



**SAGE Computing Services**

**Consulting and customised training workshops**

# **Deploying ADF Applications to Oracle WebLogic Server (a practical walkthrough)**

*Ray Tindall*

Senior Systems Consultant

[www.sagecomputing.com.au](http://www.sagecomputing.com.au)

# Agenda Overview

- WebLogic
  - Why
  - Concepts
- Installation & ADF preparation
  - WebLogic install
  - ADF runtime libraries
  - Creating/configuring a deployment target
- JDev connection to WebLogic
  - JDev/WebLogic connection
  - JDBC Data Sources
- Deploying ADF Applications
  - Auto Deploy
- Testing
- Pitfalls & further resources

# Why

- Why deploy?

JDeveloper integrated WLS

vs

“The Real Thing”

- Why WebLogic?

Oracle Fusion Middleware's strategic  
application server



# WebLogic Fundamentals

- Domains

Core encompassing unit comprising Administration server, one or more Managed servers, Machines and maybe Clusters

- Servers

- Administration Server – for Domain configuration and management
- Managed Server/s – providing a platform and resources for applications, EJBs, Web Services...

- Machines

WebLogic definition that maps servers to physical hardware

- Clusters?

Means of high availability implementations;  
logical grouping for administration of managed servers

# WebLogic vs OAS

Weblogic	OAS (& comments)
Domain	EM or Grid control and a Farm
Admin Server	OAS instance and/or “home” oc4j - that is about the closest (ableit quite loose) equivalent. OAS home oc4j is the core admin process
Managed servers	dedicated purpose oc4j eg, oc4j_portal, oc4j_mycustomapps
Machines	errrr - Host (& not sure that is strictly valid). In OAS there is no definite component that represents a separation of physical hardware from the processes. In WebLogic that is exactly what a Machine is; it maps servers to physical hardware and is actively used by Node Manager to handle High Availability tasks such as failover.
Clusters	OAS multi instance tiers ie. a multi mid-tier architecture to support higher performance and/or failover.

# Walkthrough

- WebLogic Installation & Preparation
  - WebLogic Package installation
  - Install ADF runtime libraries
  - Create ADF ready domain & target server
  - Sharing ADF libraries
- Connecting JDev & WebLogic
  - JDBC Data Sources
- Deployment (light theory)
  - Auto-Deploy
- Testing



# WebLogic Package Installation

## Requirements:

- Download latest platform specific WLS installer
- NET or PACKAGE
- Virus/Firewall – config/disable/shutdown



# WebLogic Package Installation

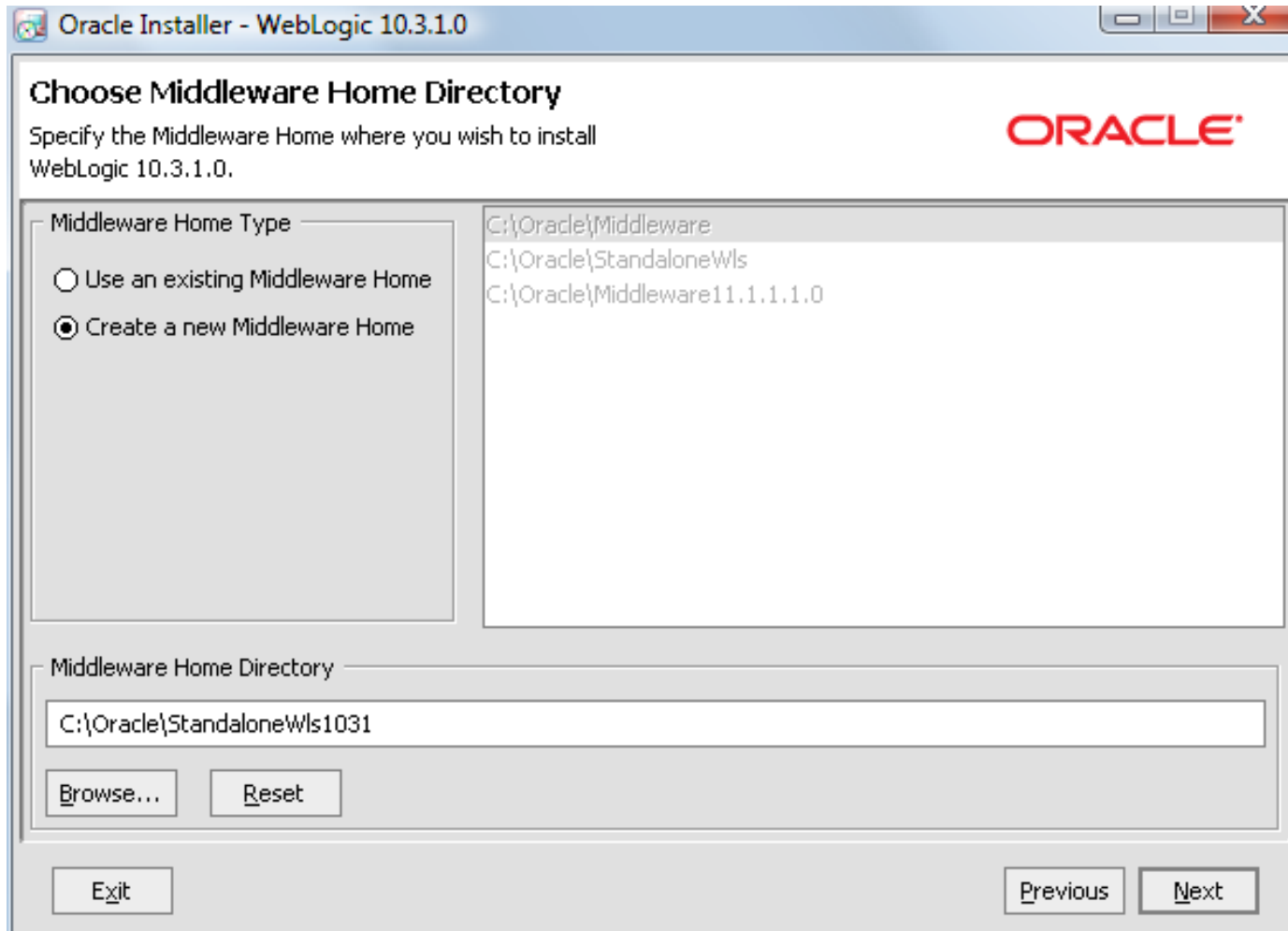
## Installation Steps:

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 directory in the BEA Home Directory field.





# WebLogic Package Installation



We have chosen **c:\Oracle\StandaloneWls1031**.

Note no spaces in the directory names. Click **Next**.

# WebLogic Package Installation

4. Register for Security Updates page – **uncheck** then **Next**.
5. Choose Install Type page - select **Custom** then **Next**.
6. Choose Product and Components page -



# WebLogic Package Installation

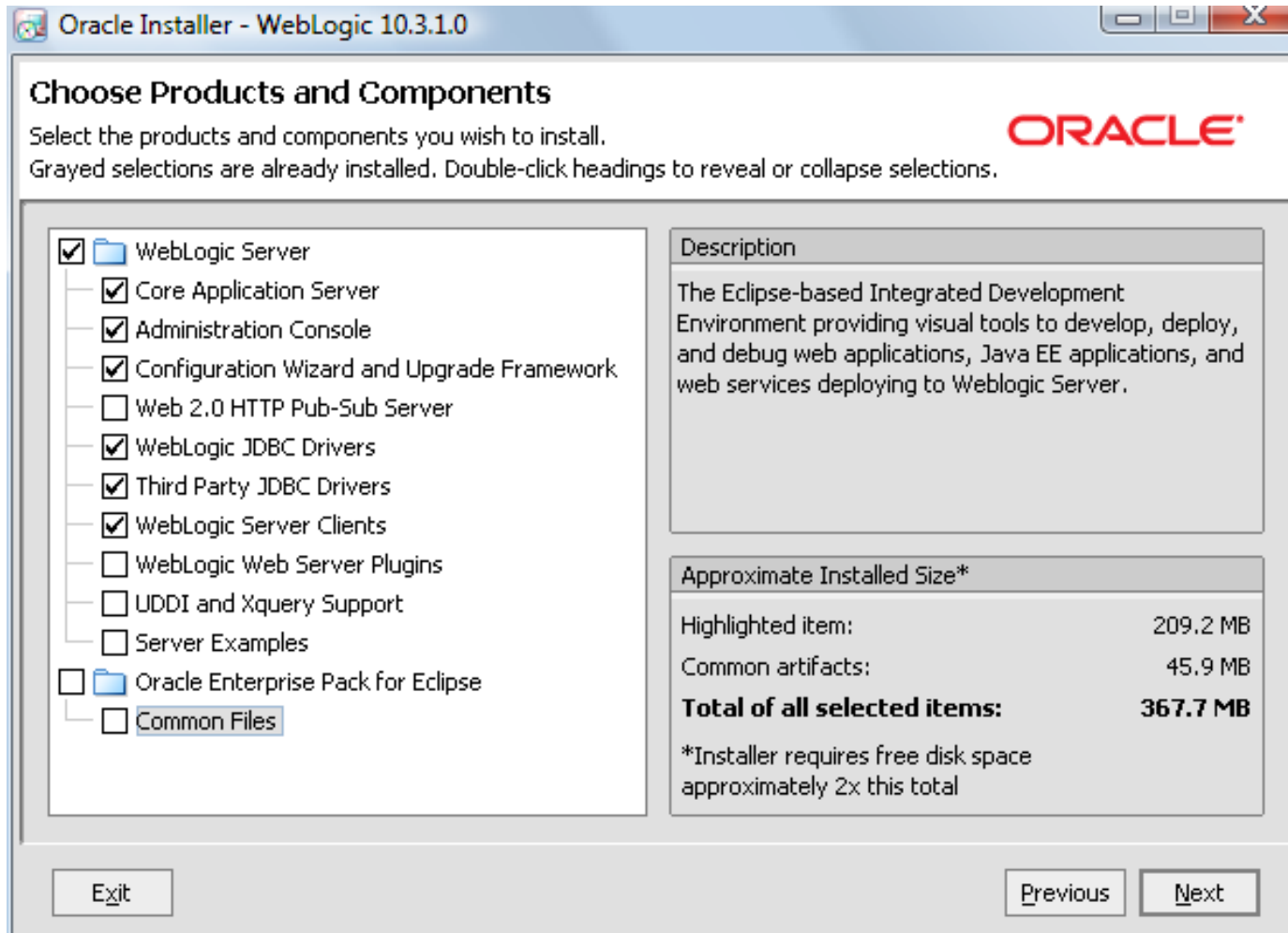
**unselect** the following options:

- Web 2.0 HTTP Pub-Sub Server
- WebLogic Web Server Plugins\*
- UDDI and Xquery Support
- Server Examples
- Common Files

and then Click **Next**.



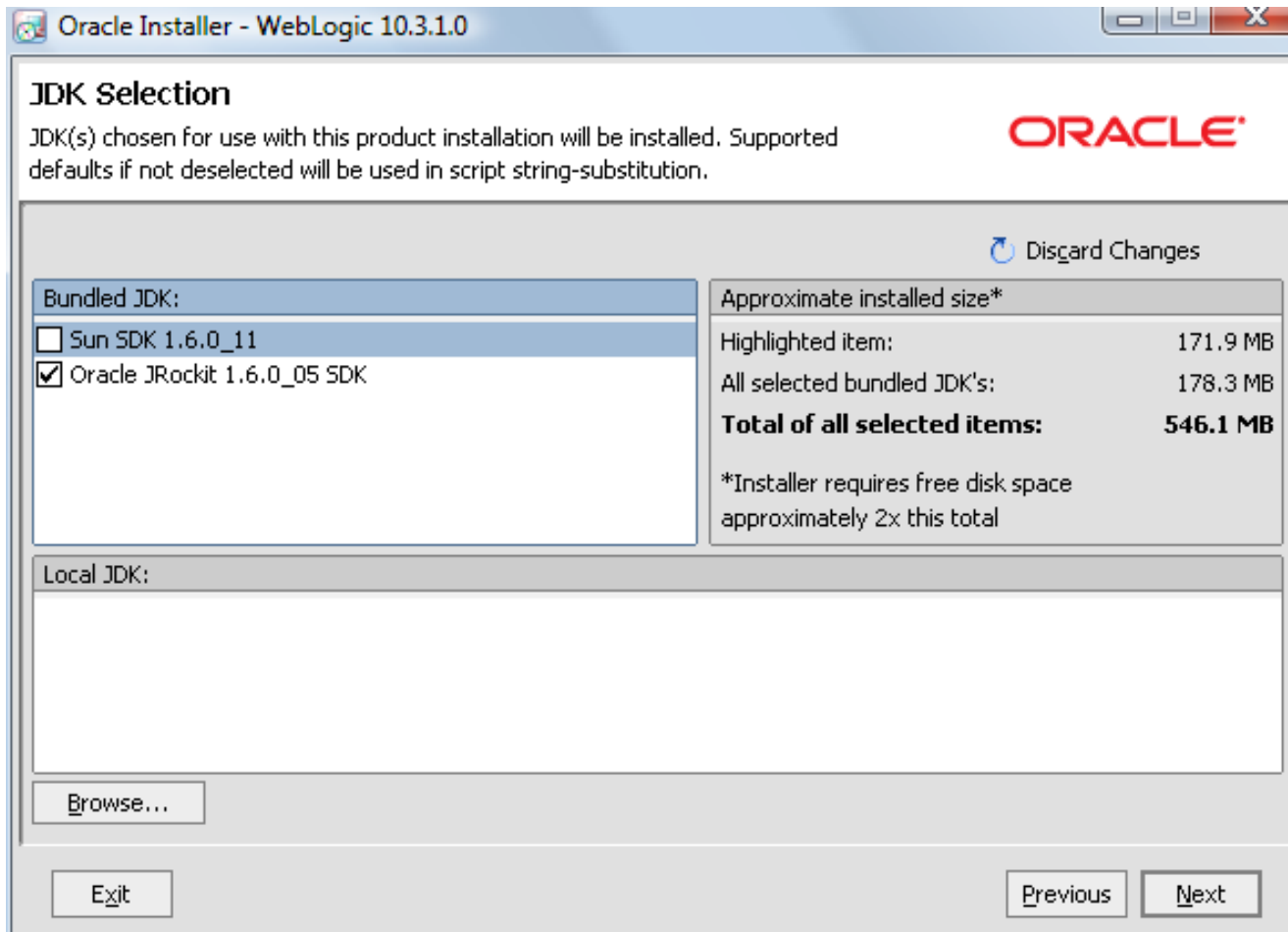
# WebLogic Package Installation



\* The WebLogic Web Server Plugins option may be required for production purposes if you use WLS clustering.

# WebLogic Package Installation

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

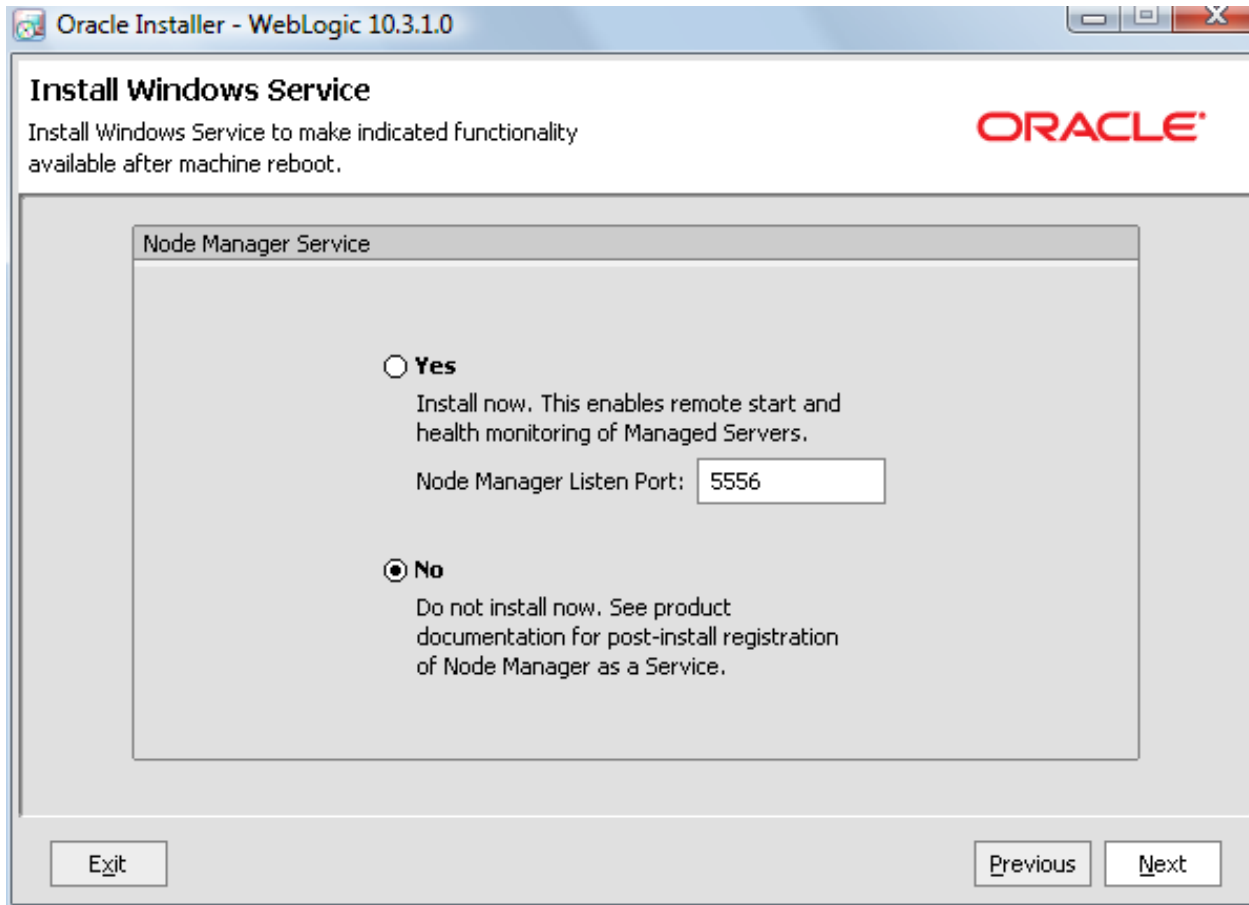


# WebLogic Package Installation

8. Choose Product Installation Directories page - leave the defaults and Click Next.
9. Install Windows Service page – we will not install the node manager so leave the No option and click Next.



# WebLogic Package Installation



- \* Those users interested in clustering may want the node manager option, however Duncan Mill's post mentions a classpath issue and so, if required, it is suggested to log an SR with Oracle Support to see how to configure it with a JDeveloper production install.

# WebLogic Package Installation

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

11. Installation Summary page – just click **Next**.

The installer will now runs the installation.

Once complete **unselect the Run Quickstart** option then select the **Done** button.





# Install ADF Runtime Libraries

## Requirements:

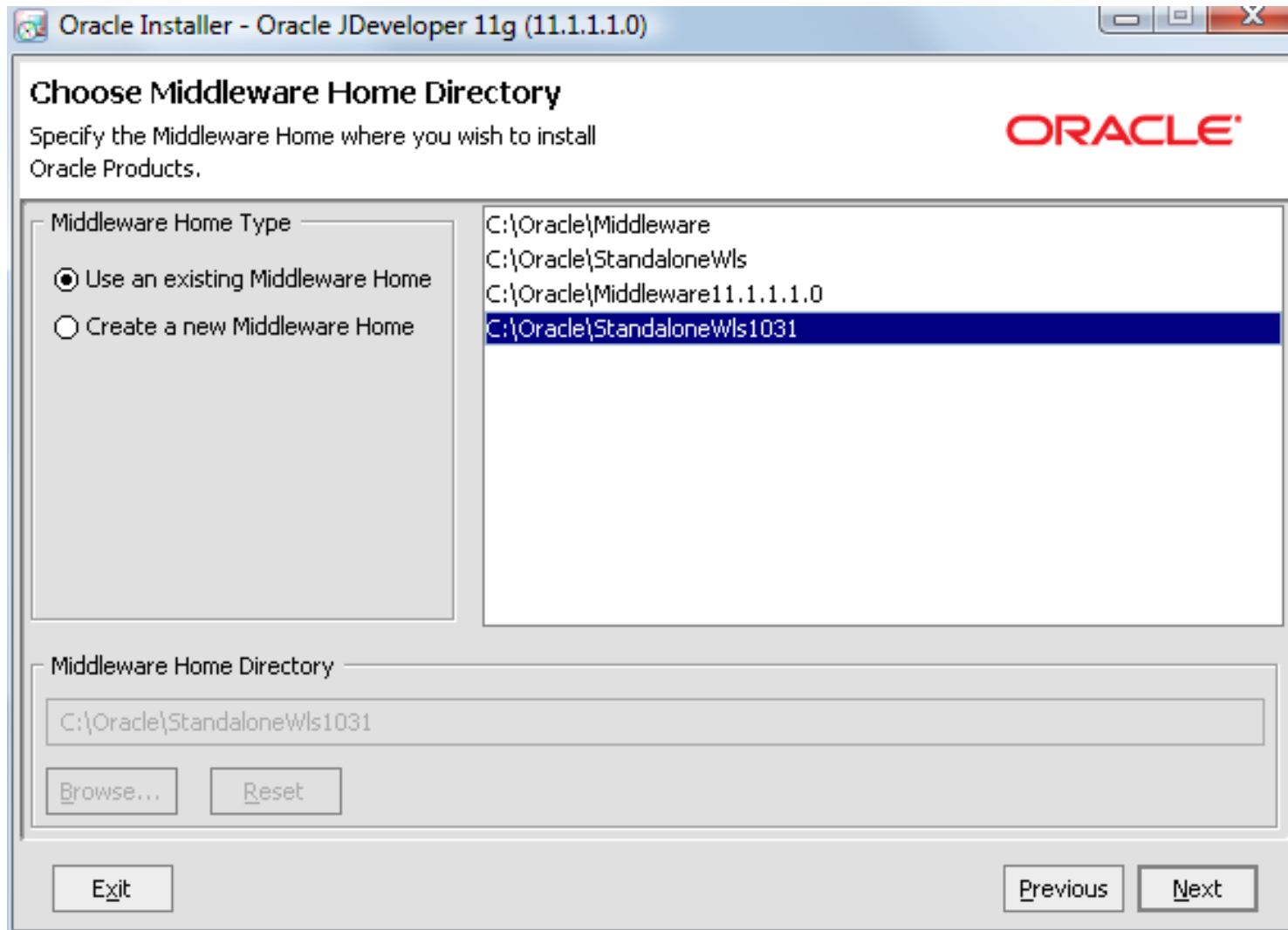
- WebLogic requires ADF Runtime libraries to support ADF Applications
- No standalone Runtimes installer  
    JDev Studio installer  
        (for ADF runtimes “only”)  
    or  
    Fusion Middleware Application Developer installer  
        (includes Enterprise Manager)

# Install ADF Runtime Libraries

## Installation Steps:

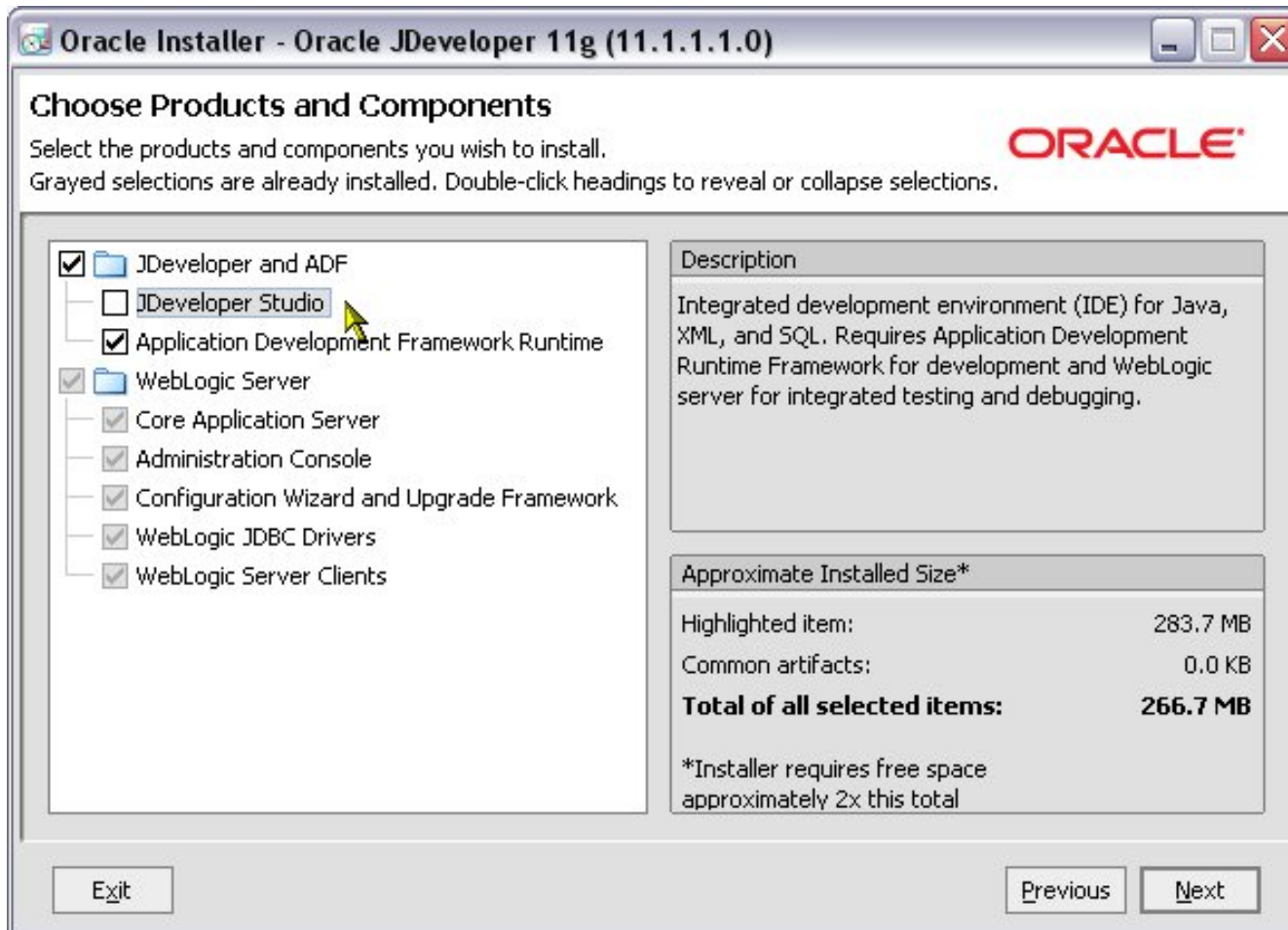
1. Run the JDeveloper 11g (11.1110) complete installer – `jdevstudio11111install.exe`
2. Welcome page - following unpacking the installer dialog will display starting with the Welcome page. Click **Next**.
3. Choose Middleware Home Directory page – select the **Use an existing Middleware Home** option and select the directory you previously installed WLS (e.g. `c:\Oracle\StandaloneWls1031`). Click **Next**.

# Install ADF Runtime Libraries



# Install ADF Runtime Libraries

4. Choose Products and Components page – you only need **Application Development Framework**. Click **Next**.



# Install ADF Runtime Libraries

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. **Unselect Run Quickstart** and then click the **Done** button.

**Note:** JDK Selection page – WLS is already installed and configured, WLS will continue to use JRocket. The choice on this screen is for Jdeveloper/ADF JDK version only; in which case leaving the default Sun JDK is fine.

# Create the Deployment Target Server

## Requirements:

- Use WebLogic Configuration Wizard
- Create an ADF ready Domain, including
  - default Admin Server
  - and
  - target Managed Server

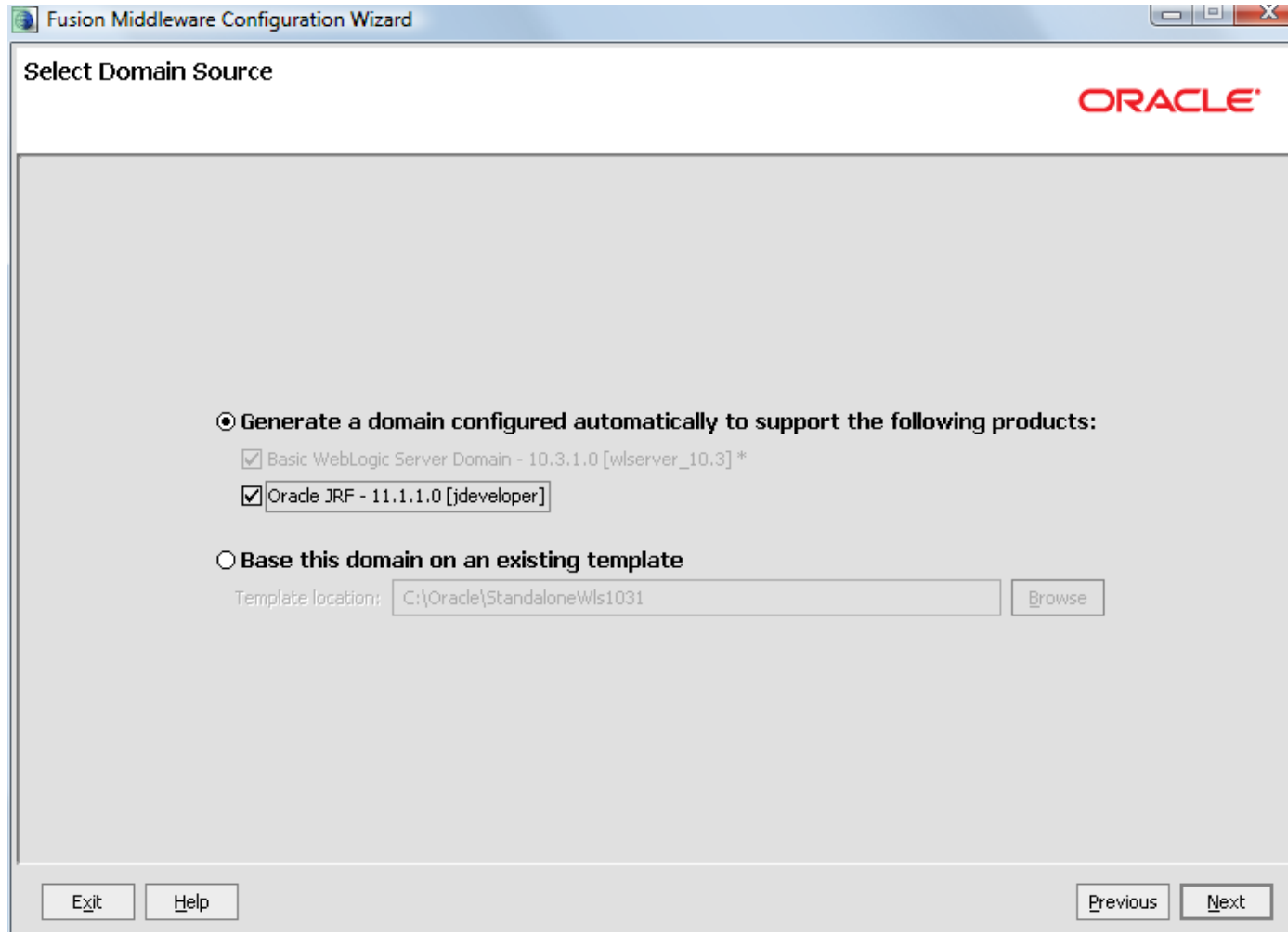


# Create the Deployment Target Server

The Configuration Wizard is available from  
Start → Programs → Oracle WebLogic → WebLogic  
Server 11gR1 → Tools → Configuration Wizard.

1. Welcome Page – there will be Create and Extend options. 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.


# Create the Deployment Target Server





# Create the Deployment Target Server

4. Specify Domain Name and Location page – change the domain name to **ADFDomain**, leave the location **default**.



The screenshot shows a window titled "Fusion Middleware Configuration Wizard" with a sub-header "Specify Domain Name and Location" and the Oracle logo. The main instruction is "Enter the name and location for the domain:". Below this, there are two input fields: "Domain name:" with the text "ADFDomain" and "Domain location:" with the text "C:\Oracle\StandaloneWls1031\user\_projects\domains". A "Browse" button is next to the domain location field. At the bottom, there are "Exit", "Help", "Previous", and "Next" buttons.

Fusion Middleware Configuration Wizard

Specify Domain Name and Location

ORACLE

Enter the name and location for the domain:

Domain name: ADFDomain

Domain location: C:\Oracle\StandaloneWls1031\user\_projects\domains Browse

Exit Help Previous Next

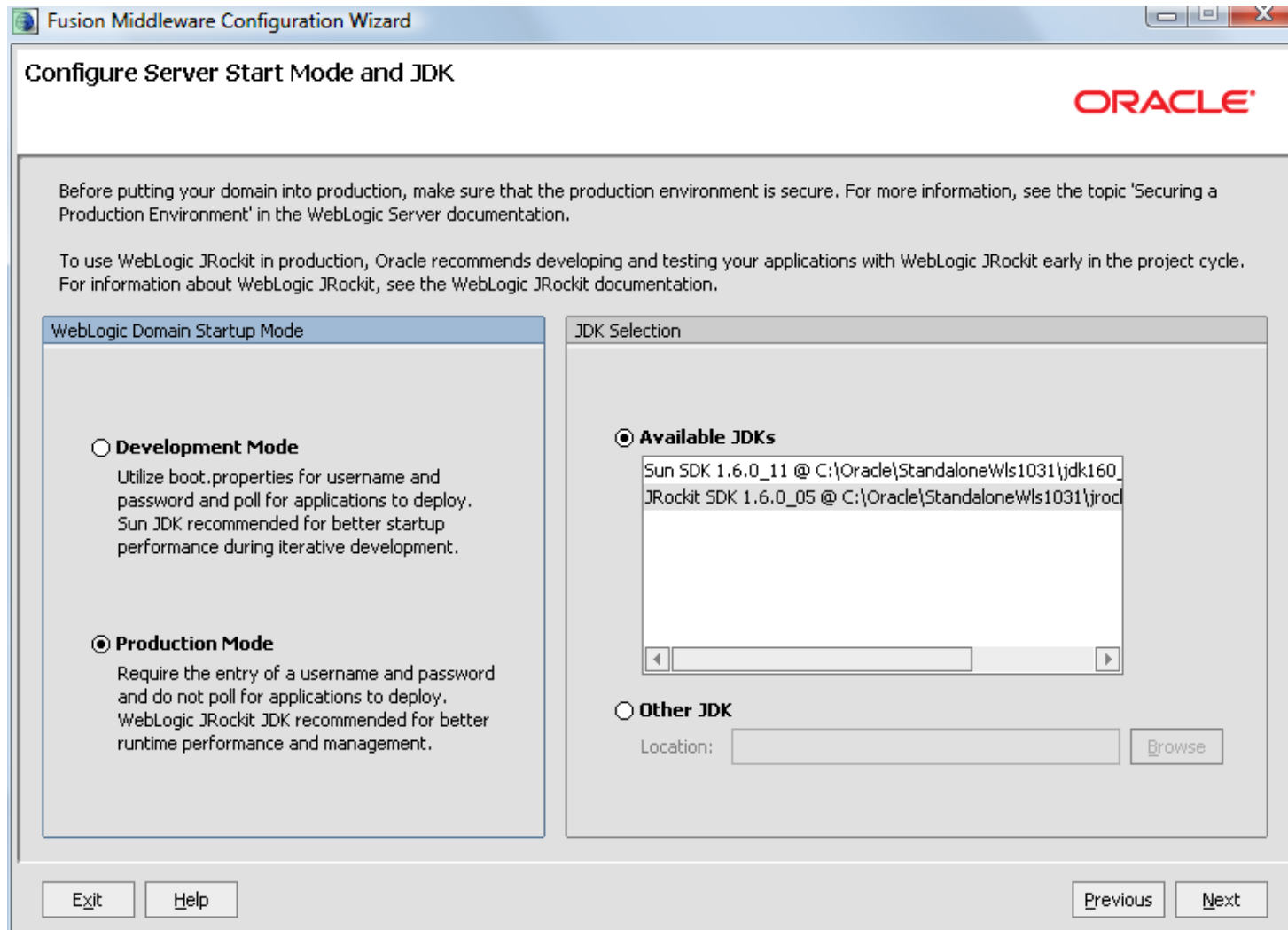
# Create the Deployment Target Server

5. Configure Administrator Username and Password page – enter a **user name** and **password** for the default WebLogic Server administrator.
6. 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 **JRocket** previously for the WLS install, this should be your choice here. Click **Next**.



# Create the Deployment Target Server



The image shows a screenshot of the 'Fusion Middleware Configuration Wizard' window, specifically the 'Configure Server Start Mode and JDK' step. The window has a title bar with the text 'Fusion Middleware Configuration Wizard' and standard Windows window controls. The main content area is divided into two panes. The left pane, titled 'WebLogic Domain Startup Mode', contains two radio button options: 'Development Mode' and 'Production Mode'. The 'Production Mode' option is selected. The right pane, titled 'JDK Selection', contains a radio button option 'Available JDKs' which is selected, and a list box showing two entries: 'Sun SDK 1.6.0\_11 @ C:\Oracle\StandaloneWls1031\jdk160\_' and 'JRockit SDK 1.6.0\_05 @ C:\Oracle\StandaloneWls1031\jrockit\_1.6.0\_05'. Below the list box is a horizontal scrollbar. There is also an 'Other JDK' option with a 'Location' text box and a 'Browse' button. At the bottom of the window are four buttons: 'Exit', 'Help', 'Previous', and 'Next'. The Oracle logo is visible in the top right corner of the main content area.

**Fusion Middleware Configuration Wizard**

**Configure Server Start Mode and JDK**

ORACLE

Before putting your domain into production, make sure that the production environment is secure. For more information, see the topic 'Securing a Production Environment' in the WebLogic Server documentation.

To use WebLogic JRockit in production, Oracle recommends developing and testing your applications with WebLogic JRockit early in the project cycle. For information about WebLogic JRockit, see the WebLogic JRockit documentation.

**WebLogic Domain Startup Mode**

☐ **Development Mode**  
Utilize boot.properties for username and password and poll for applications to deploy. Sun JDK recommended for better startup performance during iterative development.

☒ **Production Mode**  
Require the entry of a username and password and do not poll for applications to deploy. WebLogic JRockit JDK recommended for better runtime performance and management.

**JDK Selection**

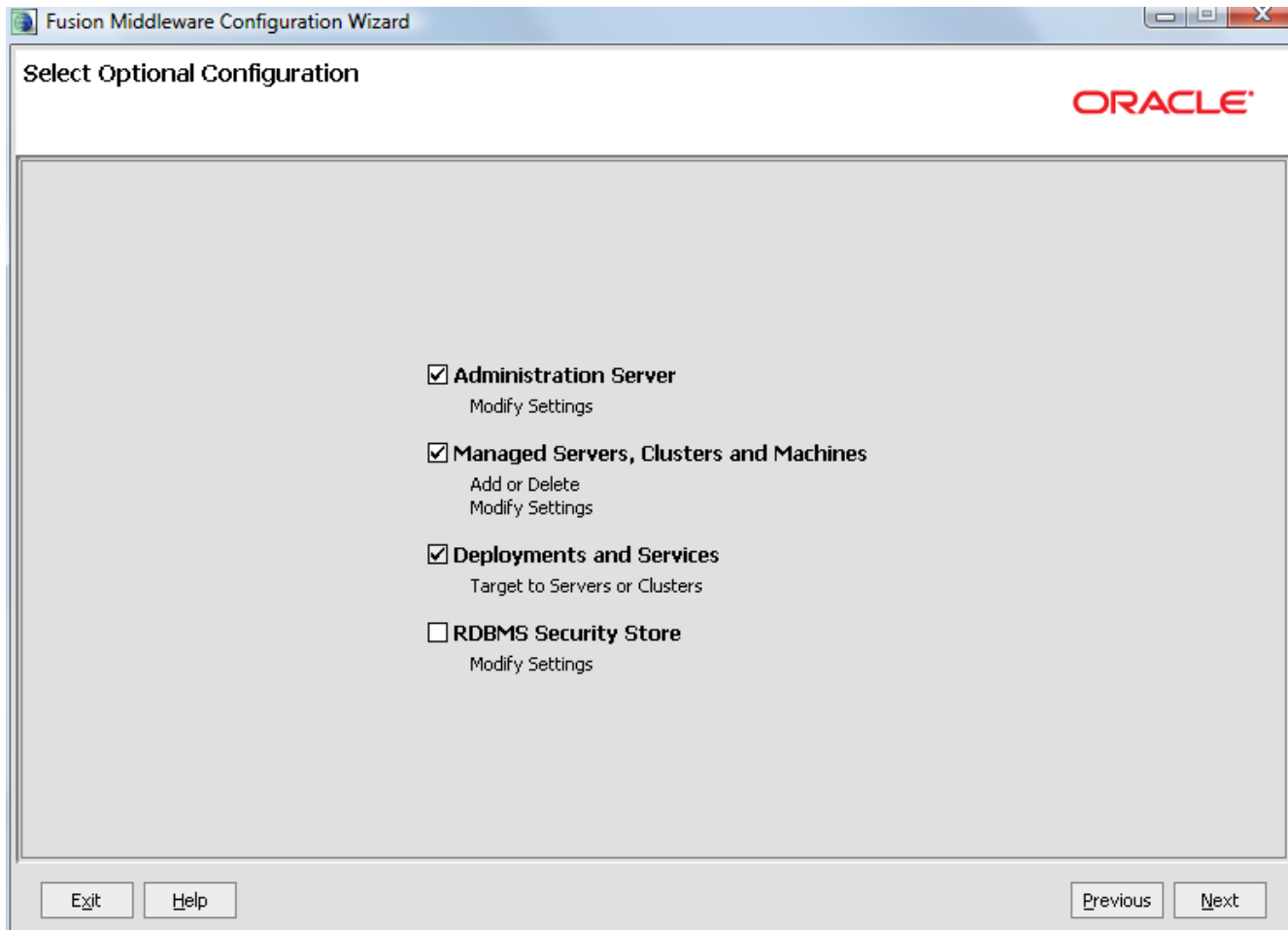
☒ **Available JDKs**

- Sun SDK 1.6.0\_11 @ C:\Oracle\StandaloneWls1031\jdk160\_
- JRockit SDK 1.6.0\_05 @ C:\Oracle\StandaloneWls1031\jrockit\_1.6.0\_05

☐ **Other JDK**  
Location:

# Create the Deployment Target Server

7. Select Optional Configuration page – select **yes** to all options except the RDBMS Security Store option.



# Create the Deployment Target Server

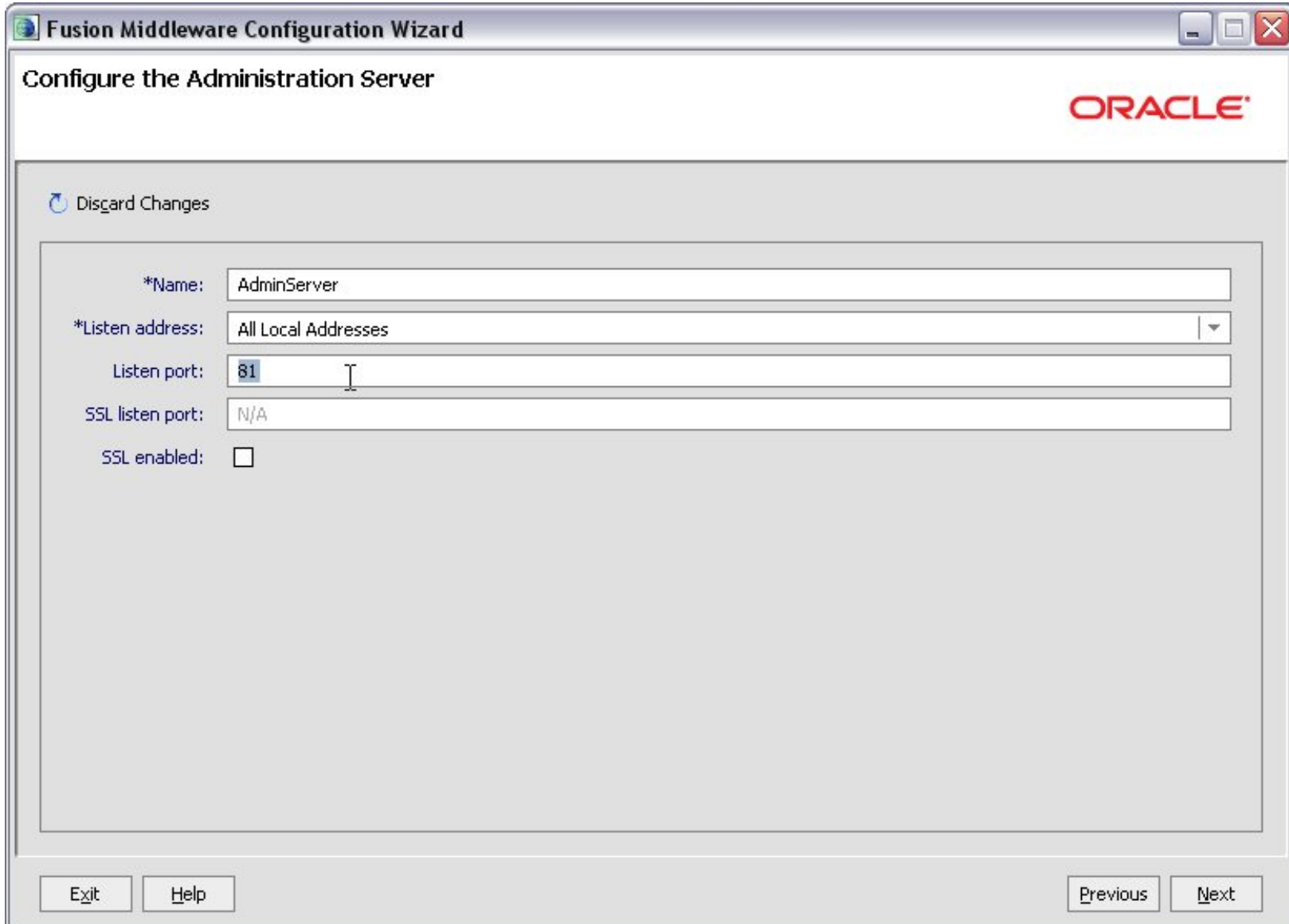
8. Configure the Administration Server page – each WLS domain includes 1 administration server by default referred to as the Admin Server.

The Admin Server's main role is to configure and maintain the other server instances under WLS.

This page allows you to configure the Admin Server. You can override these settings or leave the defaults. We have chosen to use **port 81**. Click **Next**.



# Create the Deployment Target Server



The image shows a screenshot of the 'Fusion Middleware Configuration Wizard' window. The title bar reads 'Fusion Middleware Configuration Wizard'. The main heading is 'Configure the Administration Server'. The Oracle logo is in the top right corner. Below the heading, there is a 'Discard Changes' button with a circular arrow icon. The configuration area contains several 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 four buttons: 'Exit', 'Help', 'Previous', and 'Next'.

Fusion Middleware Configuration Wizard

Configure the Administration Server

ORACLE

Discard Changes

\*Name: AdminServer

\*Listen address: All Local Addresses

Listen port: 81

SSL listen port: N/A

SSL enabled: ☐

Exit Help Previous Next

# Create the Deployment Target Server

9. Configure Managed Servers page – we will create a separate Managed Server to host our applications.

Click **Add** to add a new Managed Server definition. Check the **port number is different** from the Admin Server.

In our example we **override the defaults as shown**. Click **Next**.



# Create the Deployment Target Server

Fusion Middleware Configuration Wizard

Configure Managed Servers

ORACLE

+ Add    ✕ Delete    ↺ Discard Changes    Switch Display

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

Exit    Help    Previous    Next



# Create the Deployment Target Server

10. Configure Clusters page – we are ignoring clustering for this exercise. Click **Next**.
11. Configure Machines page – machine definitions represent physical hardware and allow control of servers via the Admin console. However it is most useful for Clustering and requires Node Manager.

For this exercise we will just use local commands to control our servers.

**Skip** this and click **Next**.



# Create the Deployment Target Server

12. Target Deployments to Clusters or Servers – we could choose to deploy the ADF libraries to only the Managed Server here.

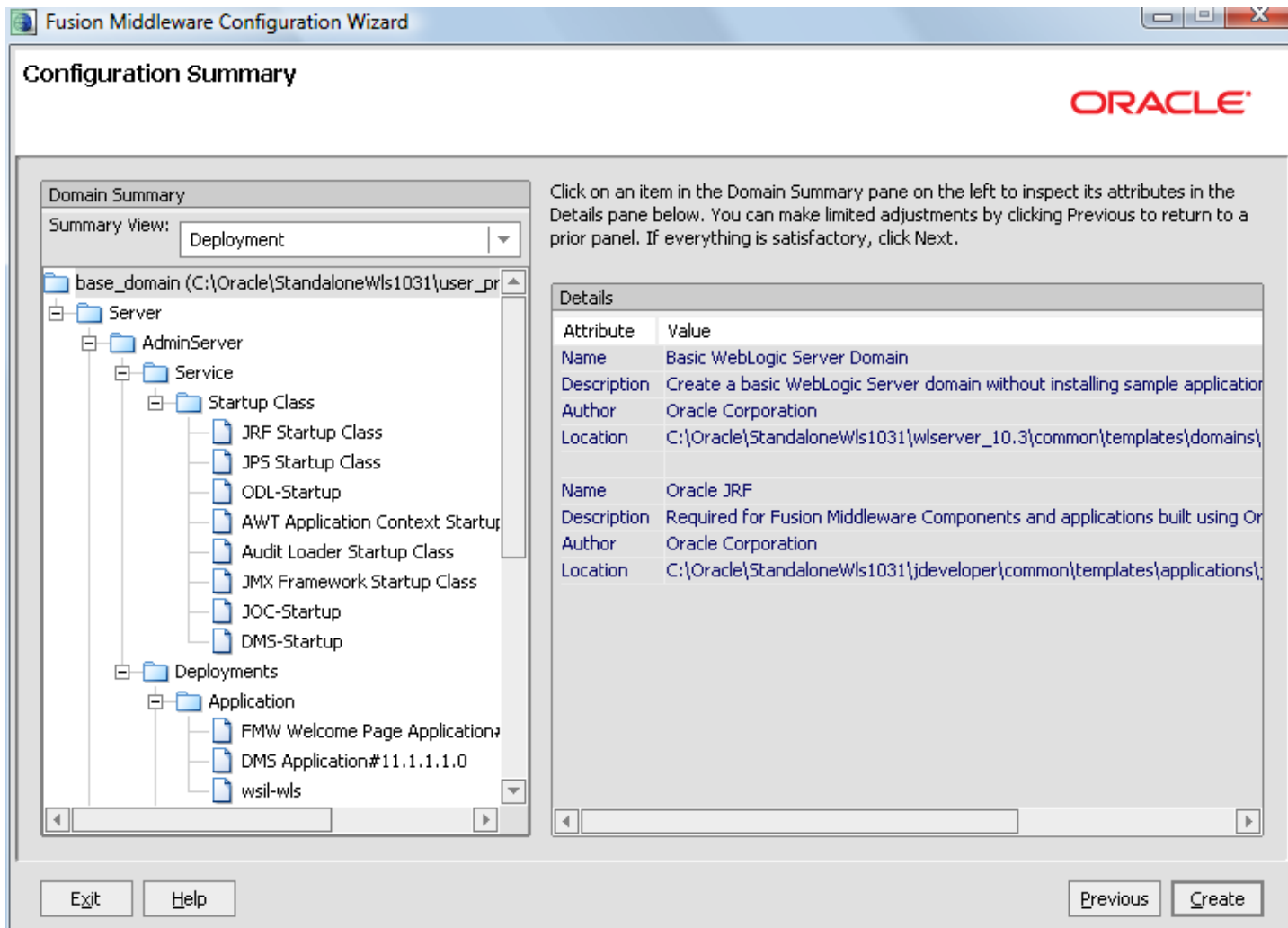
However we will leave it as the **default** and share the libraries later. Click **Next**.

13. Target Services to Clusters or Servers – **skip** this step. Click **Next**.



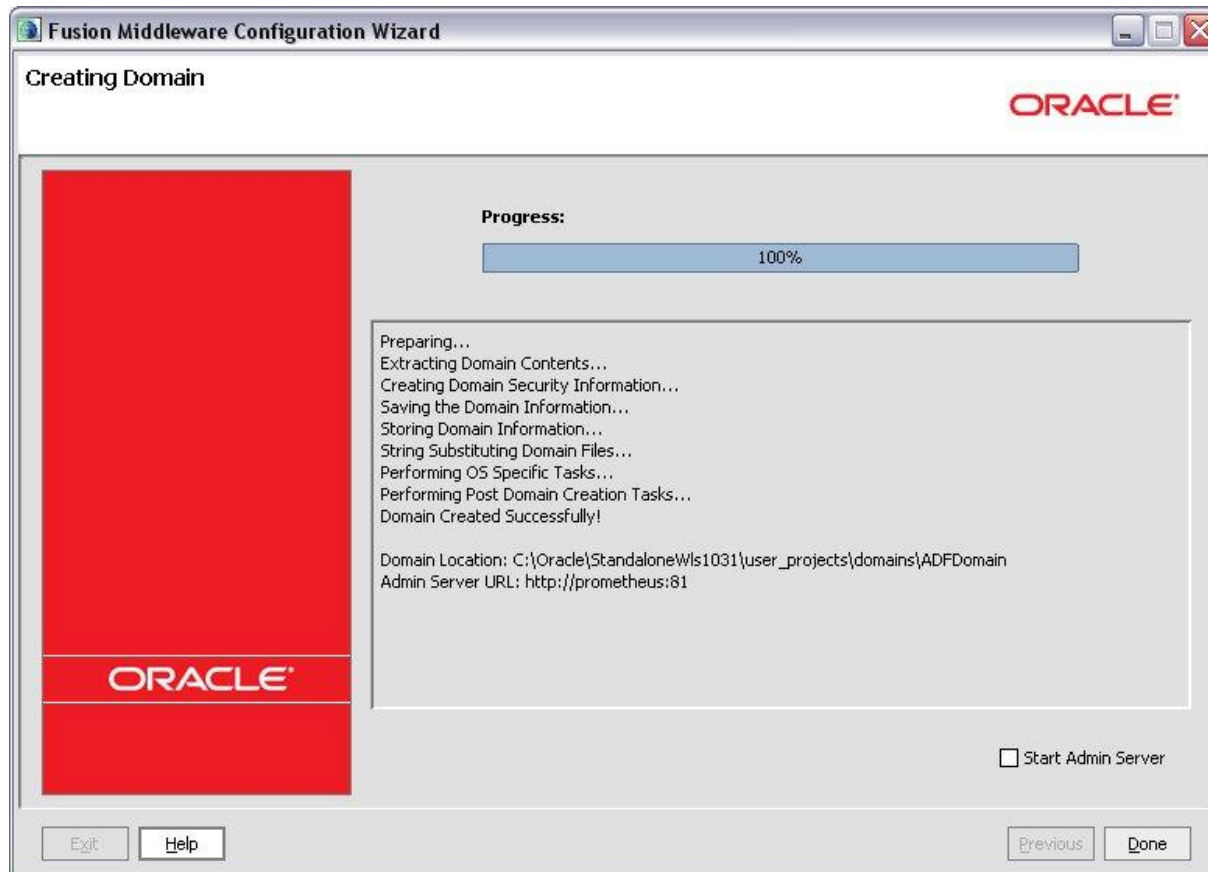
# Create the Deployment Target Server

14. Review WebLogic Domain page – Click **Next**.



# Create the Deployment Target Server

15. Creating Domain page (100%) - at this point your domain, admin & managed servers are correctly installed and configured, but not running. Click **Done**.



# Share the ADF runtime libraries

## Requirements:

- Start the Admin Server
- Review Domain configuration
- Share ADF libraries to target Managed Server

# Share the ADF runtime libraries

Local commands for your new domain will now be available within a User Project sub menu group under Oracle WebLogic.

1. Start the Admin Server using  
Start → Programs → Oracle WebLogic →  
User Projects → ADFDomain → **Start Admin Server...**
2. Enter Administrator **username & password** at the prompt in the command window.
3. Wait for the final message –  
“<Server started in RUNNING mode>”

# Share the ADF runtime libraries

Now we can Login to the WebLogic Admin Server Console.

Invoke the Admin Console using Start → Programs → Oracle WebLogic → User Projects → ADFDomain → Admin Server Console



Again enter the Administrator **username & password** you specified at installation.



# Share the ADF runtime libraries

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



Click the **Deployments** node.



# Share the ADF runtime libraries

Under the Summary of Deployments page you will see the new ADF Runtime Libraries 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 16 Previous | Next

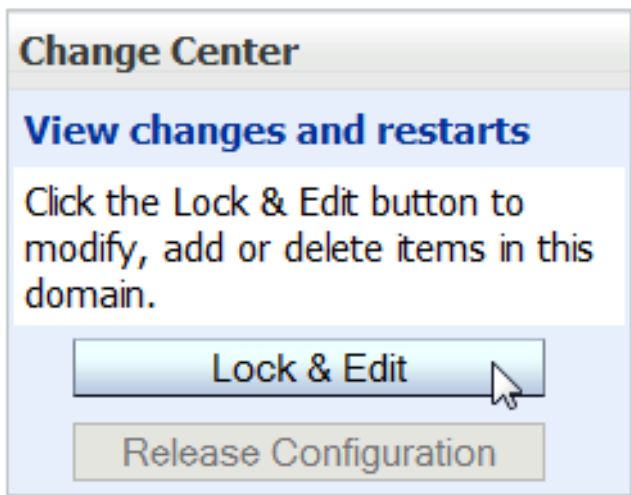
<input type="checkbox"/> Name	State	Health	Type	Deployment Order
<input type="checkbox"/> adf.oracle.domain(1.0,11.1.1.1.0)	Active		Library	100
<input type="checkbox"/> adf.oracle.domain.webapp(1.0,11.1.1.1.0)	Active		Library	100
<input type="checkbox"/> DMS Application (11.1.1.1.0)	Active	✓ OK	Web Application	190
<input type="checkbox"/> FMW Welcome Page Application (11.1.0.0.0)	Active	✓ OK	Enterprise Application	150
<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-ux(5,5.0)	Active		Library	100
<input type="checkbox"/> oracle.adf.dconfigbeans(1.0,11.1.1.0.0)	Active		Library	100
<input type="checkbox"/> 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

# Share the ADF runtime libraries

The libraries are currently only configured to be available for the Admin Server.

We now need to share them to the target host Managed Server.



Our domain is in Production mode.

To make changes we first need to invoke the “**Lock & Edit**” option at the top left of the screen.

# Share the ADF runtime libraries

For each library, [click the link](#),

**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

# Share the ADF runtime libraries

then the **Targets** tab.

Then select the **ManagedServer** and click **Save**.

The screenshot shows a web-based configuration window titled "Settings for adf.oracle.domain(1.0,11.1.1.1.0)". It has three tabs: "Overview", "Targets" (which is selected), and "Notes". Below the tabs is a "Save" button. A descriptive text says: "Use this page to select the WebLogic Server instances and clusters to which you want to deploy (target) the Java EE library." Below this is a table with the heading "Servers". The table contains two rows: "AdminServer" and "ManagedServer", both with checked checkboxes. At the bottom of the window is another "Save" button with a mouse cursor hovering over it.

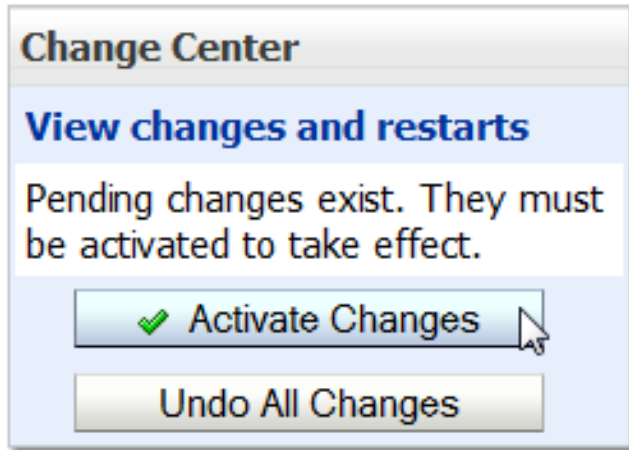
Servers
<input checked="" type="checkbox"/> AdminServer
<input checked="" type="checkbox"/> ManagedServer

You can click on the **Deployments** node in the Domain Structure portlet again to return to the libraries list.

This process needs to be repeated for **each ADF library**.

# Share the ADF runtime libraries

Once all libraries are shared, we need to make the changes permanent.



Our domain is in Production mode.

To finalise changes we need to confirm with “**Activate Changes**” at the top left of the screen.

The ADF libraries will now be shared and available to both the Admin Server and our target host Managed Server for our ADF applications.

# Start the target Managed Server

We should now start the Managed Server ready for Deployment of our application from JDeveloper.

1. Run the following command in a command window to start our Managed Server:

```
<wls_home_dir>\user_projects\domains\  
(your domain name)\bin\startManagedWeblogic.cmd <your managed  
server name> http://<admin server host>:<admin  
server port>
```

For our exercise that is:

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

# Start the target Managed Server

2. Enter Administrator **username & password** at the prompt in the command window.
3. Wait for the final message –  
“<Server started in RUNNING mode>”

You can now inspect the summary and state of your Servers including the now running Managed Server via the Admin Console.

Return to the Admin Console and select the **Environment** → **Servers** node in the Domain Structure portlet.


# Start the target Managed Server

**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

<input type="checkbox"/>	Name 	Cluster	Machine	State	Health	Listen Port
<input type="checkbox"/>	AdminServer(admin)			RUNNING	✓ OK	81
<input type="checkbox"/>	ManagedServer			RUNNING	✓ OK	80

New Clone Delete

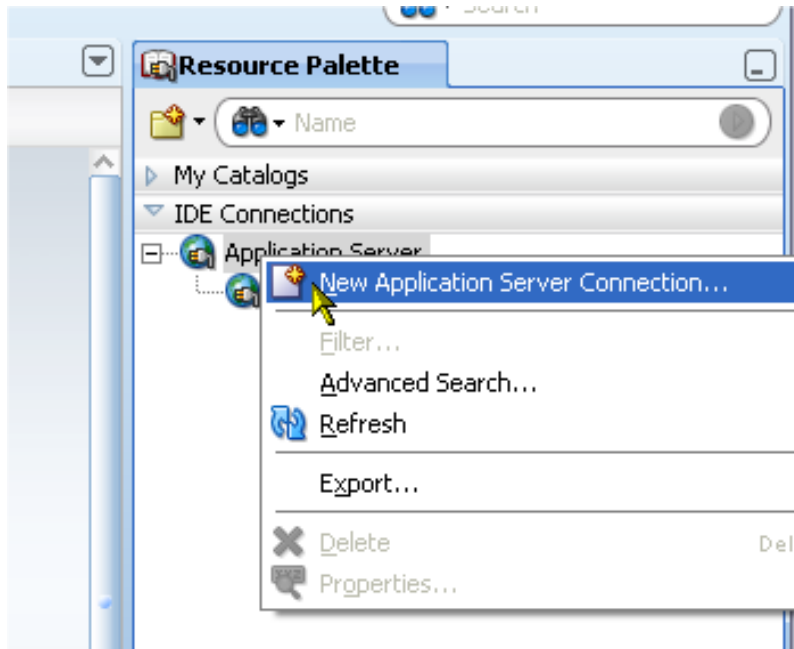
Showing 1 to 2 of 2 Previous | Next

To shut down both the new Managed server and the AdminServer, simply **Ctrl-C** them in the command prompts, upon which they will gracefully shut down.



# Connect JDev to WebLogic

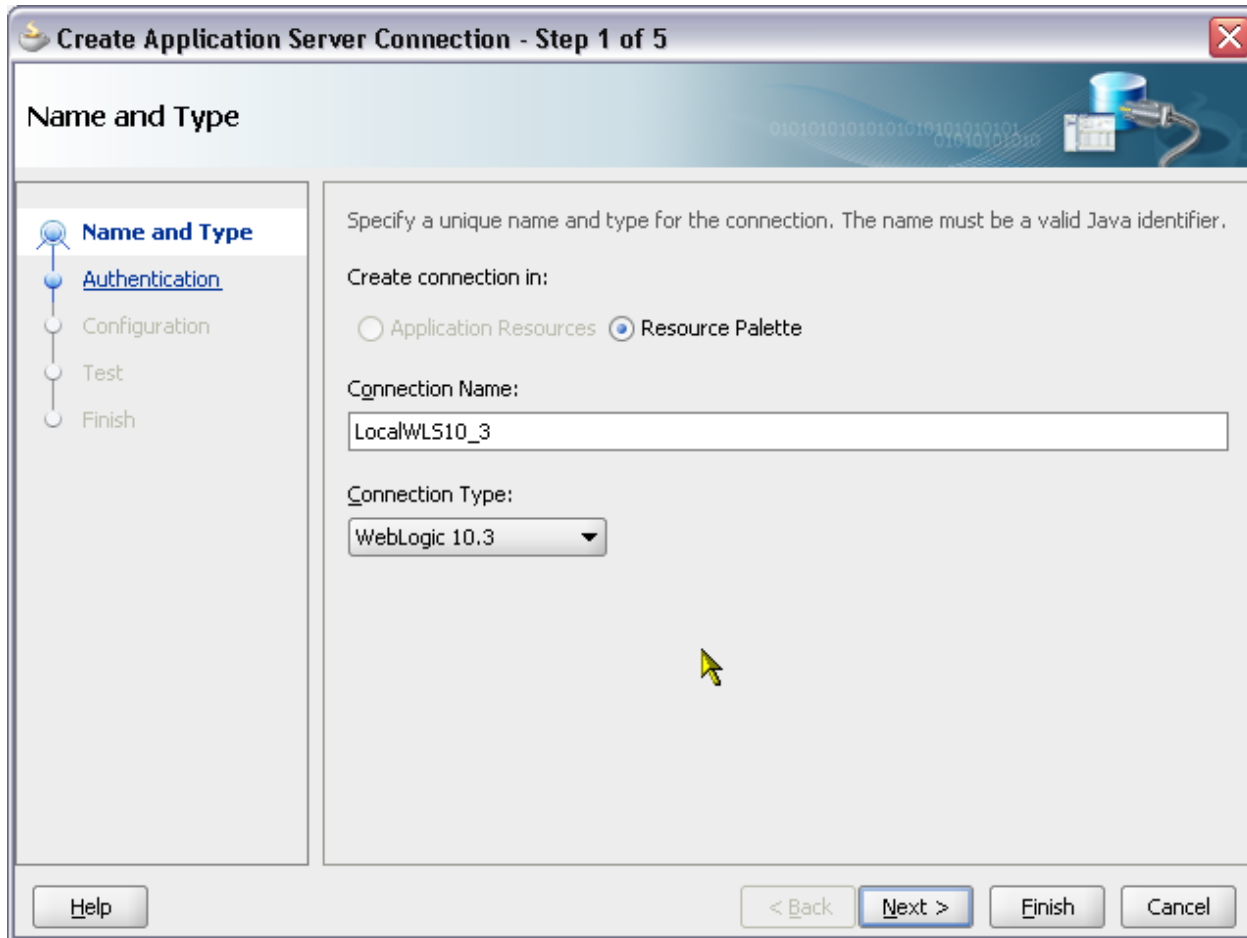
In order to deploy our application from “within” JDeveloper we need to create an Application Server Connection in the JDev IDE Connections Navigator.



1. Create a new Application Server connection from the context menu  
Right click on the Application Server node in the IDE Connections Navigator.  
By default the Navigator is on the upper right side of the IDE.  
Choose the New Application Server Connection option.

# Connect JDev to WebLogic

2. Give the connection a **name** and select the connection type of **WebLogic 10.3**. Click **Next**.



# Connect JDev to WebLogic

3. Provide the **credentials** to connect to the AdminServer. **Ignore the Deploy Password** checkbox, it is only relevant for an Application Resource connection and it would generally be inadvisable to deploy the password for a production application resource connection anyway. Click **Next**.
4. Specify the AdminServer context connection details including **hostname, AdminServer port and Domain**.



# Connect JDev to WebLogic

Create Application Server Connection - Step 3 of 5

Configuration

WebLogic Server connections use a host name and port to establish a connection. The Domain of the target will be verified

Weblogic Hostname (Administration Server):  
localhost

Port: 81 SSL Port: 7002

☐ Always use SSL

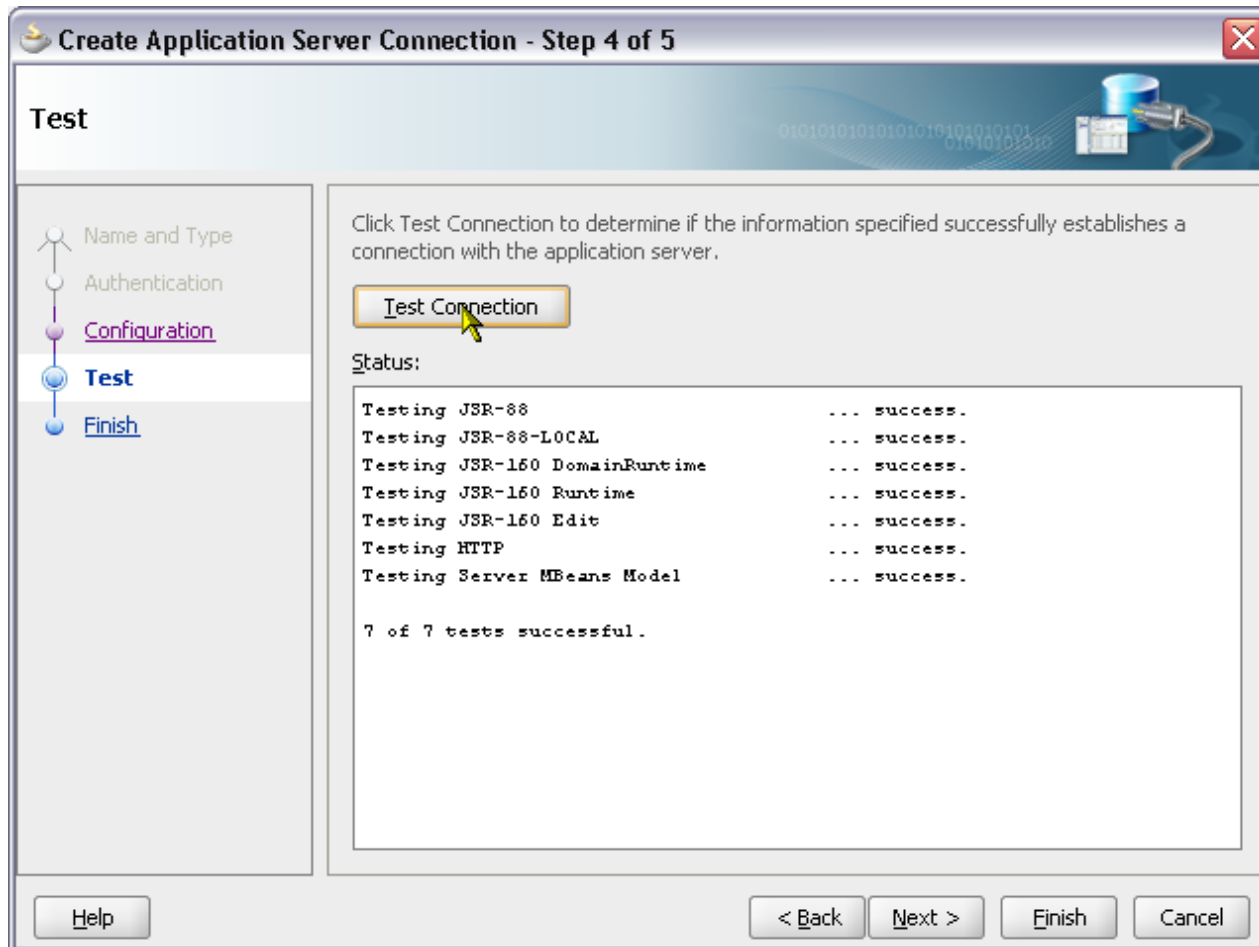
WLS Domain:  
ADFDomain

Help < Back Next > Finish Cancel

Following on from previous installation and configurations sections our values are as shown. Ignore SSL for the exercise. Click Next.

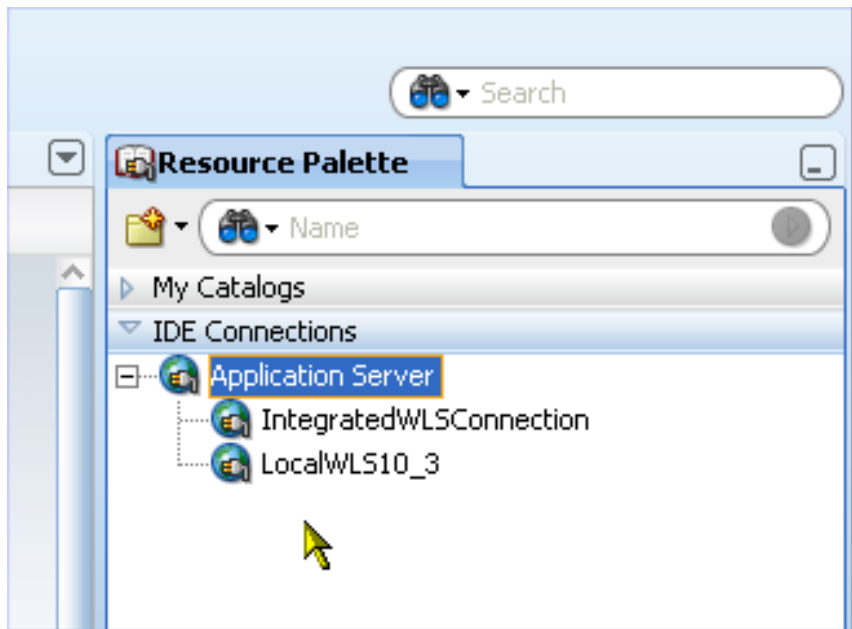
# Connect JDev to WebLogic

5. Click Test Connection for JDev to run tests to validate the WLS connection. Click **Next or Finish**.



# Connect JDev to WebLogic

If you clicked Next rather than Finish from the Test page then you will get a confirmation of the connection creation. Then click **Finish** to return to the JDev IDE.



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

# JDBC Data Sources

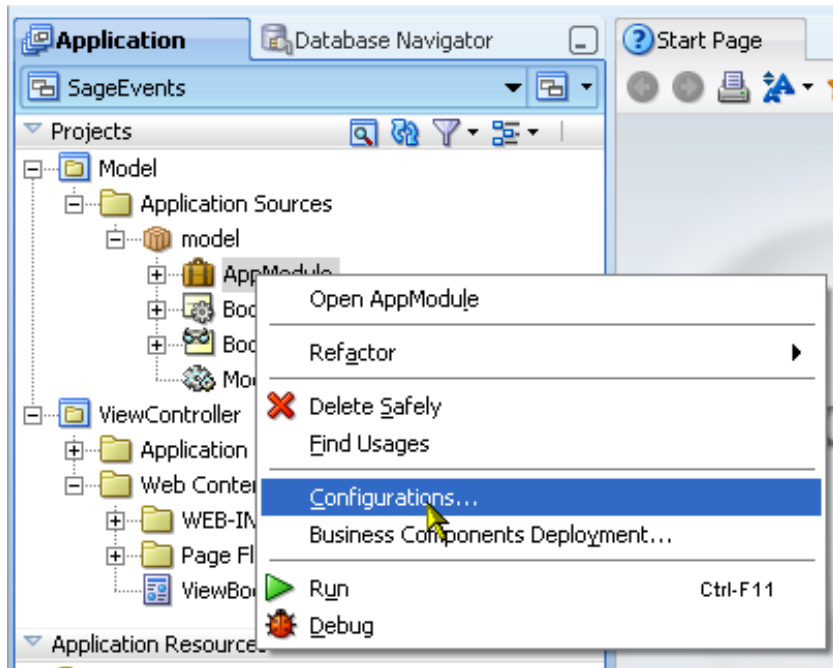
## Requirements:

- Application Database connectivity, options  
JDBC URL  
(for Development & within JDev IDE)  
or  
JDBC Data Sources  
(Production & generic)
  - No embedded credentials
  - Dev to Test to Prod without rebuild
  - Secrecy
  - Better app module pooling support...



# JDBC Data Sources

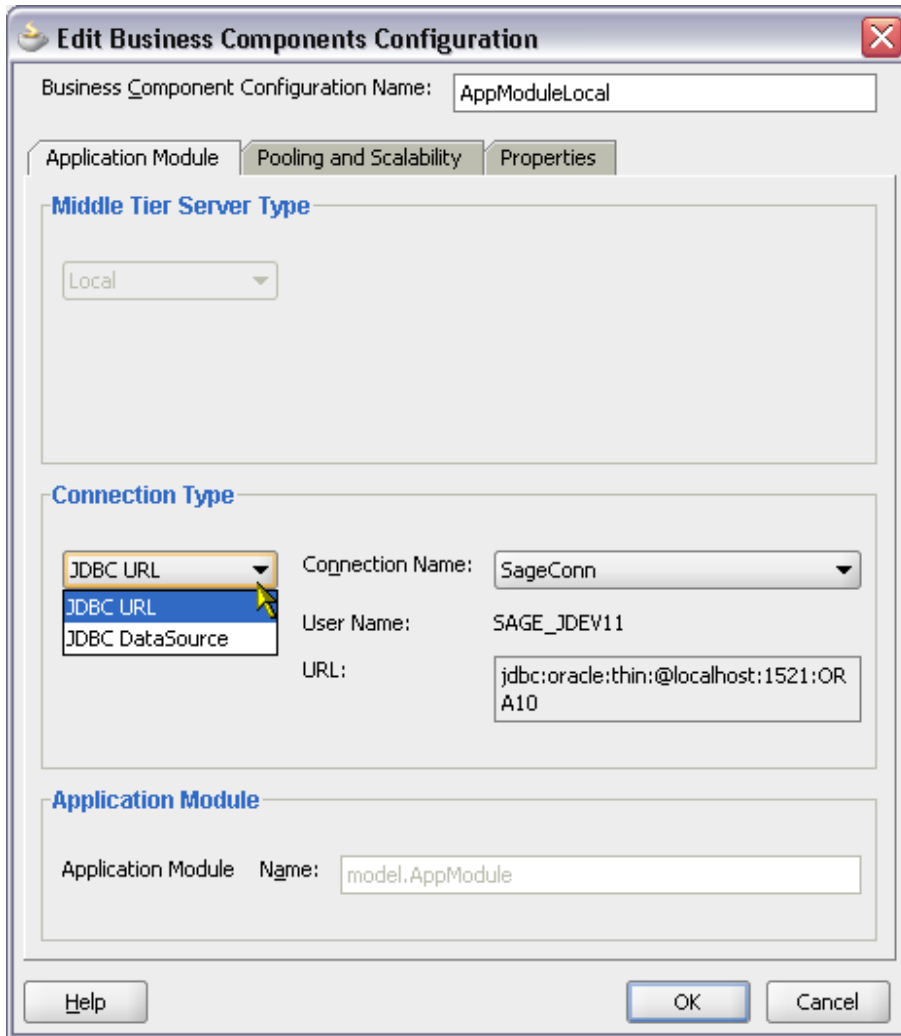
We need to check & if needed change our application to use a Data Source & then create a matching Data Source in WebLogic.



1. 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.

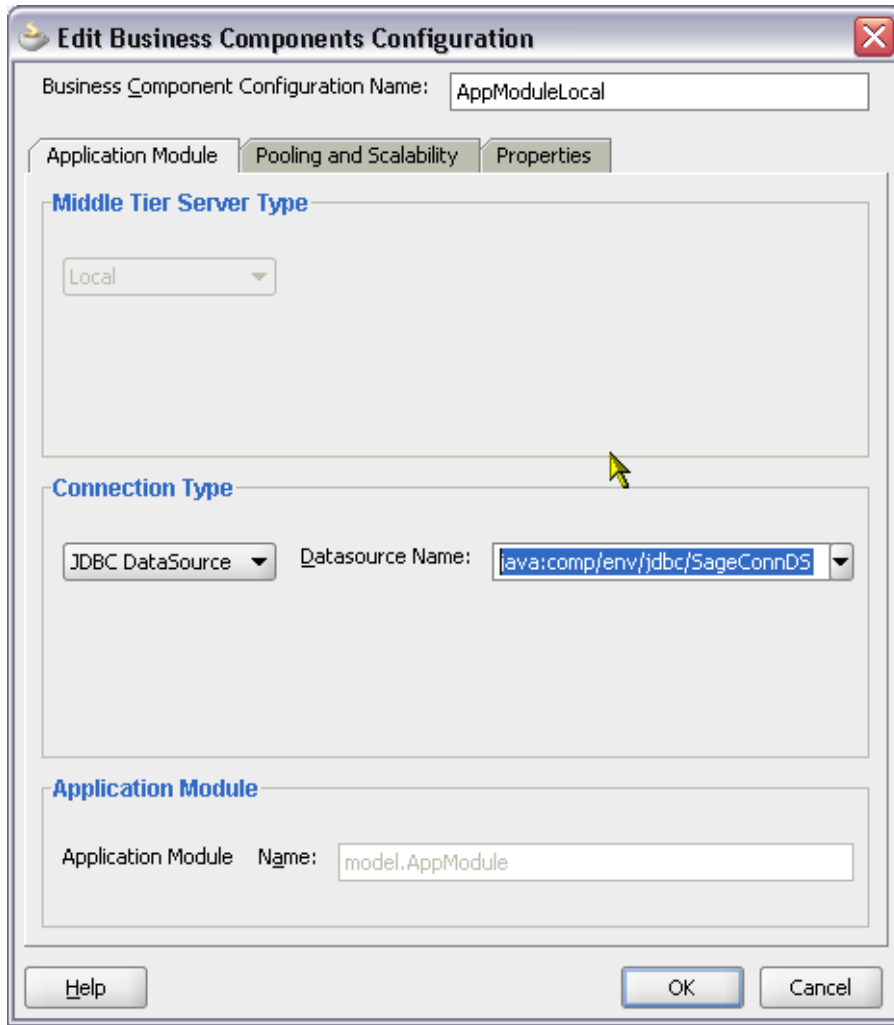


# JDBC Data Sources



2. Click on the **Edit** button for **AppModuleLocal**.
3. On the **Application Module** Tab in the **Connection Type** panel, assuming the type is currently set to JDBC URL, change the type to **JDBC Data Source**.

# JDBC Data Sources



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 exit.



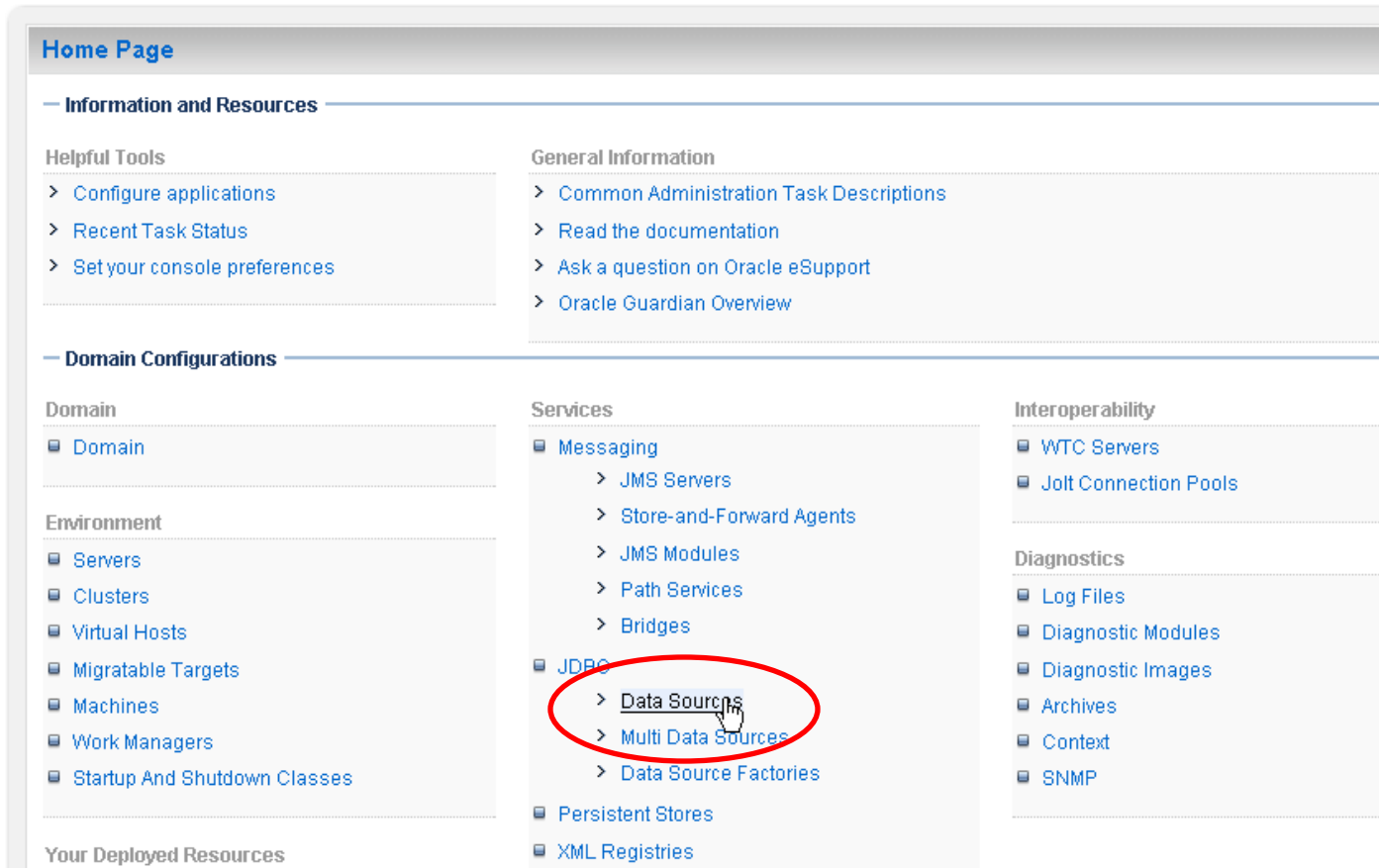
# JDBC Data Sources

We now configure a WLS Data Source to match the “name” defined for our application.

1. **Login** to the Admin Console (as previously covered).
2. 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.



# JDBC Data Sources



As previously covered our domain is in Production mode.

Again we first need to invoke the “**Lock & Edit**” option at the top left of the screen.

# JDBC Data Sources

3. Click **New** on the Summary of JDBC Data Sources panel to create a new definition.
4. JDBC Data Source Properties page - enter the **name** and **JNDI name** as defined for your application AppModuleLocal configuration within JDeveloper.

**Note:** specify **only the name without the prefix** such as `java:comp/env/jdbc/`. And in the **JNDI Name** field enter the Data Source name in the format **`jdbc/<Datasource_name>`**;  
do not use the full `java:comp/env/jdbc/` prefix,  
only `jdbc/`



# JDBC Data Sources

**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:**

---

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:**  ▼

---

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:**  ▼

Back Next Finish Cancel

Select **Oracle**  
as the **Database Type**  
and  
**Oracle's Driver (Thin) for Instance...**  
as the **Database Driver**.  
Click **Next**.

# JDBC Data Sources

Specifying the Datasource name within the JDeveloper IDE and the Datasource Name and JNDI Name in the formats described 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  
as well as the  
target standalone WLS Managed Server

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

# JDBC Data Sources

5. Transaction Options page – leave the **defaults** and Click **Next**.
6. Connection Properties page – enter **standard connection details** for the Database connection that your application requires. Click **Next**.





# JDBC Data Sources

**Create a New JDBC Data Source**

Back

Next

Finish

Cancel

---

**Connection Properties**

Define Connection Properties.

---

What is the name of the database you would like to connect to?

**Database Name:**

---

What is the name or IP address of the database server?

**Host Name:**

---

What is the port on the database server used to connect to the database?

**Port:**

---

What database account user name do you want to use to create database connections?

**Database User Name:**

---

What is the database account password to use to create database connections?

**Password:**

**Confirm Password:**

Back

Next

Finish

Cancel

# JDBC Data Sources

7. Test Database Connection page – on this page you can click the **Test Connection** button to test the connection.

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

8. Select Targets page – ensure that the **Data Source** is made available to the **Managed Server** and optionally the Admin Server. Click **Finish**.

# JDBC Data Sources

9. Summary of JDBC Data Sources page – you are now returned where you see your **new Data Source** listed.

### Summary of JDBC Data Sources

A JDBC data source is an object bound to the JNDI tree that provides database connectivity through a pool of JDBC connections. . source on the JNDI tree and then borrow a database connection from a data source.

This page summarizes the JDBC data source objects that have been created in this domain.

[Customize this table](#)

**Data Sources(Filtered - More Columns Exist)**

NewDelete

<input type="checkbox"/>	Name ^	JNDI Name	Targets
<input type="checkbox"/>	SageConnDS	jdbc/SageConnDS	AdminServer, ManagedServer

NewDelete

Sho

As previously covered our domain is in Production mode.

Again to finalise changes we need to confirm with “**Activate Changes**” at the top left of the screen.

# Deployment (theory)

- Archive files - JARs, WARs, EARs

The JEE specification defines standards for the creation of archive files. Archives contain standard file packages, locations and configuration files. JEE Servers make use of these when deploying an application.

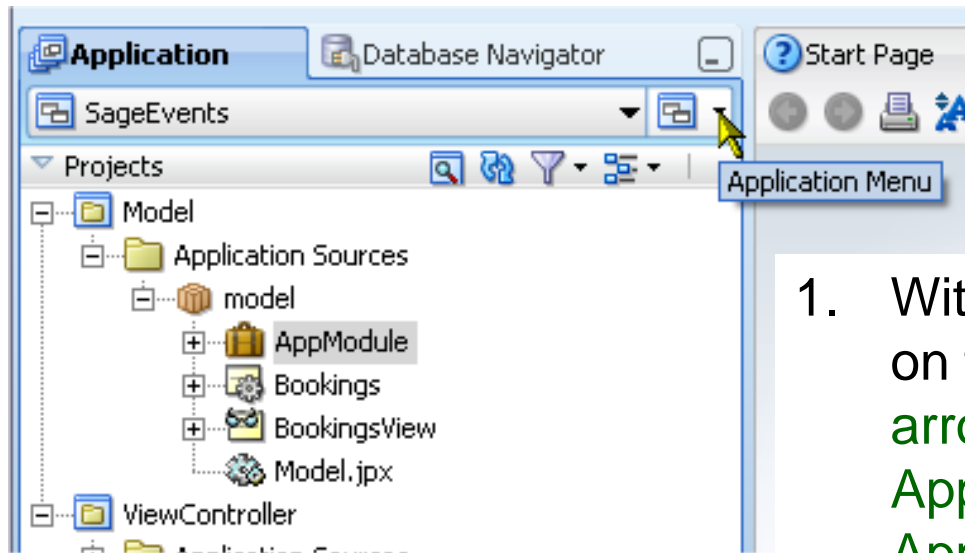
- Deployment Profiles

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.



# Deployment (Auto-Deploy)

After all.... this preparation we finally get to Deploy; with a single click, 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 WebLogic server.



1. Within the JDeveloper IDE click on the **Application Menu** down arrow to the right of your **Application** in the **Application Navigator**.

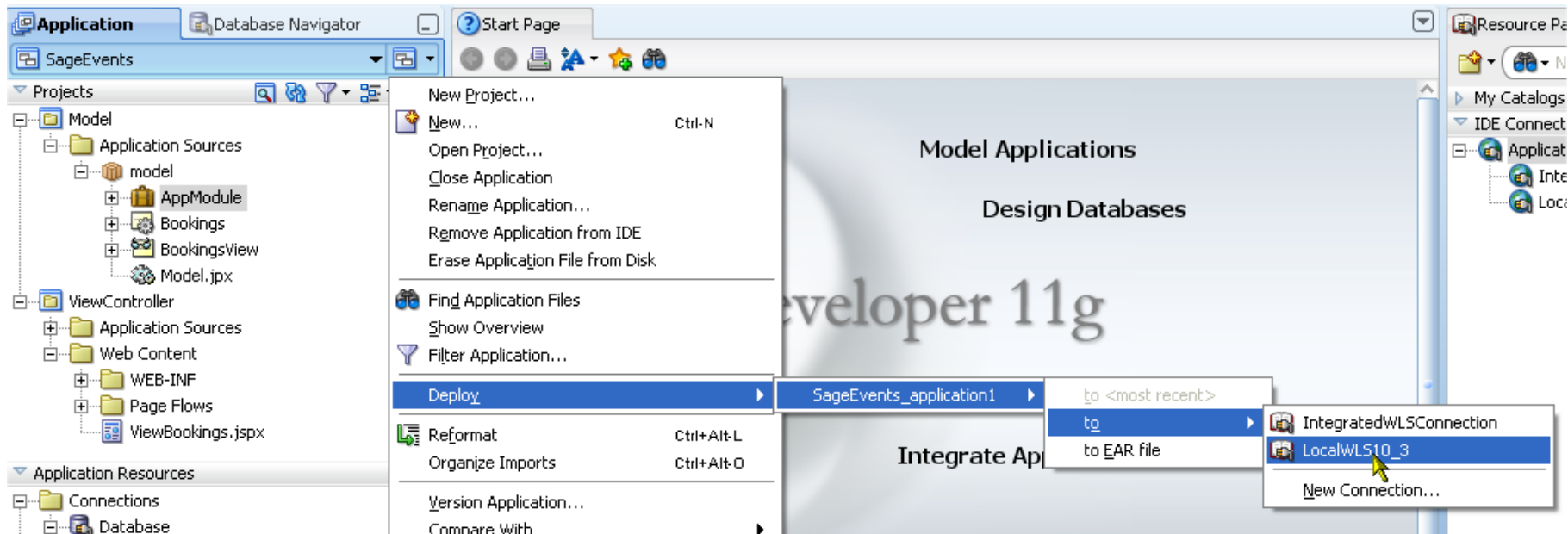
# Deployment (Auto-Deploy)

- 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..



# Deployment (Auto-Deploy)

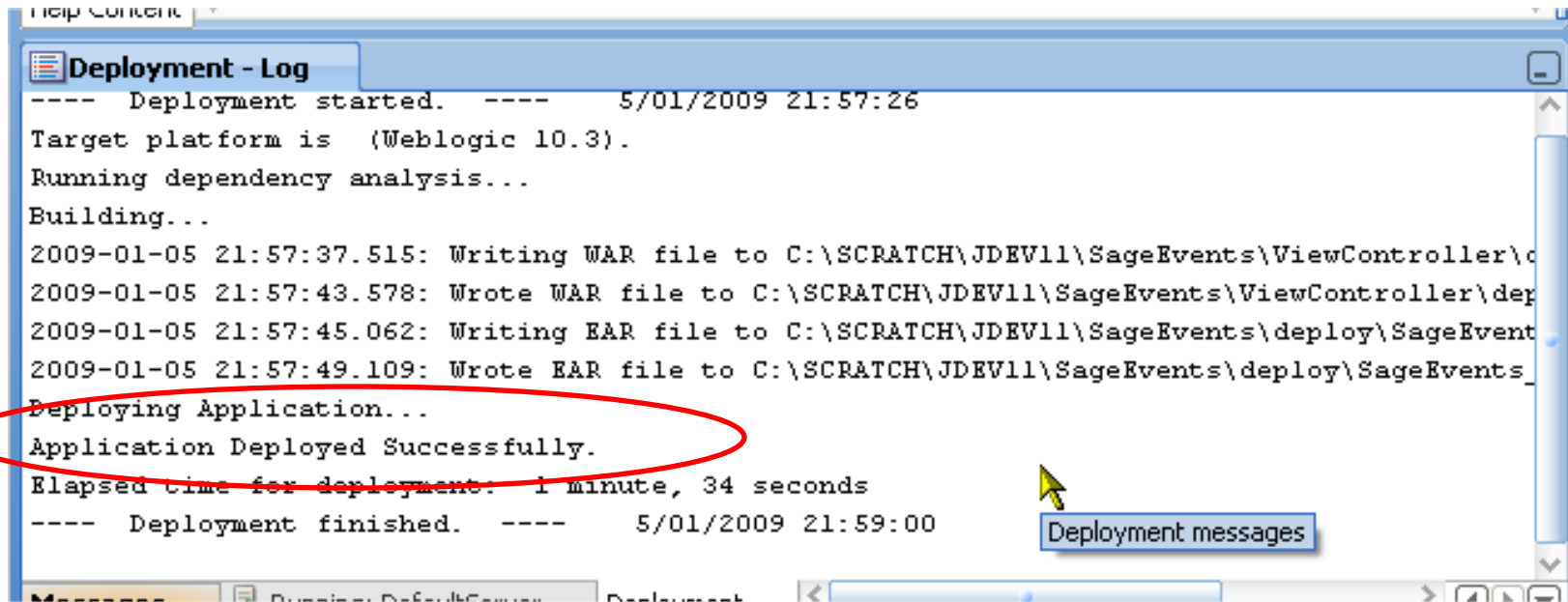
JDeveloper now compiles and builds the application, creates the ViewController WAR file, the ViewController EAR file and then begins deployment to WLS. You can watch the progress in the Deployment Log panel.



3. JDeveloper will prompt you for the **target Server** that you wish to deploy to. Typically this would be the **ManagedServer** instead of the AdminServer.

# Deployment (Auto-Deploy)

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





# Test

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

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



# Test

- Summary of Deployments page – we previously used this page to see the ADF runtime libraries.

Along with the ADF Libraries, you should now see your **new application** listed, 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

# Test

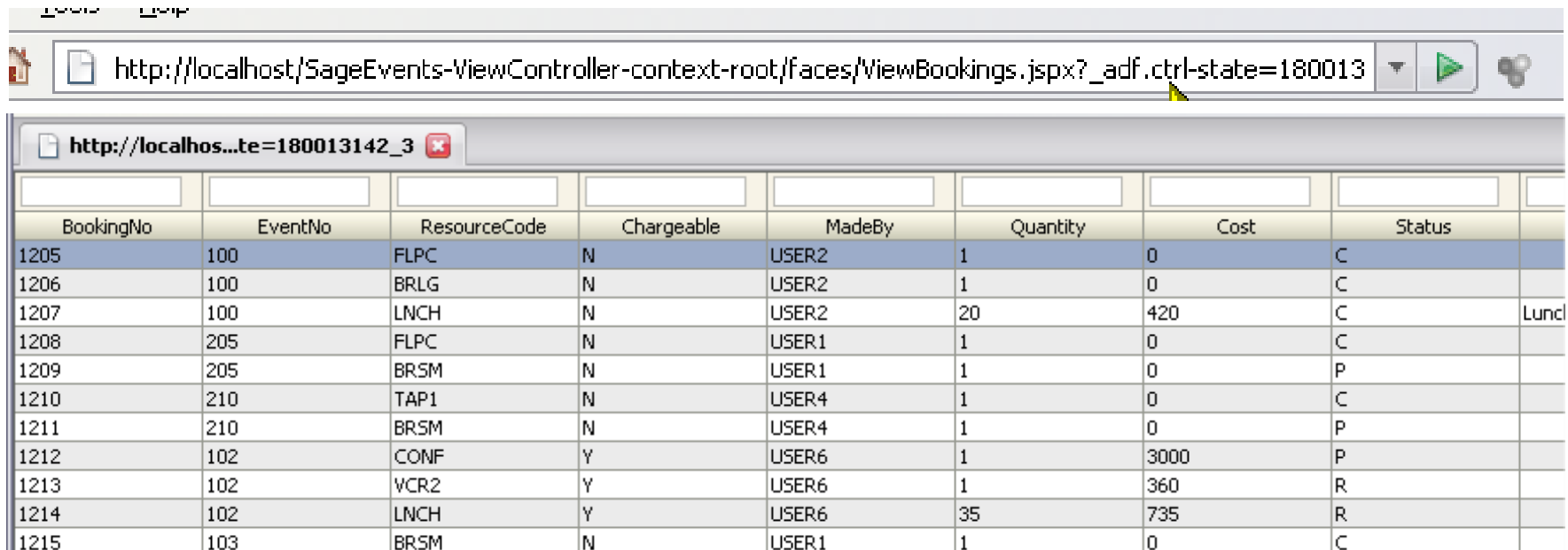
4. Settings for <your application> page - this page has a number of tabs that specify the full configuration details of your deployed application. Click on the **Testing** tab.
5. Deployment Tests page – this is mostly for SOA process testing. However, **take note of the context root** for your application,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



# Ta-Dahh!

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://<managed server host>:<managed server port>/<app context root>/faces/<your page>**



The screenshot shows a web browser window with the address bar displaying the URL: `http://localhost/SageEvents-ViewController-context-root/faces/ViewBookings.jspx?_adf.ctrl-state=180013`. Below the browser window, a table of bookings is displayed. The table has columns for BookingNo, EventNo, ResourceCode, Chargeable, MadeBy, Quantity, Cost, and Status. The data is as follows:

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

# Is it really that easy?

For the most part – Yes

Of course some trial & error plus referring to the odd blog & forum post to find out the tricks & workarounds for the traps & pitfalls didn't go astray either.

**Shay Shmeltzer**

“Common pitfalls when deploying from JDeveloper 11g to WebLogic 10.3”



# The Pitfalls shortlist

- Configure/disable/shutdown Virus/Firewall
- At least triple 1 JDev installer – jdevstudio1111install.exe
- JDK for the JDev install doesn't matter
  - For WebLogic Domains it does – match the initial install
- Don't bother with QuickStarts
- Need at least local Machine definition for Admin console control
  - Also need Node Manager
- Take note of differences in specifying JDBC definitions
  - JDev vs WebLogic
- If you deploy using JDBC URL “it will stick for good”?
- Go Auto-Deploy – then change if needed
- Deployed app testing in WLS?
  - But you can use it to get your context URL
- TimeZone gotcha



# Updates / Further Resources

- New installer for ADF runtimes (inc EM)

Oracle Fusion Middleware 11g  
Application Developer Installer

- Official Doco

[Oracle® Fusion Middleware](#)  
[Fusion Developer's Guide for](#)  
[Oracle Application Development Framework 11g](#)  
[Release 1 \(11.1.1\)](#)

Chapter 35 – Deploying Fusion Web Applications

- Oracle by Example (OBE)

Deploying a JDeveloper SOA Application  
to Oracle WebLogic Server 11g

# Credit & big Thank You

## Duncan Mills

“A Rough Guide To Installing and Setting up WebLogic 10.3 Production for Running ADF Applications”

+

Blog this, blog that  
Forum this, forum that







# **SAGE Computing Services**

**Consulting and customised training workshops**

## **Questions?**

Presentations are available from our website:

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

*[ray@sagecomputing.com.au](mailto:ray@sagecomputing.com.au)*