEM_ARGS -XX:+UseParNewGC "
USER_MEM_ARGS="$USER_MEM_ARGS -XX:+CMSIncrementalMode "
USER_MEM_ARGS="$USER_MEM_ARGS -XX:+ExplicitGCInvokesConcurrent "
USER_MEM_ARGS="$USER_MEM_ARGS -XX:+CMSClassUnloadingEnabled "
USER_MEM_ARGS="$USER_MEM_ARGS -XX:+UseCMSCompactAtFullCollection "
USER_MEM_ARGS="$USER_MEM_ARGS -XX:+CMSPermGenSweepingEnabled "
export USER_MEM_ARGS

if [ "${ADMIN_URL}" = "" ] ; then

EXTRA_JAVA_PROPERTIES="-Dlogback.configurationFile=/home/aia/software/oracle/wls1032/properties/logback.xml"
export EXTRA_JAVA_PROPERTIES
else
EXTRA_JAVA_PROPERTIES=" "
EXTRA_JAVA_PROPERTIES="$EXTRA_JAVA_PROPERTIES
-Dlogback.configurationFile=/home/aia/software/oracle/wls1032/properties/logback.xml "
EXTRA_JAVA_PROPERTIES="$EXTRA_JAVA_PROPERTIES
-Dohi.properties.file=/home/aia/software/oracle/wls1032/properties/ohi-claims.properties "
EXTRA_JAVA_PROPERTIES="$EXTRA_JAVA_PROPERTIES
-Dtangosol.coherence.override=file:/home/aia/software/oracle/wls1032/properties/single-server-tangosol-coherence-override.xml "
EXTRA_JAVA_PROPERTIES="$EXTRA_JAVA_PROPERTIES
```

4. <http://coherence.oracle.com/display/COH34UG/Production+Checklist#ProductionChecklist-CoherenceEditionsandModes>
6. <http://download.oracle.com/javase/1.4.2/docs/guide/awt/AWTChanges.html#headless>
7. http://download.oracle.com/docs/cd/E12839_01/web.1111/e13709/load_balancing.htm#CHDGFIBD

```
-Dtangosol.coherence.cluster=<cluster_name> "
EXTRA_JAVA_PROPERTIES="$EXTRA_JAVA_PROPERTIES
-Dtangosol.coherence.member=<member_name> "
EXTRA_JAVA_PROPERTIES="$EXTRA_JAVA_PROPERTIES -Dohi.mds.country=US "
EXTRA_JAVA_PROPERTIES="$EXTRA_JAVA_PROPERTIES
-Dcom.sun.org.apache.xml.internal.dtm.DTManager=com.sun.org.apache.xml.in
ternal.dtm.ref.DTManagerDefault "
EXTRA_JAVA_PROPERTIES="$EXTRA_JAVA_PROPERTIES
-Djavax.xml.datatype.DatatypeFactory=com.sun.org.apache.xerces.internal.ja
xp.datatype.DatatypeFactoryImpl "
export EXTRA_JAVA_PROPERTIES
fi
```

USER_MEM_ARGS Explanation:

- `-Xms<SOME_SIZE>m -Xmx<SOME_SIZE>m` -- this represents the heap size allocated for the JVM. `SOME_SIZE` should always be the same number.
 - Determining what these sizes should be in production environments requires a full JVM sizing exercise. More on JVM sizing for production is available at [OHI-Claims JVM Sizing](#) (page44).
- `-XX:PermSize=<SMALLER_SIZE>m -XX:MaxPermSize=<SMALLER_SIZE>m` -- this sets the size for the permanent generation of the JVM's heap. This should be set to 1024m.
- `-XX:NewSize=2048m -XX:MaxNewSize=2048m` -- sets the size for the JVM's new generation. As a rule of thumb, set it to 30% of the heap size.
- `-XX:+UseConcMarkSweepGC -XX:+UseParNewGC`
`-XX:+ExplicitGCInvokesConcurrent -XX:+CMSIncrementalMode`
`-XX:+CMSClassUnloadingEnabled`
`-XX:+UseCMSCompactAtFullCollection`
`-XX:+CMSPermGenSweepingEnabled` -- these are the garbage collector settings recommended for use with the OHI-Claims application. More information on JVM options and garbage collector settings is at [Java HotSpot VM Options](#)⁵.

EXTRA_JAVA_PROPERTIES Explanation:

- `tangosol.coherence.mode`: use this property for production environments only.
- `tangosol.coherence.cluster`: the same name needs to be specified by all members in order to join a specific cluster.
- `tangosol.coherence.member`: the member-name element contains the name of the member itself. This name makes it possible to easily differentiate among members, such as when multiple members run on the same machine. If a name is not specified, the node will fail to start (`IllegalArgumentException`). Suggested naming convention: `OHI-<systemproperty.ohi.environment.identifier>-<machinename_or_ip-address>-<unique-identifier>`.

5. <http://www.oracle.com/technetwork/java/javase/tech/vmoptions-jsp-140102.html>

6. <http://download.oracle.com/javase/1.4.2/docs/guide/awt/AWTChanges.html#headless>

7. http://download.oracle.com/docs/cd/E12839_01/web.1111/e13709/load_balancing.htm#CHDGFIBD

Limited support for the X window system

For displaying gauges in UI pages on systems that do not have the X windows system or have limited access to it add the following EXTRA_JAVA_PROPERTIES option to the setDomainEnv script:

```
EXTRA_JAVA_PROPERTIES="$EXTRA_JAVA_PROPERTIES -Djava.awt.headless=true"
```

See the [Java documentation](#)⁶ for more information. Typically, on a system that lacks X windows support and that does not have this option specified, gauges will not display correctly and the following exception will be in the logs (formatted for displaying it in this guide):

```
<Aug 31, 2011 12:39:00 PM CEST> <Error> <
oracle.adfinternal.view.faces.config.rich.RegistrationConfigurator>
<BEA-000000> <ADF_FACES-60096:Server Exception during PPR, #1
javax.servlet.ServletException:
  java.lang.InternalError: Can't connect to X11 window server using ':0'
as the value of the DISPLAY variable.
```

SETTING UP A WEBLOGIC CLUSTER FOR RUNNING OHI CLAIMS ON MULTIPLE NODES

A WebLogic Server cluster consists of multiple WebLogic Server Managed server instances running simultaneously and working together to provide increased scalability and reliability. A cluster appears to clients to be a single WebLogic Server instance. The server instances that constitute a cluster can run on the same machine, or be located on different machines.

A cluster's capacity can be increased by adding additional Managed server instances to the cluster on an existing machine, or by adding machines to the cluster to host the incremental Managed server instances. Each server instance in a cluster must run the same version of WebLogic Server.

Typically, the administration for the WebLogic Server instance is done through an Administration Server or Admin Server. The Managed Servers do not require the Administration Server to be up and running.

Prerequisites

Make sure that the following prerequisites are met before configuring a WebLogic cluster:

- Experience in setting up a WebLogic Server cluster is required!
- The WebLogic software needs to be installed on all the machines that will be part of the cluster (that will run WebLogic server instances). Make sure that the same version of the WebLogic software is installed on all nodes.

The OHI Claims release bundle contains scripts that may be used to automate the creation of a WebLogic Cluster.

6. <http://download.oracle.com/javase/1.4.2/docs/guide/awt/AWTChanges.html#headless>

7. http://download.oracle.com/docs/cd/E12839_01/web.1111/e13709/load_balancing.htm#CHDGFIBD

Note Using these requires experience setting up a WebLogic Server cluster. The scripts are located in `<OHI_ROOT>\util\wlst`. Note that the scripts are provided "as is".

Before using the scripts, stage these to the environment in which they will be applied and make sure that the scripts can be executed.

Note If the cluster setup is for a distributed environment, make sure to stage the files on a shared disk so that all machines can access these.

Before executing the steps to create a WebLogic Cluster, the following must be done in preparation:

- Change the `setEnv.sh` script to match the settings of the environment in which the scripts will be applied, e.g. set the correct Middleware Home (MW_HOME) and reference a Java Home.
- Populate the `wlst\properties\createOHIDomain.properties` file with the values for the desired setup.

The OHI Domain creation script supports the following Domain Topologies:

1. Admin Server only
2. Admin Server + single Managed Server (single host)
3. Admin Server + single Managed Server (distributed)
4. Admin Server + multiple Managed Servers (single host)
5. Admin Server + multiple Managed Servers (distributed)
6. Admin Server + multiple Clustered Managed Servers (single host)
7. Admin Server + multiple Clustered Managed Servers (distributed)

Sample configuration files are provided in `<OHI_ROOT>\util\wlst\properties\samples` for all Domain Topologies mentioned.

Steps for setting up a WebLogic Cluster

Perform the following steps for setting up a WebLogic Cluster:

- Set up a Node Manager on all hosts in the cluster.
- Create a WebLogic domain for OHI Claims.
- Generate node manager boot & startup properties.
- Register the Domain with the Node Manager.
- Optional: Create WebLogic Domain Template for secondary hosts.
- Set up a load balancer to distribute requests to different managed servers in the cluster.

Warning Before putting the domain into production, make sure that the environment is secure. See the specific WebLogic documentation with respect to "Securing a Production Environment".

Starting and stopping WebLogic Server is covered in the Operations Guide.

7. http://download.oracle.com/docs/cd/E12839_01/web.1111/e13709/load_balancing.htm#CHDGFIBD

Set up a Node Manager for all nodes in the cluster

Note This step must be performed on all hosts (primary and secondary) that will be part of the WebLogic Server domain.

Node Manager is a WebLogic Server utility that controls start, shut down, and restart of Administration Server and Managed Server instances from a remote location. A Node Manager process is not associated with a specific WebLogic domain but with a machine. The same Node Manager process can be used to control server instances in any WebLogic Server domain, as long as the server instances reside on the same machine as the Node Manager process. Node Manager must run on each computer that hosts WebLogic Server instances -whether Administration Server or Managed Server- that need to be controlled with Node Manager.

Before a domain is created set up a Node Manager. The Node Manager will run as "init.d" service. Use the `<OHI_ROOT>\util\wlst\registerNodeManagerService.sh` script (as *root*) to create the `nodemgrservice` file and to set the correct property values in the `nodemanager.properties` file:

- `StopScriptEnabled=true`
- `CrashRecoveryEnabled=true`
- `StartScriptEnabled=true`

Note All scripts are driven from properties for which the values are specified in the `<OHI_ROOT>\util\wlst\properties\createOHIDomain.properties` file.

Create a WebLogic domain for OHI Claims in the cluster

Note This step must be performed on the primary host only.

Use the WebLogic Configuration Wizard to create a domain for OHI Claims. Alternatively, use the `<OHI_ROOT>\util\wlst\createOHIDomain.sh` script (as *oracle* user).

Oracle suggested values for configuration of the WebLogic Cluster are listed in the following table:

Parameter	Suggested Value
Domain Name	ohi_domain
Administration Server Name	ohi_admin_server
Managed Server Name(s)	ohi_managed_serverX (where X is an integer value that starts with 1)
Cluster Name	ohi_cluster

Note that these values can be set in the `<OHI_ROOT>\util\wlst\properties\createOHIDomain.properties` file.

Required setting: make sure that the Server Start Mode for the domain is set to *Production Mode*.

Generate node manager boot & startup properties

Make sure that the server is up and running.

The easiest way to do is by using the `<OHI_ROOT>\util\wlst\generateNMPPropsOHIDomain.sh` script (as *oracle* user). Make sure that the `<OHI_ROOT>\util\wlst\properties\createOHIDomain.properties` file has all required values.

Note The `generateNMPPropsOHIDomain.sh` script needs to be executed from the root directory of the WebLogic domain that was created.

Verify that the `boot.properties` and `startup.properties` files were created correctly for all server instances and in the proper location (`$DOMAIN_HOME/servers/[SERVER_NAME]/data/nodemanager`).

Register the Domain with the Node Manager

Make sure that the server is up and running.

Enroll the domain (i.e. register the domain with the node manger service) by running `<OHI_ROOT>\util\wlst\properties\enrollOHIDomain.sh` (as *oracle* user). Verify that the enroll operation was successful, by checking the script output for "Successfully enrolled...".

Note The `enrollOHIDomain.sh` script needs to be executed from the root directory of the WebLogic domain that was created.

Optional: Create WebLogic Domain Template for secondary hosts

This step is only required if Managed Servers are defined that run on other hosts than the Admin Server.

Execute the "pack" command to create a WebLogic Domain Template for all secondary host machines. Alternatively, use the `<OHI_ROOT>\util\wlst\packOHIDomain.sh` script to do that. The scripts requires the fully qualified root directory of the WebLogic domain that was created as an input parameter. Transfer the generated WebLogic Domain Template to all secondary host machines. The template can now be removed from the primary host.

On any secondary host machine, use the "unpack" command to create the WLS Domain Directory. Alternatively, use the `<OHI_ROOT>\util\wlst\unpackOHIDomain.sh` script to do that. The scripts requires two arguments:

- a reference to the generated WebLogic Domain Template
- the fully qualified root directory of the WebLogic domain that was created as an input parameter

Set up a Load Balancer

A load balancer is needed to distribute incoming requests to the participating nodes in the cluster. Details about configuration of load balancers can be found in [Oracle Fusion Middleware Using Clusters for Oracle WebLogic Server⁷](#).

OHI Claims requires that HTTP session "stick" to the same node; that needs to be supported by the load balancer. OHI Claims maintains the session by sending a cookie to the client. The name of the cookie is OHISESSION.

7. http://download.oracle.com/docs/cd/E12839_01/web.1111/e13709/load_balancing.htm#CHDGFIBD

Note OHI Claims does not support HTTP Session state replication.

Required Web Services Patch for WebLogic clusters

To make the OHI Claims Web Services work in a WebLogic cluster, patch **9KA3** has to be applied (Bug number 10056851). The patch can be applied through **WebLogic Smart Update** utility and can be downloaded from **My Oracle Support** by searching for 10056851 in **Patches & Updates**.

Follow the steps mentioned below in order to apply this patch.

Step 1: Download the patch (zip file)

Step 2: Navigate to <WLS_INSTALL_DIR>/utils/bsu

Step 3: Create a folder named **cache_dir** under **bsu** directory and extract the contents of zip file to **cache_dir**

Step 4: Run the following command: <WLS_INSTALL_DIR>/utils/bsu/bsu.sh -install -patchlist=9KA3 -prod_dir="<WLS_INSTALL_DIR>/wlserver_10.3"

Step 5: Verify that the patch is installed successfully by running the following command: <WLS_INSTALL_DIR>/utils/bsu/bsu.sh -report

Step 6: The result should be similar to the screen-shot given below:

```
ssubiram@ssubiran-nl:~/wls1034/utlis/bsu$ ./bsu.sh -report
Patch Report
=====
Report Info
Report Options
  bea_home..... ## OPTION NOT SET
  product_mask..... ## OPTION NOT SET
  release_mask..... ## OPTION NOT SET
  profile_mask..... ## OPTION NOT SET
  patch_id_mask..... ## OPTION NOT SET
Report Messages
BEA Home..... /home/ssubiran/wls1034

Product Description
Product Name..... WebLogic Server
Product Version..... 10.3.4.0
Installed Components..... Core Application Server, Administration Console, Configuration Wizard and Upgrade Framework, Web 2.0 HTTP Pub-Sub Server, WebLogic
SCA, WebLogic JDBC Drivers, WebLogic Server Clients, Server Examples, Evaluation Database, Workshop Code Completion Support
Product Install Directory. /home/ssubiran/wls1034/wlserver_10.3
Java Home..... null
Java Vendor..... Sun
Java Version..... 1.6.0_21
Patch Directory..... /home/ssubiran/wls1034/patch_wls1034

Profile..... Default

Patch ID..... 9KA3
CR(s).....
Description..... JAX-WS RI DOESN'T WORK IN WLS CLUSTER ENVIRONMENT
Classpath
  Classpath type..... SYSTEM
  Classpath control jar.... weblogic_patch.jar
  Jar..... BUG10056851_1034.jar
  File..... META-INF/MANIFEST.MF
  File..... weblogic/wsee/jaxws/cluster/ClusterRoutingTubeUtils.class
ssubiram@ssubiran-nl:~/wls1034/utlis/bsu$
```

NoteThis bug is also fixed in Oracle WebLogic Server Patch Set Update 10.3.4.0.2 & Oracle WebLogic Server Patch Set Update 10.3.4.0.3. See the following links for more details: <https://support.us.oracle.com/oip/faces/secure/km/DocumentDisplay.jspx?id=1348977.1&h=Y> & <https://support.us.oracle.com/oip/faces/secure/km/DocumentDisplay.jspx?id=1328645.1&h=Y#aref2>

Final steps

The domain is almost ready to deploy the application. Perform these final steps before deployment:

- Update the setDomainEnv.sh scripts.
- Set the ohi-claims.properties, logback.xml and ohi-claims-security.config files.
- Configure the Coherence cluster.

Set Domain Environment Variables for OHI Claims

Note This step must be performed on the primary host only.

Edit the file `setDomainEnv.sh` in that directory and add the following lines at the beginning as shown in this sample:

```

USER_MEM_ARGS="-Xms8192m -Xmx8192m -XX:PermSize=1024m
-XX:MaxPermSize=1024m"
export USER_MEM_ARGS

if [ "${ADMIN_URL}" = "" ] ; then

EXTRA_JAVA_PROPERTIES="-Dlogback.configurationFile=/home/aia/software/oracle/wls1032/properties/logback.xml"
export EXTRA_JAVA_PROPERTIES
else
EXTRA_JAVA_PROPERTIES="<for a managed server, apply the settings as
listed earlier in this chapter>"
export EXTRA_JAVA_PROPERTIES
fi

```

Note: the `EXTRA_JAVA_PROPERTIES` need to be specified on one line.

Warning Before putting a domain into production, make sure that the environment is secure. See the specific WebLogic documentation with respect to "Securing a Production Environment".

INITIAL CONFIGURATION FOR OHI CLAIMS IN ORACLE FUSION MIDDLEWARE

LOGGING CONFIGURATION

OHI Claims makes use of Logback library for generating log output. That log output is controlled by `logback.xml` file that is referenced in the WebLogic Server configuration. Through the configuration file, the logging level can be controlled as well as the output channels (referred to as 'appenders') for log messages. An example of an output channel for logging is a file.

Predefined logging configurations

OHI Claims comes bundled with a number of predefined logback configurations:

- `logback.xml`: a default logging config file.
- `production-logback.xml`: for maximum performance, will reveal errors.
- `trace-logback.xml`: provides trace-level output (most detailed).

By default, `logback.xml` is used. To use one of the others, use the `-Dlogback.configurationFile` Java option in the `setDomainenv` script:

```

-Dlogback.configurationFile=production-logback.xml
or
-Dlogback.configurationFile=trace-logback.xml

```

The OHI Claims Operations Guide describes log files and how to control log output.

Logging Configuration For Web Services

To enable logging web services request and response, enable debug logging on DefaultServerSOAPHandler class. Add the following entries in logback.xml:

```
<logger
name="com.oracle.healthinsurance.jaxws.ext.handlers.DefaultServerSOAPHandler"
level="debug" />
```

SET REQUIRED DEFAULTS

The application requires default settings for a number of objects. Before default settings can be applied, users must be provisioned in order to access the system. Make sure the following prerequisites are met:

- Set up users in an LDAP Directory Server as outlined below.
- Provision the users in OHI Claims. For this purpose, a Provisioning service is provided.

Countries

Log into the application and navigate to the Relation Management / Countries page. Execute a query on the page.

If no countries are available, set up at least a default country using the directions in the following table:

Field	Description	Example
Code	The country code that conforms to the ISO 3166-1 standard for representation of countries.	US
Name	Name of the country	United States
Primary Date format	Set the primary date format for the country.	MM/dd/yyyy
Default	There is one and only one default country. The first country that is entered must be the default.	Check the check box.
Active	Make sure the default country is activated in order to be used.	Check the check box.

In case a list of countries was optionally loaded as sample data, verify that a default country is set and that the primary date format is set.

SET UP A DIRECTORY FOR FILE EXCHANGE

In a number of scenarios OHI Claims processes files, for example for the File Import integration points. It is recommended to set up a shared directory structure that can be accessed by any machine that executes the system.

For example:

- For inbound files: /<MOUNT_POINT>/ohi-claims/transfer/in
- For outbound or response files: /<MOUNT_POINT>/ohi-claims/transfer/out

AUTHENTICATION AND USER PROVISIONING

Before users can access the OHI Claims application, the following prerequisites must be met:

- Users need to authenticate themselves by entering a valid combination of username and password credentials. All pages (other than the login page) are only available to authenticated (and properly authorized) users.
- A user must be provisioned to access the OHI Claims application. The main purpose of OHI Claims user accounts is authorization: the administration of (role-based) access rights for users is handled in the OHI Claims application.

The following paragraphs provide details on authentication and provisioning.

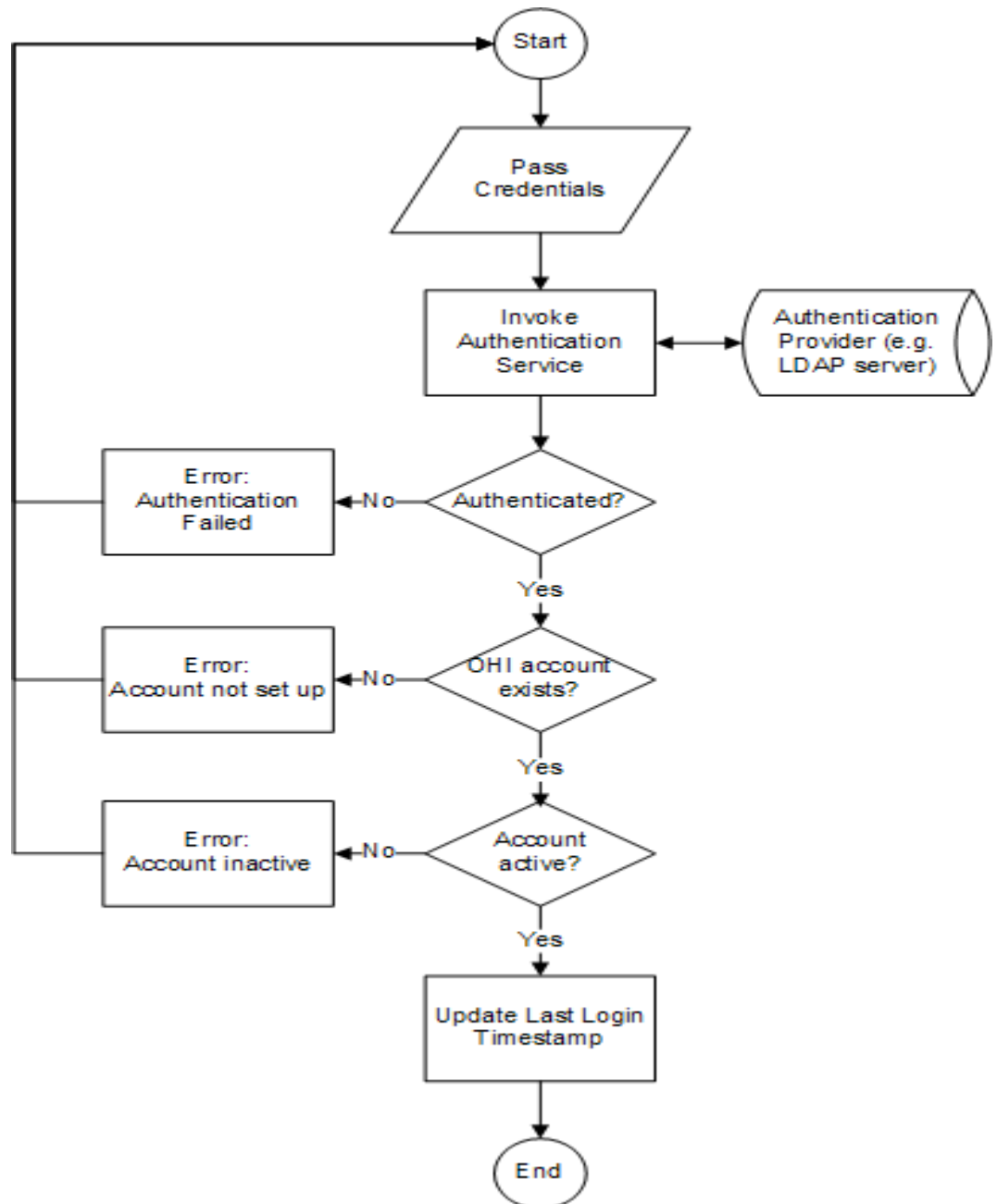
Authentication

Although user accounts are stored in the application, user passwords are not. As a result, the application relies on external services for authentication. It provides support for LDAP based authentication (LDAP version 3).

The application supports LDAP authentication by binding to the LDAP server using the user-supplied credentials. This way, no LDAP-specific account info needs to be stored in OHI Claims.

Users in the LDAP server are expected to be defined using the industry standard *inetOrgPerson* object class (which is derived from the *organizationalPerson* object class). Typically, in that class, the properties *uid* and *userpassword* are used to store the credentials used for logging in.

The following picture shows the flow of the authentication process:



Credentials are passed by the user via the OHI Claims Login page.

In the authentication process, the user account data that is stored in OHI Claims is accessed, for example for logging the last time the user successfully logged in to the system. Before someone can authenticate and subsequently access OHI Claims, an account has to be set up. For that purpose, OHI Claims offers a user provisioning service which is documented in the Integration Guide.

LDAP access configuration

Configure the properties as described in [Configuration Parameters](#) (page0).

Authentication using SSL: Steps for LDAPS Configuration

The system can authenticate using SSL-secured traffic by changing the LDAP connect string to **ldaps://<machine>.<domain>:<ssl_port>**. The SSL port is usually 636.

This paragraph describes the configuration for enabling SSL encrypted traffic between OHI Claims and Oracle Internet Directory (OID). OID supports three SSL Modes that are listed in the following table.

Note Mode 1 (No SSL Authentication) is not supported by OHI CLaims.

SSL Authentication Method	Description	Supported by OHI Claims?
Mode 1: No SSL Authentication	Neither the client nor the server authenticates itself to the other. No certificates are sent or exchanged. Only SSL encryption and decryption is used.	No
Mode 2: SSL Server Authentication	The directory server authenticates itself to the client. The directory server sends the client a certificate verifying that the server is authentic.	Yes
Mode 3: SSL Client and Server Authentication	The client and server authenticate themselves to each other and send certificates to each other.	Yes

To use the LDAPS feature, an SSL certificate needs to be obtained and installed on the Directory Server. Recommended steps for configuring Oracle Internet Directory 11g (OID) SSL Server Authentication (mode 2) are listed in this paragraph. The listed process is applicable for OID releases 11.1.1.2 to 11.1.1.4 and is based on Support Article 1203271.1 that is published on the Oracle Support website (and takes precedence over the product documentation). Article 1203271.1 covers steps 1 to 4 in the following list:

1. Support Article 1203271.1 suggests to create an additional OID Instance / Configset. Rationale as given in the article: "By default, the SSL authentication mode is set to authentication mode 1 (encryption only, no authentication). Be sure at least one Oracle Internet Directory server instance has this default authentication mode. Otherwise, you break Oracle Delegated Administration Services and other applications that expect to communicate with Oracle Internet Directory on the encrypted SSL port.". Create an additional OID instance (requires migrating the data of the original instance) or make sure that a configuration set is configured to also support authentication mode 1.
2. Use the Fusion Middleware Enterprise Manager to create a Wallet. For test systems Self-Signed Wallets are sufficient. For production systems Self-Signed Certificates are not recommended: Self-Signed Certificates typically lead to Certificate Trust messages. Users could react to these messages but in OHI Claims the user authentication process will fail as a result of an error in the SSL handshake. Create a proper Wallet for production systems.

For a production setup, generate a certificate request and send that to a Certificate Authority (CA). Import the SSL certificate that was issued by the CA before continuing with the following step.

3. Enable SSL for the OID server using the Wallet that was created in the previous step.
4. Restart the OID instance.
5. Stop the WebLogic (managed) servers that execute OHI Claims.
6. If a Self-Signed certificate was used, prevent Certificate Trust warnings that will break the authentication process by importing the self-signed root certificate in the cacerts certificates store of the JVM that executes OHI Claims.
 - Export the Self-Signed root certificate from the Self-Signed Wallet using the Fusion Middleware Enterprise Manager.
 - Make a backup of the JVM's cacerts file.
 - Import the root certificate into the cacerts certificate store using the keytool:

```
keytool -import -trustcacerts -keystore cacerts -storepass  
changeit -noprompt -alias <alias>  
-file <path_to_exported_root_certificate_file>
```

- where alias is a self-chosen, meaningful name for the root certificate (note: the alias has to be unique within the cacerts file!).
7. Start the WebLogic (managed) servers that execute OHI Claims.
 8. In the WebLogic Console, in the "Provider Specific" settings tab of the OHIClaimsAuthenticationProvider, set the SSLEnabled flag (restart of WebLogic server required).
 9. Test the setup. If an additional OID instance was created and the original instance is no longer needed, the original OID Instance / Configuration set can be stopped using opmnctl. Optionally, it can be removed.

Note The OHI Claims User Interface is browser-based. Network traffic between browsers on user workstations and the application servers executing OHI Claims uses the HTTP protocol. It is highly recommended to also secure the HTTP traffic between user workstations and the application servers, for example by using HTTPS (secure HTTP traffic). For user credentials and sensitive data to be encrypted always, properly secure all channels.

INTERNAL SYSTEM USER

During installation, an account for the Internal System User is created in the OHI_USERS table with the following characteristics:

- ID=10
- IND_ACTIVE=Y
- DISPLAY_NAME='Internal System User'
- LOGIN_NAME=null

8. <http://www.oracle.com/technetwork/java/javase/tech/vmoptions-jsp-140102.html>

This user cannot be used to log in to the application via the UI pages, because the LOGIN_NAME is null. The Internal System User is used for the internal processing. For example, records created or updated by an Integration Point, will have CREATED_BY and/or LAST_UPDATED_BY = 10 (the id of the internal system user).

Seeded access roles

As said in the previous section, the seeded Internal System User cannot be used to log in to the application to use the UI pages. So after installation, new users should be created with appropriate roles.

There is a bootstrap issue here: new roles should be defined first in the OHI Claims application using the Setup access role page. To be able to access the setup access role page, a user should exist with a role that gives access to this page.

To solve the bootstrap issue, role SETUP_ACCESS_ROLE is seeded during installation as follows:

Access Role Attribute	Value
Code	SETUP_ACCESS_ROLE
Name	Setup Access Role
Description	System role that gives access to setup access role page only.
Active	Y
Enabled	Y
Ohi specific?	Y

Table 2-1: Access Restriction Grants for SETUP_ACCESS_ROLE

Access Restriction Grant Attribute	Value
Access Restriction	AccessRoles
Create?	Y
Retrieve?	Y
Update?	Y
Delete?	Y
OHI specific?	Y

So the role SETUP_ACCESS_ROLE gives access to the setup access role page only.

After installation, the following steps need to be taken to setup a new user with the SETUP_ACCESS_ROLE granted:

1. Create a new access role SETUP_ACCESS_ROLE in the external identity store.
2. Create a new user in the external identity store and grant the SETUP_ACCESS_ROLE to that user.

3. Provision the user with the SETUP_ACCESS_ROLE granted to the OHI Claims application.

For explanation of these steps, see [Function Authorization](#) (page0).

To facilitate testing, role ALL_FUNCTIONS_ACCESS_ROLE is seeded also. This role gives access to all pages of the application. This role is not intended to be used in production environments, so this role is disabled by default.

Acces Role Attribute	Value
Code	ALL_FUNCTIONS_ACCESS_ROLE
Name	All Functions Access Role
Description	System role that gives access to all pages (disabled by default)
Active	Y
Enabled	N
Ohi specific?	Y

Table 2-2: Access Restriction Grants for ALL_FUNCTIONS_ACCESS_ROLE

Access Restriction Grant Attribute	Value
Access Restriction	All access restrictions of type 'Function'
Create?	Y
Retrieve?	Y
Update?	Y
Delete?	Y
OHI specific?	Y

After installation, the following steps needs to be taken to setup a new user with the ALL_FUNCTIONS_ACCESS_ROLE granted:

1. Create a new access role ALL_FUNCTIONS_ACCESS_ROLE in the external identity store.
2. Create a new user in the external identity store and grant the ALL_FUNCTIONS_ACCESS_ROLE to that user.
3. Provision the user with the ALL_FUNCTIONS_ACCESS_ROLE granted to the OHI Claims application.
4. Enable to access role ALL_FUNCTIONS_ACCESS_ROLE.

SYSTEM SIZING GUIDE

OVERVIEW

Getting the best performance in production environments requires a properly configured system. The most critical elements of that configuration surround JVM memory usage and garbage collection settings. This section covers:

8. <http://www.oracle.com/technetwork/java/javase/tech/vmoptions-jsp-140102.html>

- a subset of JVM options relevant to the sizing and tuning of the OHI Claims application
- sizing guidelines for the JDBC Connection Pool

JVM OPTIONS

A full list of JVM options can be found at [Java HotSpot VM Options](#).⁸

Setting	Description
-Xms	Initial java heap size.
-Xmn	Minimum java heap size.
-Xmx	Maximum java heap size.
-XX:PermSize	The default value is 64MB for a server JVM. Setting it to a more appropriate value eliminates the overhead of increasing this part of the heap.
-XX:MaxPermSize	Maximum size of the permanent generation.
-XX:NewSize	Size of the new generation.
-XX:MaxNewSize	Maximum size of the new generation.

Heap size does not determine the amount of memory your JVM will consume.

Java will allocate a certain amount of memory for the native part of the JVM, as well as a per thread call stack and reserved code cache for JIT compilation. The native part of JVM allocation can not be influenced and depends on a number of factors including platform and heap heuristics.

-XX:PermSize

The perm size should be set to 1024 Megabytes.

-XX:MaxPermSize

The maximum perm size should be set to 1024 Megabytes.

-Xms -Xmn -Xmx

Oracle recommends that -Xmn and -Xmx be set to the same value. This eliminates potentially costly heap reallocations, and can reduce the amount of heap fragmentation that can occur. Setting -Xms is then unnecessary since the heap size itself is static.

-XX:NewSize and -XX:MaxNewSize

Oracle recommends that these be set to the same value. As a rule of thumb, make sure that these are set to 30% of the heap size.

UI Related Memory Sizing

Oracle recommends allocating 10 Megabytes of heap per tab per user session as a minimum. A session maps mostly directly to a user, and a tab is an OHI Claims major UI element. For example: if someone is using the OHI Claims application and has 3 tabs open in the browser (e.g. a tab for searching claims and 2 tabs for changing claims) the total heap size allocated for the users session is 30

8. <http://www.oracle.com/technetwork/java/javase/tech/vmoptions-jsp-140102.html>

Megabytes. Note that the number of tabs is limited to a maximum of 15 for Claims pages and also 15 for non-Claims pages so a total of 30 tabs per user.

Processing Related Memory Sizing

Oracle recommends allocating 30 Megabytes of heap per Claims processing thread. There are 10 processing threads that will always be allocated (JMS related threads) and by default an additional 15 threads will be allocated unless the `ohi.processing.maxThreads` is used to change that number.

Note Consult Oracle before changing the default value for `ohi.processing.maxThreads`.

Web Services Related Memory Sizing

Similar to the sizing for Claims processing threads, Oracle recommends allocating 30 Megabytes of heap for servicing a Web Service request.

Working Space Memory

Once the UI and processing contributions have been calculated, the entire value should be increased by 30%. This accounts for working space for the garbage collector. Oracle recommends the use of the through put collector (concurrent mark and sweep or CMS) as well as the parallel new generation collector. Both of these collectors require extra working space for carry over objects during concurrent collection.

Sample Worksheet for heap size calculations

PermSize	1024m	
	Users	10
	Tabs per User	3
	Megabytes per tab per user	10
UI Memory	300m	$10 * 3 * 10$
	JMS Threads	10
	Processing Threads	15
	Threads for servicing Web Services requests	15
	Megabytes per Thread	30
Processing Memory	1200m	$(10 + 15 + 15) * 30$
Total Memory	2524m	PermSize + UI Memory + Processing Memory
Heap Setting (-Xmn -Xmx)	3282m	2524Mb * 1.3 (garbage collector overhead)
New Size (-XX:NewSize -XX:MaxNewSize)	1078m	3282Mb / 3

In this scenario, the `-Xmn` and `-Xmx` values would be set to a minimum of 3282M in the `setDomainEnv.sh` script for the OHI Claims installation.

Allocating too little Memory

Allocating too little memory can have significant performance impacts. It can lead to garbage collector issues, system freeze and severe system performance issues.

Allocating too much Memory

Allocating too much memory can lead to lengthy garbage collection pauses and lengthy memory defragmentation (also known as compaction). That in turn may lead to system failures. Make sure that the maximum heap size does not exceed 8192 megabytes. If more memory is required to service all requests then set up additional managed servers in the same domain / cluster for running the system.

CONNECTION POOL SIZING GUIDELINES

OHI Claims utilizes a WebLogic Data Source & Connection Pool to connect to the database. The number of connections in the Connection Pool should be properly sized. This paragraph contains guidelines for that sizing exercise.

Connection Pool management

The connection pool is managed by the WebLogic application server. For example, WebLogic may temporarily test a connection before it is handed to the process that requested it. For determining the size of the connection pool the total amount of connections is corrected for this overhead by increasing it with 30%.

UI Related Connection Pool Sizing

The number of connections for servicing UI requests depends largely on the number of (concurrent) users. Roughly, servicing a user's request requires the use of one connection. After the system's response is completed "think time" (before the user initiates a follow-up request) needs to be taken into account. Oracle assumes that a user spends 75% of his time evaluating the response and for initiating the next activity. The formula for determining the number of connections for servicing UI requests is: number of concurrent users * 75% with a minimum of 50 connections for handling UI requests.

Processing Related Connection Pool Sizing

For Claims processing, the number of database connections required is similar to the amount of processing threads.

Web Services Related Connection Pool Sizing

For handling a Web Service request, OHI Claims uses more than one database connection. For auditing purposes, information is logged using a separate connection. As a result, for handling Web Services requests the number of database connections required is: the number of concurrent requests * 2.

Sample Worksheet for connection pool calculations

	Users	40
UI Database Connections	50	40 * 0.75 = 30 (with a minimum of 50)

	JMS Threads	10
	Processing Threads	15
Processing Database Connections	25	(10 + 15)
	Number of concurrent Web Services Requests	15
Web Services Database Connections	30	15 * 2
Total number of Database Connections	137	(50 + 25 + 30) * 1.3 (management overhead)

As a result, the maximum number of database connections in the pool should be set to 137. Make sure that the number of sessions that the database can accommodate is larger than that amount.

Release installation

In this chapter, the generic process for Installing an OHI Claims release is described.

Release specific instructions are documented in the Release Notes for that specific release.

INSTALL DATABASE OBJECTS

CHANGE INSTALLATION CONFIGURATION

1. In `<OHI_ROOT>\util\install`, make a copy of `ohi_install.cfg.template` and name it `ohi_install.cfg`.
2. Edit `ohi_install.cfg` to contain your specific database connection data and other configuration settings. The settings are explained in the file itself.

Note By default, the schema passwords will be similar to the schema user names. The `ohi_install.cfg` files allows the specification of different passwords. Alternatively, specify empty string passwords to have the option of entering the passwords at the command prompt. In the latter case, the passwords will not end up in a configuration file.

Warning Default schema passwords should not be used.

Warning Oracle recommends that schema passwords are entered at the command prompt. Never store passwords in configuration files.

Configure Instance Discriminator

In accordance with the concepts explained in the paragraph Enabling Replication of Setup Data, the correct environment or instance must be configured during a new installation. The data that is entered for `ohiInstances` in the `ohi_install.cfg` file is stored in the database when a fresh install is performed. Make sure to assign unique discriminator values for each environment.

RUN INSTALLER

1. Open a command window and browse to `<OHI_ROOT>\util\install`.
2. Run the installer by typing `ohi_install.bat`. This will assume that the config file `ohi_install.cfg` is present in the same directory where `ohi_install.bat` is present and uses the default configuration from `ohi_install.cfg`. To specify a

different location of ohi_install.cfg or to specify a different environment from ohi_install.cfg, follow the next step.

3. Specify the command line options to specify the location of ohi_install.cfg file and environment to use from ohi_install.cfg.
 - Eg: **ohi_install.bat -c /home/oracle/someLocation/ohi_install.cfg -e dev**
4. The command line arguments are explained below:

Option		Argument	Description
Short	Long		
-c	--cfg	config file path	The location of the configuration file. Default is ohi_install.cfg
-e	--env	environment name	The name that specifies which of the environment settings from the config file to use

Install Seed Data

Part of the database objects installation is the installation of Seed Data

Types of Seed Data

Generic Seed Data

Seed data is maintained by Oracle. Customers should not change this data. It is delivered as part of a release and may be updated by software upgrades.

Tables containing Seed Data include:

Localization Seed Data

This category covers specific data that is required by localizations. The data is maintained by Oracle. There is currently no data in this category. Examples:

- Flex code definitions for a specific page
- Specific messages for localizations.

Sample Data

Sample data is provided by Oracle to give you a headstart during configuration. You can opt to install this data. It is *not* modified during future upgrades. Tables containing Sample Data include:

Table Name	Remarks
To be determined	

1. http://download.oracle.com/docs/cd/E12840_01/wls/docs103/config_wls/self_tuned.html

Restrictions on using Seed Data

Because Seed Data is maintained by Oracle, it may be modified or even deleted as part of an upgrade. Customers should therefore exercise caution when using seed data in their configuration by abiding these rules.

1. Do not remove (delete) Seed Data rows. A patch may re-insert the row.
2. Do not update columns, other than those indicated as updateble below.
3. Do not make references to rows that may be deleted by Oracle (see table below).

Violations of the rules above (especially rule 3) may lead to failures during the installation of upgrades.

The table below lists the Seed Data tables.

- **Data:** The table or logical entity
- **Updatable columns:** The customer may update the values in these columns. They will not be overwritten by upgrade scripts. Other columns should not be updated by the customer.
- **Physical Delete:** Upgrade scripts may delete this data. The customer should not create references to this data. Example: Do not use OHI messages for your own dynamic checks.

Data	Updatable columns	Physical Delete	Remarks
Access Restrictions		Yes	Also deletes Access Restriction Grants referring to this row
Access Restriction Grants		Yes	
Access Roles		No	Two roles are seeded
Boilerplate Texts		Yes	
Claim Forms		No	
Countries	all _b columns	No	
Country Regions		No	
Coverage Labels		No	
Dynamic Field Usages		No	
Dynamic Logic		No	
Fields (+ dynamic logic)		No	
Flex Codes		No	
Flex Code Sets (+ details)		No	
Flex Code Systems		No	
Languages	ind_default ind_installed	No	

1. http://download.oracle.com/docs/cd/E12840_01/wls/docs103/config_wls/self_tuned.html

Data	Updatable columns	Physical Delete	Remarks
Messages	ind_suppress_log_in_ui ind_suppress_log_in_ext ind_mark external_code	Yes	
Single Flex Code Definitions (+ usage)		No	
Task Types		Yes	Customer is not allowed to change anything in base table
Task Type Attributes	value_char value_number value_datetime value_clob	Yes	
Users		No	One User will be seeded (system user)

ENABLE TOTAL RECALL (OPTIONAL)

When Total Recall Option is activated, you should decide if one or more of the new tables should be added to a Flashback Data Archive.

Syntax to enable history tracking for a table is:

```
ALTER TABLE <tablename> FLASHBACK ARCHIVE [<Flashback Data Archive name>];
```

Note that the FDA name is required only when adding the table to a non-default FDA.

To disable history tracking for a table use:

```
ALTER TABLE <tablename> NO FLASHBACK ARCHIVE;
```

For convenience, we provide an *example* script that can help configuring archived tables:

```
import groovy.sql.Sql
import java.util.logging.*

def logger = Logger.getLogger(this.class.getName());

def config = new Config(args, 'setFlashbackArchive.cfg')
// Override the default level for the top-level logger
Logger.getLogger("").setLevel(config.log.level);
config.log()

def db = Sql.newInstance( config.db.url
                        , config.db.user
                        , config.db.passwd
                        , 'oracle.jdbc.driver.OracleDriver')

this.db = db

// Get the set of tables For which FBA needs to be switched off
def getTablesToSwitchOff(tables) {
    def tableList = tables.join(", ")
    switch_off = db.rows("""select fba.owner_name as owner
                            ,      fba.table_name
                            from    dba_flashback_archive_tables fba
                            where   fba.status = 'ENABLED'
                            and     fba.owner_name||'.'||fba.table_name not
                            in ('"" + tableList + ""')
```

```

        """)
    }

    for (table in config.flashback.tables) {
        switch_on = db.firstRow("""select fba.owner_name as owner
                                ,      fba.table_name
                                from    dba_flashback_archive_tables fba
                                where   fba.owner_name||'.'||fba.table_name
= $table
                                and    ( fba.status is null
                                or fba.status != 'ENABLED'
                                )
                                """)

        if (switch_on) {
            logger.info "Switching Flashback Archiving on for table ${table}"
            def stmt = "alter table " + table + " flashback archive
${config.flashback.archive}"
            try {
                println stmt
                db.execute(stmt)
            } catch (java.sql.SQLException e) {
                logger.warning "Error occurred while executing SQL " + stmt
                logger.warning "Error was " + e.getMessage()
                println "Unable to set Flashback Archive to
${config.flashback.archive} for table ${table} " + e.getMessage()
                println "Press enter to continue"
                new InputStreamReader(System.in).readLine()
            }
        }
    }

    // For every table that is not in the config-list but has FBA turned on, a
    // confirmation is asked before actually
    // turning off FBA
    getTablesToSwitchOff(config.flashback.tables).each { it ->
        def tableName = "${it.owner}.${it.table_name}"
        println "Switching Flashback Archiving OFF for table ${tableName}"
        println "Are you sure you want to do that? This will purge all history
for this table!! (Y/N)"
        response = new InputStreamReader(System.in).readLine()
        while (response.toUpperCase() != "Y" && response.toUpperCase() != "N")
        {
            println "Please respond with \"Y\" or \"N\""
            response = new InputStreamReader(System.in).readLine()
        }
        if (response.toUpperCase() == "Y") {
            def stmt = "alter table " + tableName + " no flashback archive"
            try {
                println stmt
                db.execute(stmt)
                logger.warning "Flashback Archiving was switched off for table
${tableName} as per users request"
            } catch (java.sql.SQLException e) {
                logger.warning "Error occurred while executing SQL " + stmt
                logger.warning "Error was " + e.getMessage()
                println "Unable to switch off Flashback Archive for table
${tableName} " + e.getMessage()
                println "Press enter to continue"
                new InputStreamReader(System.in).readLine()
            }
        } else {
            println "Flashback Archiving was NOT switched off for table
${tableName}."
            logger.warning "Flashback Archiving was NOT switched off for table
${tableName} as per users request"
        }
    }

    class Config {
        private static Logger logger = Logger.getLogger(Config.class.getName())
        def configFile
        def db = [:]
    }

```

```
def log = [:]
def flashback

def Config(args, configFile) {
    this.configFile = configFile
    def parsedConfig
    def cl = new CliBuilder(usage:
        'setFlashbackArchive.groovy [-c configFile]')

    cl.h(longOpt:'help', 'Show usage information and quit')
    cl.c(argName:'configFile', longOpt:'cfg', args:1, required:false,
'Config file, default is ' + configFile)

    def opt = cl.parse(args)

    if (!opt) {
        // the parse failed, the usage will be shown automatically
        println "\nInvalid command line, exiting..."
        System.exit(-1)
    } else if (opt.h) {
        cl.usage()
        System.exit(0)
    }

    if (opt.c) {
        this.configFile = opt.c
    }
    try {
        parsedConfig = new ConfigSlurper().parse(new
File(this.configFile).toURL())
    } catch (FileNotFoundException e) {
        logger.severe "Config file ${this.configFile} not found"
        System.exit(-1)
    }
    db = parsedConfig.db
    db.url = db.protocol + "://" + db.hostname + ":" + db.port + "/" +
db.sid
    log = parsedConfig.log
    flashback = parsedConfig.flashback
    if (!flashback) {
        println("\nflashback is not set in Config file")
        System.exit(-1)
    }
}

def log() {
    logger.config "db=" + db.toString()
}
}
```

This script works in conjunction with a configuration file that can be created by copying the following to a file named `setFlashbackArchive.cfg` and changing its settings to appropriate values:

```
import java.util.logging.Level
// Allowable values: SEVERE | WARNING | INFO | CONFIG | FINE | FINER |
FINEST
log.level = Level.INFO

// user has to be a database user with the FLASHBACK ARCHIVE object pri
vilege granted
// on the Archive used.
db {
    protocol = 'jdbc:oracle:thin:@'
    hostname = '<your hostname>'
    port     = '<your port>'
    sid      = '<your sid>'
    user     = '<your dbuser>'
    passwd   = '<your dbpassword>'
}
```

```
// This Flashback Archive has to exist.
// It can be created by a SYSDBA or a user with
// The tables argument contains a list of tables that should be archived.
// If you remove a table from this list, archiving will be disabled.
flashback {
    archive = '<name of your FlashBack Archive>'
    tables = ['<owner.table1>', '<owner.table2>']
}
```

The script can be run using Groovy (groovy <scriptname>)

INSTALL APPLICATION

This section lists the steps that are required to install the OHI application on the Oracle Fusion Middleware WebLogic Server (WLS).

CREATING WEBLOGIC WORK MANAGER

By default, WebLogic Server uses *default* work manager to handle thread management and perform self-tuning. This *default* Work Manager is used by an application when no other Work Managers are specified in the application's deployment descriptors. For more information, refer WLS documentation [here](#)¹.

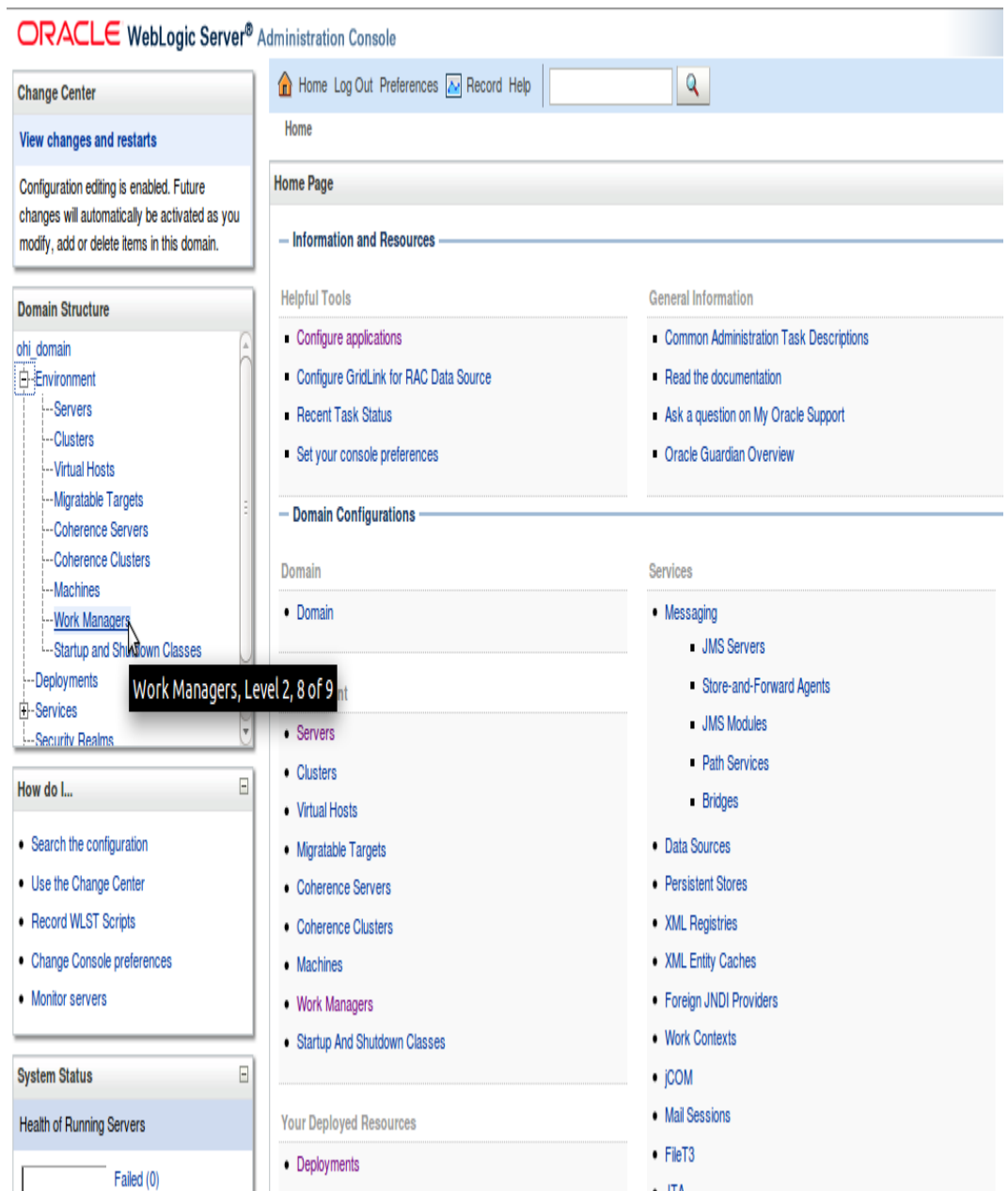
However, it is recommended to have 2 different work managers - 1 to serve UI requests and 1 to serve WebService requests. By having UI & WebService requests isolated, the predictable behavior can be improved. Follow the steps mentioned below to create work managers in WebLogic Admin Console.

Note Creating these work managers is done by WLST scripts as part of creating a new domain.

Step 1: Login to WebLogic Admin Console

Step 2: Click on **Environment > Work Managers**

1. http://download.oracle.com/docs/cd/E12840_01/wls/docs103/config_wls/self_tuned.html



Step 3: Click on **New** button to create a work manager to handle UI requests

ORACLE WebLogic Server® Administration Console

Home Log Out Preferences Record Help

Welcome, weblogic Connected to: ohi_domain

Home > Summary of Work Managers

Summary of Work Managers

A Work Manager defines a set of request classes and thread constraints that manage work performed by WebLogic Server instances. This page displays the global Work Managers, request classes and thread constraints defined for this domain.

Global Work Managers are defined at the domain level. You can also define application-level and module-level Work Managers.

[Customize this table](#)

Global Work Managers, Request Classes and Constraints

Showing 0 to 0 of 0 Previous | Next

<input type="checkbox"/>	Name ↕	Type	Targets
There are no items to display			

Showing 0 to 0 of 0 Previous | Next

Change Center

[View changes and restarts](#)

Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

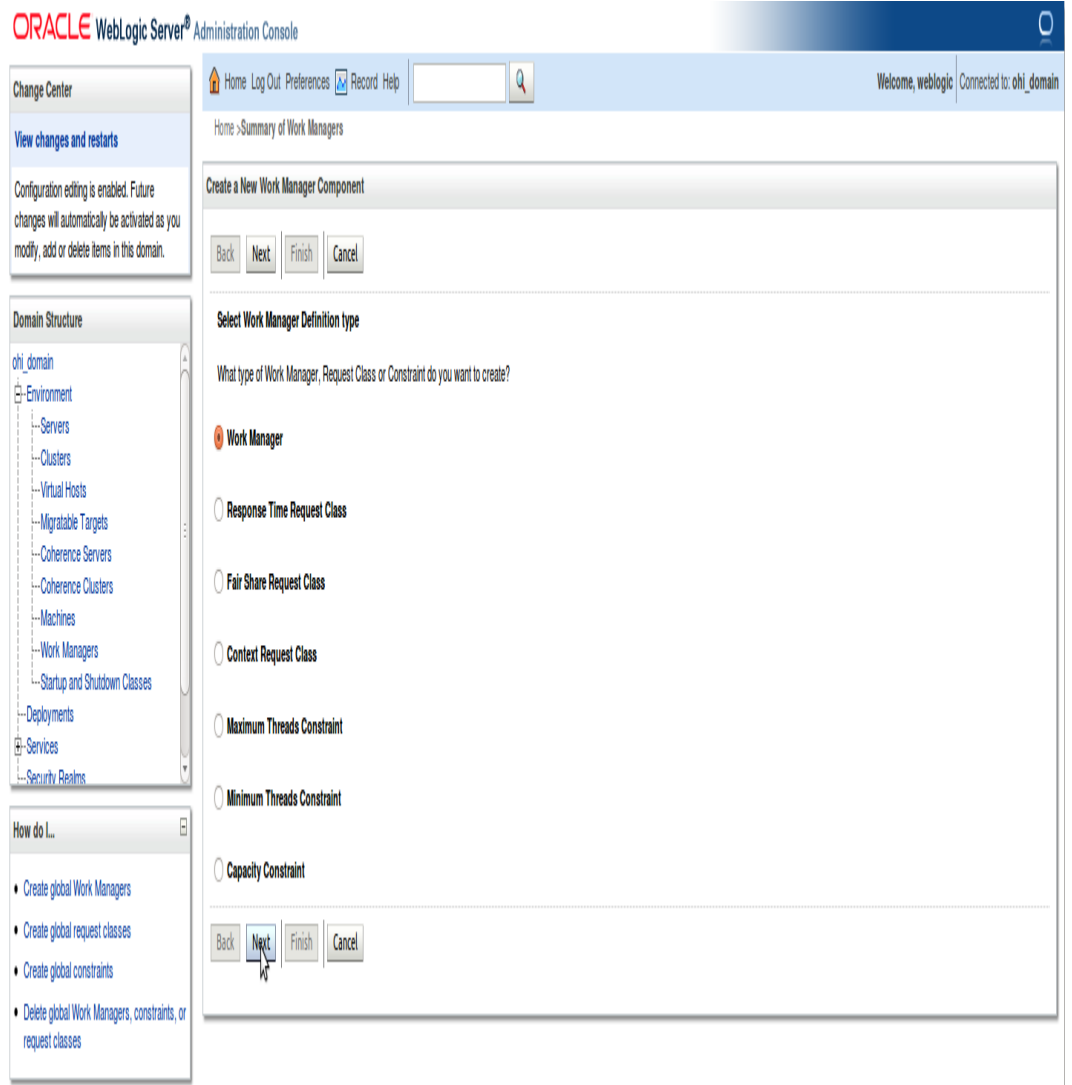
Domain Structure

- ohi_domain
 - Environment
 - Servers
 - Clusters
 - Virtual Hosts
 - Migratable Targets
 - Coherence Servers
 - Coherence Clusters
 - Machines
 - Work Managers**
 - Startup and Shutdown Classes
 - Deployments
 - Services
 - Security Realms

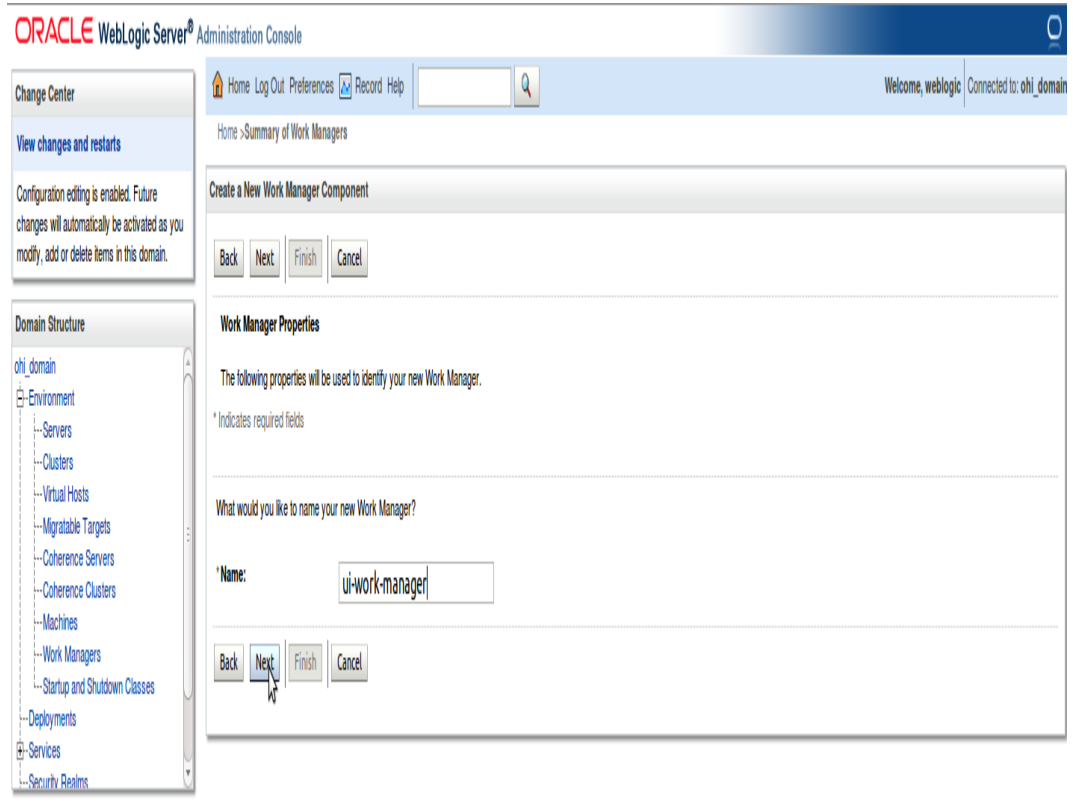
How do I...

- Create global Work Managers
- Create global request classes
- Create global constraints
- Delete global Work Managers, constraints, or request classes
- Create application-scoped Work Managers

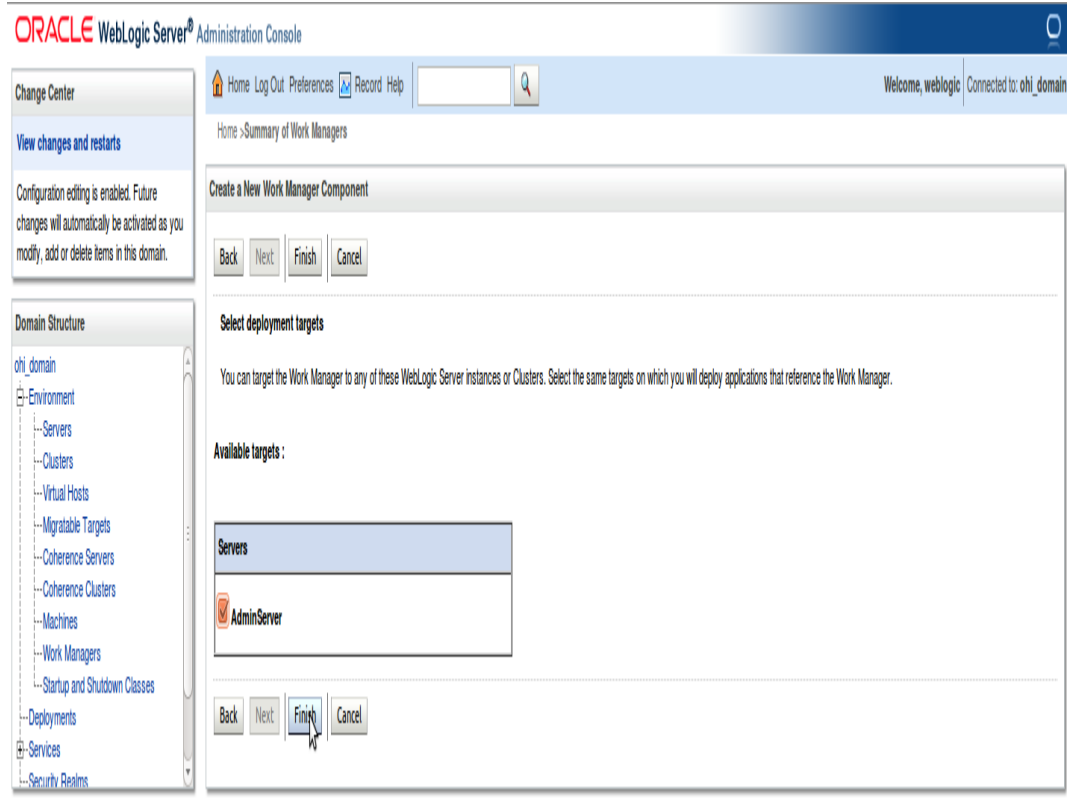
Step 4: Click on **Next** button



Step 5: Change the **Name** to **ui-work-manager** and click on **Next** button



Step 6: Select the appropriate target(s) from **Available targets** panel and click on **Finish** button



Step 7: In the **Summary of Work Managers** page, click on **ui-work-manager** link

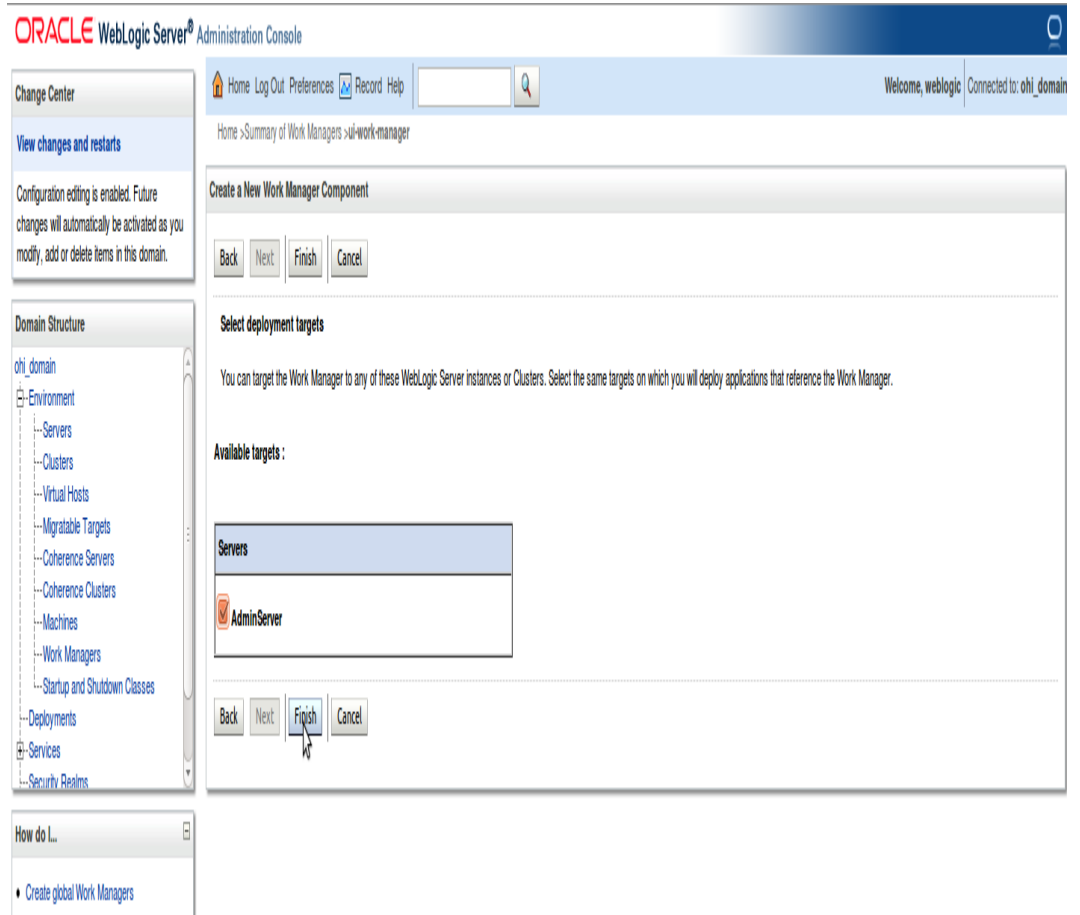
Step 8: Click on New button to create Minimum Threads Constraint

The screenshot shows the Oracle WebLogic Server Administration Console. The main content area displays the configuration for a Work Manager named 'ui-work-manager'. The 'Minimum Threads Constraint' field is set to '(None configured)', and the 'New' button next to it is being clicked. The console interface includes a navigation menu on the left, a top navigation bar, and a main content area with tabs for Configuration, Targets, and Notes.

Step 9: Change the Name and Count to UIMinThreadsConstraint and 5 respectively and click on Next button

The screenshot shows the Oracle WebLogic Server Administration Console interface. The main window displays the 'Create a New Work Manager Component' wizard. The current step is 'Minimum Threads Constraint Properties'. The wizard includes a navigation bar with 'Back', 'Next', 'Finish', and 'Cancel' buttons. Below the navigation bar, there is a section titled 'Minimum Threads Constraint Properties' with the following text: 'The following properties will be used to identify your Min Threads Request Class.' and '* Indicates required fields'. The wizard asks 'What would you like to name the new Minimum Threads Constraint?' and the 'Name' field contains 'UIMinThreadsConstraint'. It then asks 'What is the minimum number of threads to allocate for resolving deadlocks?' and the 'Count' field contains '5'. At the bottom of the wizard, there are 'Back', 'Next', 'Finish', and 'Cancel' buttons. On the left side, there is a 'Domain Structure' panel showing a tree view of the domain hierarchy, with 'Work Managers' selected. Below the 'Domain Structure' panel is a 'How do I...' panel with a list of tasks: 'Create global Work Managers', 'Create global request classes', 'Create global constraints', and 'Delete global Work Managers, constraints, or request classes'. The top of the console shows the 'ORACLE WebLogic Server Administration Console' header, a search bar, and the user 'Welcome, weblogic' connected to 'ohi_domain'.

Step 10: Select the appropriate target(s) from **Available targets** panel and click on **Finish** button



Step 11: Click on **New** button to create **Fair Share Request Class**

Step 12: Select **Fair Share Request Class** and click on **Next** button

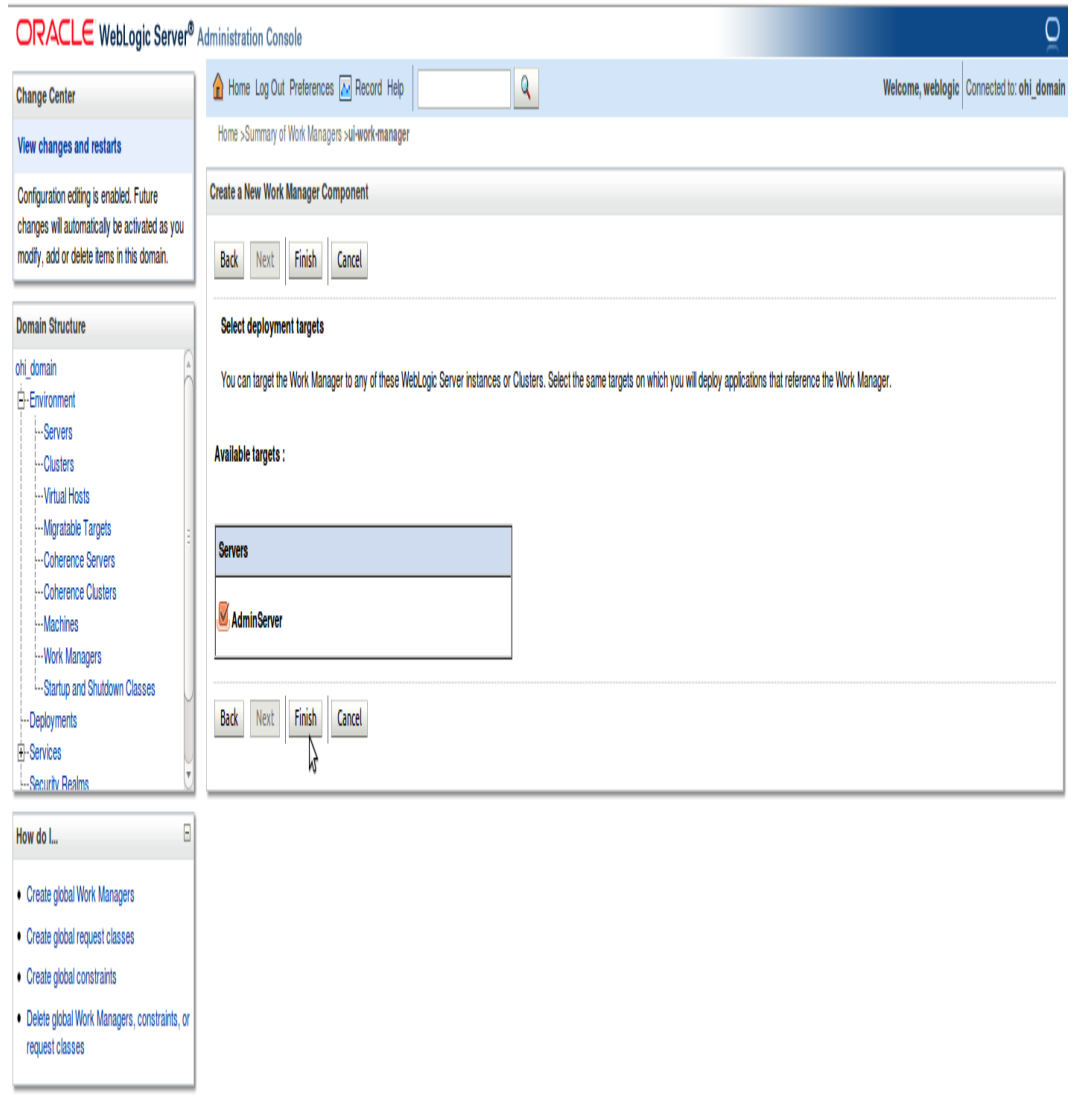
The screenshot displays the Oracle WebLogic Server Administration Console interface. The main content area shows the 'Create a New Work Manager Component' wizard. The 'Select Work Manager Definition type' step is active, with the 'Fair Share Request Class' option selected. The 'Domain Structure' tree on the left shows the hierarchy: ohi_domain > Environment > Work Managers. The 'How do I...' panel at the bottom left lists tasks such as 'Create global Work Managers' and 'Create global request classes'.

Step 13: Change the **Name** and **Fair share** to **UIFairShareReqClass** and **50** respectively.

The screenshot shows the Oracle WebLogic Server Administration Console interface. On the left, there are several panels: 'Change Center' with a 'View changes and restarts' link and a note about configuration editing; 'Domain Structure' showing a tree view of the domain hierarchy; 'How do I...' with a list of tasks; and 'System Status' showing the health of the domain. The main content area is titled 'Create a New Work Manager Component' and contains the following elements:

- Navigation buttons: Back, Next, Finish, Cancel.
- Section: **Fair Share Request Class**
- Text: 'The following properties will be used to identify your new Fair Share Request Class.'
- Note: '* Indicates required fields'
- Text: 'What would you like to name the new Fair Share Request Class?'
- Form: '*Name:' followed by a text input field containing 'UJFairShareReqClass'.
- Text: 'The fair share value determines how much time will be allocated to service requests from this class. The fair share value is a relative value (from 1 to 1000), not a percentage. Therefore if two fair share values were defined as 400 and 100, they would still have the same relative values as 80 and 20 or 4 and 1 respectively. For example, RequestClass1, RequestClass2 and RequestClass3 have the fair share values 10, 20 and 50 respectively. There is a 12.5% (10/80) chance that the next free thread will perform work for RequestClass1. Similarly, there is a 25% (20/80) chance it will next service RequestClass2 and a 62.5% (50/80) chance it will next service RequestClass3. What is the fair share value that requests from this class will use to allocate time for threads, relative to other Request Classes?'
- Form: 'Fair Share:' followed by a text input field containing '50'.
- Navigation buttons: Back, Next, Finish, Cancel.

Step 14: Select the appropriate target(s) from **Available targets** panel and click on **Finish** button



Repeat the above steps to create work manager for WebService (Refer the table below).

Configuration	Value
Work Manager Name	ws-work-manager
Minimum Threads Constraint Name	WSMinThreadsConstraint
Minimum Threads Constraint Count	5
Fair Share Request Class Name	WSFairShareReqClass
Fair Share Request Class Fair share	50

The work manager configuration like minimum threads constraint and fair share class etc can be modified at any time in WLS Admin Console to suit the needs.

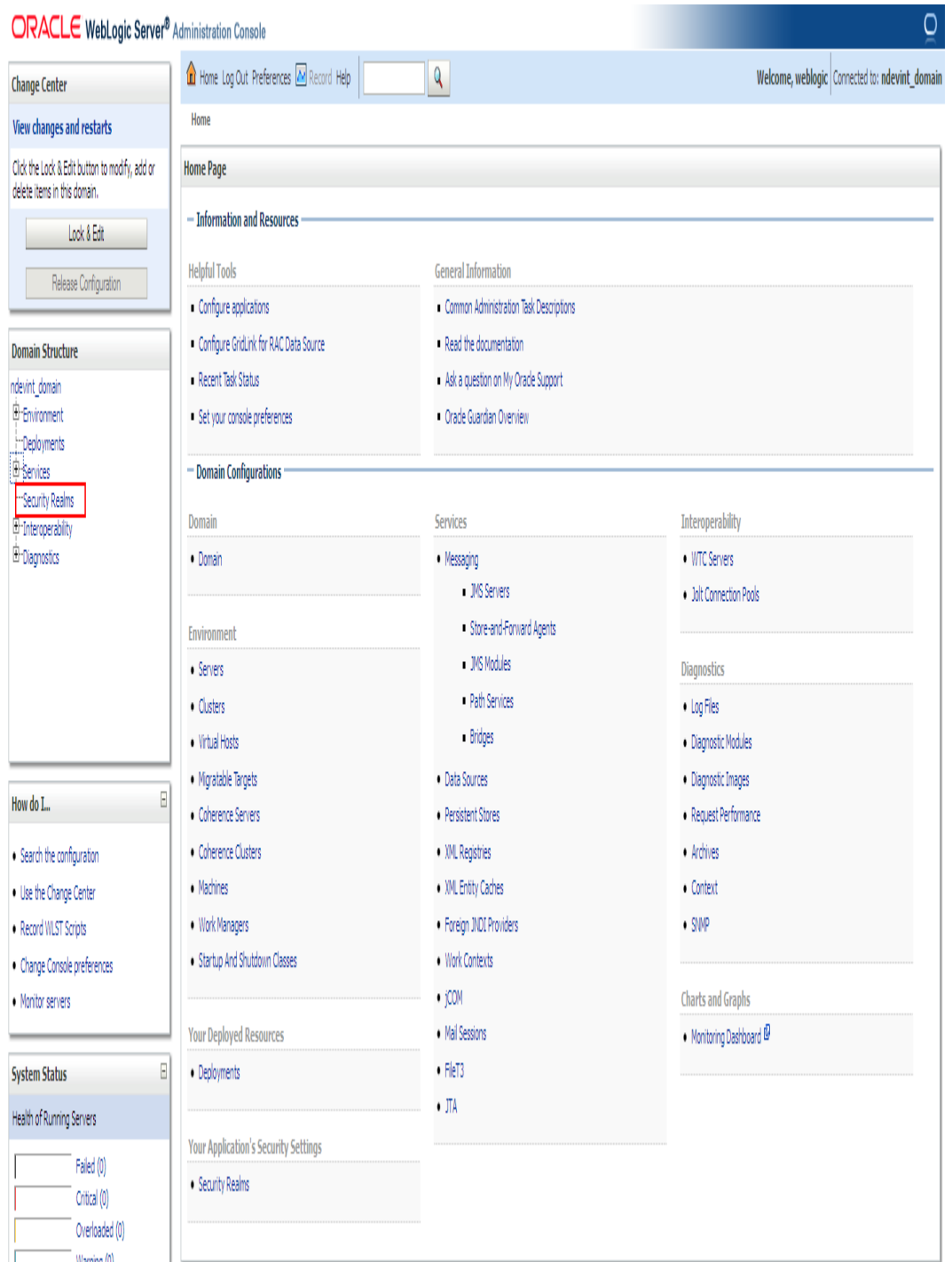
Note In these work managers, only **Min Threads Constraint** and **Fair Share Request Class** (50-50 between UI & WS) are configured. There is no need to configure Max Threads Constraint (The default is unlimited) and Capacity Constraint (The default is -1, which means the capacity is unlimited).

CONFIGURING OID AUTHENTICATION PROVIDER

The application uses a WebLogic Authentication Provider to connect to Oracle Internet Directory (OID) or to a third party LDAP server. This section describes the configuration of an OID or third party LDAP Authentication Provider.

Note Alternatively, for creating a new WebLogic domain for OHI Claims use the WLST scripts for setting up the Authentication Provider.

Step 1: Login to WLS admin console and click on **Security Realms** link.



Step 2: Click on **myrealm** link.

The screenshot shows the Oracle WebLogic Server Administration Console interface. The browser address bar displays the URL: `http://localhost:7001/console/console.portal?_nfpb=true&_pageLabel=Security/RealmRealmTablePage`. The page title is "ORACLE WebLogic Server Administration Console".

The left sidebar contains several panels:

- Change Center:** View changes and restarts. Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.
- Domain Structure:** A tree view showing the domain hierarchy: base_domain > Environment > Deployments > Services > Security Realms (highlighted) > Interoperability > Diagnostics.
- How do I...:**
 - Configure new security realms
 - Delete security realms
 - Change the default security realm
- System Status:** Health of Running Servers.

Failed	(0)
Critical	(0)
Overloaded	(0)
Warning	(0)
OK	(1)

The main content area is titled "Summary of Security Realms". It contains the following text:

A security realm is a container for the mechanisms—including users, groups, security roles, security policies, and security providers—that are used to protect WebLogic resources. You can have multiple security realms in a WebLogic Server domain, but only one can be set as the default (active) realm.

This Security Realms page lists each security realm that has been configured in this WebLogic Server domain. Click the name of the realm to explore and configure that realm.

Below the text is a section titled "Customize this table" and a table of realms:

Realms (Filtered - More Columns Exist)

Name	Default Realm
myrealm	true

The table includes "New" and "Delete" buttons for each row. The "Showing 1 to 1 of 1" and "Previous | Next" navigation options are also present.

The footer of the console displays the following information:

WebLogic Server Version: 10.3.4.0
 Copyright © 1996-2010, Oracle and/or its affiliates. All rights reserved.
 Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Note In WLS Production-mode use the **Lock & Edit** button before clicking on the **New** button.

Step 3: Click on **Providers** tab.

The screenshot shows the Oracle WebLogic Server Administration Console interface. The main content area is titled 'Settings for myrealm' and has several tabs: 'Configuration', 'Users and Groups', 'Roles and Policies', 'Credential Mappings', 'Providers', and 'Migration'. The 'Providers' tab is selected, and within it, the 'General' sub-tab is active. The configuration options include:

- Name:** myrealm (The name of this security realm. [More Info...](#))
- Security Model Default:** DD Only (Specifies the default security model for Web applications or EJBs that are secured by this security realm. You can override this default during deployment. [More Info...](#))
- Combined Role Mapping Enabled** (Determines how the role mappings in the Enterprise Application, Web application, and EJB containers interact. This setting is valid only for Web applications and EJBs that use the Advanced security model and that initialize roles from deployment descriptors. [More Info...](#))
- Use Authorization Providers to Protect JMX Access** (Configures the WebLogic Server MBean servers to use the security realm's Authorization providers to determine whether a JMX client has permission to access an MBean attribute or invoke an MBean operation. [More Info...](#))

At the bottom of the configuration area, there is an 'Advanced' section and a 'Save' button. On the left side, there are navigation panels for 'Change Center', 'Domain Structure', 'How do I...', and 'System Status'. The footer of the console displays 'WebLogic Server Version: 10.3.4.0' and 'Copyright © 1996, 2010, Oracle and/or its affiliates. All rights reserved.'

Step 4: Click on New button.

ORACLE WebLogic Server® Administration Console

Home Log Out Preferences Record Help

Welcome, weblogic Connected to: base_domain

Home > Summary of Security Realms > myrealm > Providers

Settings for myrealm

Configuration Users and Groups Roles and Policies Credential Mappings **Providers** Migration

Authentication Password Validation Authorization Adjudication Role Mapping Auditing Credential Mapping Certification Path Keystores

An Authentication provider allows WebLogic Server to establish trust by validating a user. You must have one Authentication provider in a security realm, and you can configure multiple Authentication providers in a security realm. Different types of Authentication providers are designed to access different data stores, such as LDAP servers or DBMS. You can also configure a Realm Adapter Authentication provider that allows you to work with users and groups from previous releases of WebLogic Server.

Customize this table

Authentication Providers

Showing 1 to 2 of 2 Previous | Next

<input type="checkbox"/>	Name	Description	Version
<input type="checkbox"/>	DefaultAuthenticator	WebLogic Authentication Provider	1.0
<input type="checkbox"/>	DefaultIdentityAsserter	WebLogic Identity Assertion provider	1.0

Showing 1 to 2 of 2 Previous | Next

Change Center

View changes and restarts

Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

Domain Structure

- base_domain
 - Environment
 - Deployments
 - Services
 - Security Realms
 - Interoperability
 - Diagnostics

How do I...

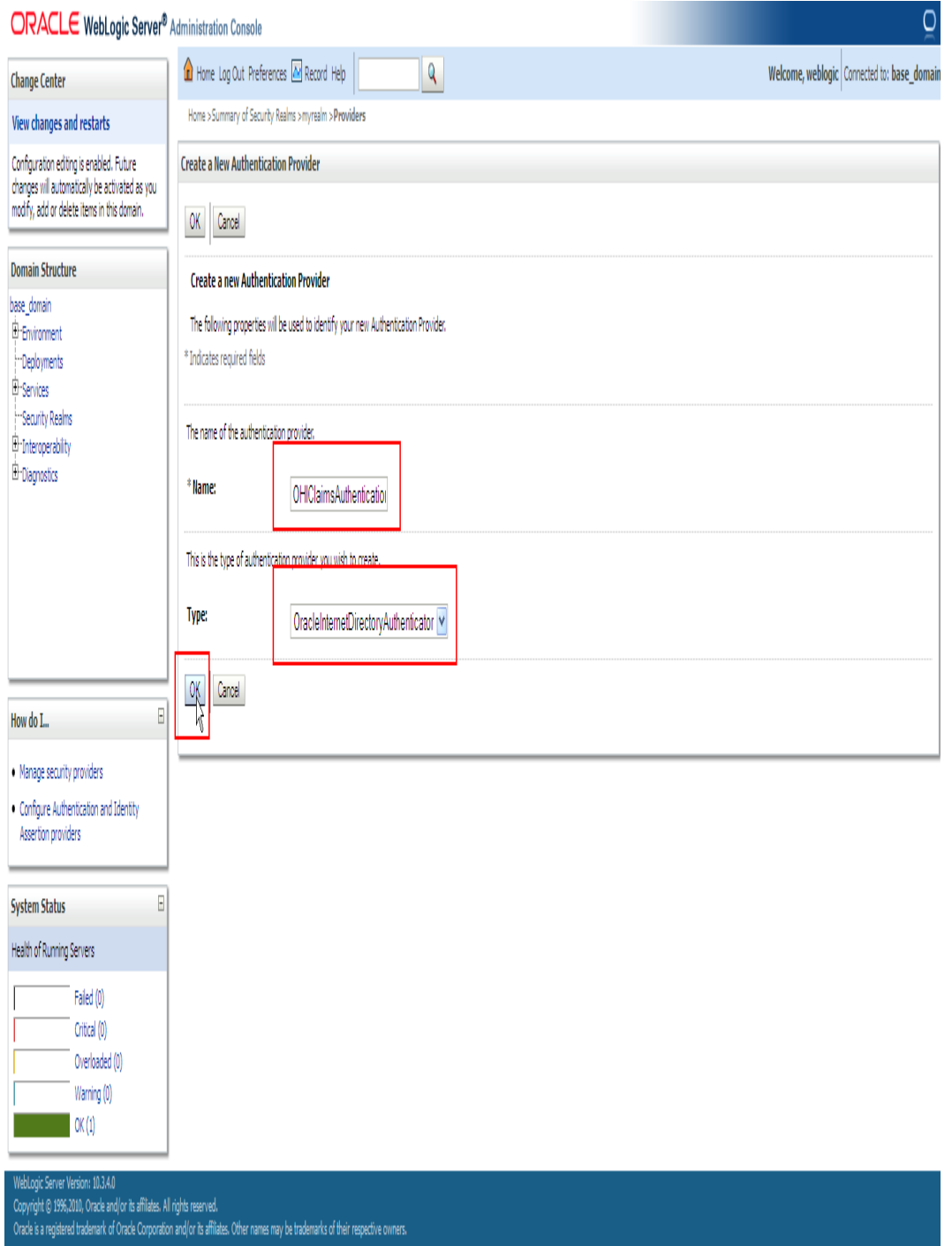
- Configure Authentication and Identity Assertion providers
- Configure the Password Validation provider
- Manage security providers
- Set the JAAS control flag
- Re-order Authentication providers

System Status

Health of Running Servers

- Failed (0)
- Critical (0)
- Overloaded (0)
- Warning (0)
- OK (1)

Step 5: Change **Name** and **Type** to **OHIClaimsAuthenticationProvider** and **OracleInternetDirectoryAuthenticator** (or to **LDAPAuthenticator** in case a third party LDAP server is used) respectively in **Create a new Authentication Provider** page. Click on **OK** button.



Step 6: Click on OHIClaimsAuthenticationProvider link.

ORACLE WebLogic Server® Administration Console

Home Log Out Preferences Record Help Welcome, weblogic Connected to: base_domain

Home > Summary of Security Realms > myrealm > Providers

Messages

All changes have been activated. However 1 items must be restarted for the changes to take effect.

Settings for myrealm

Configuration Users and Groups Roles and Policies Credential Mappings **Providers** Migration

Authentication Password Validation Authorization Adjudication Role Mapping Auditing Credential Mapping Certification Path Keystores

An Authentication provider allows WebLogic Server to establish trust by validating a user. You must have one Authentication provider in a security realm, and you can configure multiple Authentication providers in a security realm. Different types of Authentication providers are designed to access different data stores, such as LDAP servers or DBMS. You can also configure a Realm Adapter Authentication provider that allows you to work with users and groups from previous releases of WebLogic Server.

Customize this table

Authentication Providers

New Delete Reorder Showing 1 to 3 of 3 Previous Next

<input type="checkbox"/>	Name	Description	Version
<input type="checkbox"/>	DefaultAuthenticator	WebLogic Authentication Provider	1.0
<input type="checkbox"/>	DefaultIdentityAsserter	WebLogic Identity Assertion provider	1.0
<input type="checkbox"/>	OHClaimsAuthenticationProvider	Provider that performs LDAP authentication	1.0

New Delete Reorder Showing 1 to 3 of 3 Previous Next

System Status

Health of Running Servers

Failed (0)

Critical (0)

Overloaded (0)

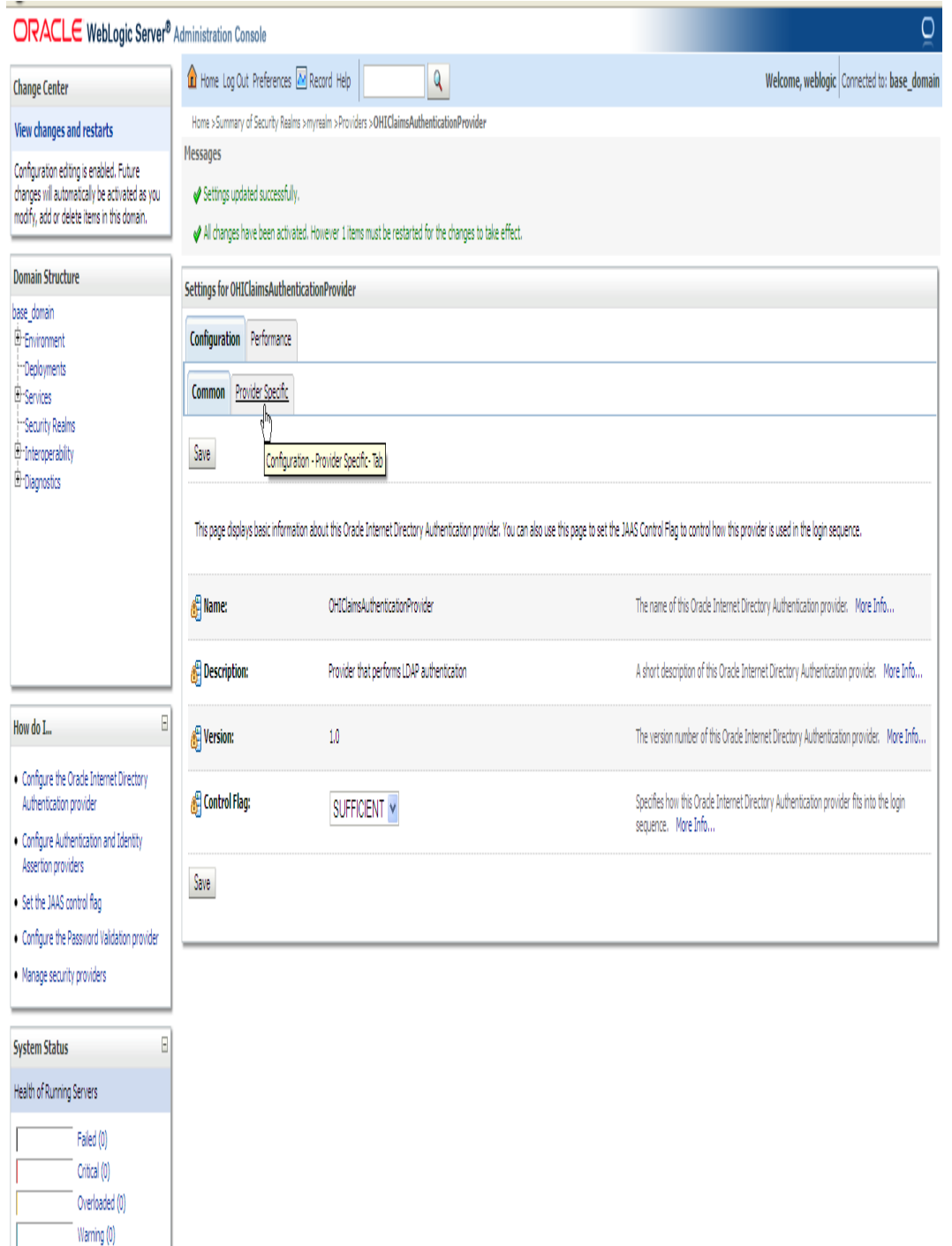
Warning (0)

OK (1)

Step 7: Change the **Control Flag** to **SUFFICIENT** and click on **Save** button.

The screenshot shows the Oracle WebLogic Server Administration Console interface. The main content area displays the 'Settings for OHIClaimsAuthenticationProvider' page. The 'Configuration' tab is active, and the 'Provider Specific' sub-tab is selected. The 'Control Flag' dropdown menu is highlighted with a red box and is set to 'SUFFICIENT'. The page also shows a 'Save' button and a description of the provider's role in the login sequence.

Step 8: Click on Provider Specific tab.



Step 9: Enter/change the values for various fields as shown below and select the option **Propagate Cause For Login Exception**. Click on **Save** button.

Field	Value
Host	LDAP hostname or IP address
Port	LDAP Port or SSL Port if the LDAP is SSL enabled. E.g.: 3060. In case LDAPS is used, make sure to check the SslEnabled flag as well.
Principal	LDAP admin principal: E.g.: cn=orcladmin
Credential	LDAP admin password
Confirm Credential	LDAP admin password

Field	Value
User Base DN	User Base distinguished name. E.g.: ou=Users,dc=healthinsurance,dc=oracle,dc=com
All Users Filter	E.g.: (&(uid=*)(objectclass=person))
User From Name Filter	E.g.: (&(uid=%u)(objectclass=person))
User Name Attribute	E.g.: uid
Group Base DN	If there are no groups in the LDAP, leave this field empty.

Note There are a few more properties (or fields in the page) which are not mentioned in the table above. Change the values of those fields to suit your LDAP settings.

General

Connection Pool Size: The LDAP connection pool size. Default is 6. [More Info...](#)

Connect Timeout: The maximum time in seconds to wait for the connection to the LDAP server to be established. If this attribute is set to 0, there is not a maximum time limit. [More Info...](#)

Connection Retry Limit: Specifies the number of times to attempt to connect to the LDAP server if the initial connection failed. [More Info...](#)

Parallel Connect Delay: The delay in seconds when making concurrent attempts to connect to multiple LDAP servers. [More Info...](#)

Results Time Limit: The maximum number of milliseconds for the LDAP server to wait for results before timing out. If this attribute is set to 0, there is no maximum time limit. [More Info...](#)

Keep Alive Enabled [More Info...](#)

Follow Referrals [More Info...](#)

Bind Anonymously On Referrals [More Info...](#)

Propagate Cause For Login Exception [More Info...](#)

Cache Enabled [More Info...](#)

Cache Size: The size of the cache (in kilobytes) that is used with the LDAP server. [More Info...](#)

Cache TTL: The time-to-live of the cache (in seconds) that is used with the LDAP server. [More Info...](#)

GUID Attribute: Specifies the name of the GUID attribute defined in the Oracle Internet Directory LDAP server. The default value is orclguid. [More Info...](#)

2. http://download.oracle.com/docs/cd/E17904_01/web.1111/e13737/gridlink_datasources.htm

Step 10: Click on **myrealm** link and then **DefaultAuthenticator** link. Change the **Control Flag** to **SUFFICIENT** and click on **Save** button.

The screenshot shows the Oracle WebLogic Server Administration Console interface. The main content area displays the 'Settings for DefaultAuthenticator' page, which is divided into 'Configuration', 'Performance', and 'Migration' tabs. The 'Configuration' tab is active, and the 'Common' sub-tab is selected. The 'Control Flag' is set to 'SUFFICIENT', which is highlighted with a red box. The 'Save' button is visible at the bottom of the configuration area.

On the left side, there are several panels: 'Change Center' with a 'View changes and restarts' link; 'Domain Structure' showing a tree view of the domain hierarchy; 'How do I...' with a list of configuration tasks; and 'System Status' showing the health of running servers.

Step 11: Make sure that file **<OHI_ROOT>/util/security/ohi-claims-security.config** is available in the **<PROPERTIES_ROOT>** directory (that also contains the **ohi-claim.properties** and **logback.xml** files).

Step 12: Add the following system property to the **JAVA_OPTIONS** in the **setDomainEnv.sh** script:

```
-Djava.security.auth.login.config=<PROPERTIES_ROOT>/ohi-claims-security.config
```

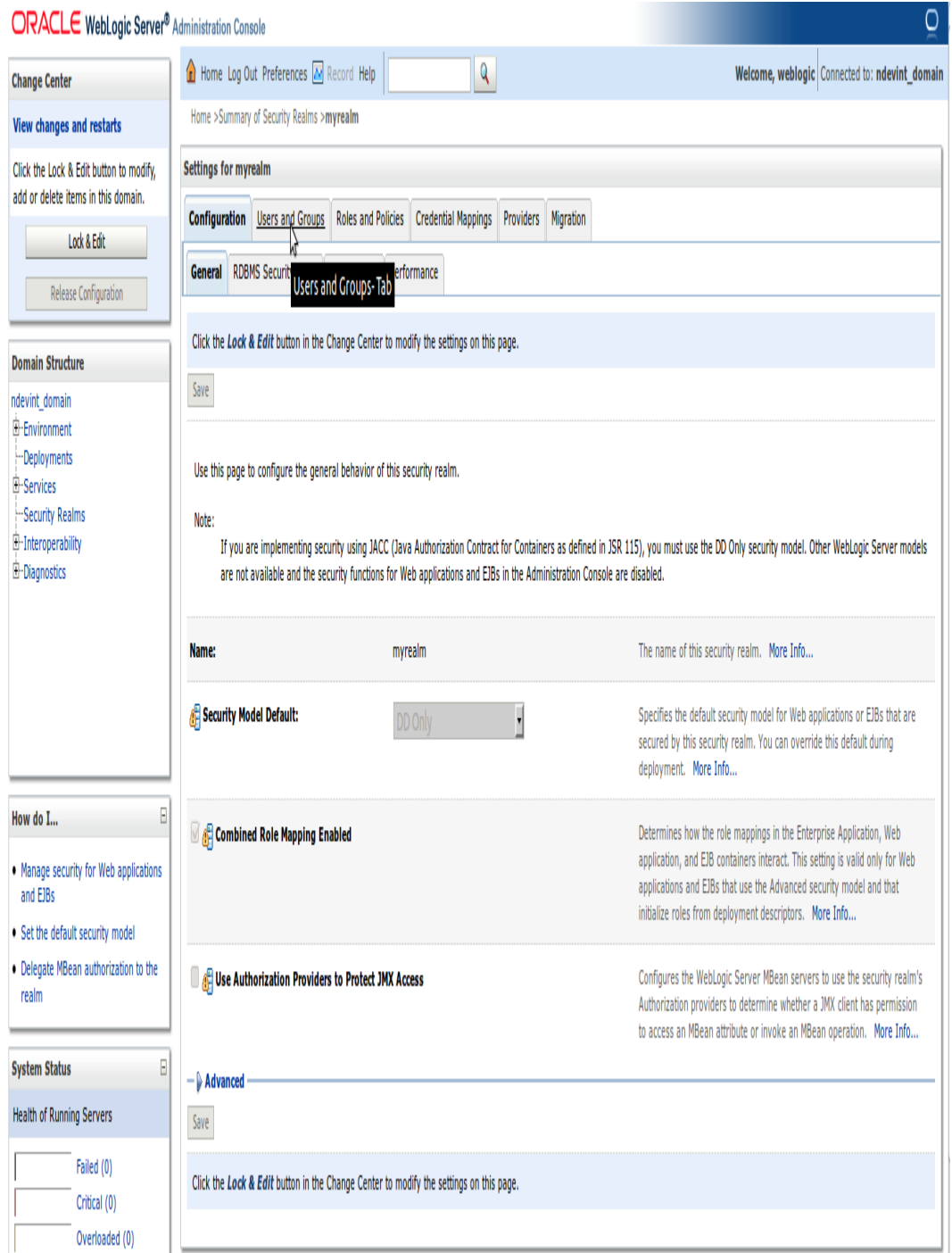
Step 12: Restart the WebLogic Server.

Optionally, verify that the authentication provider is configured successfully (after the WebLogic Server is restarted) by following the steps mentioned below:

Step 1: Login to WLS Admin Console and click on **Security Realms**

Step 2: Click on **myrealm**

Step 3: Click on **Users and Groups** tab



Step 4: You should be able to see the list of users from **OHIClaimsAuthenticationProvider** (in addition to the default users from **DefaultAuthenticator**).

ORACLE WebLogic Server® Administration Console

Home Log Out Preferences Record Help Welcome, weblogic Connected to: ndevint_domain

Home > Summary of Security Realms > myrealm > Users and Groups

Settings for myrealm

Configuration **Users and Groups** Roles and Policies Credential Mappings Providers Migration

Users Groups

This page displays information about each user that has been configured in this security realm.

[Customize this table](#)

Users

New Delete Showing 1 to 10 of 42 Previous Next

Name	Description	Provider
boosthoo		OHIClaimsAuthenticationProvider
configdata		OHIClaimsAuthenticationProvider
eblair		OHIClaimsAuthenticationProvider
ftollena		OHIClaimsAuthenticationProvider
gmailotr		OHIClaimsAuthenticationProvider
guruvenk		OHIClaimsAuthenticationProvider
hchang		OHIClaimsAuthenticationProvider
ibleljen		OHIClaimsAuthenticationProvider
jverheul		OHIClaimsAuthenticationProvider
jvissters		OHIClaimsAuthenticationProvider

New Delete Showing 1 to 10 of 42 Previous Next

Change Center

View changes and restarts

Click the Lock & Edit button to modify, add or delete items in this domain.

Lock & Edit

Release Configuration

Domain Structure

- ndevint_domain
 - Environment
 - Deployments
 - Services
 - Security Realms
 - Interoperability
 - Diagnostics

How do I...

- Manage users and groups
- Create users
- Modify users
- Delete users

System Status

Health of Running Servers

- Failed (0)
- Critical (0)
- Overloaded (0)
- Warning (0)
- OK (1)

TESTING LDAP CONFIGURATION WITHOUT DEPLOYING OHI CLAIMS APPLICATION (OPTIONAL)

To quickly test the configuration of the WebLogic Authentication Provider, a sample web application is bundled with the OHI Claims release for convenience. It can be used to test the LDAP configuration without having to deploy the OHI Claims application.

Note Rationale: for every LDAP configuration change the WebLogic server needs to be restarted before the authentication can be tested again. Restarting the WebLogic server while the OHI

Claims application is deployed takes a significant amount. To reduce the amount of WebLogic restart time, this sample web application will be useful - it is very easy to install & test.

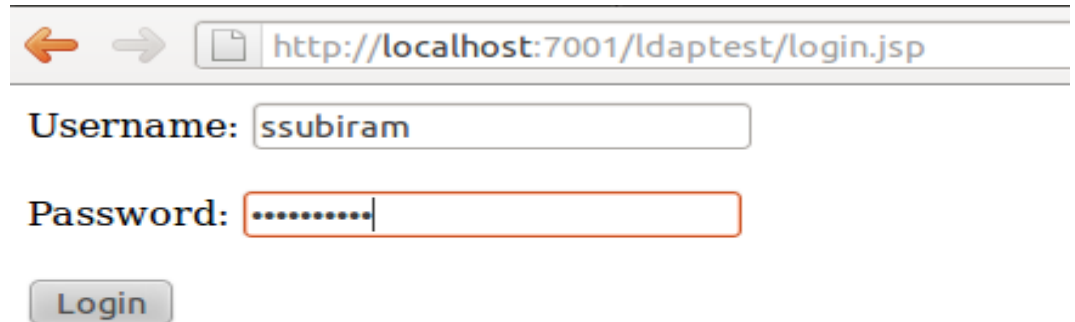
Follow the steps mentioned below to install and use the LDAP tester application:

Step 1: Login to the WebLogic AdminServer Console

Step 2: Install <OHI_ROOT>/util/ldap/ldaptest.war

Step 3: Navigate to http://<MACHINE_NAME/IP_ADDRESS>:<PORT>/ldaptest/login.jsp

Step 4: Enter a valid *Username & Password*. Click on *Login* button



← → <http://localhost:7001/ldaptest/login.jsp>

Username:

Password:

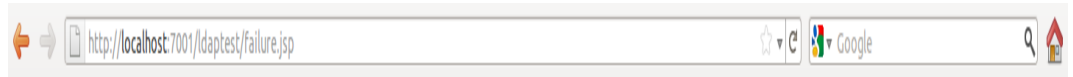
Login

Step 5: Upon successful authentication a page similar to the following is shown:



Step 6: If the authentication fails, the following page is shown:

2. http://download.oracle.com/docs/cd/E17904_01/web.1111/e13737/gridlink_datasources.htm



Login Failed

Stack trace:

```

javax.security.auth.login.LoginException: javax.security.auth.login.LoginException: java.lang.SecurityException: [Security:090304]Authentication Failed: User ssubiram
javax.security.auth.login.FailedLoginException: [Security:090302]Authentication Failed: User ssubiram denied at
weblogic.security.auth.login.UsernamePasswordLoginModule.login(UsernamePasswordLoginModule.java:199) at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39) at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25) at
java.lang.reflect.Method.invoke(Method.java:597) at javax.security.auth.login.LoginContext.invoke(LoginContext.java:769) at
javax.security.auth.login.LoginContext.access$000(LoginContext.java:186) at javax.security.auth.login.LoginContext$4.run(LoginContext.java:683) at
java.security.AccessController.doPrivileged(Native Method) at javax.security.auth.login.LoginContext.invokePriv(LoginContext.java:680) at
javax.security.auth.login.LoginContext.login(LoginContext.java:379) at com.oracle.healthinsurance.ldap.AuthenticationServiceImpl.authenticate(AuthenticationServiceImpl.java:40) at
jsp_servlet._authenticate._jspService(_authenticate.java:99) at weblogic.servlet.jsp.JspBase.service(JspBase.java:34) at
weblogic.servlet.internal.StubSecurityHelper$ServletServiceAction.run(StubSecurityHelper.java:227) at
weblogic.servlet.internal.StubSecurityHelper.invokeServlet(StubSecurityHelper.java:125) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:300) at
weblogic.servlet.internal.TailFilter.doFilter(TailFilter.java:26) at weblogic.servlet.internal.FilterChainImpl.doFilter(FilterChainImpl.java:56) at
oracle.security.jps.ee.http.JpsAbsFilter$1.run(JpsAbsFilter.java:111) at java.security.AccessController.doPrivileged(Native Method) at
oracle.security.jps.util.JpsSubject.doAsPrivileged(JpsSubject.java:313) at oracle.security.jps.util.JpsPlatformUtil.runJaasMode(JpsPlatformUtil.java:413) at
oracle.security.jps.ee.http.JpsAbsFilter.runJaasMode(JpsAbsFilter.java:94) at oracle.security.jps.ee.http.JpsAbsFilter.doFilter(JpsAbsFilter.java:161) at
oracle.security.jps.ee.http.JpsFilter.doFilter(JpsFilter.java:71) at weblogic.servlet.internal.FilterChainImpl.doFilter(FilterChainImpl.java:56) at
oracle.dms.servlet.DMServletFilter.doFilter(DMServletFilter.java:136) at weblogic.servlet.internal.FilterChainImpl.doFilter(FilterChainImpl.java:56) at
weblogic.servlet.internal.WebAppServletContext$ServletInvocationAction.wrapRun(WebAppServletContext.java:3715) at
weblogic.servlet.internal.WebAppServletContext$ServletInvocationAction.run(WebAppServletContext.java:3681) at
weblogic.security.acl.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:120) at
weblogic.servlet.internal.WebAppServletContext.securedExecute(WebAppServletContext.java:2277) at
weblogic.servlet.internal.WebAppServletContext.execute(WebAppServletContext.java:2183) at weblogic.servlet.internal.ServletRequestImpl.run(ServletRequestImpl.java:1454) at
weblogic.work.ExecuteThread.execute(ExecuteThread.java:207) at weblogic.work.ExecuteThread.run(ExecuteThread.java:176)

```

SET UP JDBC DATA SOURCES

The application connects to the Oracle database through a Data Source that need to be specified in the WLS Server.

Note For security reasons, the database connections used by the application connect to database schemas that do not own database objects. These schemas are only granted the required privileges to use the objects.

The following sections describe setting up data sources for connecting to:

- an Oracle database that is running on a single machine
- a RAC-enabled Oracle database that is running on multiple machines

Data Source for connecting to an Oracle database that is running on a single machine

The following table lists the Data Source that must be configured in WLS before installing the application for use with an Oracle database that is executed on a single machine (not clustered):

Data Source Parameters	Non-clustered database	Explanation
Data Source Name	ohi-application-datasource	Logical name
JNDI Name	jdbc/claimsUserOhiApplicationDS	Used by the application to resolve the Data Source
Database Type	Oracle	
Database Driver	Oracle's Driver (Thin) for Instance connections; Versions:9.0.1,9.2.0,10,11 or Oracle's Driver (Thin) for Service connections; Versions:9.0.1,9.2.0,10,11	
Database Name	SID or service name of the database If the name of the Oracle driver that was selected contains the words "for Instance connections" enter the SID. If the name of the Oracle driver contains the words "for Service connections" enter the service name.	
Host Name	Name or IP address of the machine where the database is running	
Port	Port on which the database is running	
Database User Name	ohi_claims_user	Fixed value, do not change
Password & Confirm Password	Password of "ohi_claims_user"	The schema password as selected during the installation
Service Name	Service name of the database	SID or service name

The data sources can be created by either

1. using the `<OHI_ROOT>\util\wlst\createOHIDomain.sh` script (i.e. the data sources are created at the time the domain is created) or
2. creating them through WLS Admin Server console (see sample below).

2. http://download.oracle.com/docs/cd/E17904_01/web.1111/e13737/gridlink_datasources.htm

Data Source for connecting to an Oracle RAC database that is running on multiple machines

To support Oracle RAC features within Oracle WebLogic Server, Oracle recommends using Oracle WebLogic Server **GridLink Data Source**. A single GridLink data source provides connectivity between WebLogic Server and an Oracle Database service targeted to an Oracle RAC cluster. It uses the Oracle Notification Service (ONS) to adaptively respond to state changes in an Oracle RAC instance. An Oracle Database service represents a workload with common attributes that enables administrators to manage the workload as a single entity.

To configure this, the following steps need to be performed. For more details about GridLink Data Source configuration, see the Oracle WebLogic Server documentation [here](#)².

Configuring GridLink Data Source

Step 1: Login to WLS admin console and click the **Services/Data Sources** link.

Step 2: Click on **New** button and select the option **GridLink Data Source**

2. http://download.oracle.com/docs/cd/E17904_01/web.1111/e13737/gridlink_datasources.htm

The screenshot shows the Oracle WebLogic Server Administration Console interface. The top navigation bar includes 'Home', 'Log Out', 'Preferences', 'Record', and 'Help'. The user is logged in as 'weblogic' and connected to the 'ohi_domain'. The main content area is titled 'Summary of JDBC Data Sources' and has two tabs: 'Configuration' (selected) and 'Monitoring'. Below the tabs, there is a descriptive text about JDBC data sources and a 'Customize this table' link. A table titled 'Data Sources (Filtered - More Columns Exist)' is shown, but it is empty, displaying 'Showing 0 to 0 of 0' and 'Previous | Next'. A 'New' button dropdown menu is open, showing three options: 'Generic Data Source', 'GridLink Data Source', and 'Multi Data Source'. The 'GridLink Data Source' option is highlighted by the mouse cursor. On the left side, there is a 'Change Center' panel with 'View changes and restarts' and 'Release Configuration' buttons, and a 'Domain Structure' tree showing the hierarchy of the 'ohi_domain'.

Note In WLS Production-mode use the **Lock & Edit** button before clicking on the **New** button.

Step 3: Change the value of **Name** to **ohi-application-datasource** and enter **jdbc/claimsUserOhiApplicationDS** in **JNDI Name**. Click the **Next** button.

The screenshot shows the Oracle WebLogic Server Administration Console interface. The main content area displays the 'Create a New JDBC GridLink Data Source' wizard. The wizard is currently at the 'Transaction Options' step. The 'Name' field is filled with 'ohi-application-datasource'. The 'JNDI Name' field is filled with 'jdbc/c/claimsUserOhiApplicationDS'. The 'Database Type' is set to 'Oracle'. The 'Is this XA driver?' question is followed by an unchecked 'XA Driver' checkbox. The 'Next' button is highlighted with a mouse cursor. The left sidebar shows the 'Domain Structure' tree with 'ohi_domain' selected, and the 'System Status' section showing 'OK (11)'.

Step 4: In **Transaction Options** page, accept the default settings (Supports Global Transactions and One-Phase Commit) and click the **Next** button.

The screenshot displays the Oracle WebLogic Server Administration Console interface. The main content area is titled "Create a New JDBC GridLink Data Source". Below the title bar, there are navigation buttons: "Back", "Next" (highlighted by a mouse cursor), "Finish", and "Cancel".

The "Transaction Options" section is expanded, showing the following options:

- Supports Global Transactions**
Select this option if you want to enable non-XA JDBC connections from the data source to participate in global transactions using the Logging Last Resource (LLR) transaction optimization. Recommended in place of Emulate Two-Phase Commit.
- Logging Last Resource**
Select this option if you want to enable non-XA JDBC connections from the data source to emulate participation in global transactions using JTA. Select this option only if your application can tolerate heuristic conditions.
- Emulate Two-Phase Commit**
Select this option if you want to enable non-XA JDBC connections from the data source to participate in global transactions using the one-phase commit transaction processing. With this option, no other resources can participate in the global transaction.
- One-Phase Commit**

At the bottom of the wizard, there are navigation buttons: "Back", "Next" (highlighted by a mouse cursor), "Finish", and "Cancel".

On the left side of the console, there are several panels:

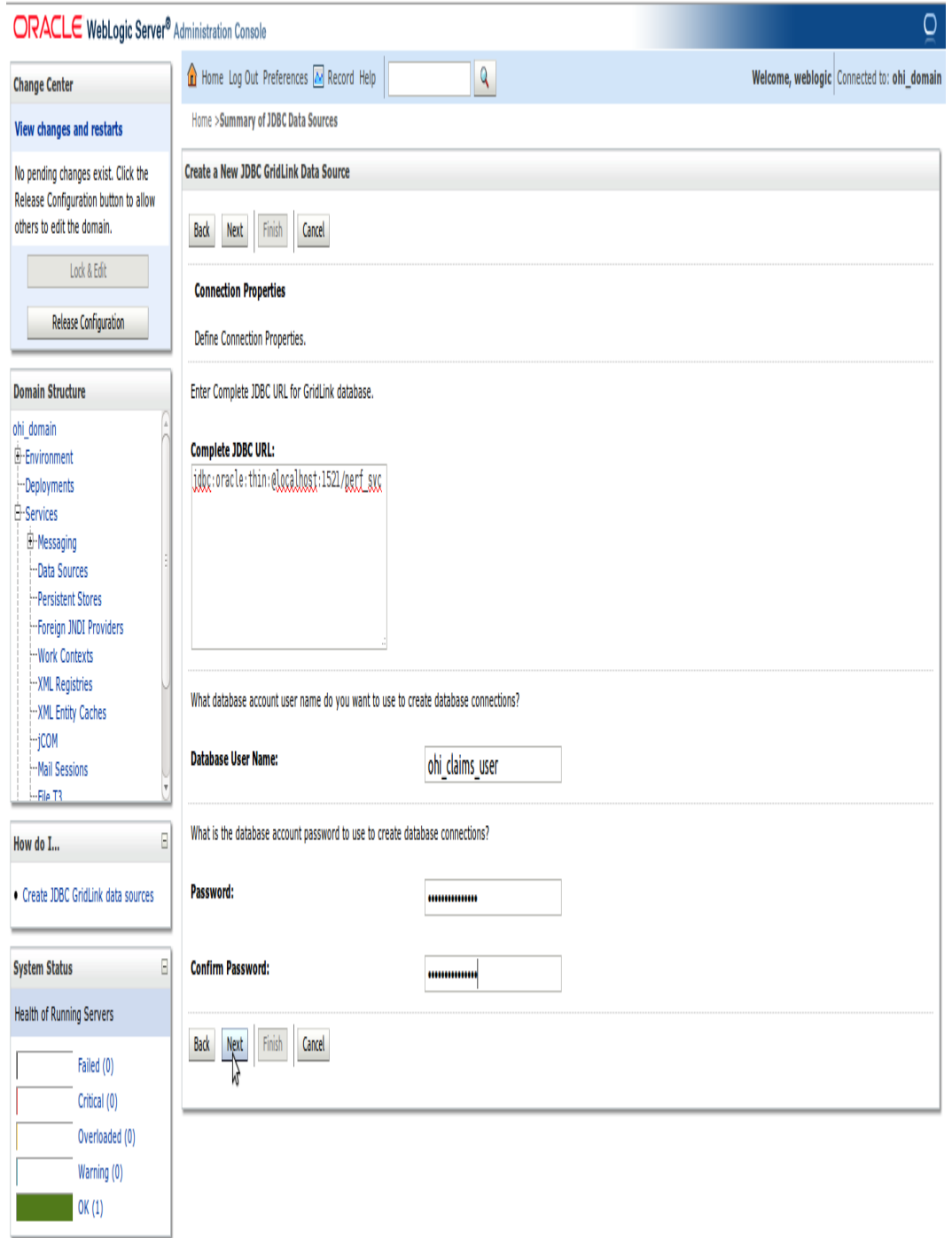
- Change Center:** Shows "View changes and restarts" and "No pending changes exist. Click the Release Configuration button to allow others to edit the domain." Buttons for "Lock & Edit" and "Release Configuration" are visible.
- Domain Structure:** A tree view showing the hierarchy: ohi_domain > Environment > Deployments > Services > Messaging > Data Sources.
- How do I...:** A list of tasks, with "Create JDBC GridLink data sources" selected.
- System Status:** Shows "Health of Running Servers" with "Failed (0)" and "Critical (0)" indicators.

Step 5: If SCAN (Single Client Access Name) is used for the Oracle RAC database, select the option **Enter complete JDBC URL**. Else, select the option **Enter individual listener information**.

The screenshot shows the Oracle WebLogic Server Administration Console interface. The main content area displays the 'Create a New JDBC GridLink Data Source' wizard. The 'Enter complete JDBC URL' option is selected, and the 'Next' button is highlighted with a mouse cursor. The left sidebar contains the 'Change Center', 'Domain Structure', 'How do I...', and 'System Status' panels. The 'System Status' panel shows the health of running servers, with 1 OK and 0 in other states.

Step 6: In Connection Properties page either

- enter the values of various fields as outlined in the table below if option **Enter complete JDBC URL** is selected:



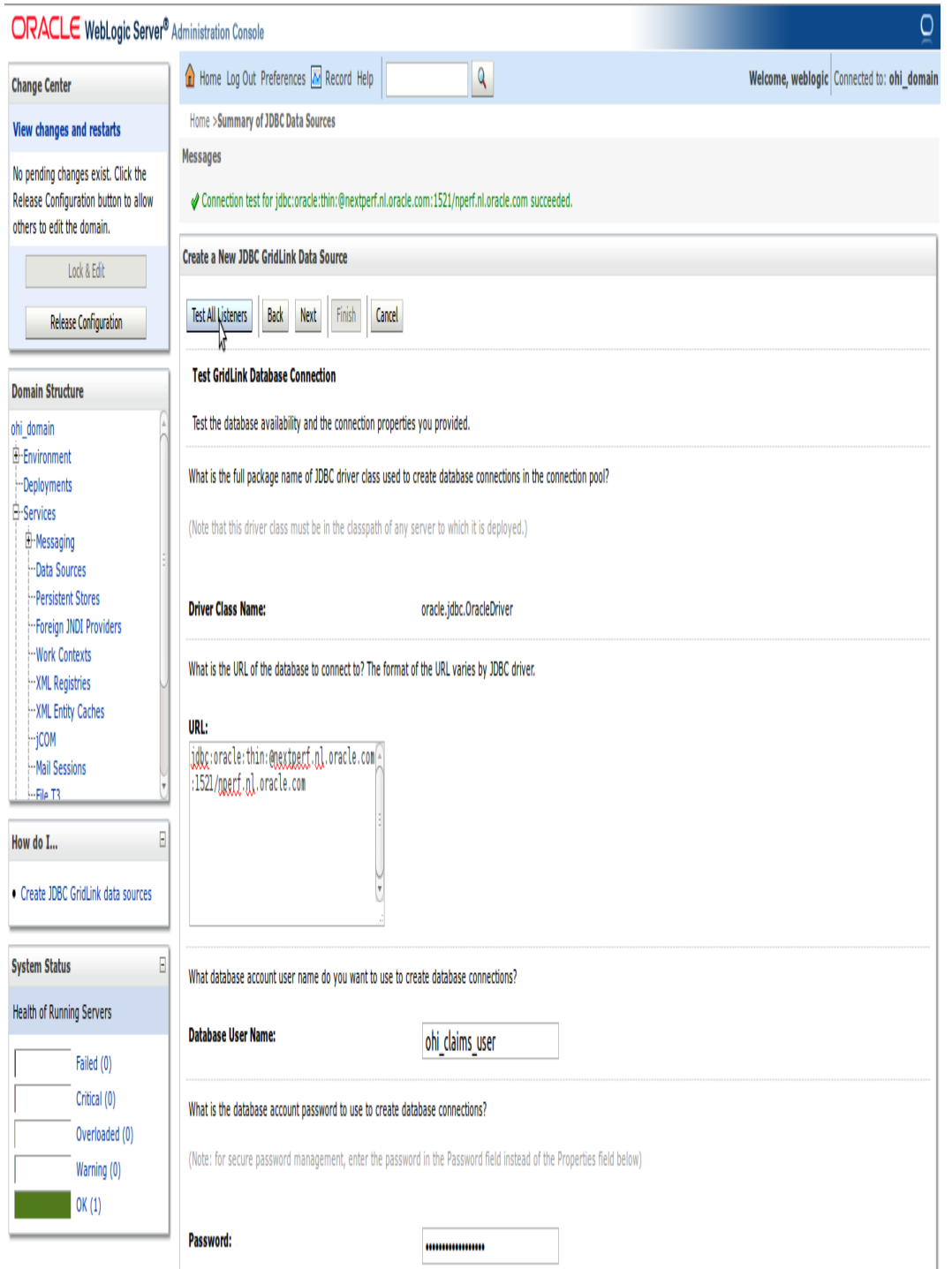
Parameters	Value	Explanation
Complete JDBC URL	jdbc:oracle:thin:@{scan-listener-host}:{scan-listener-port}/{service-name}	JDBC URL using SCAN
Database User Name	ohi_claims_user	Fixed value, do not change
Password & Confirm Password	Password of "ohi_claims_user"	The schema password as selected during the installation

- or enter the values of various fields as outlined in the table below if option **Enter individual listener information** is selected:

The screenshot shows the Oracle WebLogic Server Administration Console interface. The main window displays the 'Create a New JDBC GridLink Data Source' wizard. The wizard is currently at the 'Connection Properties' step. The 'Service Name' is set to 'nperf'. Under 'Host and Port', two entries have been added: 'nloz12.nl.oracle.com:1527' and 'nloz14.nl.oracle.com:1527'. The 'Database User Name' is set to 'ohi_claims_user'. The 'Password' and 'Confirm Password' fields are masked with dots. The left sidebar shows the 'Domain Structure' tree with 'ohi_domain' selected, and the 'System Status' section showing 'OK (1)' servers.

Parameters	Value	Explanation
Service Name		Oracle RAC service name
Host and Port	hostname1:port hostname2:port	Individual RAC node details. The format is <HOSTNAME>:<PORT >
Database User Name	ohi_claims_user	Fixed value, do not change
Password & Confirm Password	Password of "ohi_claims_user"	The schema password as selected during the installation

Step 7: In **Test GridLink Database Connection** page, click on **Test All Listeners** to see if the connection is successful. Once the test connection succeeds, click on **Next** button.



Step 8: Enter the details of ONS client configuration as outlined in the table below and click the **Next** button.

The screenshot shows the Oracle WebLogic Server Administration Console interface. The main content area is titled 'Create a New JDBC GridLink Data Source'. It includes a progress bar with 'Back', 'Next', 'Finish', and 'Cancel' buttons. The 'ONS Client Configuration' section is active, showing a checked 'FAN Enabled' checkbox. Below this, there is a text input field for 'ONS host and port' with an 'Add' button. A list box below the input field contains two entries: 'nloz12.nl.oracle.com:6200' and 'nloz14.nl.oracle.com:6200', with a 'Remove' button to the right. Further down, there are fields for 'ONS Wallet File Directory' and 'ONS Wallet Password', with a 'Confirm ONS Wallet Password' field below. The left sidebar shows the 'Domain Structure' tree with 'ohi_domain' selected, and the 'System Status' section showing 'Health of Running Servers' with 'OK (1)'.

Parameters	Value	Explanation
Fan Enabled	Check-box selected	Enables the data source to subscribe to and process Oracle FAN events. This attribute is only applicable for RAC configurations that publish FAN notification events using the ONS protocol.
ONS Nodes	Eg: hostname1:6200,hostname2:6200	A comma-separated list of ONS daemon listen addresses and ports to connect to for receiving ONS-based FAN events.

Parameters	Value	Explanation
ONS Wallet File	Location of ONS Wallet File (including the file name)	The location of the Oracle wallet file in which the SSL certificates are stored. Only required when the ONS client is configured to communicate with ONS daemons using SSL.
ONS Wallet Password & Confirm ONS Wallet Password	The wallet password	The wallet password attribute that is included as part of the ONS client configuration string. This attribute is only required when ONS is configured to use the SSL protocol.

Step 9: Click on **Test All ONS Nodes** to see if the connection is successful. Once the connection test succeeds, click the **Next** button.

ORACLE WebLogic Server® Administration Console

Home Log Out Preferences Record Help Welcome, weblogic Connected to: ohi_domain

Home > Summary of JDBC Data Sources

Messages

- Connection test for nloz12.nl.oracle.com:6200 succeeded.
- Connection test for nloz14.nl.oracle.com:6200 succeeded.

Create a New JDBC GridLink Data Source

Test ANS Nodes Back Next Finish Cancel

Test ONS client configuration

Test the ONS client configuration properties you provided

Check to enable the data source to subscribe to and process Oracle FAN events

FAN Enabled

Enter host and port of each ONS node separated by colon and click the add button.

ONS host and port:

nloz12.nl.oracle.com:6200, nloz14.nl.oracle.com:6200

Click the test button to test ONS node.

Test ONS Node nloz12.nl.oracle.com:6200

Test ONS Node nloz14.nl.oracle.com:6200

The location of the Oracle wallet file in which the SSL certificates are stored.

ONS Wallet File Directory:

The wallet password attribute that is included as part of the ONS client configuration string. This attribute is only required when ONS is configured to use the SSL protocol.

Change Center

View changes and restarts

No pending changes exist. Click the Release Configuration button to allow others to edit the domain.

Lock & Edit

Release Configuration

Domain Structure

- ohi_domain
 - Environment
 - Deployments
 - Services
 - Messaging
 - Data Sources
 - Persistent Stores
 - Foreign JNDI Providers
 - Work Contexts
 - XML Registries
 - XML Entity Caches
 - jCOM
 - Mail Sessions

File T3

How do I...

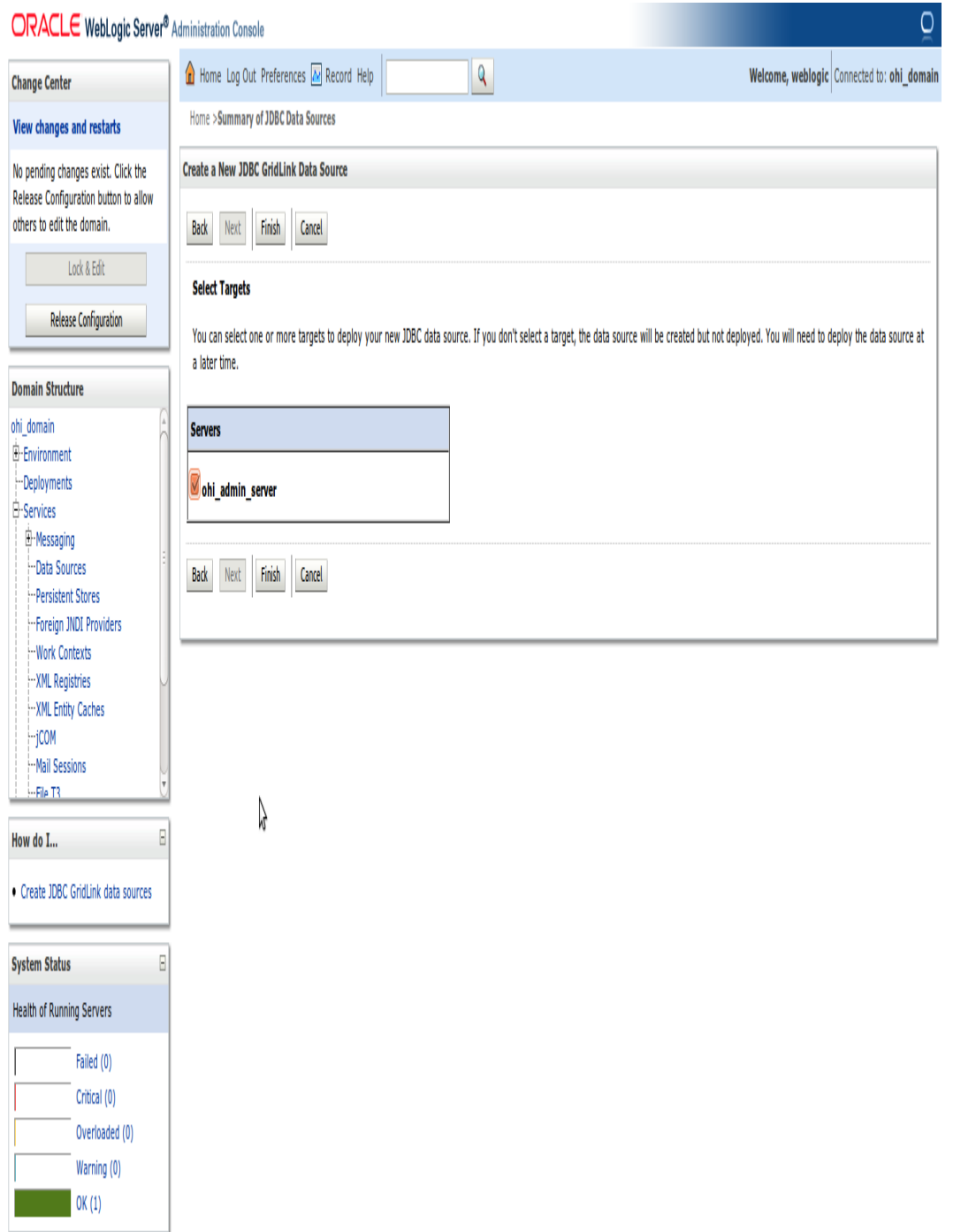
- Create JDBC GridLink data sources

System Status

Health of Running Servers

- Failed (0)
- Critical (0)
- Overloaded (0)
- Warning (0)
- OK (1)

Step 10: Select the Target(s) in the next page and click the **Finish** button.



Make sure to specify the managed server as target for the GridLink Data Source and change the connection pool settings by executing the following steps:

1. Select the newly created GridLink Data Source
2. Click on the tab **Connection Pool**
3. Expand the **Advanced** node at the bottom of the page to display all properties and set the following:

Property	Value
Initial Capacity	0

Property	Value
Test Connections On Reserve	Checked
Test Frequency	300
Connection Creation Retry Frequency	30
Seconds to Trust an Idle Pool Connection	10

Set the following driver property:

Property	Value
oracle.net.CONNECT_TIMEOUT	10000

INSTALLING THE UI CUSTOMIZATION LIBRARY THROUGH WLS ADMIN SERVER CONSOLE

To enable the creation of site-level UI Customizations, without having to change the OHI Application itself, an initially empty library called `custom.oracle.healthinsurance` needs to be installed before the OHI application can be installed.

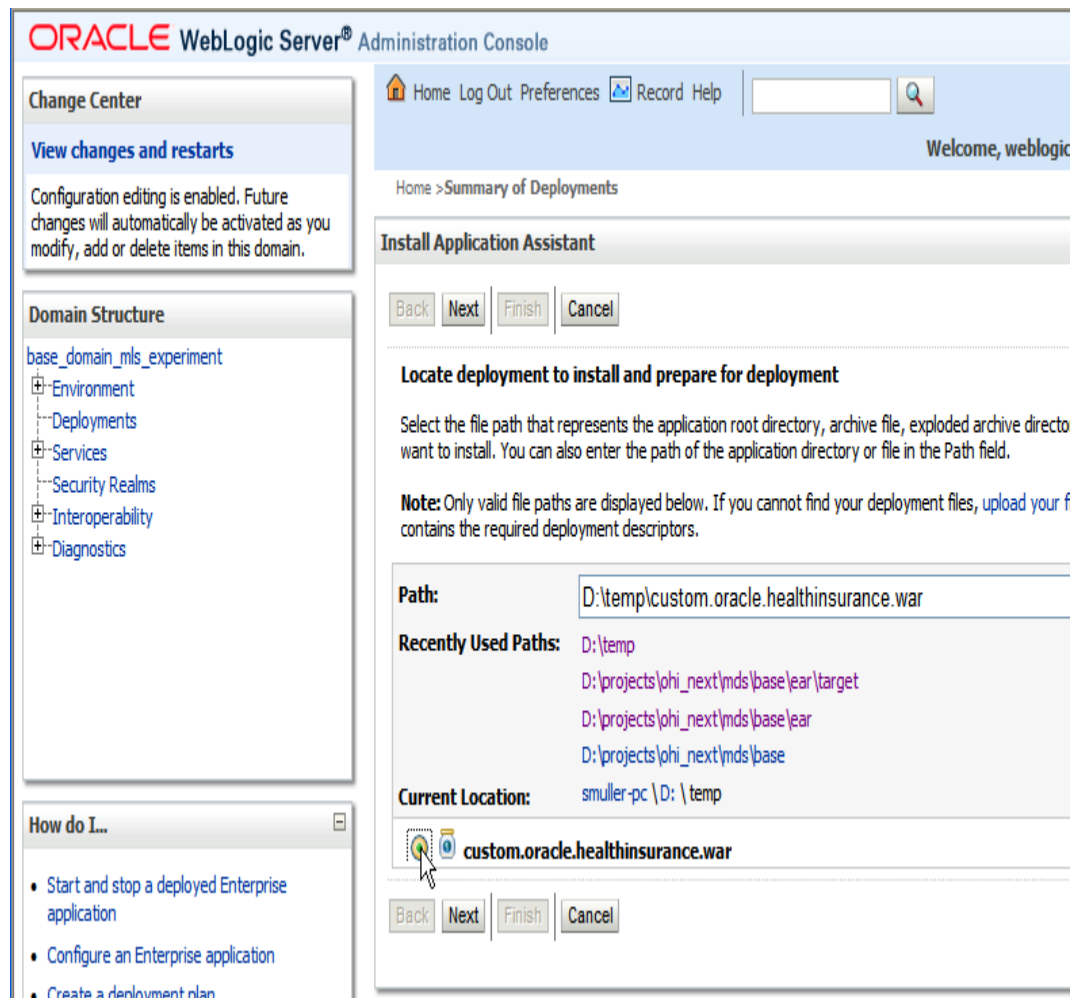
Step 1: Login to the Admin Server console (for example: `http://machine.domain:port/console`).

Step 2: Click the "**Deployment**" link and then click on the "**Install**" button as shown in the following screen shot. If the Install button is disabled, click the Lock & Edit button first (in the upper left section of the page).

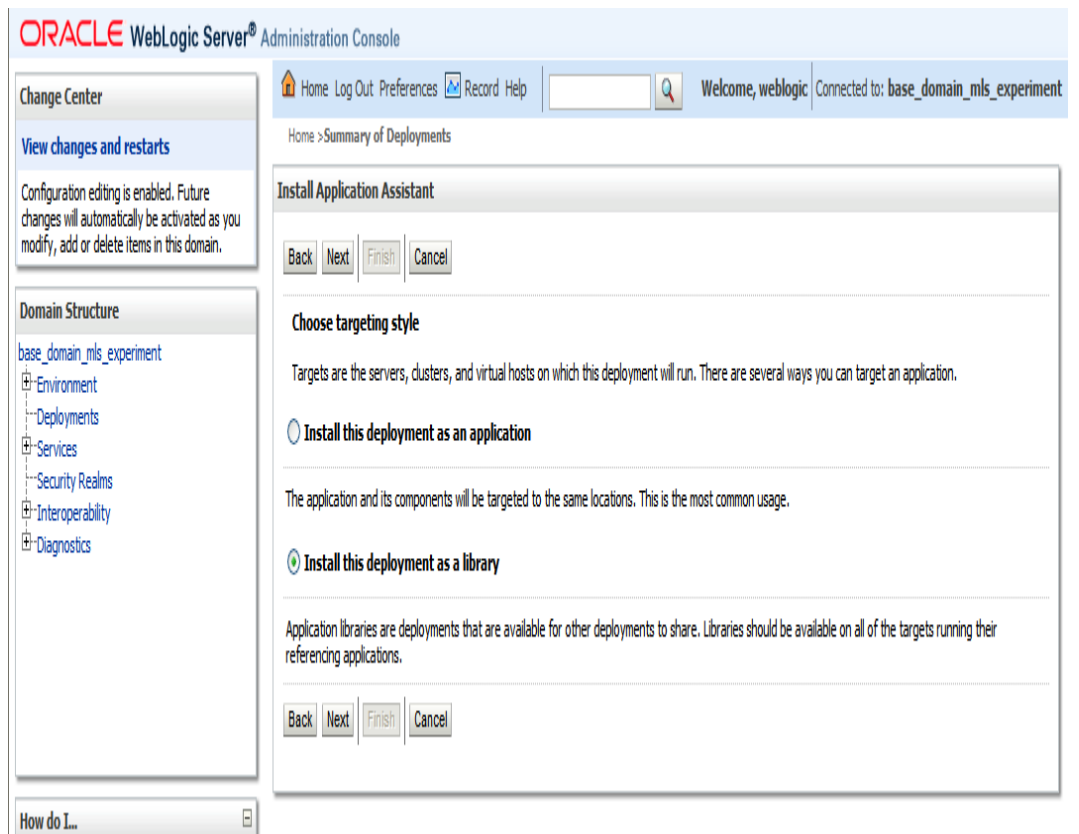
The screenshot shows the Oracle WebLogic Server Administration Console. The main content area is titled "Summary of Deployments" and contains a table of installed applications and modules. The table has the following columns: Name, State, Health, Type, and Deployment Order. The table contains five rows of data, all with a State of "Active" and Type of "Library". The "Install" button is visible at the bottom of the table.

Name	State	Health	Type	Deployment Order
adf.oracle.domain(1.0,11.1.1.1.0)	Active		Library	100
adf.oracle.domain.webapp(1.0,11.1.1.1.0)	Active		Library	100
adf(1.2,1.2.9.0)	Active		Library	100
adf(1.2,1.2.0.1)	Active		Library	100
oracle.jsp.mvst(11.1.1,11.1.1)	Active		Library	100

Step 3: Select the path where the library `custom.oracle.healthinsurance.war` file is located (`<OHI_ROOT>\lib`) and click the "**Next**" button as shown in the following sample screen shot:



Step 4: Select the option "Install this deployment as a library" and click on "Next" button as shown below:



Step 5: Ensure that the General - Name is set to **custom.oracle.healthinsurance** as shown below. This is the name the OHI Application refers to when loading the library, so this is the name under which it must have been installed. The version numbers may differ from what is shown in the screen shot below. The OHI Application will automatically load the highest version of all installed libraries with this deployment name. Then click on the "**Next**" button.

ORACLE WebLogic Server® Administration Console

Home Log Out Preferences Record Help

Welcome, weblogic Connected to: b

Home > Summary of Deployments

Install Application Assistant

Back Next Finish Cancel

Optional Settings

You can modify these settings or accept the defaults

General

What do you want to name this deployment?

Name:

Specification Version: 0.2

Implementation Version: 0.2.0.1

Security

What security model do you want to use with this application?

DD Only: Use only roles and policies that are defined in the deployment descriptors.

Custom Roles: Use roles that are defined in the Administration Console; use policies that are defined in the deployment descriptor.

Custom Roles and Policies: Use only roles and policies that are defined in the Administration Console.

Advanced: Use a custom model that you have configured on the realm's configuration page.

Source accessibility

How should the source files be made accessible?

Use the defaults defined by the deployment's targets

Change Center

View changes and restarts

Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

Domain Structure

base_domain_mls_experiment

- Environment
- Deployments
- Services
- Security Realms
- Interoperability
- Diagnostics

How do I...

- Start and stop a deployed Enterprise application
- Configure an Enterprise application
- Create a deployment plan
- Target an Enterprise application to a server
- Test the modules in an Enterprise application

System Status

Health of Running Servers

- Failed (0)
- Critical (0)
- Overloaded (0)

Step 6: Click on **"Finish"** button. You should see a success message as shown below. The library is now installed. Note: if you had to click Lock & Edit in step 1, you now have to click Activate Changes (just below the Lock & Edit).

ORACLE WebLogic Server® Administration Console

Home Log Out Preferences Record Help

Welcome, weblogic Connected to: base_domain_mls_experiment

Home > Summary of Deployments

Messages

- ✓ All changes have been activated. No restarts are necessary.
- ✓ The deployment has been successfully installed.

Summary of Deployments

Control Monitoring

This page displays a list of Java EE applications and stand-alone application modules that have been installed to this domain. Installed applications and modules can be started, stopped, updated (redeployed), or deleted from the domain by first selecting the application name and using the controls on this page.

To install a new application or module for deployment to targets in this domain, click the Install button.

[Customize this table](#)

Deployments

Install Update Delete Start Stop

Showing 1 to 10 of 20 Previous Next

Name	State	Health	Type	Deployment Order
adf.oracle.domain(1.0,11.1.1.2.0)	Active		Library	100
adf.oracle.domain.webapp(1.0,11.1.1.2.0)	Active		Library	100
custom.oracle.healthinsurance(0.2,0.2.0.1)	Active		Library	100
DMS Application (11.1.1.1.0)	Active	OK	Web Application	5

The following section describes the installation of the OHI application.

INSTALLING THE OHI APPLICATION THROUGH WLS ADMIN SERVER CONSOLE

The OHI applications are delivered in a so called Java Enterprise Archive (EAR) which will be installed through the WLS Admin Server Console. In order to do that, perform the following steps.

Step 1: Login to the Admin Server console (for example: <http://machine.domain:port/console>).

Step 2: Click the "**Deployment**" link and then click on the "**Install**" button as shown in the following screen shot:

The screenshot shows the Oracle WebLogic Server Administration Console interface. The main content area is titled "Summary of Deployments" and includes a "Control" tab. Below the tab, there is a table of deployed applications. The "Install" button in the table's control bar is highlighted with a mouse cursor.

Summary of Deployments

This page displays a list of Java EE applications and stand-alone application modules that have been installed to this domain. Installed applications and modules can be started, stopped, updated (redeployed), or deleted from the domain by first selecting the application name and using the controls on this page.

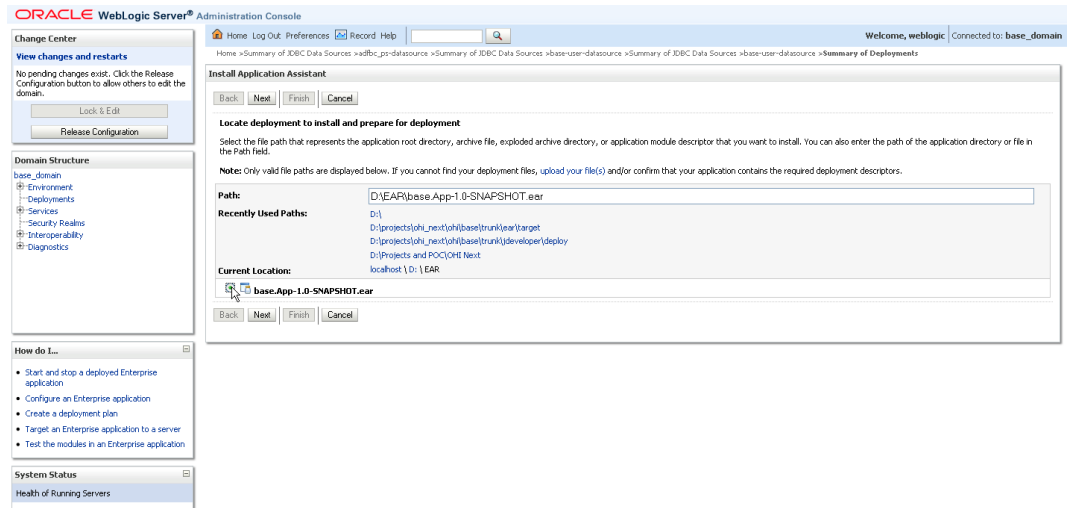
To install a new application or module for deployment to targets in this domain, click the Install button.

Deployments

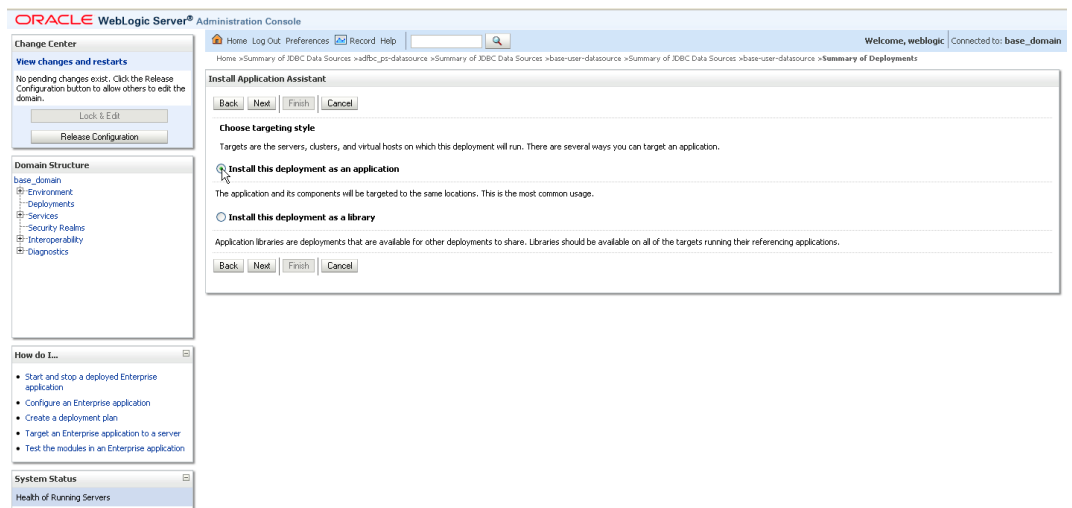
Showing 1 to 10 of 20 Previous | Next

<input type="checkbox"/>	Name	State	Health	Type	Deployment Order
<input type="checkbox"/>	adf.oracle.domain(1.0,11.1.1.2.0)	Active		Library	100
<input type="checkbox"/>	adf.oracle.domain.webapp(1.0,11.1.1.2.0)	Active		Library	100
<input type="checkbox"/>	custom.oracle.healthinsurance(0.2,0.2.0.1)	Active		Library	100
<input type="checkbox"/>	DMS Application (11.1.1.1.0)	Active	OK	Web Application	5
<input type="checkbox"/>	FMW Welcome Page Application (11.1.0.0.0)	Active	OK	Enterprise Application	5
<input type="checkbox"/>	jsf(1.2,1.2.9.0)	Active		Library	100
<input type="checkbox"/>	jstl(1.2,1.2.0.1)	Active		Library	100
<input type="checkbox"/>	ohw-rcf(5,5.0)	Active		Library	100
<input type="checkbox"/>	ohw-uix(5,5.0)	Active		Library	100

Step 3: Select the path where the EAR file is located and click the "Next" button as shown in the following screen shot:



Step 4: Select the option "Install this deployment as an application" and click on "Next" button as shown below:



Step 5: Click on "Finish" button. The OHI Application is now installed.

Step 6: If you are deploying the application to cluster, in "Select deployment targets" page, select the Clusters target as shown below:

The screenshot shows the Oracle WebLogic Server Administration Console interface. The main window displays the 'Install Application Assistant' wizard for the application 'OHI_BackOffice_build'. The wizard is currently at the 'Select deployment targets' step. The 'Available targets for OHI_BackOffice_build' section shows two categories: 'Servers' and 'Clusters'. Under 'Servers', the 'AdminServer' is listed with an unchecked checkbox. Under 'Clusters', the 'nperf' cluster is selected with a checked checkbox. Below 'nperf', there are three options: 'All servers in the cluster' (selected with a radio button), 'Part of the cluster' (unchecked), and three individual servers: 'mloz16', 'mloz17', and 'mloz13', each with an unchecked checkbox. The left sidebar contains several panels: 'Change Center' (with 'View changes and restarts' and 'Release Configuration' buttons), 'Domain Structure' (showing a tree view of the domain), 'How do I...' (with a list of tasks), and 'System Status' (with 'Health of Running Servers'). The top navigation bar includes 'Home', 'Log Out', 'Preferences', 'Record', and 'Help'.

INSTALLING OHI CLAIMS APPLICATION TO USE CUSTOM WORK MANAGER (OR) TO CHANGE CONTEXT-ROOT

If you want to,

- Make use of the work managers **ui-work-manager** & **ws-work-manager** created in section "**Creating WebLogic Work Manager**"
- Change the default context-root of OHI Claims web application and web service

Then, DO NOT follow the steps mentioned in "**Installing The OHI Application Through WLS Admin Server Console**", but follow the steps mentioned below:

Step 1: The EAR by default contains the following context-root:

- UI.war - **base**
- OHI-WEB-SERVICES.war - **ohi-web-services**

Step 2: Edit the values of the variables **UI_CONTEXT_ROOT** and **WS_CONTEXT_ROOT** in `<OHI_ROOT>/application/plan/Plan.xml` to suit your requirements.

Step 3: The EAR and Plan.xml (deployment plan) are packaged under a directory named "**application**" in the release bundle (See the directory structure below). It is recommended to copy the "**application**" directory to a location (this directory will be referred as *<INSTALL-ROOT>* hereafter) and optionally rename the directory (for example, rename to OHIClaims).

Appendices

APPENDIX A - WEB SERVICE LOCATIONS

The URLs of the OHI Web Services depend on the name of the machine on which the application installed (or alternatively a load balancer).

The BASE is `http://machine.domain:port/ohi-web-services`

Web Service	WSDL	Endpoint
Authorizations	BASE/AuthorizationService?wsdl	BASE/AuthorizationService
Claims In	BASE/ClaimsInService?wsdl	BASE/ClaimsInService
Claims Pre Finalized Out	BASE/ClaimPreFinalizedOutService?wsdl	BASE/ClaimPreFinalizedOutService
Claims Reprocessing	BASE/ClaimsReprocessingService?wsdl	BASE/ClaimsReprocessingService
Claims Transactions	BASE/ClaimTransactionService?wsdl	BASE/ClaimTransactionService
Claims Update	BASE/ClaimsUpdateService?wsdl	BASE/ClaimsUpdateService
Counters	BASE/CountersService?wsdl	BASE/CountersService
Data Access Group Import	BASE/DataAccessGroupService?wsdl	BASE/DataAccessGroupService
Enrollment Response	BASE/EnrollmentResponseService?wsdl	BASE/EnrollmentResponseService
File Import	BASE/FileImportService?wsdl	BASE/FileImportService
Financial Messages	BASE/FinancialMessageService?wsdl	BASE/FinancialMessageService
Payment Status Response	BASE/PaymentStatusResponseService?wsdl	BASE/PaymentStatusResponseService
Providers	BASE/ProviderImportService?wsdl	BASE/ProviderImportService
Provisioning	BASE/ProvisioningService?wsdl	BASE/ProvisioningService
Relations	BASE/RelationImportService?wsdl	BASE/RelationImportService
ExternalClaimsData	BASE/ExternalClaimsDataService?wsdl	BASE/ExternalClaimsDataImportService
Medical Case	BASE/MedicalCaseService?wsdl	BASE/MedicalCaseService

Note Please ensure that the exchange of messages with OHI Web Services is properly secured before these are used.

OHI calls out to external Web Services in a number of cases. Endpoints for the external or outbound Web Services are defined in the [OHI Claims properties file](#)¹.

APPENDIX B - SEED DATA

TYPES OF SEED DATA

Generic Seed Data

Seed data is maintained by Oracle. It is delivered as part of a release and may be updated by software upgrades. In general, customers should not change this data (see the Restrictions on using Seed Data section below).

Localization Seed Data

This category covers specific data that is required by localizations. The data is maintained by Oracle. Examples:

- Flex code definitions for a specific page
- Specific messages for localizations.

Sample Data

Sample data is provided by Oracle to give you a headstart during configuration. You can opt to install this data. It is *not* modified during future upgrades. Tables containing Sample Data include:

Table Name	Remarks
To be determined	

RESTRICTIONS ON USING SEED DATA

Because Seed Data is maintained by Oracle, it may be modified or even deleted as part of an upgrade. Customers should therefore exercise caution when using seed data in their configuration by abiding these rules.

1. Do not remove (delete) Seed Data rows. A patch may re-insert the row.
2. Do not update columns, other than those indicated as updateable below.
3. Do not make references to rows that may be deleted by Oracle (see table below).

Violations of the rules above (especially rule 3) may lead to failures during the installation of upgrades.

The table below lists the Seed Data tables.

- **Data:** The table or logical entity
- **Updatable columns:** The customer may update the values in these columns. They will not be overwritten by upgrade scripts. Other columns should not be updated by the customer.
- **Physical Delete:** Upgrade scripts may delete this data. The customer should not create references to this data. Example: Do not use OHI messages for your own dynamic checks.

1. <http://nloz13.nl.oracle.com:8888/OHI-Main/4990-DSY>

Data	Updatable columns	Physical Delete	Remarks
Access Restrictions		Yes	Also deletes Access Restriction Grants referring to this row
Access Restriction Grants		Yes	
Access Roles		No	Two roles are seeded
Boilerplate Texts		Yes	
Claim Forms		No	
Countries	all _b columns	No	
Country Regions		No	
Coverage Labels		No	
Dynamic Field Usages		No	
Dynamic Logic		No	
Fields (+ dynamic logic)		No	
Flex Codes		No	
Flex Code Sets (+ details)		No	
Flex Code Systems		No	
Languages	ind_default ind_installed	No	
Messages	ind_suppress_log_in_ui ind_suppress_log_in_ext ind_mark external_code	Yes	
Single Flex Code Definitions (+ usage)		No	
Task Types		Yes	Customer is not allowed to change anything in base table
Task Type Attributes	value_char value_number value_datetime value_clob	Yes	
Users		No	One User will be seeded (system user)

