



**SAGE Computing Services**

---

---

# **Oracle JDeveloper 11gR1**

## **ADF Faces Rich Client Workshop – Deploying to WebLogic Server**

---

**0419 904 458  
[www.sagecomputing.com.au](http://www.sagecomputing.com.au)**

Oracle JDeveloper 11gR1 ADF Faces Rich Client Workshop - – Deploying to WebLogic Server

Edition 11gR1 ADF Faces RC v1.3 September 2009

© Copyright SAGE Computing Services 2006 - 2009

SAGE Computing Services believes the information in this document is accurate as of its publication date. The information is subject to change without notice.

SAGE Computing Services assumes no responsibility for any errors that may appear in this document.

Apache, Ant, MyFaces, Struts and Tomcat are registered trademarks of the Apache Software Foundation.

Apple QuickTime is a registered trademark of Apple Computer Inc.

JDeveloper, JPublisher, Oracle, Oracle 9, Oracle 10g, Oracle 11g, Oracle Application Server, WebLogic Server, Oracle Internet Directory and Toplink are registered trademarks of Oracle Corporation.

Microsoft, Microsoft Excel, Microsoft Internet Explorer, Microsoft Windows, Microsoft Word, ODBC, SQLServer, MS Active Directory and Windows Media Player are registered trademarks of Microsoft Corporation.

Enterprise Java Beans, JEE, JME, JSE, J2EE, J2ME, J2SE, Java, JavaBeans, Javadoc, JavaServer Faces, JavaServer Pages, JDBC, JSP, JSF, JSR, JVM, Sun and Sun Java are registered trademarks of Sun Microsystems.

CVS is a registered trademark of the Free Software Foundation.

JBoss is a registered trademark of JBoss Inc.

MySQL is a registered trademark of MySQL AB.

WinZip is a registered trademark of WinZip International LLC.

Real Player is a registered trademark of RealNetworks Inc.

Google.com is a registered trademark of Google.

All trade names referenced are the service mark, trademark, or registered trademark of the respective manufacturer.

<b>DEPLOYING AN ADF APPLICATION .....</b>	<b>5</b>
ORACLE'S PREFERRED JEE PLATFORM - WEBLOGIC .....	7
WEBLOGIC INSTALLATION & PREPARATION.....	8
<i>WebLogic Concepts – Domains / Servers / Machines / Clusters</i> .....	9
<i>WebLogic Package Installation</i> .....	10
<i>Installing the ADF Runtime Libraries</i> .....	12
<i>Creating an ADF ready Domain and Managed Server</i> .....	14
CONNECTING JDEVELOPER TO YOUR APPLICATION SERVER.....	26
<i>WebLogic JDBC Data Sources</i> .....	29
DEPLOYMENT .....	34
<i>JARs, WARs, EARs and JDeveloper</i> .....	34
Deployment Profiles.....	34
<i>JDeveloper Auto Deployment</i> .....	35
TESTING YOUR APPLICATION.....	37

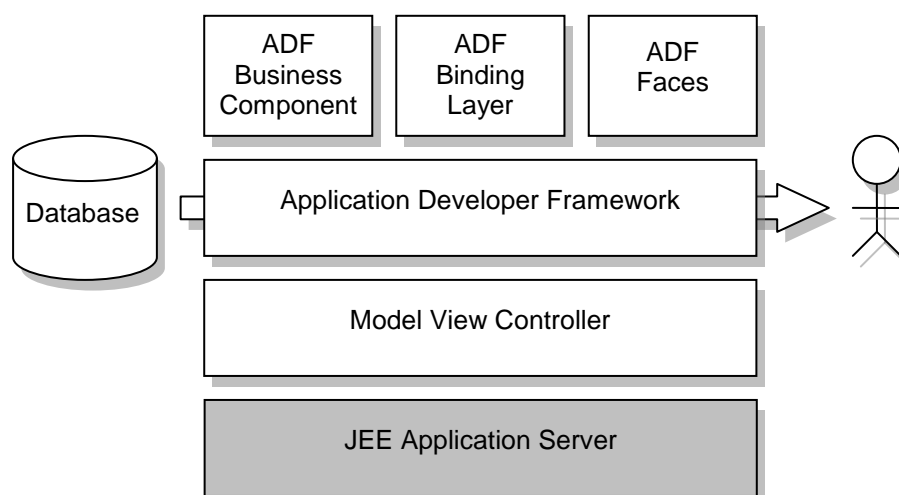


# **Deploying an ADF Application**

# Introduction

In this unit we will be looking at the following:

- Deploying an ADF Application
- Weblogic – The New Preferred Deployment Platform
- Weblogic Concepts – Domains / Servers / Machines
- Weblogic Package Installation
- Weblogic ADF Preparation – ADF / JDev Runtimes
- Creating an ADF ready Domain and Managed Server
- JDev Weblogic Connection
- Weblogic JDBC Data Sources
- JARs, WARs, EARs and JDeveloper
- JDev Auto Deployment
- Testing your Application



## Deploying an ADF Application

While running an ADF application under JDeveloper makes sense for development and initial testing, the need to deploy your application to QA/Production level environments where your QA Team and eventually your users can access your application will be required.

For this unit we assume you have the following software packages available:

- Oracle JDeveloper 11gR1 (Build 5407) => jdevstudio11111install.exe
- Oracle WebLogic Server 10.3.1 (Package Installer) => oep11\_wls1031.exe

The unit is based on the Windows platform.

## Oracle's Preferred JEE Platform - WebLogic

Oracle JDeveloper 11g ADF applications based on JEE 1.5, are by designed like other JEE applications to run within a JEE container. Oracle supplies such a container via WebLogic Server (WLS). ADF applications can also be deployed to other JEE compliant servers such as Apache Tomcat or JBoss with the proviso that the server is JEE 1.5 capable.

WLS is now the preferred deployment platform for JDeveloper ADF applications. This begs the question where did OC4j go?

Could it simply be the fact that ADF 11g applications require a JEE 1.5 capable server and WLS fits the bill, whereas even the latest OAS release, Oracle iAS 10.1.3, only supports JEE 1.4?

While the JEE 1.5 requirement is fundamental, there may be quite a bit more to it. Following the BEA acquisition, WLS has become the new strategic direction that Oracle is taking for its Fusion Middleware (FMW) application server offering, so it stands to reason that it is the preferred deployment platform for the premier Java development suite, JDeveloper.

Hot on the heels of the BEA acquisition, on July 1 2008 Oracle conducted the Fusion Middleware Strategy Briefing, which is still available in podcast form on Oracle.com. In that briefing it was indicated that Oracle's future application server offering would be born from "BEA WebLogic Server, a Strategic Java EE 5.0 Application Server" and that features from OC4J would be integrated into the WebLogic Server.

To back that up with current marketing statements the following quote is taken directly from the Fusion Middleware application server homepage on Oracle.com:

"Oracle WebLogic Suite and Oracle WebLogic Application Grid are Oracle Fusion Middleware's strategic application server products. Learn more about these products and their role in Oracle's middleware strategy. Oracle Application Server customers will benefit from ongoing development and Oracle's commitment to integrate its best capabilities with Oracle WebLogic Server over time."

In line with WebLogic being the preferred ADF Fusion application deployment platform, JDeveloper makes use of an integrated WLS in testing and running your JEE application during development from within the JDeveloper Suite. Alternatively you can make use of the integrated Server as a standalone server in deploying your application for testing purposes.

For QA/Production purposes however, you will use a full package installation of WLS that is separate to JDeveloper.

## WebLogic Installation & Preparation

The remainder of this chapter primarily comprises a walkthrough of the steps required to install and prepare WebLogic and then connect JDeveloper to WebLogic and deploy an ADF Application.

The practical steps outlined here largely represent a reviewed and expanded version of a Blog post by Duncan Mills'; "A Rough Guide To Installing and Setting up WebLogic 10.3 Production for Running ADF Applications". All credit for the basic principles and steps go to Duncan. The post can be found at:

[http://groundside.com/blog/DuncanMills.php?title=a\\_rough\\_guide\\_to](http://groundside.com/blog/DuncanMills.php?title=a_rough_guide_to)

***Please note this exercise is aimed at providing familiarity with the concepts and major steps involved in the process. This chapter is not meant to be a production guide. There are various facets that would be required for a Production ADF hosting environment that are abbreviated and/or skipped altogether. In addition individual site and application requirements will vary.***

### ***Virus/Firewall issues***

If you are going to follow this exercise on a personal machine such as laptop or PC, it may be advisable to turn off your Virus Scanner and any aggressive firewall software to avoid installation errors or failures. At the very least if your Virus suite monitors HTTP traffic it is suggested to disable the feature while installing and running WebLogic. For a true Production install this would likely just require liaison with sys admins to configure port rules and such, as is the norm for running any Oracle web enabled products.



## ***WebLogic Concepts – Domains / Servers / Machines / Clusters***

Key WebLogic Server building blocks include Domains, Admin and Managed Servers, Machines and Clusters.

### ***Domain***

A Domain is the core architectural unit of a WebLogic Server environment. It comprises a single Administration Server, usually one or more Managed Servers, Machines and optionally Clusters.

### ***Servers***

#### ***Administration Server***

The key controlling entity of a Domain is a special Server called the Administration Server (or Admin Server), that has the primary purpose of providing Domain configuration and management.

#### ***Managed Server/s***

One or more Managed Servers can be hosted by the Domain and these provide the platform and resources to host your JEE applications.

Although the Admin Server can also play the part of a Managed Server, particularly in a production environment, it is generally preferable to restrict the Admin Server to configuration and management purposes only.

### ***Machines***

A Machine is a WebLogic definition that maps to real physical hardware. Machines associate Servers to physical hosts and may be used in Highly Available (HA) environments to facilitate server restarts and optimisation of replicated session state data.

### ***Clusters***

Finally Clusters provide the means to build Highly Available environments with increased scalability and reliability. A Cluster is a logical grouping of Managed Servers that are managed and run simultaneously. While Clusters may reside and run from a single Machine, or from multiple Machines, Clusters will appear as a single Managed Server to clients.

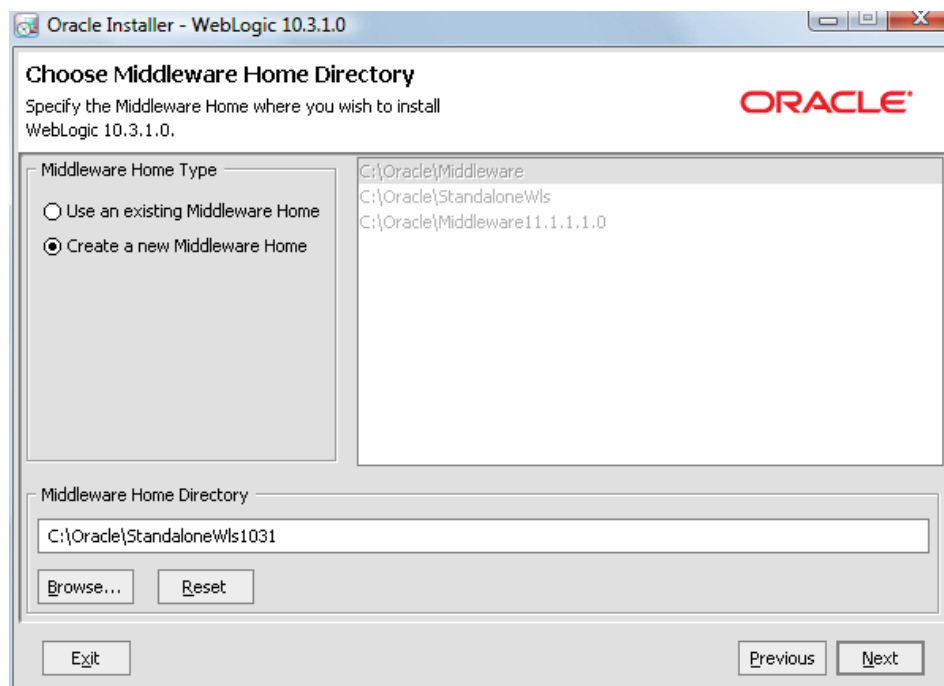
## WebLogic Package Installation

The following describes the steps to install a standalone WLS server.

There is a NET installer and a PACKAGE installer available. The steps described here are for the PACKAGE installer.

### Installation Steps:

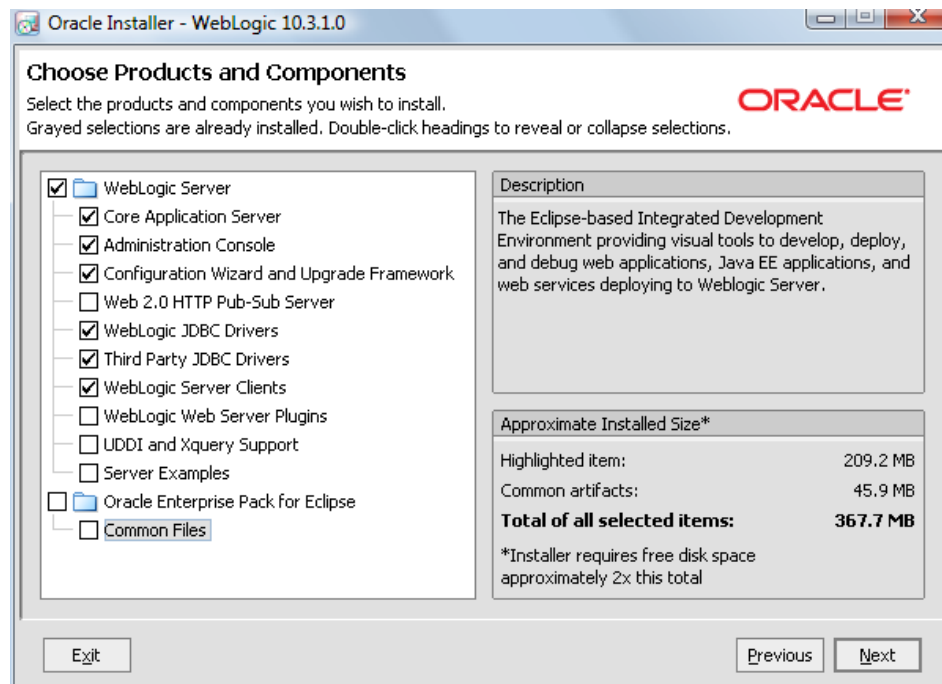
1. Run the WLS installer – `oepe11_wls1031.exe`
2. Welcome page - Click Next.
3. Choose BEA Home Directory page - select the Create a new BEA Home option, and enter an appropriate path & directory in the BEA Home Directory field. Click Next.



We have chosen `c:\Oracle\StandaloneWls1031`. Note that there are no spaces in the directory names.

4. Register for Security Updates page – uncheck the updates option and Click Next.
5. Choose Install Type page - select Custom so that we get the Products and Components page. Click Next.
6. Choose Product and Components page - unselect the following options:
  - Web 2.0 HTTP Pub-Sub Server
  - WebLogic Web Server Plugins\*
  - UDDI and Xquery Support
  - Server Examples
  - Common Files

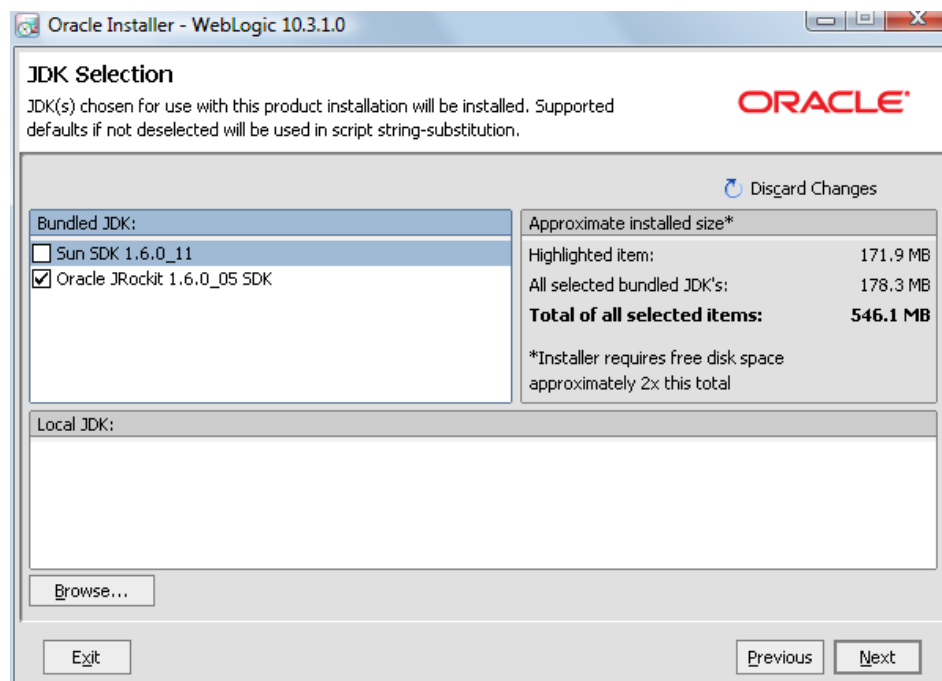
and then Click Next.



The components chosen represent the minimum typical options required for ADF Application hosting.

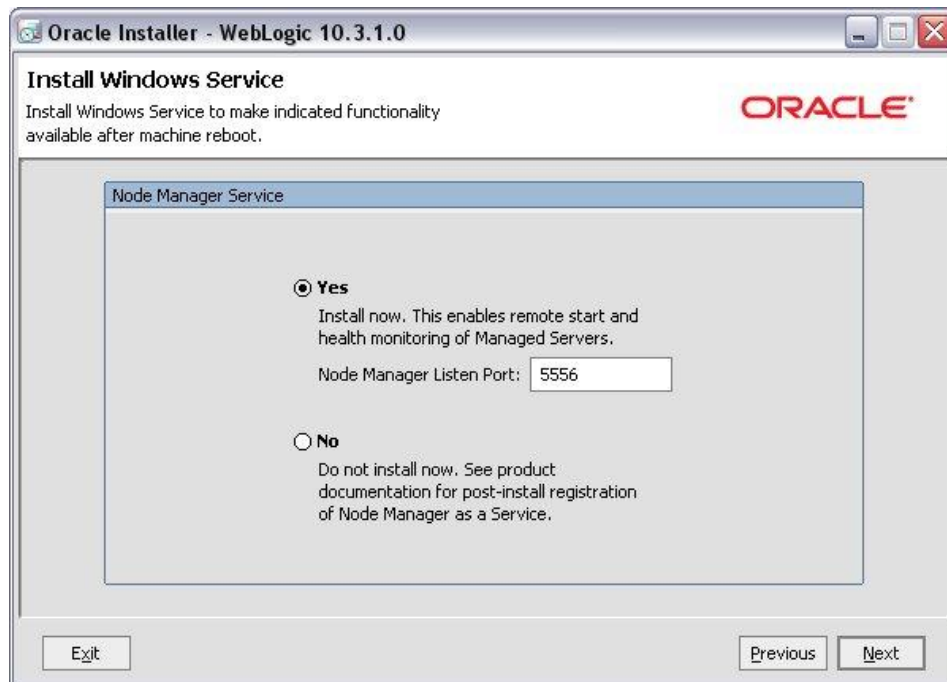
The WebLogic Web Server Plugins option may be required if you plan on implementing clustering.

7. JDK Selection page - select one of the JDKs. By default we have picked the JRockit option. Click Next.



8. Choose Product Installation Directories page - leave the defaults and Click Next.

9. Install Windows Service page – select Yes, to install Node Manager on an available port. Click Next.



Node Manager is primarily required for HA environments including Clustering. Node Manager is also required if you wish to have remote control of your servers via the WLS Admin Console.

*There is a classpath issue with Node Manager and ADF enabled Domains & Servers which will cause deployment errors and failures. We will address and workaround this problem later in the process. Duncan Mill's mentions this issue in his post and suggests to log an SR with Oracle Support in the case of Production configurations.*

10. Choose Shortcut Location page – leave defaults and click Next.

11. Installation Summary page – review your choices and click Next.

The installer will now install the software as specified. Once complete unselect the Run Quickstart option as we need to perform further installations/configuration and do not want to run the Quickstart Wizard or tutorials. Click Done.

## ***Installing the ADF Runtime Libraries***

In order for a WebLogic server installation to support ADF applications, the ADF Runtime libraries must be installed to the WebLogic home. Currently there is no standalone ADF Runtime installer available for the 11g ADF Runtimes. The options are:

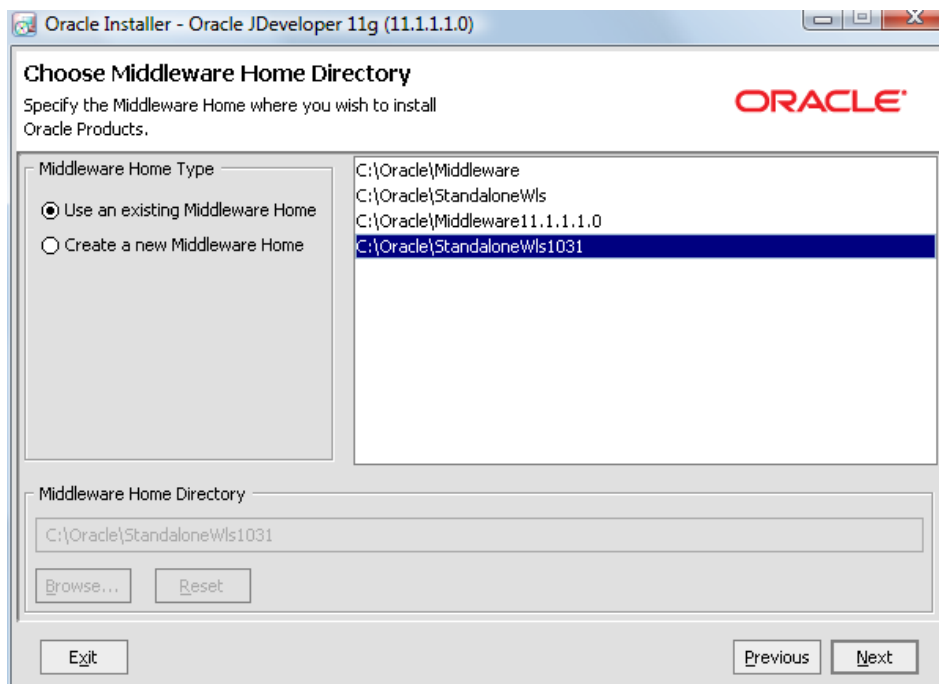
- Fusion Middleware 11g Application Developer installer  
or
- JDeveloper Studio installer (at least version 11.1.110)

The Fusion Middleware installer includes and will install Enterprise Manager. If you do not require Enterprise Manager then you should use the JDeveloper Studio installer against the new WebLogic Server package home to make the WebLogic Server home ADF capable. Here we will describe the steps using the JDeveloper Studio installer.

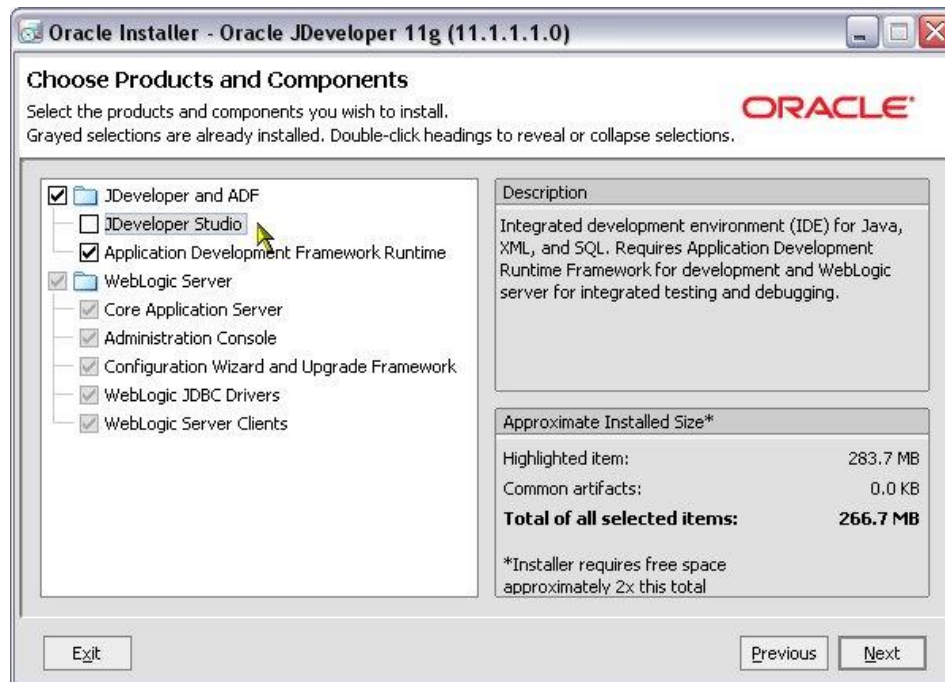
Note that previous JDeveloper installers will not work correctly; specifically the 11.1.10 installer required the “full” JDeveloper software in addition to the ADF runtimes to be installed to WebLogic. Using a previous version installer with the steps described here will result in ADF deployment/runtime errors and failures.

### Installation Steps:

1. Run the JDeveloper 11g Studio installer – `jdevstudio11111install.exe`
2. Welcome page - following unpacking, the installer dialog will display starting with the Welcome page. Click Next.
3. Choose Middleware Home Directory page – select the Use an existing Middleware Home option and select the directory where you previously installed WLS (e.g. `c:\Oracle\StandaloneWls1031`). Click Next.



4. Choose Products and Components page – with the 11.1110 installer (or better) we only need to select the Application Development Framework Runtime option. Click Next.



5. For the rest of the wizard accept the defaults through to the Installation Summary page. Click Next to execute the installation.
6. Installation Complete page - once completed you'll see this page. Again unselect the Run Quickstart and then click Done.

Note for the JDK Selection page – JDK selection is for the JDeveloper IDE/ADF only. We installed WLS with JRocket, and this selection will not impact on that, in which case we can leave the option as the default JDeveloper Sun JDK.

## ***Creating an ADF ready Domain and Managed Server***

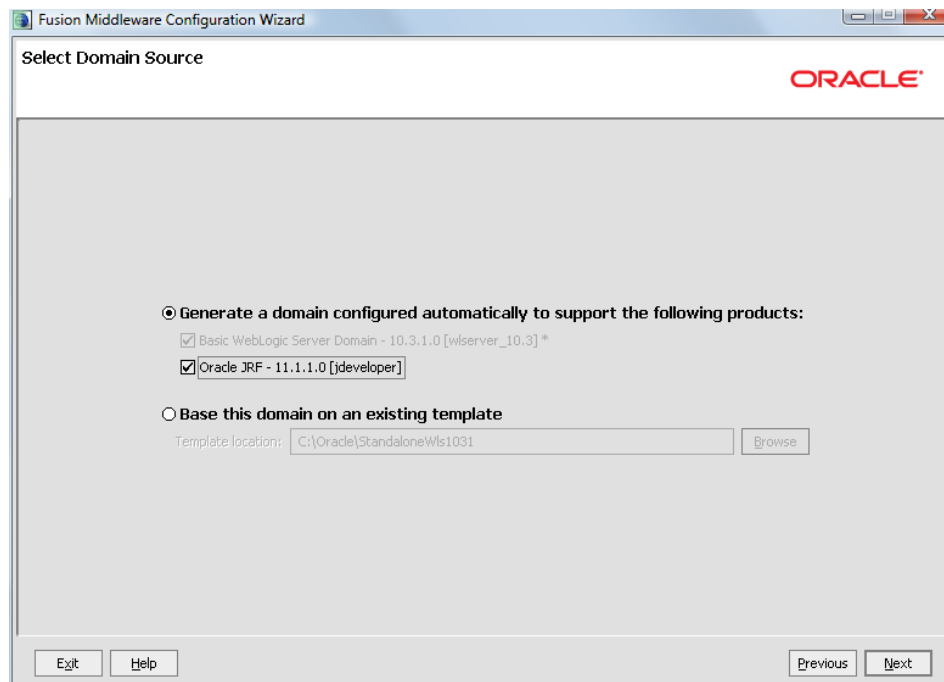
Now that our WebLogic home is ADF capable we need to create an ADF capable Domain including default Admin Server and a Managed Server to host our application. The new Domain and Managed Server must be created using the Configuration Wizard after the ADF Runtimes are installed to the WLS home.

### **Configuration Steps:**

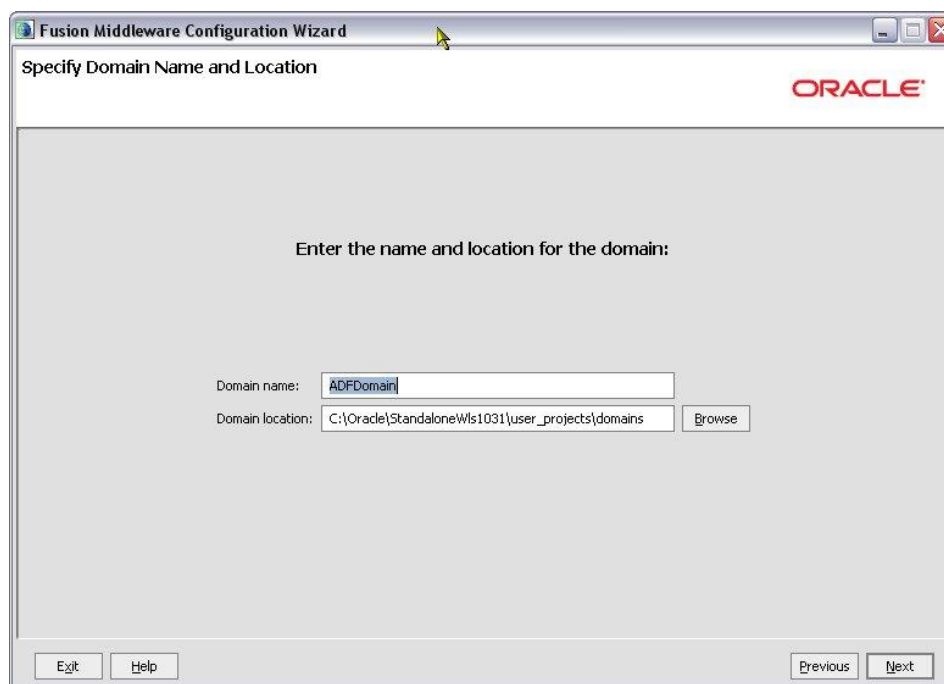
The Configuration Wizard is available from the Windows Start menu → Oracle WebLogic → WebLogic Server 11gR1 → Tools → Configuration Wizard option:

1. Welcome Page – there will be options for Create and for Extend. Select Create a new WebLogic Domain. Click Next.

2. Select Domain Source page – select Generate a domain configured automatically to support the following products, and in addition select the Oracle JRF option. Click Next.

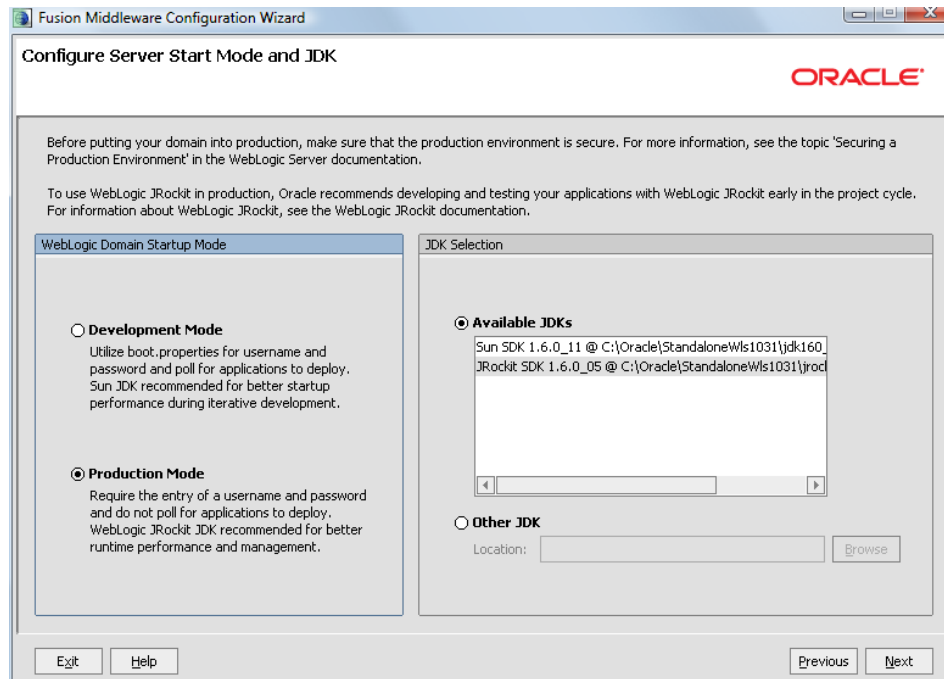


3. Specify Domain Name and Location page – change the domain name to ADFDomain and leave the location default. Click Next.



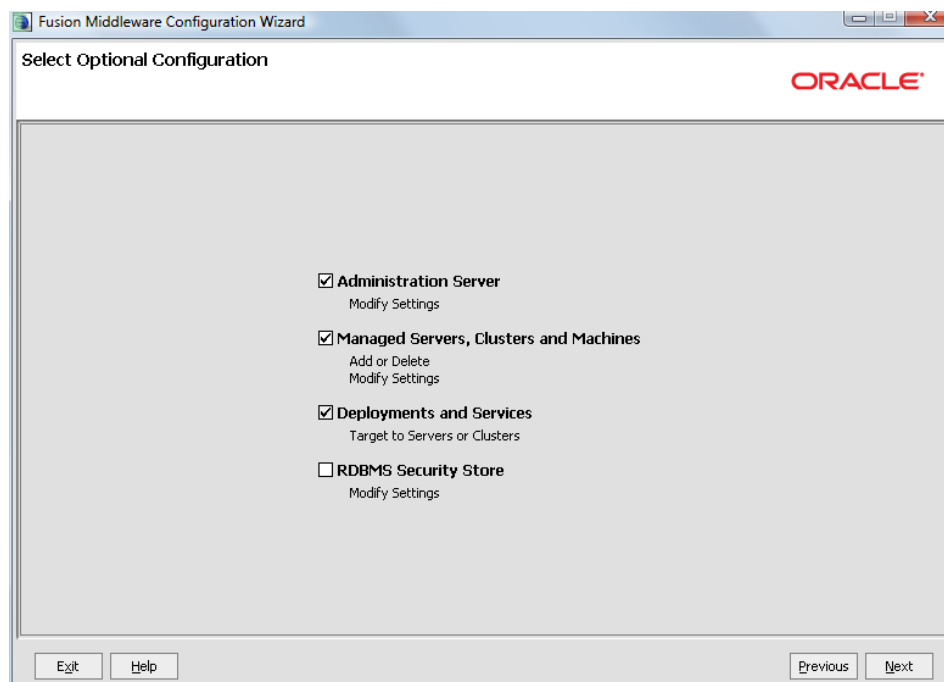
4. Configure Administrator Username and Password page – enter a user name and password for the default WebLogic Server administrator. Click Next.

5. Configure Server Start Mode and JDK page – for a production install select the Production Mode option. In addition select the JDK for the Domain to use. If you picked JRockit previously for the WLS install this should be your choice here. Click Next.



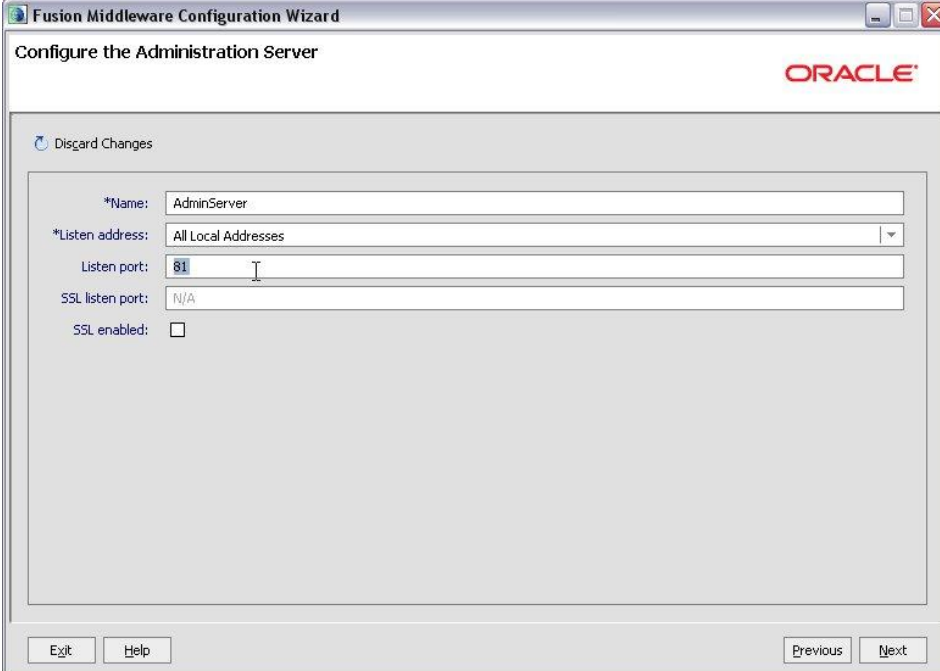
Production mode requires authentication for administrative tasks such as startup. Production mode also uses a “Lock & Edit” measure to avoid inadvertent changes being made to the Domain. We will look at the Lock & Edit feature later.

6. Select Optional Configuration page – select yes to all options except the RDBMS Security Store option. Click Next.





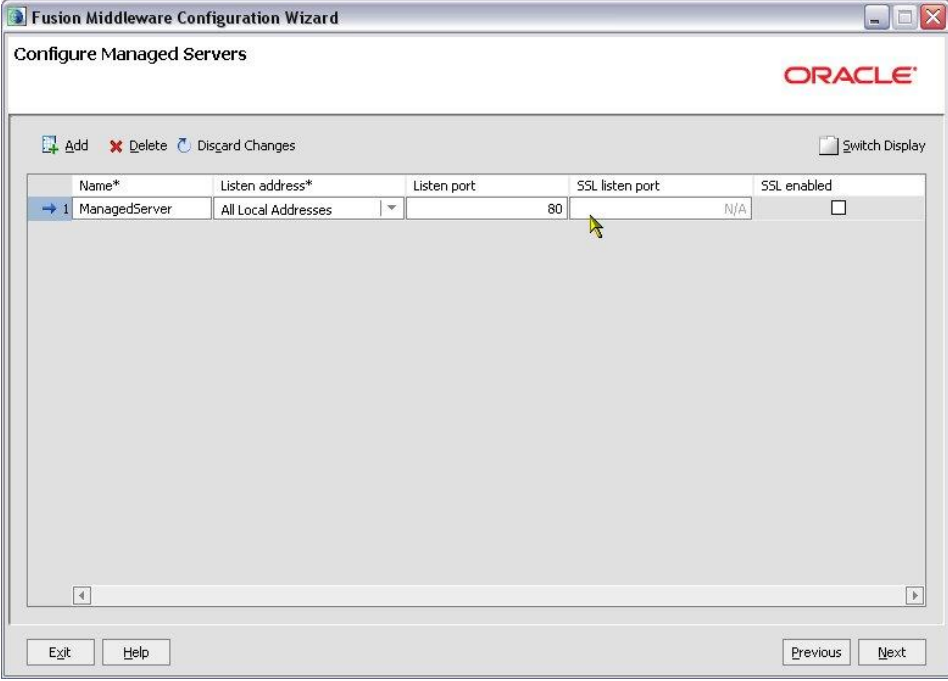
7. Configure the Administration Server page – this page allows you to configure the AdminServer. You can override these settings or leave the defaults. We have chosen to use port 81. Click Next.



The screenshot shows the 'Configure the Administration Server' window of the Fusion Middleware Configuration Wizard. The window has a title bar with the Oracle logo and standard window controls. Below the title bar, there is a 'Disard Changes' link. The main area contains several input fields: '\*Name:' with the value 'AdminServer', '\*Listen address:' with a dropdown menu showing 'All Local Addresses', 'Listen port:' with the value '81', 'SSL listen port:' with the value 'N/A', and 'SSL enabled:' with an unchecked checkbox. At the bottom, there are buttons for 'Exit', 'Help', 'Previous', and 'Next'.

Each WLS domain includes 1 administration server by default referred to as the AdminServer. The AdminServer's primary role is to configure and maintain the other server instances within the Domain.

8. Configure Managed Servers page – here we configure the new managed server. Click Add to add a new Managed Server definition. You can keep the default settings, but ensure the port number is different from the AdminServer port. In our example we override the defaults as shown. Click Next.

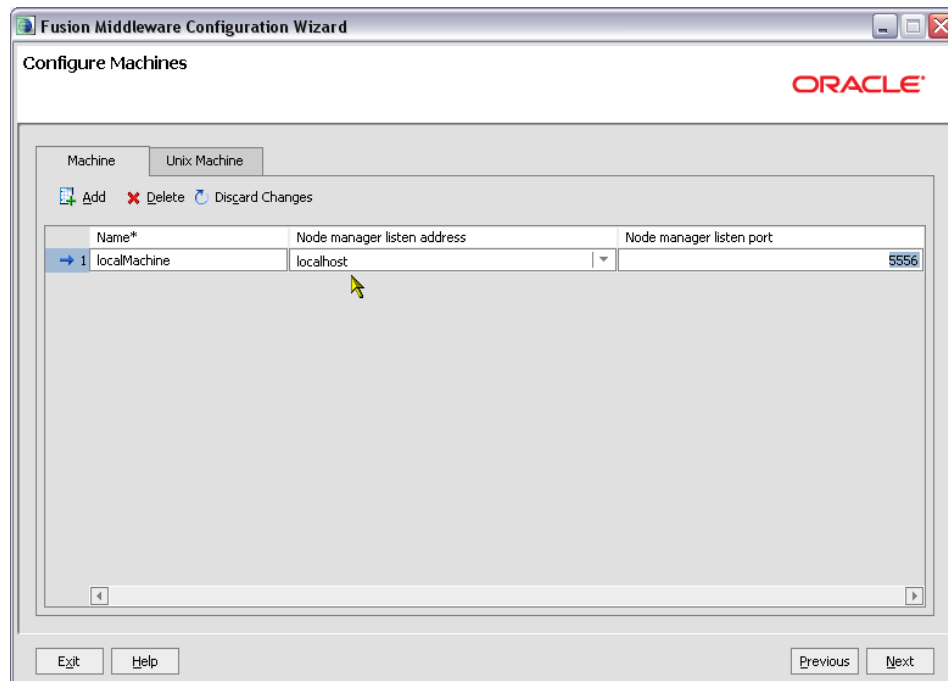


The screenshot shows the 'Configure Managed Servers' window of the Fusion Middleware Configuration Wizard. The window has a title bar with the Oracle logo and standard window controls. Below the title bar, there are links for 'Add', 'Delete', and 'Disard Changes', and a 'Switch Display' button. The main area contains a table with the following columns: 'Name\*', 'Listen address\*', 'Listen port', 'SSL listen port', and 'SSL enabled'. There is one row in the table with the following values: '1 ManagedServer', 'All Local Addresses', '80', 'N/A', and an unchecked checkbox. A mouse cursor is pointing at the '80' value in the 'Listen port' column. At the bottom, there are buttons for 'Exit', 'Help', 'Previous', and 'Next'.

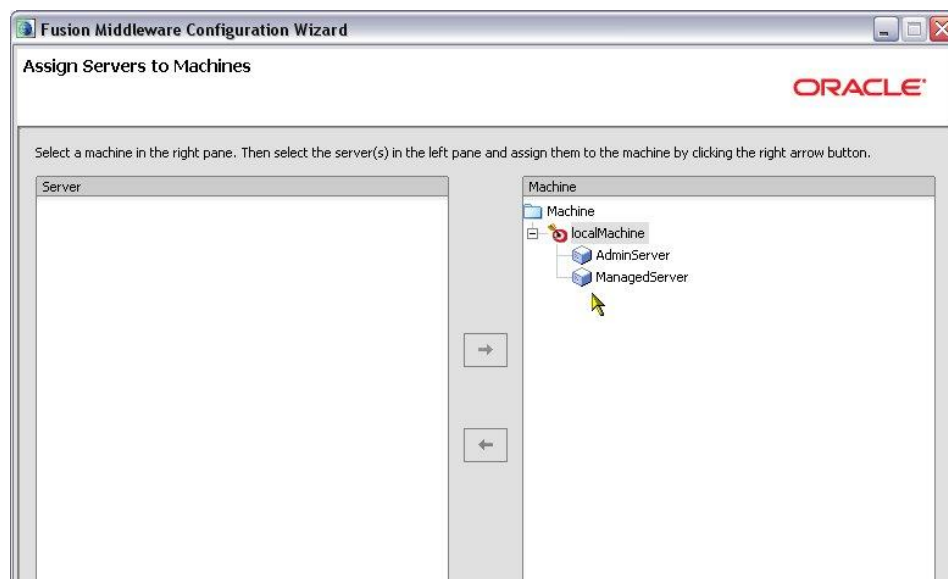
Name*	Listen address*	Listen port	SSL listen port	SSL enabled
1 ManagedServer	All Local Addresses	80	N/A	<input type="checkbox"/>

As previously described, while it's possible to install JDeveloper applications into the AdminServer, it is not ideal, as the AdminServer is intended for administration only. Additionally, using a managed server for your application means if required you have control over server configuration that specifically suits your application.

9. Configure Clusters page – Clustering is not covered in this course. Skip this page. Click Next.
10. Configure Machines page – here we create a definition for our local machine. Click Add to add a new Machine definition. Enter the values as shown below. Click Next.



11. Assign Servers to Machines page – we now assign our default Admin Server and defined Managed Server to the localMachine. Move both servers in the shuttle to the right under the localMachine node. Click Next.



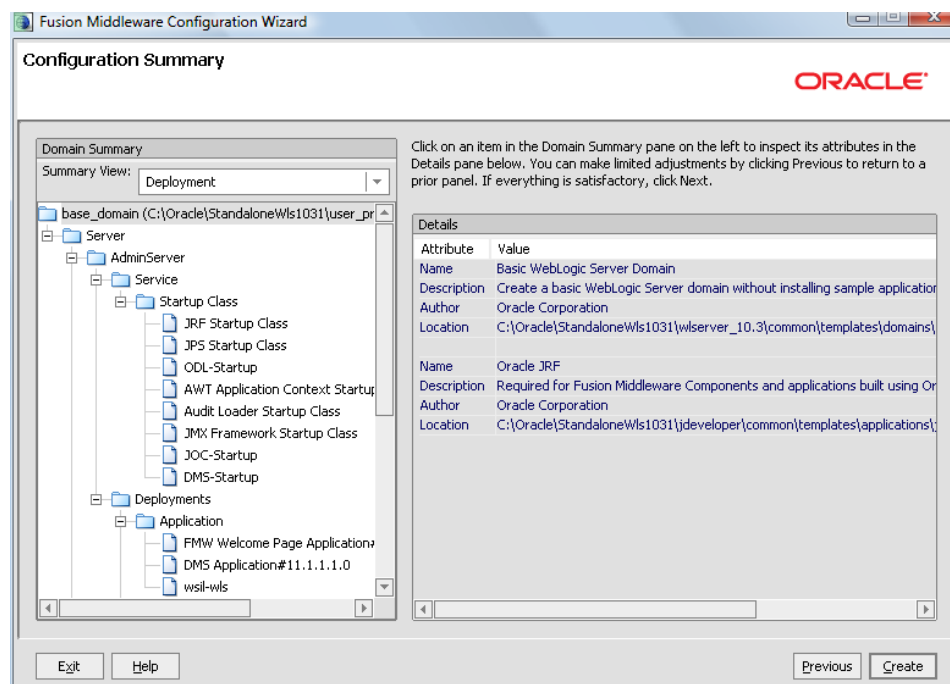
Machines represent the physical hardware that the servers run on. This allows control of the servers via the Admin Console and is most useful for clustering. Admin console control and clustering also requires Node Manager to be installed. Although we are not looking at clustering, the local machine definition and server mappings we have created will allow us to control our managed server via the admin console.

12. Target Deployments to Clusters or Servers – leave this as the default. Click Next.

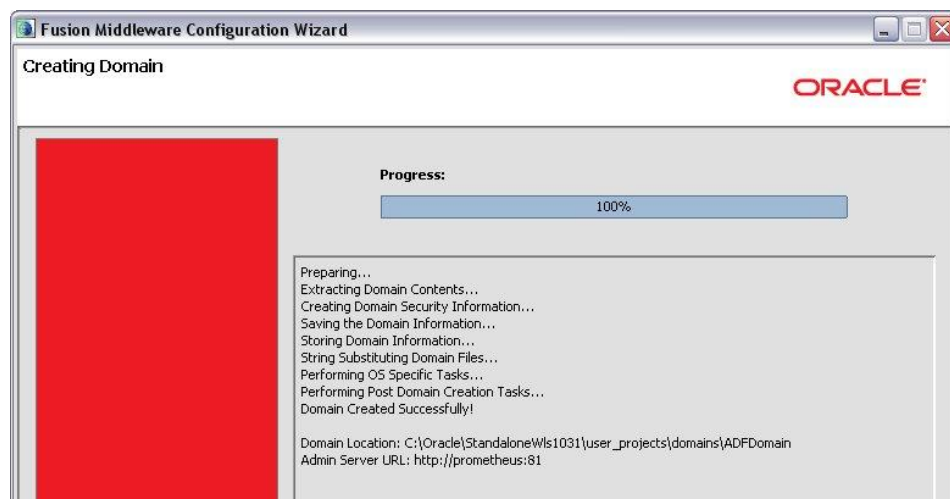
We could choose to deploy the ADF libraries to only the ManagedServer here; however we will leave it as the default and share the libraries to the ManagedServer later.

13. Target Services to Clusters or Servers – leave this as the default. Click Next.

14. Review WebLogic Domain page – Click Next.



15. Creating Domain page (100%) - at this point your additional server and machine are installed and configured, but not running. Click Done.



**WORKAROUND - Node Manager ADF classpath issue**

As previously mentioned during the WebLogic core package installation, there is a classpath issue with Node Manager and ADF enabled Domains & Servers which will cause deployment errors and failures. The problem occurs due to Node Manager starting a managed server without setting the classpath appropriately to provide access for the server to ADF shared libraries.

We need to make the following changes to workaround this problem. Duncan Mill's mentions this issue in his post and suggests to log an SR with Oracle Support in the case of Production configurations.

Locate the nodemanager.properties configuration file. The file should be in the following location: `<wls_home_dir>\wlserver_10.3\common\nodemanager`

Change the following declarations as shown:

```
StartScriptName=startWebLogic.cmd
StartScriptEnabled=false
To
StartScriptName=startManagedWebLogic.cmd
#StartScriptName=startWebLogic.cmd
StartScriptEnabled=true
#StartScriptEnabled=false
```

These changes will declare that Node Manager should use the command scripts to start Managed Servers. The command script will ensure that the environment is set correctly including the requirements in support of ADF.

Note that in contrast to the official Node Manager documentation you do not need to copy or move the command scripts. Node Manager will find the start script within the default Domain bin directory.

To ensure the changes are immediately applied use the Windows Services admin console to restart the Node Manager.

The alternative to this workaround would be to avoid the use of Node Manager (stop the service or don't install it in the first place) and start your Managed Server/s via the command scripts directly. The start, stop and other command scripts for your Domain can be found at: `<wls_home_dir>\user_projects\domains\<your domain name>\bin\`

***Java Options & other environment requirements***

If you need to set specific Java Options for a server, for example setting a specific server timezone on startup, then you can add your requirements to the start, stop and configuration scripts mentioned above, at: `<wls_home_dir>\user_projects\domains\<your domain name>\bin\`

## Sharing the ADF libraries with the new servers via the Admin Console

The ADF capable Domain, Admin Server and a Managed Server are now installed & configured. However we need to share or assign the deployed libraries to the Managed Server of the newly created ADFDomain so that they will be accessible to our deployed application/s. We will now start the Admin Server and then share the libraries via the Admin Console.

You will now have a new Windows submenu group under Oracle WebLogic → User Projects → ADFDomain.

You can start the Admin Server with the following menu option: Windows Start menu → Oracle WebLogic → User Projects → ADFDomain → Start Admin Server for WebLogic...

This invokes a command window and runs the following script:

```
<wls_home_dir>\user_projects\domains\<your domain name>\bin\startWebLogic.cmd
```

With the server configured in Production Mode, you will be prompted by the script for the administrator username and password that you specified on the Configure Administrator Username and Password page during the Domain creation phase.

Once successfully running within the command prompt window you should see the message <Server started in RUNNING mode>. Wait until you see that message and then you can minimize the command window. Do not close the window or you will shut the server down.

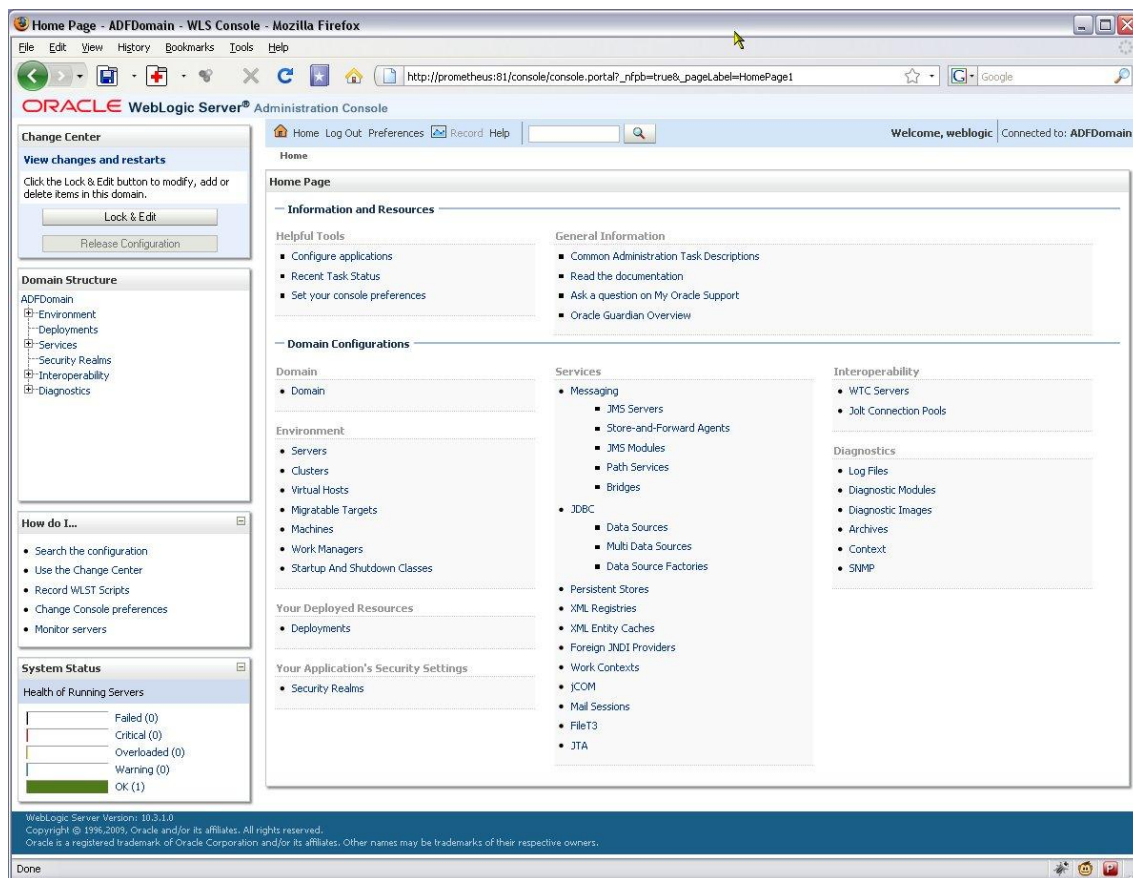
Open the WebLogic AdminServer console with the following menu option: Windows Start menu → Oracle WebLogic → User Projects → ADFDomain → Admin Server Console.

This will launch your browser opening the console at the following URL:

```
http://localhost:<port>/console    eg. http://localhost:81/console
```

....where <port> is the **AdminServer** port entered on the Configure the Administration Server page of the Configuration Wizard. The login screen again requires your WLS admin username and password.

Following login, the Weblogic Admin Console home page will be displayed.



The Domain Structure portlet on the left side of the console shows the primary review & control options for you new Domain



Within the Domain Structure portlet click the Deployments node. Under the Summary of Deployments page you will see the new ADF Runtime Libraries installed along with other default applications and libraries.

**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 16 Previous | Next

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
DMS Application (11.1.1.1.0)	Active	✓ OK	Web Application	190
FMW Welcome Page Application (11.1.0.0.0)	Active	✓ OK	Enterprise Application	150
jsf(1.2,1.2.9.0)	Active		Library	100
jstl(1.2,1.2.0.1)	Active		Library	100
ohw-rcf(5,5.0)	Active		Library	100
ohw-uix(5,5.0)	Active		Library	100
oracle.adf.dconfigbeans(1.0,11.1.1.0.0)	Active		Library	100
oracle.adf.management(1.0,11.1.1.1.0)	Active		Library	100

Install | Update | Delete | Start ▾ | Stop ▾

Showing 1 to 10 of 16 Previous | Next

Remember, when creating the domain we only deployed the libraries to the Admin Server. We must now share them with the Managed Server as well. We are going to share the libraries here.

Assuming WLS is running in production mode, first click the Lock and Edit option at the top left of your screen to allow domain changes to be made. This is unnecessary if your server is in deployment mode.

**Change Center**

**View changes and restarts**

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

Lock & Edit

Release Configuration

For each library, select the library hyperlink, followed by the Targets tab, and then select the ManagedServer. Click Save for each.

**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 16 Previous | Next

<input type="checkbox"/>	Name	State	Health	Type	Deployment Order
<input type="checkbox"/>	<a href="#">adf.oracle.domain(1.0,11.1.1.1.0)</a>	Active		Library	100
<input type="checkbox"/>	<a href="#">adf.oracle.domain.webapp(1.0,11.1.1.1.0)</a>	Active		Library	100
<input type="checkbox"/>	<a href="#">DMS Application (11.1.1.1.0)</a>	Active	✓ OK	Web Application	190
<input type="checkbox"/>	<a href="#">FMW Welcome Page Application (11.1.0.0.0)</a>	Active	✓ OK	Enterprise Application	150

**Settings for adf.oracle.domain(1.0,11.1.1.1.0)**

Overview | **Targets** | Notes

Save

Use this page to select the WebLogic Server instances and clusters to which you want to deploy (target) the Java EE library.

**Servers**

<input checked="" type="checkbox"/>	AdminServer
<input checked="" type="checkbox"/>	ManagedServer

Save

Return to the Summary of Deployments (list of libraries) via the Domain Structure Portlet, Deployments node again and repeat the process for each ADF library, until all ADF libraries are shared to the ManagedServer.

Assuming WLS is running in production mode, you will need to click the Activate Changes button at the top left of the console to make the changes permanent. This step is unnecessary if your WLS server is running in deployment mode.

**Change Center**

**View changes and restarts**

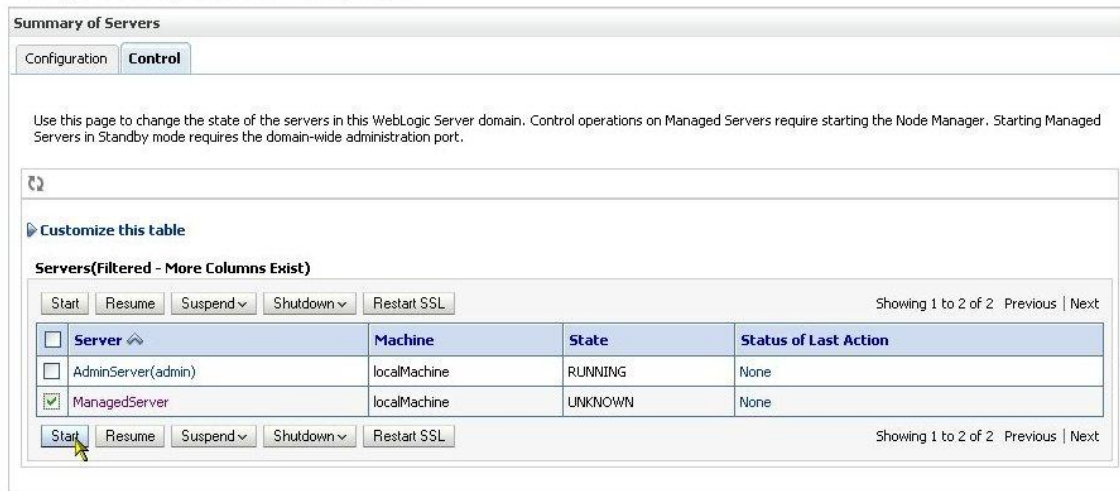
Pending changes exist. They must be activated to take effect.



## Starting the new WebLogic Managed Server

Now we can start our Managed Server. Since we installed Node Manager, defined a machine definition for our local machine and mapped the ManagedServer to the machine, we can start our managed server via the Admin Console.

Within the Domain Structure portlet click the Environment node followed by the Servers and then the Control tab. On the Summary of Servers page Control tab, check the box for our ManagedServer and then click the Start button. Confirm you wish to start the server. Click the refresh icon if you wish to monitor the startup status of the server.



Alternatively, to start the Managed Server (ManagedServer), via the command line, execute the following command:

```
<wls_home_dir>\user_projects\domains\<your domain name>\bin\startManagedWeblogic.cmd <your server name>
http://localhost:<port>
```

....where your server name is the name of the new server you created in the Configure Managed Servers page of the Configuration Wizard (e.g. ManagedServer), and the port is the port configured for the AdminServer (**not the new managed server port**). Eg:

```
c:\oracle\wls10_3\user_projects\domains\ADFDomain\bin\startManagedWebLogic.cmd ManagedServer http://localhost:81
```

When starting the server via the command line, again you will need to enter the administrator username and password, and wait for the message <Server started in RUNNING mode>.

Following the server startup you should now see the state as **RUNNING** and health as **OK** in the Summary of Servers page in the Admin Console. As described above you will find the Summary of Servers page within the Structure Portlet by clicking the Environment → Servers node.

**Summary of Servers**

**Configuration** Control

A server is an instance of WebLogic Server that runs in its own Java Virtual Machine (JVM) and has its own configuration.  
This page summarizes each server that has been configured in the current WebLogic Server domain.

[Customize this table](#)

**Servers (Filtered - More Columns Exist)**  
Click the **Lock & Edit** button in the Change Center to activate all the buttons on this page.

New Clone Delete Showing 1 to 2 of 2 Previous Next

Name	Cluster	Machine	State	Health	Listen Port
AdminServer(admin)	localMachine		RUNNING	✓ OK	81
ManagedServer	localMachine		RUNNING	✓ OK	80

New Clone Delete Showing 1 to 2 of 2 Previous Next

Additionally, since we defined a machine definition for our local machine you can inspect the Summary of Machines. Select the Environment → Machine node to reveal the Summary of Machines page, listing the new machine you just created, in this case LocalMachine:

If you select the LocalMachine link, the Settings for Machine page will be displayed where you can in turn select the Configuration tab followed by the Servers tab, where you will see the AdminServer and ManagedServer are installed for the LocalMachine.

You now have an ADF capable domain with an ADF ready Managed Server running. The ManagedServer can now be used as the deployment target for our ADF Fusion Application/s.

To shut down both the new Managed server and the AdminServer, simply **Ctrl-C** the command prompt window/s, causing the servers to gracefully shut down. Alternatively you can select the server/s and click Shutdown in the Summary of Servers, Control tab in the Admin Console.

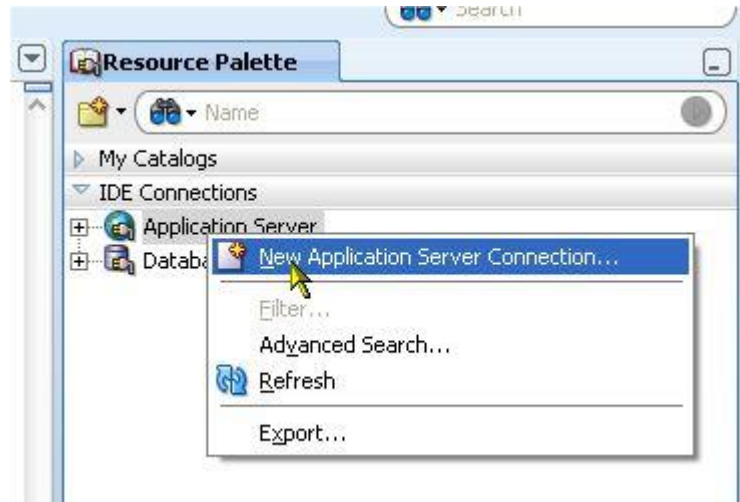
## Connecting JDeveloper to your application server

In order to deploy an application from “within” JDeveloper to your WLS you need to establish an Application Server Connection in the IDE Connections Navigator.

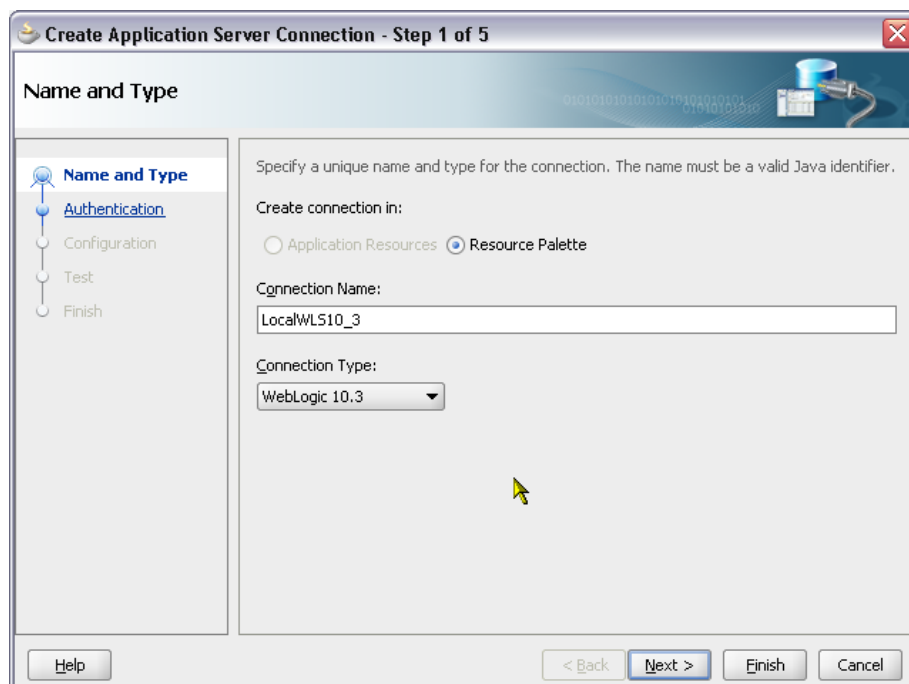
**Configuration Steps:**

Create a new Application Server connection from the context menu by right clicking on the Application Server node in the IDE Connections Navigator.

By default this Navigator is on the upper right side of the IDE. Choose the New Application Server Connection option.



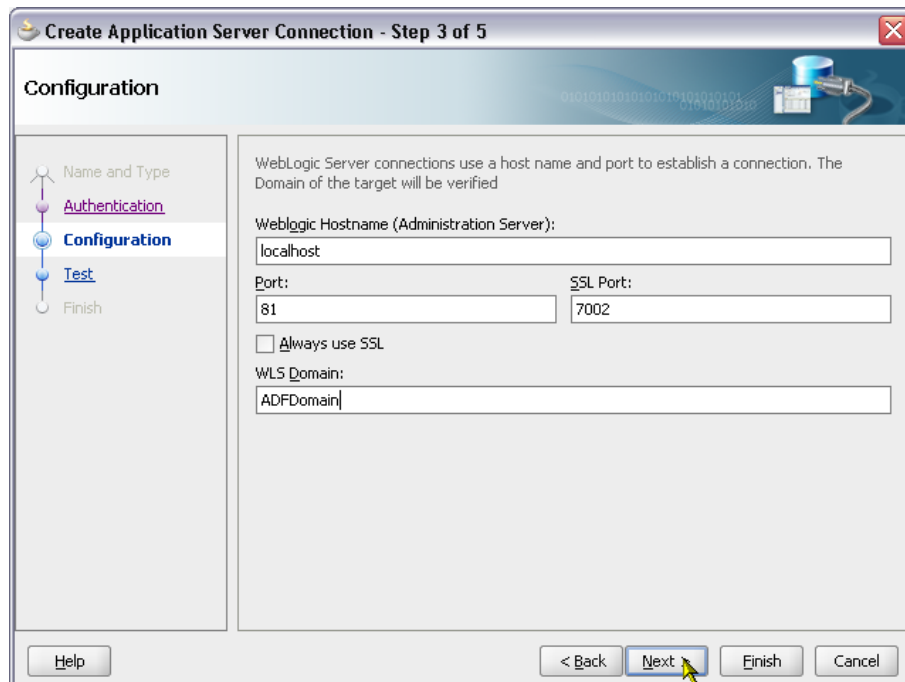
1. Give the connection a name; we have used LocalWLS10\_3, and select the connection type of WebLogic 10.3. Click Next.



Since we are creating an IDE Connection it will be created within the Resource Palette.

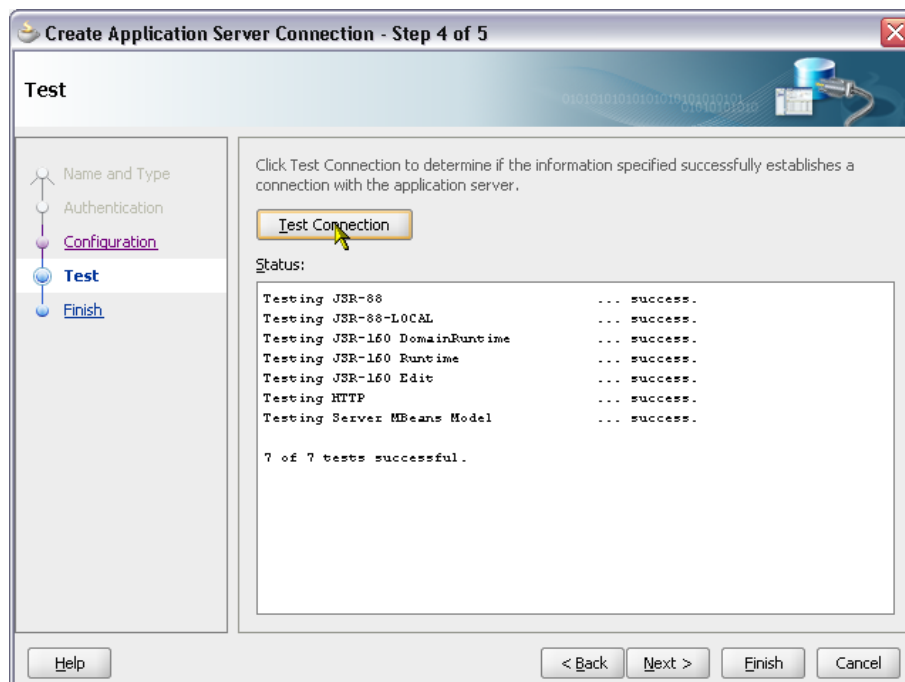
2. Provide the credentials to connect to the AdminServer. Ignore the Deploy Password checkbox, it is only relevant for a Application Resource connection and it would generally be inadvisable to deploy the password for a production application resource connection anyway. Click Next.

- Specify the AdminServer context connection details including hostname, AdminServer port and Domain.



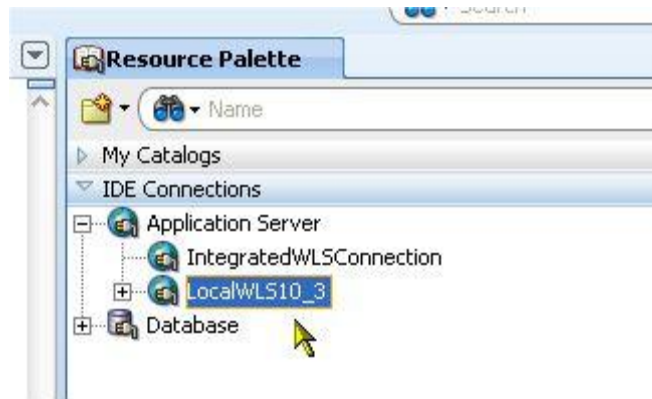
Following on from the previous installation and configurations sections our values are as shown to the right. For our purposes here we are ignoring SSL. Click Next.

- Click Test Connection for JDeveloper to run various tests to validate the WLS connection.



You can now Click Finish to create the connection and return to the JDeveloper IDE.

You should now see the new Connection in the IDE Connections palette.



## WebLogic JDBC Data Sources

To provide Database connectivity to our applications there are two principal approaches available, those being, embedded JDBC URLs or container based JDBC Data Sources.

During development and within the JDeveloper IDE, by default, you most likely will have been using the embedded JDBC URL. It is preferable, particularly for QA/Production, to use container based connections defined as WebLogic JDBC Data Sources.

Some key benefits of using container based connections over embedded JDBC URL connections include:

- Avoiding the hassle of embedded connection details; embedded URLs would require you to change the database name and possibly credentials and rebuild to deploy to development vs to test vs to production<sup>1</sup>.
- Avoiding the security implications of embedded credentials; yes the connection username and password will be deployed
- Avoiding the need for the development team of even knowing the production password, which in many cases would be a matter of security/audit policy
- better connection and application module pooling support, management and easier monitoring by Application Server Administrators

We shall now run through the steps required to configure a WebLogic Data Source equivalent to the JDBC URL that you have been using during development of your application.

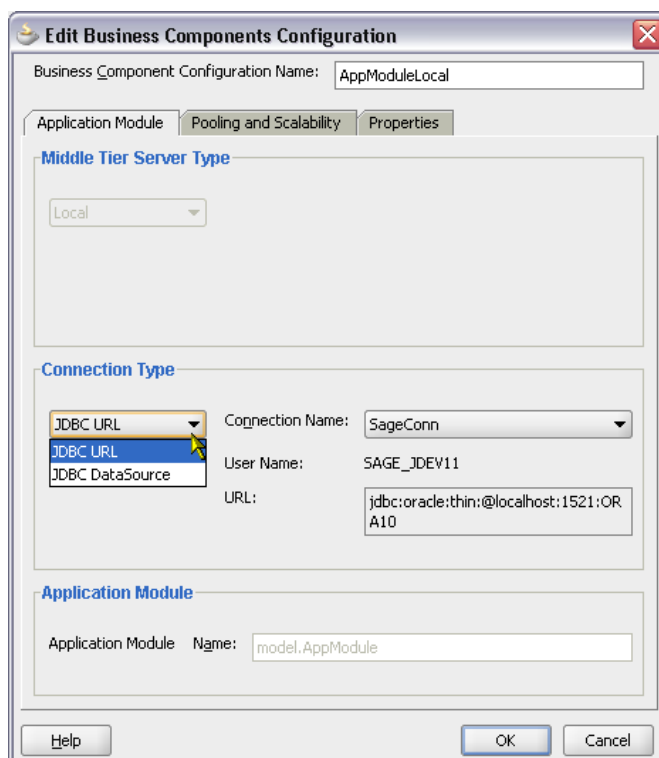
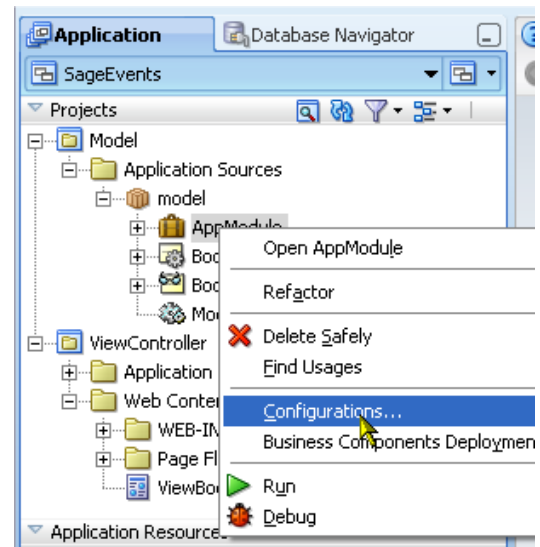
---

<sup>1</sup> It is best to avoid **ever** deploying to your QA/PROD environments with a JDBC URL. Even if you change to a Datasource later, the embedded credentials will remain in your application configuration files; with the only option to remove them manually.

## Configuration Steps:

First we check the connection currently used by our application and if required change the connection from a JDBC URL to a JDBC Datasource.

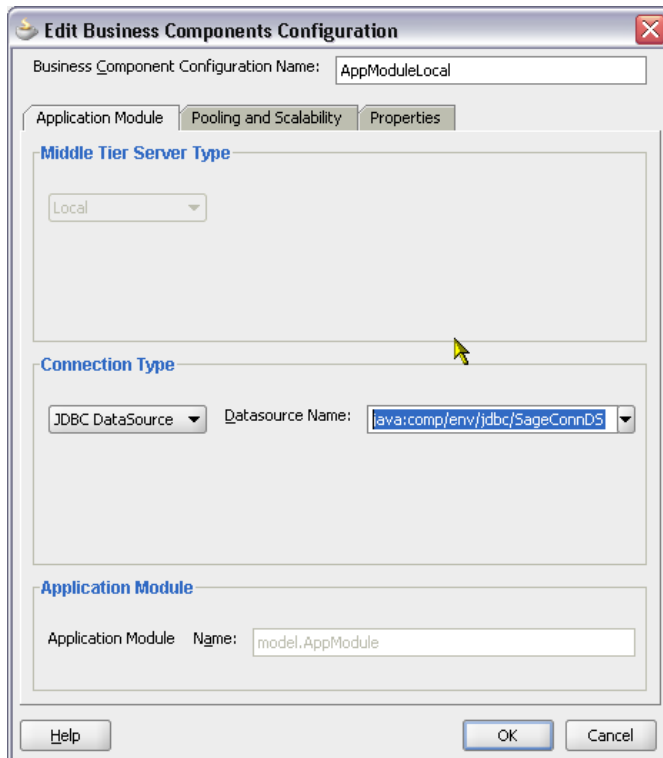
In the JDeveloper IDE, expand your model project, right click on the ADF BC Application Module and in the context menu choose the Configurations option. This will open the Manage Configurations dialog.



Click on the Edit button for AppModuleLocal.

On the Application Module Tab in the Connection Type panel, assuming the type is currently set to JDBC URL, change the type to JDBC Datasource.

This should give you a default datasource name based on the previously defined URL.



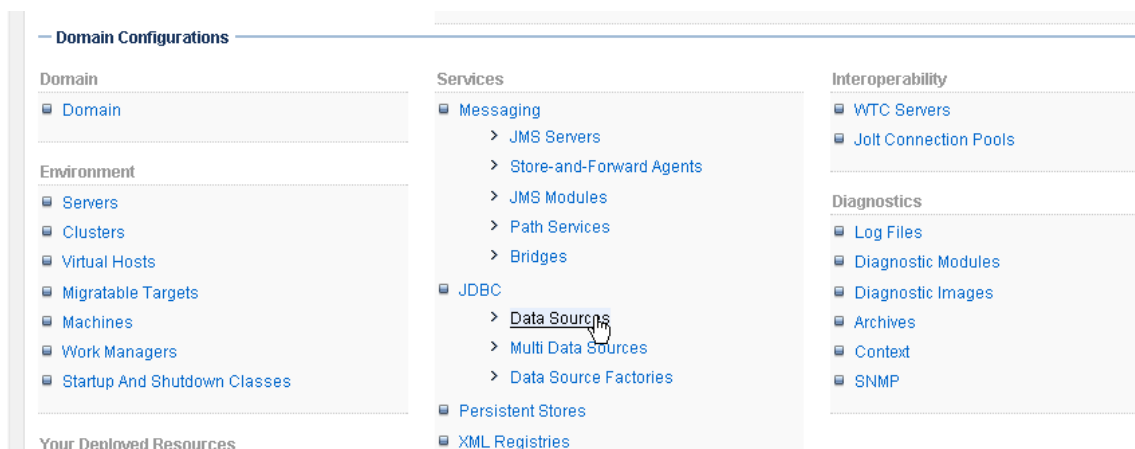
In this example the default datasource is  
`java:comp/env/jdbc/SageConnDS`.

Take note of the name and then click OK to save and exit.

We now configure a WLS Data Source to match the name required by our application. Note that the actual connection details be different, according to the needs of the target environment; however for this exercise we will keep the connection details the same as within JDeveloper.

Login to the WLS Admin Console by clicking on the Oracle WebLogic → User Projects → ADFDomain → Admin Server Console option under your Windows start menu. Alternatively you can open your browser and manually enter the address URL of the new Domain Admin Console (ADFDomain). Eg: `http://localhost:81/console`

On the main Admin console homepage under the Domain Configurations click on the Data Sources link listed under JDBC in the approximate centre of the page.



Assuming WLS is running in production mode, as previously detailed above under “Sharing the ADF libraries with the new servers via the Admin Console”, first select the Lock and Edit option at the top left of your screen.

1. Click New on the Summary of JDBC Data Sources panel to create a new definition.
2. JDBC Data Source Properties page - enter the name as defined for your application AppModuleLocal configuration within JDeveloper. Note: specify only the name without the prefix: “java:comp/env/jdbc/”. Also enter the DataSource name in the format jdbc/<Datasource\_name> in the JNDI Name field; in this case again do not use the full java:comp/env/jdbc/ prefix, but include only the prefix of jdbc/.

Specifying the Datasource name within the JDeveloper IDE and the Datasource Name and JNDI Name in the formats specified here, including/excluding the appropriate prefixes, will allow the IDE to still serve a connection for you when running your application with the IDE Integrated WLS<sup>2</sup> as well as the target standalone WLS.

Select Oracle as the Database Type and Oracle’s Driver (Thin) for Instance... as the Database Driver. The values for our application are shown. Click Next.

Create a New JDBC Data Source

Back Next Finish Cancel

**JDBC Data Source Properties**

The following properties will be used to identify your new JDBC data source.

\* Indicates required fields

What would you like to name your new JDBC data source?

Name: SageConnDS

What JNDI name would you like to assign to your new JDBC Data Source?

JNDI Name: jdbc/SageConnDS

What database type would you like to select?

Database Type: Oracle

What database driver would you like to use to create database connections? Note: \* indicates that the driver is explicitly supported by Oracle WebLogic Server.

Database Driver: \*Oracle's Driver (Thin) for Instance connections; Versions:9.0.1,9.2.0,10,11

Back Next Finish Cancel

3. Transaction Options page – leave the defaults and Click Next.

<sup>2</sup> With the proviso that "Auto Generate JDBC Connections When Running Application in JDeveloper" is enabled in the Application Properties / Deployment configuration



4. Connection Properties page – enter connection details for the Database connection that your application requires in the target environment (DEV, Test, QA or PROD). The values for our application are shown. Click Next.

The screenshot shows the 'Create a New JDBC Data Source' wizard, specifically the 'Connection Properties' step. The page has a title bar 'Create a New JDBC Data Source' and navigation buttons 'Back', 'Next', 'Finish', and 'Cancel'. Below the title bar, the section 'Connection Properties' is followed by the instruction 'Define Connection Properties.' The form contains several fields with labels and prompts:

- Database Name:** Prompt: 'What is the name of the database you would like to connect to?'. Value: 'ORADB102'.
- Host Name:** Prompt: 'What is the name or IP address of the database server?'. Value: 'localhost'.
- Port:** Prompt: 'What is the port on the database server used to connect to the database?'. Value: '1521'.
- Database User Name:** Prompt: 'What database account user name do you want to use to create database connections?'. Value: 'train1'.
- Password:** Prompt: 'What is the database account password to use to create database connections?'. Value: masked with dots.
- Confirm Password:** Value: masked with dots.

At the bottom, there are navigation buttons 'Back', 'Next', 'Finish', and 'Cancel'.

5. Test Database Connection page – on this page you can click the Test Connection button to have WLS test that the definition and values provided result in a successful database connection.

*Do not click Finish here or you will not get to choose Targets for your Datasource resulting in your Datasource not being available to the Servers and in turn to your Applications. Click Next.*

6. Select Targets page – ensure that the Data Source is made available to the Managed Server (ManagedServer) and optionally the AdminServer. Click Finish.
7. Summary of JDBC Data Sources page – you are now returned to this page where you should see your new Data Source listed.

The screenshot shows the 'Summary of JDBC Data Sources' page. It has a title bar 'Summary of JDBC Data Sources'. Below the title bar, there is a descriptive paragraph: 'A JDBC data source is an object bound to the JNDI tree that provides database connectivity through a pool of JDBC connections. Applications can look up a data source on the JNDI tree as follows. This page summarizes the JDBC data source objects that have been created in this domain.' Below this, there is a link 'Customize this table'. The main content is a table titled 'Data Sources(Filtered - More Columns Exist)'. The table has three columns: 'Name', 'JNDI Name', and 'Targets'. There is a 'New' and 'Delete' button above the table. The table contains one row with the following data:

Name	JNDI Name	Targets
SageConnDS	jdbc/SageConnDS	AdminServer, ManagedServer

Below the table, there are 'New' and 'Delete' buttons. A mouse cursor is pointing at the 'SageConnDS' row.

Assuming WLS is running in production mode, as previously detailed above under “Sharing the ADF libraries with the new server/s via the Admin Console”, you need to press the Activate Changes button at the top left of the console to make the changes permanent.

## Deployment

### *JARs, WARs, EARs and JDeveloper*

In order to make the deployment of JEE applications to JEE Servers easier, the JEE specification defines standards for the creation of archive files. The archives contain standard file packages, standard locations and configuration files of which JEE Servers can make use in deploying an application.

The JEE archive file types include:

- **JAR file** – an abbreviation for Java Archive, typically a compressed file with a `.jar` extension. Can be opened by such tools as Winzip. Designed to be run using the java command line. eg. `java -jar myapp.jar`, or to contain libraries of support code.
- **WAR file** – an abbreviation for Web Archive, a compressed file with a `.war` extension. Contains a web application comprised of web pages, including JSPs and Faces files. Within a `WEB-INF` directory includes a standard set of configuration files such as `web.xml`. On deployment the WAR file is expanded into its component directories and files on the server, while a copy of the WAR file is kept in the project root directory.
- **EAR file** – an abbreviation for Enterprise Archive, a compressed file with a `.ear` extension. Intended to contain all applications required for an enterprise, including one or more JAR or WAR files.

Within JDeveloper you may inspect the contents of JEE archive files. Select the File → Open menu option, then the archive file in the Open File dialog. A page showing the archive's content will be displayed.

### Deployment Profiles

Deployment Profiles define source files, deployment descriptors, configuration files, the type and name of the archive file to be created, dependency information and platform-specific instructions required in deploying your application to a JEE archive; ie. all that is required for deployment.

Thankfully, when creating a new Fusion Application, JDeveloper auto generates Deployment Profiles suitable for development purposes for you. Archive files are also generated by JDeveloper upon deployment within the IDE.

Opening the Project Properties for each project you can see that under the Deployment node JDeveloper has created an ADF Library JAR Archive for your Model project, and a WAR Archive for your ViewController project. At the Application level JDeveloper has created an EAR archive that includes the Model JAR and ViewController WAR.

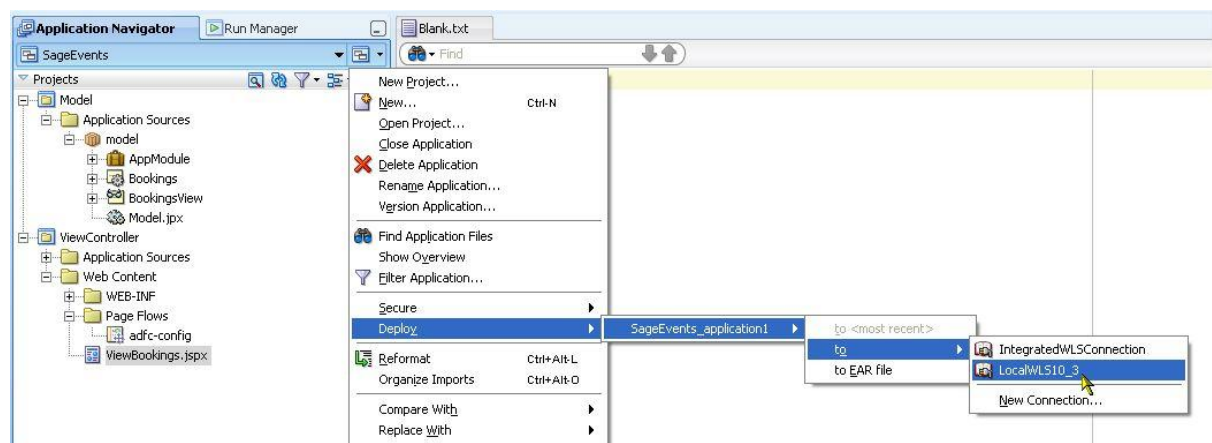
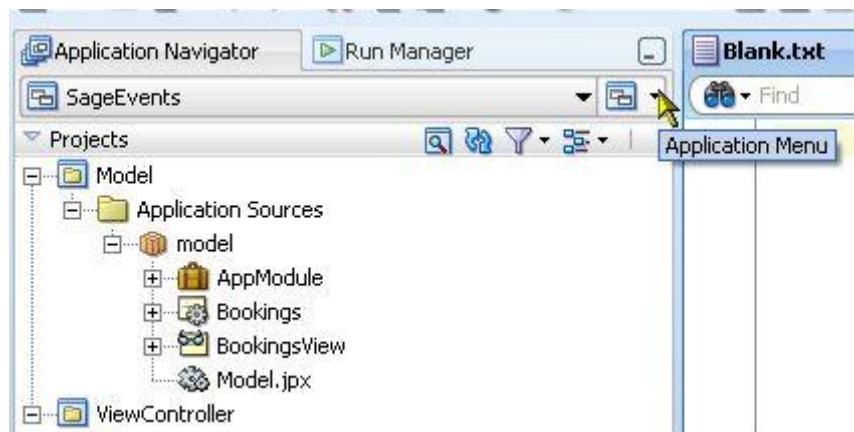
If you wish to make changes to the deployed application name, context-root (access URL) or included libraries, among a myriad of other things, this is the place to do it. A prerequisite to creating Deployment Profiles is the creation of an Application Server Connection definition in the Connections Navigator as we covered earlier.

## JDeveloper Auto Deployment

As described above, we can get JDeveloper to do most of the Deployment work for us. JDeveloper can build the application, create and write-out the appropriate WAR files, generate the ViewController EAR file and deploy straight to your standalone packaged WLS with a single click.

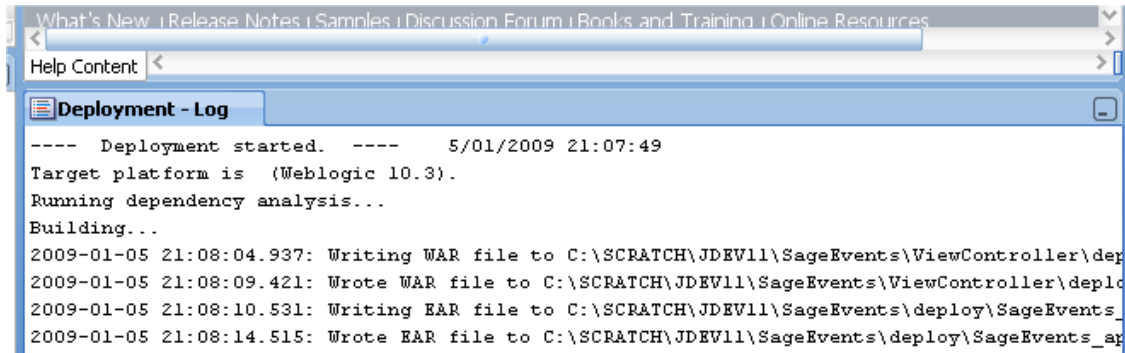
### Deployment Steps:

1. Within the JDeveloper IDE click on the Application Menu down arrow to the right of your Application in the Application Navigator.
2. Expand out from the Deploy option to Deploy → <your application → to → <your WebLogic connection> for example: Application Menu → Deploy → SageEvents\_application1 → to → LocalWLS10\_3

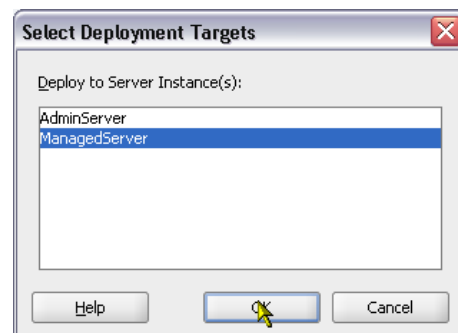


The Deployment will immediately start.

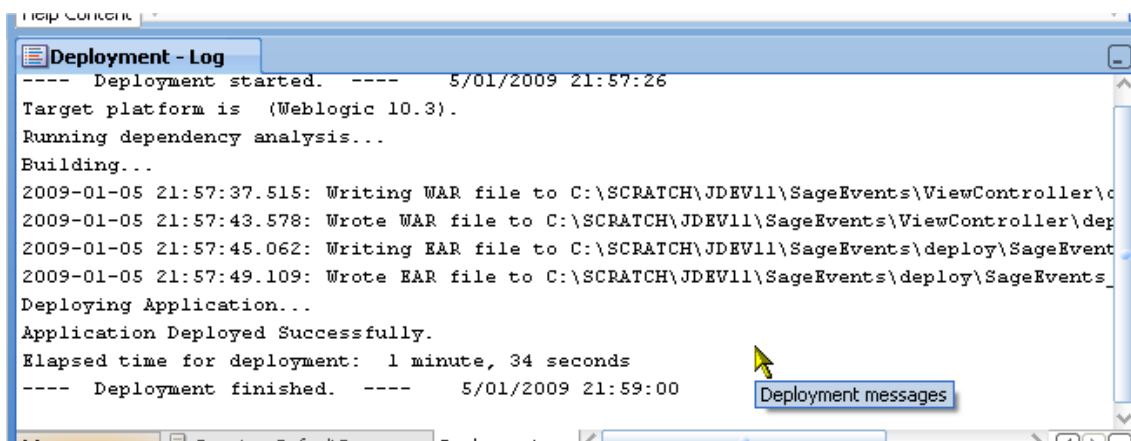
JDeveloper will automatically compile and build the application and then create the ViewController WAR file followed by the ViewController EAR file and then begin deployment to the WLS. You can watch the progress in the Deployment Log panel.



3. Since during the course of this unit we have configured both the AdminServer and the ManagedServer for the ADFDomain, JDeveloper will prompt you for the target Server that you wish to deploy to. Typically this would be a ManagedServer instead of the AdminServer.



You should then see confirmation of the successful deployment in the Deployment Log panel.

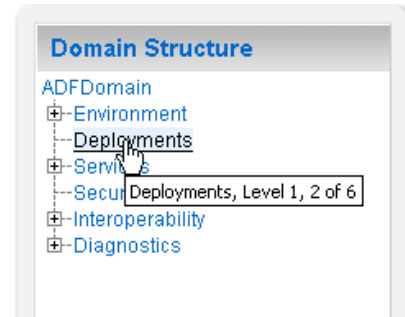


## Testing your Application

Once you have deployed your application to the WLS you can check the status of the deployment and finally access and test run your application.

### Testing Steps:

1. Login to the WLS Admin Console as previously covered previously.
2. Click on Deployments in the Domain Structure portlet on the left.



3. Summary of Deployments page – you should see your new application listed, along with the ADF Libraries with a state of Active and Health of OK. Click on your application link.

**Deployments**

Install Update Delete Start Stop Showing 1 t

<input type="checkbox"/>	Name	State	Health	Type
<input type="checkbox"/>	adf.oracle.domain(1.0.11.1.1.0.0)	Active		Library
<input type="checkbox"/>	jsf(1.2.1.2.7.1)	Active		Library
<input type="checkbox"/>	jstl(1.2.1.2.0.1)	Active		Library
<input type="checkbox"/>	SageEvents_application1	Active	OK	Enterprise Application

Install Update Delete Start Stop Showing 1 t

4. Settings for <your application> page - this page has a number of tabs that specify the full configuration details of your deployed application. Click on the Testing tab.
5. Deployment Tests page – this is mostly for SOA process testing. However, take note of the context root for your application, particularly if you have used one click deployment via JDeveloper, as we have here.

**Settings for SageEvents\_application1**

Overview Deployment Plan Configuration Security Targets Control **Testing** Monitoring Notes

Some deployment types support test points you can use to verify that a deployment was successful and that the module is ready for use.  
The following table includes all of the test points available for this application or module.

**Deployment Tests**

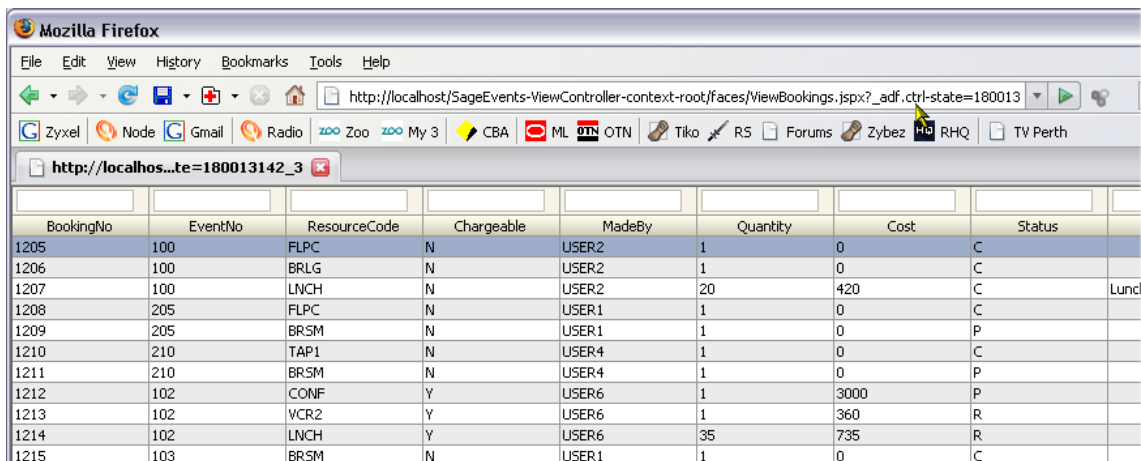
Name	Test Point	Comments
SageEvents_application1		
SageConn		There are no test points for this JDBC module.
SageEvents-ViewController-context-root		

SageEvents-ViewController-context-root, Level 2, 2 of 2

6. With the context root for you application in mind, you can now open a browser session and test your application using a URL with the following format:

```
http://localhost:<target-server-port>/
    <app-context-root>/faces/<your page>
```

....where the port is the port configured for the Managed Server, the app context root is the root as shown in WLS and your page is the faces.jspx that you would like to run.



The screenshot shows a Mozilla Firefox browser window with the address bar displaying the URL: `http://localhost:SageEvents-ViewController-context-root/faces/ViewBookings.jspx?_adf.ctrl-state=180013`. The browser's bookmark bar contains several links including Zyxel, Node, Gmail, Radio, Zoo, My 3, CBA, ML, OTN, Tiko, R5, Forums, Zybez, RHQ, and TV Perth. The main content area displays a table with the following data:

BookingNo	EventNo	ResourceCode	Chargeable	MadeBy	Quantity	Cost	Status
1205	100	FLPC	N	USER2	1	0	C
1206	100	BRLG	N	USER2	1	0	C
1207	100	LNCH	N	USER2	20	420	C
1208	205	FLPC	N	USER1	1	0	C
1209	205	BRSM	N	USER1	1	0	P
1210	210	TAP1	N	USER4	1	0	C
1211	210	BRSM	N	USER4	1	0	P
1212	102	CONF	Y	USER6	1	3000	P
1213	102	VCR2	Y	USER6	1	360	R
1214	102	LNCH	Y	USER6	35	735	R
1215	103	BRSM	N	USER1	1	0	C