Understanding The Basics of ADF and Jdeveloper From an Oracle Forms Perspective

Author: Thomas Korbecki



Tom Korbecki has worked with Oracle Applications for more than 15 years. His first assignment on Oracle Applications project was to develop custom applications in Release 10. He currently is solution architect/developer that designs and develops versatile, customer-centric solutions for Oracle Application Products.

White Paper Presentation OAUG Connection Point - Apps Tech 2013 Pittsburgh, PA

Abstract

This presentation was created to help Oracle Forms Developers transition their current Oracle Forms skill set to the next version of forms development called Application Development Framework (ADF). ADF has been around since 2004 but most Oracle Forms Developers are just now having to learn ADF because Oracle is developing future applications and products using a new technology stack which supports ADF. ADF is based on the Java 2 Platform, Enterprise Edition (J2EE) set of standards. This means that Oracle Forms Developers will need to transition their existing skill sets and/or acquire new skill sets to develop applications and forms in a Web based environment. In others words, Oracle Forms Developers will need to be become Web Developers and this white paper will help you start your journey into becoming a Web Developer via ADF.

In order to build ADF applications, Oracle has provided a new integrated development environment (IDE) called JDeveloper. JDeveloper is similar to Oracle Forms in that its purpose is to simplify application development by providing a visual and declarative approach to building your application or form. Unlike the Oracle Forms tool which can build Oracle Forms applications, JDeveloper includes a development environment that supports SQL, PL/SQL, Java, XML, HTML, JavaScript, BPEL, and PHP. Since JDeveloper is a multi-purpose IDE tool, you will be intimidated by the layout, menu options, profile options and wizards, as compared to the existing E-Business Suite environment where there is a separate IDE for each Oracle Forms, Oracle Reports, Oracle Workflow and XML Gateway.

From my experience, some of your Oracle Forms skill sets can transition very easily while others will be a challenge. In addition, there are several new skills to learn and based on your past experience and/or education with Web Development, the learning curve can be fairly short but in most cases it will require an extended amount of effort. Remember when you first started developing in Oracle Forms, it took only a few days to build a simple form but it took you months or years to master it. The same can be applied to building ADF forms and applications. In fact, it will take longer to master ADF because there are hundreds of more components and services to learn.

This white paper will highlight the skill sets required to build your ADF application, steps required to set up your development environment, basic navigation of the Oracle JDeveloper software, building an ADF application with comparison to Oracle Form development and provide some understanding on how to deploy an ADF application.

Objective

- 1. What are the challenges with transitioning from Oracle Forms to Oracle ADF?
- 2. How to learn the required programming skill set to build ADF applications?
- 3. What are the basic ADF concepts?
- 4. Provide side by side comparisons between Oracle Forms and Oracle ADF.
- 5. Provide links to relevant websites explaining key concepts.

What are the challenges associated with the transition from Oracle Forms to Oracle ADF?

The same basic building blocks required in Oracle Forms development are still required with Oracle ADF development. The basic building blocks include gathering the requirements, designing the application User Interface (UI) and Data Model, building the application and then releasing the application in a production environment; however, there are new challenges encountered along the way.

Your first challenge will be designing your new application because it will take time and experience to understand the ADF framework of Model, View and Controller. The ADF framework separates the database layer (Model) from the UI layer (View) while introducing a new navigation concept called a Task Flow (Controller).

Task Flow is a modular and reusable unit of business navigation between views and nonvisual activities like routers and methods. Task Flow can be compared to an Oracle EBS workflow. An Oracle EBS workflow can execute PL/SQL code and route a transaction, whereas, an ADF Task Flow can execute Java code, route a transaction; as well as call other ADF Pages or call another Task Flow.

Your second challenge will be navigating within the JDeveloper software and selecting the correct set JDeveloper configuration for your ADF application. The confusion is due to JDeveloper being a multipurpose development tool and ADF is just one of tools within the same software.

For comparison, it would be like Oracle combining Oracle Forms, Oracle Reports and Oracle Workflow into one common development tool. When you start your new development task, the Developer has to select the correct configuration so the development software only offers the tools available for that configuration.

Your third challenge will be learning the Java programming language and how to research and use the pre-built Java APIs. It will take time to build your understanding of the most commonly used Java APIs and Oracle specific APIs. I would compare it to learning how to research and use the correct Oracle EBS public APIs. For non-Java Developers, it is initially overwhelming; however, it is only a matter of time before it all makes sense.

Your final challenge will be deploying and administering your ADF applications in the WebLogic Server. JDeveloper provides two ways to deploy ADF applications: 1) Deploy directly from the JDevloper application; 2) Deploy using and Enterprise Application Archive (EAR File). In addition, basic navigation skills are required for the WebLogic console and enterprise manager screens.

How to learn the required programming languages and software tools to build an application in ADFs?

The most difficult and frustrating part is trying to find the equivalent Oracle Forms action in ADF. Initially, it might take hours or days trying to understand how to perform a certain task in ADF. I would rely on my database knowledge to solve a certain problem in PL/SQL when the same task could have been performed in ordinary Java. In order to help minimize the learning curve, I will attempt to draw comparisons between Oracle Forms (PL/SQL) and ADF (Java).

Oracle Forms programming language is based on PL/SQL; whereas, ADF is based on Java. Most ADF applications will require some programming so a basic understanding of Java or Groovy (Java-like Syntax) is required. Most of the programming concepts transfer from PL/SQL to Java, but the syntax will be different. Based on my experience, the following Java topics/concepts are required to build basic ADF applications:

Primitive data types
Data types / variables
Strings
Variable scope
Operator
Ternary operator
Relational and conditional operations
If statement
Loops
Arrays / Lists
Maps
What causes Java null pointer exception
Try Catch Finally
Logging
Import statements

These concepts can be found in many online sites and here are a couple of sites for reference:

Oracle Java Tutorial	http://docs.oracle.com/javase/tutorial/
IBM Java Tutorial	http://www.ibm.com/developerworks/java/tutorials/
	j-introtojava1/j-introtojava1-pdf.pdf

Oracle provides pre-built Java APIs just like Oracle EBS provides pre-built PL/SQL APIs, so take advantage of them. In order to use the pre-built Java APIs, the class needs to be added to the code via the import statement. Oracle's pre-built Java APIs can be referenced at this link:

Java APIs	http://docs.oracle.com/javase/6/docs/api
Oracle APIs	oracle.jbo.* package

In addition to Java programming language, there is a Java-like scripting language called Groovy. Groovy is a dynamic language for the Java platform that is checked and executed at runtime as opposed to Java, which is checked at compile-time. In ADF applications, Groovy is typically used to validate business rules, set a database sequence on a table, build error messages, referencing built-in calls like setting current date and time (i.e. sysdate), and aggregate functions in view objects (i.e. sum, count, min, max, etc.).

Introduction to Groovy	http://www.oracle.com/technetwork/developer-
Support in JDeveloper and	tools/jdev/introduction-to-groovy-128837.pdf
Oracle ADF 11g	

On occasion, you will need JavaScript to react to some client event to get information from the Clients device. JavaScript was originally implemented as part of the web browser so the client-side scripts could interact with the User, control the browser, communicate asynchronously, and alter content that is displayed. In the case of ADF applications, I have used JavaScript to determine the preferred language in the Web browser, determine the Users Time Zone and store and retrieve cookies.

Introduction To JavaScript <u>http://www.w3schools.com/js/js_intro.asp</u>

What are the basic ADF concepts?

ADF Concepts

What is the ADF Framework?

Oracle Application Development Framework (ADF) is a Java-based development tool (much like Oracle Forms is a PL/SQL-based tool) designed to take full advantage of Java Enterprise Edition or Java EE. ADF Technology simplifies interaction with "Java" EE and Oracle's Fusion Middleware.



- The view layer contains the UI pages used to view or modify that data
- The controller layer processes user input and determines page navigation
- The model layer represents the data values related to the current page
- The business service layer handles data access and encapsulates business logic

ADF has many components, but the ADF BC (Data & Links) and the ADF Faces (UI) can be thought of as Form Data Blocks (Data & Relationships) and Canvases (UI).

What is the Oracle Forms Framework?



Compare ADF vs Oracle Framework



Model

<u>Entity object</u> – For the context of this white paper, an entity object is a database table on which DML operations are performed. In Oracle forms, this is the Record Manager.

The full ADF definition of an entity object is business components that encapsulate the business model, including data, business rules (When-Validate-Record), and persistence behavior for items/columns that are used in your application. Entity object definitions map to single objects in the data source and it is where DML operations are performed. In most cases, it is a database table or snapshot in a database, but it can be a spreadsheet, XML or a flat file.

<u>View object</u> – For the context of this white paper, a view object can be a single database table or a group of database tables linked together via relationships (i.e. parent table and child table) or a view only database view that can be used in a "List of Values" (LOV). In Oracle Forms, this could be considered the Data Block. For example, in a Forms Data Block, we specify what columns we are using and we can also set the "where by" or "order by" clause.

The full ADF definition of a view object is a group of business components (ie. Database columns) that are a collection of data from a data source. It can represent an individual table or a group of tables as well as a database view used by a LOV. View objects must have a process for retrieving data from the data source (database) and a method of retrieving the data (SQL based query). Oracle ADF Business Components can automatically use JDBC API to pass this query to the database and receive the result.

<u>View Link</u>– For the context of this white paper, a view link is a relationship between two view objects that are joined by one or many database columns. This enforces business rules related to the data. A view link is used to keep data in-sync in the View Layer. Note: If an association link exists, then use that relationship in the view link, so you join based on the association link and not the individual field(s).

<u>Association "Link"</u> – This is a relationship between two entity objects, which can be represented by a primary – foreign key relationship in a database. If the relationship is defined in the database, then ADF will automatically build an association link between the two entity objects. If the relationship is not defined, then you have the option to create it manually. The association link is used to keep data in-sync in the Model layer.

The Data Model is somewhat comparable to Data Blocks in Oracle Forms. The Data Block controls create, read, update and delete (CRUD) operations as well as link the UI data fields.

View

For the basis of the white paper, the View layer is the user interface that displays data from the Model layer. The View layer, in its simplest form, could be a page with input fields, buttons, input boxes and tables to display data; however, ADF provides other components that allow for the creation of rich and reusable user interface.

ADF Faces provides over 100 rich components, including hierarchical data tables, tree menus, in-page dialogs, accordions, dividers, and sortable tables. ADF Faces also provides ADF Data Visualization components, which are Flash- and SVG-enabled components capable of rendering dynamic charts, graphs, gauges, and other graphics that can provide a realtime view of underlying data. Each component also supports customization and skinning, along with internationalization and accessibility.

In order to understand these components, Oracle provides a tool called "ADF Faces Rich Client Demos". The tool is located at:

Oracle ADF Faces Rich Client	http://jdevadf.oracle.com/adf-richclient-
	<u>demo/faces/index.jspx</u>

Controller

The Controller layer processes user input and determines page navigation. JDeveloper provides a declarative way, or Task Flow, to pass application control between different types of activities such as pages, methods within managed beans, transaction support, save points, or calls to other task flows. In terms of Oracle Forms, it is the ability to navigate within your application using PL/SQL code to navigate to a block (GO_BLOCK) or to open another form (CALL_FORM).

There are two types of Task Flows:

Unbounded Task Flow is essentially the entry point into your application or home page. There can be only one unbounded task flow per application

Bounded Task Flow is a modular and reusable application flow with a defined entry point (i.e. default activity), but can have zero to many exit points. Additional information about Task Flows is as follows:

Task Flow Design Fundamentals (An	http://www.oracle.com/technetwork/developer-
Oracle White Paper April 2011)	tools/jdev/adf-task-flow-design-132904.pdf

Oracle® Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework 11g Release 1 (11.1.1.6.0)

- Section 14 Getting Started With Task Flows http://bit.ly/adfdevguide111160s14

- Section 16.4 Sharing Data Control Instances http://bit.ly/adfdevguide111160s164

- Section 18 Introduction to Complex Task Flows http://bit.ly/adfdevguide111160s18

- Section 18.3 Managing Transactions http://bit.ly/adfdevguide111160s183

Additional Concepts

Application Module (AM)

This is a collection of business rules and transactions (view objects and view links defined in the Model layer) that are related to a certain function or use case. When building your application, the view and view links need to be added to a container or application module (AM) so they can be exposed as "data controls" and consumed by the View layer.

An additional feature of ADF is the ability to expose the collection of views as a service so it can be consumed by other applications; thus reusing code and enforcing same business rules across multiple applications.

As for Oracle Forms, the Form itself is an application module because it is a collection of business rules and transactions that are combined together to archive a business requirement.

What is a Data Binding?

Data binding is a connection between UI components to data control exposed via the application module. Once the UI consumes a view object or method exposed in the data control, a binding is created between the Model layer and View layer. The following diagram illustrates the data binding concept.



Object Scope Lifecycles

When you run your application, parameters are usually passed to the application and that data is stored in an object scope. Once you place an object in a scope, it can be accessed from the scope using an expression language. For example, you might create a managed bean named myPageRequestBean and define the bean to live in the Request scope.

To access that bean, you would use the expression language #{requestScope. myPageRequestBean }. For a Forms Developer, this is a way of calling the procedure or function in a database package.

applicationScope	The object is available for the duration of the application.
sessionScope	The object is available for the duration of the session.
requestScope	The object is available for the duration between the time an
	HTTP request is sent until a response is sent back to the
	client.

There are three types of scopes in a standard JSF application:

In addition to the standard JSF scopes, ADF Faces provides the following additional scopes:

pageFlowScope	The object is available as long as the user continues
	navigating from one page to another. If the user opens a
	new browser window and begins navigating, that series of
	windows will have its own pageFlowScope. Store your
	application parameters in this scope.
backingBeanScope	Used for managed beans for page fragments and
	declarative components only. The object is available for
	the duration between the time an HTTP request is sent
	until a response is sent back to the client. This scope is
	needed because there may be more than one page fragment
	or declarative component on a page, and to avoid collisions
	between values, any values must be kept in separate scope
	instances. Use backingBeanScope for any managed bean
	created for a page fragment or declarative component.
viewScope	The object is available until the ID for the current view
	changes. Use viewScope to hold values for a given page.

Object scopes are analogous to global and local variable scopes in programming languages. The wider the scope the higher the availability of an object. During their lifespan, these objects may expose certain interfaces, hold information, or pass variables and parameters to other objects.

Relationship between Scopes and Page Flow



For your applications, you will most likely choose the pageFlowScope to store your Task Flow parameters. These variables are stored in a hash map that can be referenced in the UI managed bean or in the Application Module Java Methods. Referencing the hash map object related to a specific object scope is one way to reference variables between the Model layer and View layer in your application. In your application, you will most likely use backingBeanScope to store your methods used by your page fragments.

Expression Language

In Java Server Faces (JSF), you can use a simple expression language (EL) to access application data stored in Java Bean components. The syntax of EL is as follows:

#{<Binding or Bean Name>.<Variable/Method><Operation>}

For example, if we want to display a credit card number field only when the payment method is "Credit Card", then we could apply the following EL to the visible property on the credit card field in the UI:

#{bindings.PreferredPaymentMethod.attributeValue == 'Credit Card'}

In the reference document below, the tutorial uses the \$ sign instead of the # sign. For our ADF development, we will be using the # sign. The following explains the difference between the \$ and # sign.

\$ - \$ syntax executes expressions eagerly/immediately, which means that the result is returned immediately when the page renders.

- # syntax defers the expression evaluation to a point defined by the implementing technology. In general, JSF uses deferred EL evaluation because of its multiple lifecycle phases in which events are handled. To ensure the model is prepared before the values are accessed by EL, it must defer EL evaluation until the appropriate point in the lifecycle. I have provided a EL tutorial:

The J2EE 1.4	http://docs.oracle.com/javaee/1.4/tutorial/doc/JSPIntro7.html
Tutorial	

Note: Some Developers try to avoid using expression language statements because the statement is checked at runtime as opposed to a managed bean where the statement is checked at compile time. It is similar to calling dynamic SQL where you don't know the value of the variable you are referencing until runtime.

Managed Bean

At some point in your application, you will need a more powerful tool than expression language (EL) to programmatically modify the UI or control the behavior or your application so you will need to create a managed bean. A managed bean is a reusable software component for Java that represents a manageable resource including components, applications or devices. They are used to encapsulate many objects into a single object (the bean), so that they can be passed around as a single bean object instead of as multiple individual objects.

Managed beans are Java classes that you register with the application using various configuration files. When the JSF application starts up, it parses these configuration files and the beans listed within them are made available. The managed beans can be referenced in an EL expression, allowing access to the beans' properties and methods.

Below is an example of a change listener method that is controlled by a managed bean:

```
public void PreferredPaymentMethodChangeListner(ValueChangeEvent valueChangeEvent) {
     valueChangeEvent.getComponent().processUpdates(FacesContext.getCurrentInstance());
     Integer newValue = (Integer) valueChangeEvent.getNewValue();
     BindingContext bctx = BindingContext.getCurrent();
     BindingContainer bindings = bctx.getCurrentBindingsEntry();
     // Get The List Of Values Related To The Selected Payment Method
     JUCtrlListBinding list =(JUCtrlListBinding) bindings.get("PreferredPaymentMethod");
      // Find The Selected Values In The List Of Values
      Row selectedRow = (Row)list.getSelectedValue();
      String selectedValue = list.getAttributeValue().toString();
     if (selectedValue.equals(SiConstants.CreditCard)) {
         // Turn On Credit Card Panel Group
          this.creditCardPanelGroupLayout.setVisible(Boolean.TRUE);
     } else {
         // Turn off Credit Card Panel Group
          this.creditCardPanelGroupLayout.setVisible(Boolean.FALSE);
     }
```

As comparison to Oracle Forms, a managed bean would contain all the PL/SQL related UI behavior. For example, we can display or hide the credit card number field based on a When-Validate-Item of the payment method field.

Note: When you create your managed bean you need to specify a scope for the bean. (See Object Scope Section).

Where Should Your Custom Code Reside?

There are various Java classes and "levels" to write your custom code and based on the requirement it should be written at a certain "level". When I first started building applications, I was never quite sure where to place my custom code. I used the following guide to help me determine where to write the code.

Location	Guide
Application Module	Application module class as the place where you can write your service-level application logic.
	Execute database command or block of PL/SQL
	Access Existing View Objects
	Access DBTransaction Object
EntityImpl Class	The EntityImpl class is the base class for entity objects,
	which encapsulate the data, validation rules, and business

	behavior for your business domain objects.
	• Get an attribute
	• Set an attribute
	• Validate data before posting to the database
	• doDML Operations (Like Triggers)
	// In PLSQLEntityImpl.java
	protected void doDML(int operation,
	TransactionEvent e) {
	<pre>// super.doDML(operation, e);</pre>
	if (operation == DML_INSERT)
	callInsertProcedure(e);
	else if (operation == DML_UPDATE)
	callUpdateProcedure(e);
	else if (operation == DML_DELETE) callDeleteProcedure(e);
	J
	In most cases, Developers user DML operations to set
	"who" columns and retrieve database sequences. For
	example, translate the user login information to a
	User Id.
	Note: Add custom code to set a value after it is retrieved
	from the database and before it is passed to the view object
	or to override the view object value before storing the value
	into the database.
ViewObjectImpl Class	A view object is a base class for view objects
	• Set Where By Clause
	Set Where By Clause Set Order By Clause
	Set Order By Clause
	For Form Developers, this is where your block triggers exists.
	r of r of a Developers, and is where your block anggers exists.
	Note: I have also seen Developers set the Where By and
	Order By in an Application Module method
ViewRowImpl Class	A view object is a base class for view row objects
	Get Attribute
	Set Attribute
	Note: I normally use this class to set transient values
	attributes before they are rendered to the page and to process
	values from the page before passing the row-set back to the
	Entity class.

Managed Bean	This is where you store all your UI based code that controls the page as well as call methods in the application module.
	 Initialize code for your session Call methods defined in the application module Conditionally Rendering Set variables
	For Form Developers, this is where all your UI events triggers

Note: Oracle provides a comprehensive list of what typical code is placed in each area:

Document	Chapter
Developer's Guide for Oracle	Most Commonly Used ADF Business
Application Development	Components Methods
Framework	

Debugging (JDeveloper Debug / SOP / ADF Logger)

ADF and JDeveloper provide several ways to debug your code.

- 1. System.out.println, or SOP is equivalent to dbms_output in PL/SQL. It provides debugging capability while developing your application but ADF Logger should be used when the application is deployed to production. By default the SOP messages are displayed in the Log Window (Figure A: JDeveloper Navigation).
- 2. ADF Logger is a logging mechanism, which is built into the ADF framework. It is a wrapper around the java.util.Logging APIs with a few convenience methods thrown in, and most importantly, some specific features integrated into both JDeveloper and Enterprise Manager.
- 3. The JDeveloper tool has a few ways to debug your application.
 - a. When you want to debug your application in detail, then add the following string to your application: -Djbo.debugoutput=console and run your application in debug mode. The debug messages will be displayed in the Log Window (Figure A: JDeveloper Navigation).

Project Source Paths ADF Model ADF View			Customize Settings Edit New Delete Bestore Defaults
JSP Tag Lbraries JSP Visual Editor Lbraries and Classpath Resource Bundle Run/Debug/thofke Technology Scope	 Edit Run Configuration "Defaul Search Launch Settings JavaScript – P.(SQL – ADF Task Flow – XST – Tool Settings – Profiler – CPU – Memote – Debugger – Debugger – Remote 	t* Launch Settings Default Run Target:	Browse
21, 2013 4:53:49 PM MST> <erro 21, 2013 4:53:49 PM MST> <erro 21, 2013 4:53:49 PM MST> <erro 21, 2012 4:53:49 PM MST> <erro< td=""><td>Help</td><td></td><td>OK Cancel</td></erro<></erro </erro </erro 	Help		OK Cancel

b. ADF Declarative Debugger provides declarative breakpoints that you can set at the ADF object level (such as task flows, page definition executables, method and action bindings, ADF lifecycle phases), as well as standard Java breakpoints. How to use the debugging utility can be the entire presentation, so reference the ADF Application Developers Guide – Chapter "31.7 Using the ADF Declarative Debugger" to learn how to use this feature.

Naming Conventions

You should determine how to name your packages and the granularity.

Example Base: com.<application short name>.<project name>

Sample: com.xx.myproject

01	
Default Package	com.xx.myproject.view
Task Flows	com.xx.myproject.view.pageflows
Pages	com.xx.myproject.view.pages
Page Fragments	com.xx.myproject.view.pagefrgments
Managed Beans	com.xx.myproject.view.bean
Overriden Component Class	com.xx.myproject.view.overriden
Page Definitions	com.xx.myproject.view.pagedefs

UI

Model	
Default Package	com.xx.myproject.model
Entities	com.xx.myproject.model entities
Associations	com.xx.myproject.model.assoc
View Links	com.xx.myproject.model.vo.link
Updatable Views	com.xx.myproject.model.vo
Read Only Views	com.xx.myproject.model.vo.readonly
Application Module (AM)	com.xx.myproject.model.am
Diagram	com.xx.myproject.model.diagram

How to set up your local PC to develop ADF applications?

Now that you have been introduced to the concept of ADF and JDeveloper, it is time to set up your local PC. You can find the download page by referencing the JDeveloper home page:

Oracle JDeveloper Main	http://www.oracle.com/technetwork/developer-
Page	tools/jdev/overview/index.html

What Version?

There are several versions of JDeveloper to download, so choose the correct version for your environment. There is a one-to-one relationship between JDeveloper and Weblogic Server and you have to install a JDeveloper with the same ADF runtime version as the Weblogic Server. There is no backward or forward compatibility. EBS Forms Developers are familiar with this concept because they have to install the compatible Oracle Forms software related to their EBS technology stack.

If you are NOT planning on deploying to a Weblogic Server (WLS), then download the latest JDeveloper version. In all other cases, reference the compatibility link below:

ADF Runtimes vs WLS versions	https://blogs.oracle.com/onesizedoesntfitall/ent
as of JDeveloper 11.1.1.6.0	ry/adf_runtimes_vs_wls_versions

Installation

Oracle provides detailed instructions on how to install a local version of. I will just highlight a few steps:

- 1. Software Download the correct version of the software (See What Version?)
- 2. Prerequisites Administrative Access
- 3. Installation is quite simple



Click On jdevstudio11122install.exe

Oracle Installer		
	ORACLE [.]	
	Preparing the installer	
	Cancel	

🕢 Oracle Installer - Oracle JDeveloper 11g (11.1.2.2.0)		
Welcome This installer will guide you through the instal Oracle JDeveloper 11g Release 2 (11.1.2.2.0		
ORACLE	Instructions Click the Next button to proceed to the next screen. If you want to change entries in a previous screen, click the Previous button. You may guit the installer at any time by clicking the Exit button.	
3		
Exit	Previous Lext	

It is recommended you always install each JDeveloper installation in its own directory.

😡 Oracle Installer - Oracle JDeveloper 11g (11.1.2.2.0)	
Choose Middleware Home Directory	
Specify the Middleware Home where you wish to install Oracle Products.	ORACLE'
Middleware Home Type Ct\JEV_11_1_6	
O Use an existing Middleware Home	
Create a new Middleware Home	
Middleware Home Directory	
D;\JEV_11_1_1_6	
Browse Reset	
Exit	Previous Next
	Dening Tan
Oracle Installer - Oracle JDeveloper 11g (11.1.2.2.0)	
Choose Install Type	
Select the type of installation you wish to perform.	ORACLE
Typical	
Install the following product(s) and component(s):	
Install the following product(s) and component(s): ODeveloper and ADF	
WebLogic Server	
O Custom	
Custom	and perform optional
O Custom	and perform optional
Custom GP Choose software products and components to install	and perform optional
Custom Choose software products and components to instal configuration.	
Custom	and perform optional Previous Next
Custom Choose software products and components to instal configuration.	
Custom Choose software products and components to instal configuration.	Previous Next
Custom Choose software products and components to instal configuration. Egt Oracle Installer - Oracle JDeveloper 11g (11.1.2.2.0)	
Custom Choose software products and components to instal configuration. Egit Oracle Installer - Oracle JDeveloper 11g (11.1.2.20) Choose Product Installation Directories	Previous Liext
Custom Choose software products and components to instal configuration. Egt Oracle Installer - Oracle JDeveloper 11g (11.1.2.2.0)	Previous Next
Custom Choose software products and components to install Egt Oracle Installer - Oracle JDeveloper 11g (11.1.2.2.0) Choose Product Installation Directories Provide the directories where you wish to install	
Custom Choose software products and components to instal Egit Oracle Installer - Oracle JDeveloper 11g (11.1.2.2.0) Choose Product Installation Directories Provide the directories where you wish to install	Previous Liext
Custom Choose software products and components to install Configuration. Egit Oracle Installer - Oracle JDeveloper 11g (11.1.2.2.0) Choose Product Installation Directories Provide the directories where you wish to install Oracle Products.	
Custom Choose software products and components to install Configuration. Egit Oracle Installer - Oracle JDeveloper 11g (11.1.2.2.0) Choose Product Installation Directories Provide the directories where you wish to install Oracle Products. Middleware Home Directory D(JEV_11_1_1_6	
Custom Choose software products and components to install Configuration. Egit Oracle Installer - Oracle JDeveloper 11g (11.1.2.2.0) Choose Product Installation Directories Provide the directories where you wish to install Oracle Products. Middleware Home Directory Dr(J2FV_11_1_1_6 Product Installation Directories The Product Installation Directories The Product Home might contain shared utilities and any products or component	Previous Lext
Custom Choose software products and components to instal configuration. Egit Oracle Installer - Oracle JDeveloper 11g (11.1.2.2.0) Choose Product Installation Directories Provide the directories where you wish to instal Oracle Products. Middleware Home Directory D(JEV_11_1_6 Product Installation Directories The Product Installation The Product Installation Directories The Pro	Previous Lext
Custom Choose software products and components to install configuration. Egit Oracle Installer - Oracle JDeveloper 11g (11.1.2.2.0) Choose Product Installation Directories Provide the directories where you wish to install Oracle Products. Middleware Home Directory Dr):EV_11_1_1_6 Product Home might contain shared utilities and any products or component are not setDeveloper and ADF:	Previous Lext
Custom Choose software products and components to instal configuration. Egit Oracle Installer - Oracle JDeveloper 11g (11.1.2.2.0) Choose Product Installation Directories Provide the directories where you wish to instal Oracle Products. Middleware Home Directory D(JEV_11_1_6 Product Installation Directories The Product Installation The Product Installation Directories The Pro	Previous Lext
Custom Choose software products and components to install configuration. Egit Cracle Installer - Oracle JDeveloper 11g (11.1.2.2.0) Choose Product Installation Directories Provide the directories where you wish to install Oracle Products. Middeware Home Directory D(JEV_11_1_1_6) Product Installation Directories The Product Installation Directories The Product Installation Directories The Product Home might contain shared utilities and any products or component are not set. JDeveloper and ADF: D(JEV_11_1_1_6)[developer	Previous Lext
Custom Choose software products and components to instal configuration. Egit Oracle Installer - Oracle JDeveloper 11g (11.1.2.2.0) Choose Product Installation Directories Provide the directories where you wish to instal Oracle Products. Middleware Home Directory D(JEV_11_1_6) Product Installation Directories The Product Installation The Product Installation The Product Installation The Product Installation The Pr	Previous Lext
Custom Choose software products and components to instal configuration. Egit Oracle Installer - Oracle JDeveloper 11g (11.1.2.2.0) Choose Product Installation Directories Provide the directories where you wish to instal Oracle Products. Middleware Home Directory D(JEV_11_1_6) Product Installation Directories The Product Installation The Product Installation The Product Installation The Product Installation The Pr	Previous Lext
Custom Choose software products and components to instal configuration. Egit Oracle Installer - Oracle JDeveloper 11g (11.1.2.2.0) Choose Product Installation Directories Provide the directories where you wish to instal Oracle Products. Middleware Home Directory D(JEV_11_1_6) Product Installation Directories The Product Installation The Product Installation The Product Installation The Product Installation The Pr	Previous Lext
Custom Choose software products and components to instal configuration. Egit Oracle Installer - Oracle JDeveloper 11g (11.1.2.2.0) Choose Product Installation Directories Provide the directories where you wish to instal Oracle Products. Middleware Home Directory D(JEV_11_1_6) Product Installation Directories The Product Installation The Product Installation The Product Installation The Product Installation The Pr	Previous Lext



😡 Oracle Installer - Oracle JDeveloper 11g (11	1.1.2.2.0)	
Installation Summary The following Products and JDKs will be installed.		ORACLE
Developer and ADF Developer and ADF Developer Studio Application Development Framework Ru WebLogic Server Core Application Server Administration Console Configuration Wizard and Upgrade Fram WebLogic SCA	Description Oracle JDeveloper and ADF is a c Service-Oriented Architecture (SC is ranked best among major Java TechRankings. As part of Oracle f "hot-pluggable" with Oracle and n supporting all major J2EE applicab	DA) and Java development that vendors in Forrester Fusion Middleware, JDeveloper is on-Oracle environments,
WebLogic 3DBC Drivers WebLogic Server Clants UDDI and Xquery Support DKs jdk:60_24	Approximate Installed Size Highlighted item: Total of all selected items:	990.6 MB 1,702.3 MB
		Eleviers Fext



Installation Complete Click the Done button to exit the installer.		ORACLE
ORACLE	Message Congratulations! Installation is complete.	
C		
2		
P		
	🗹 Run Quickstart	

If you need additional installation references, then refer to the following references:

Oracle Install Guide – Screen	http://docs.oracle.com/cd/E35521_01/install.11123
Shots	0/e17074/toc.htm
Oracle Install Guide – Video	https://blogs.oracle.com/workingwithadf/entry/inst alling and configuring jdeveloper with adf

Building Your First Applications

Now that you have installed JDeveloper, take a few hours navigating within the tool and review the Oracle tutorial on how to build an ADF application.

1. Understand basic navigation within the JDeveloper tool.

Figure A: JDeveloper Navigation



Here are some navigation short-cuts:

Ctrl + Mouse Click	Go To Declaration (Find view associations)
Ctrl + -	Go To Class
Ctrl + Alt + -	Go To File (Used when it's a large project)
Alt + Home	Locate file in app navigator
App Navigator	Find as you type (Similar to Oracle Forms)
Ctrl + F	Find
Ctrl + Shift + F	Find In Files (Similar to "Find" in Oracle Forms)
Ctrl + Alt + H	Highlight Fields

Note: In JDeveloper Version 11.1.1.6, the associations are not visible in the declarative entity object so open the source tab and search for

"AccessorAttribute". Once you find the association section, then hold down the Alt key on the accessor line, and next click on the hyperlink to view the association.



Highlight is a great way to find text within in your custom code and I use this feature all the time.

	Highlight Tool
Versioning Tools Window Help	Ingingin 1001
* -	
🕞 FormsVsAdf Overview 🗴 ট FileLibraryUtils.java 🗴 🖌	
💮 🕂 Find 🕹 🖓 🖄 🎉 🛛 🖓 🖓	S 🛃 🖫 I 🕽 🕨 🖕 🖕 📟 👘 👘
Source Versioning Tools Window Help	
• 🍉 • 🌞 •	
MyProjectAppModuleImpl.java × BoeOrderHeadersAll.xml × BOrderSalesPageFlowScopeBean.java ×	
(🍓 - Find 🕹 🚯 🖄 🍻 🖓 🖓 🖓 🖓 🖓 🖓 🖓 🖓 🐘 🐘 🐘 🔛 🔛	
24 */ 25 ■ public MyProject&ppModuleImpl() {	
25 3 public hyprojectapphodulempi() (
27	
public void setAMSessionVariables() (
29 System. out.println("Start - setAMSessionVariables");	
String SessionWarehouseId = null;	
31 CallableStatement st =null;	
32	
33 // Get the Page Flow Scope Variable From the Current Session And Assign It	To A Session Variable
34 SessionWarehouseId = ADFContext.getCurrent().getPageFlowScope().get("pWare	house").toString();
35	
36 // Set Warehouse Session Variable Which Can Be Use By Application Hodule /	Mierr Dhieste
	view objects
37 this.getSession().getUserData().put("pWarehouseId", SessionWarehouseId);	View Objects

2. Review an Oracle tutorial on how to build your first application. I referenced this URL multiple times when I was building the first applications.

Developing Rich Web	http://docs.oracle.com/cd/E18941_01/tutorials/jdtut
Applications With Oracle ADF	<u>_11r2_55/jdtut_11r2_55_1.html</u>

Now it is time to start building your first application. Once you start JDeveloper, it will prompt you to select a role. A role is a way to only display tools and components related to your project. Select the default role until you have researched how this option works.

role	s using the Roles page in preferences.	
<u>R</u> ole	2	
۲	Default Role	-
	Enables all technologies.	
0	BPM Process Analyst	
	Role for BPM Business Analyst.	55
\circ	Customization Developer	2000
	Configures the product for customizing metadata.	
0	Database Edition	
	Includes only features for core database development.	
0	Java EE Edition	
	Includes only features for core Java EE development.	-
	Always prompt for role selection on startup	

Create a new application:

🖕 Create Fusion Web Appl	ication (ADF) - Step 1 of 5
Name your applicatio	
0.1.1.1.1	Application Name:
Application Name	FormsVsAdf
Project 1 Name	Directory:
 Project 1 Java Settings 	C:\JDeveloper\1Test\FormsVsAdf Browse
Project 2 Name	Application Package Prefix:
 Project 2 Java Settings 	com.xxx myproject
	Application Template:
	project has also 504 technology F Fusion Web Application (ADF) Creates a databound ADF web application. The application consists of one project for the view and controller components (ADF Faces and ADF Task Flows), and another project for the data model (ADF Fuses Components).
	Java Desktop Application Creates an application configured for building a generic Java application. The new application will include a project that is preconfigured to use Java, Swing, and JavaBeans technologies.
Help	< <u>Back</u> Next > Enish Cancel

💩 Create Fusion Web Appl	lication (ADF) - Step 2 of 5
Name your project	
Application Name	Broject Name: MyProjectModel Dirgctory: C:\/Developer\\Test\Forms\/sAdf\/MyProjectModel Brogse
Project 1 Java Settings Project 2 Name Project 2 Java Settings	Project Technologies Generated Components Associated Libraries Available: Selected: ADF Deakhop Integration ADF Faces ADF Days Web Application Support ADF Page Flow ADF Swing Ant BMM Database (Offine) Technology Description: Technology Descriptio
Help	< Back Next > Einish Cancel

Configure Java setting	gs
Application Name Project 1 Name Project 1 Java Settir Project 2 Name	Your new project starts with a default package, a source root directory, and an output directory. Default Packagee: <u>som xx.myrcoject.mode</u> Java Source Path:
Ο Project 2 Java Settings	C:\Developer\ITest\FormsVsAdf\MyProjectModel src Brogge Qutput Directory: C:\Developer\ITest\FormsVsAdf\MyProjectModel\classes Browge
Help	<back next=""> Finish Cancel</back>

💩 Create Fusion Web Appl	ication (ADF) - S	Step 4 of	5			×
Name your project					10101010101919191919191	F
Application Name	Project Name:	MyProjec	:tUI			
Project 1 Name	Directory:	C:\JDeve	eloper\1Test\Forms	sVsAdf\My	ProjectUI	Browse
Project 1 Java Settings	Project Tech	nologies	Generated Com	ponents	Associated Libraries	
👳 Project 2 Name	Available:			Se	elected:	
Project 2 Java Settings	ADF Mobile E ADF Swing Ant BPM Database (C EJB C Technology E ADF Faces a personalizat	D Integrati Web Appl Browser Offline) Description adds very tion and s	ion lication Support	A H J S S S S S S S S S S S S S S S S S S	DF Faces DF Page Flow TML ava SF SP and Servlets ML dialog framework, as w es features includes file f a page (AJAK-stych).	upload 🧧
Help				< <u>B</u> ack	Next > Finish	h Cancel

🚖 Create Fusion Web Appl	ication (ADF) - Step 5 of 5
Configure Java settin	
Application Name Project 1 Name Project 1 Java Settings Project 2 Name	Your new project starts with a default package, a source root directory, and an output directory. Default Package: <u>com sx.mproject.New</u> <u>ava Source Path:</u>
Project 2 Java Settir	C:\Developer\ITestFormsVsAdfWyProjectUI\grc Brogge Qubut Directory: C:\Developer\ITestFormsVsAdfWyProjectUI\glasses Browge
<>	< Back Next > Enish Cancel

Once you click on finish, the application will set up the directory structure on your file system and open up the application with a Quick Start Guide. This is comparable to the Oracle Forms Wizard that helps you build your application. I would recommend using

the ADF Quick Start Guide to build the first applications. As you learn the JDeveloper tool, you will probably build your application without the Wizard.

Figure B: File Structure

♥	st 🕨 FormsVsAdf 🕨	Autor Autor Late	and a second subject of	_
le Edit View Tools Help	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1			
	urn New folder			
Þ 🍌 dev	*	Name	Date modified	Туре
Documents and Settings		🔒 .adf	2/23/2013 7:50 AM	File folder
🛛 🕌 Drivers			2/23/2013 7:50 AM	File folder
FileZilla_3.5.3_win32		MyProjectModel		
FileZilla-3.5.3		MyProjectUI	2/23/2013 7:50 AM	File folder
ldentity		src 🖉	2/23/2013 7:50 AM	File folder
> 🚹 Intel		FormsVsAdf.jws	2/23/2013 7:50 AM	JWS File
iava class				
JDeveloper				
🛛 🌛 1NewTest				
0 1Prod				
4 0 1Test				
> isvn				
4 👸 FormsVsAdf				
▷ .adf				
MyProjectModel				
> li classes				
🖌 🎍 MyProjectUI				
Lasses				
public_html				
4 🏭 src				
META-INF				

Quick Start Guide

FormsVsAdf Overview ×		
Java Files Page Flows	Fusion Web Application Quick Start Checklist Create the application by following step-by-step instructions describing how to build Fusion Web	b Applications according to Oracle
Veb Pages		Show All Hide All
Business Components Binding Files	1 Plan Your Application	Not Started
Offline Databases	2 > Connect to a Database	Not Started
	3 > Build Business Services	Not Started
	4 Besign Application Flow	Not Started
	5 🕨 Design Pages	Not Started
	6 F add Common Components (Lookups, Search and Menus)	Not Started
	7 Figure 1 Figure 1 Figure 2 Fig	Not Started
	8 Figure Your Application	Not Started
	9 P Internationalize Your Application	Not Started
	10 > Debug and Test Your Application	Not Started
	11 Package and Deploy Your Application	Not Started
	12 SOA-Enable Your Application	Not Started

Now it is time to back up your application. If you ever worked with Oracle Reports or Oracle Forms, the first thing you learn is to back-up your code often. This same holds true for Oralce ADF. There are so many additional options that you are bound to make a mistake that will require you to restore from a previous version. The simple way to back up your code is to copy your application directory structure (Figure B: File Structure) to a back-up directory. Another way is to use a Third-Party tool, Subversions, which is integrated into JDeveloper.

In this next section, I will attempt to draw some comparisons from ADF to Oracle Forms.

Database Connection

JDeveloper and Oracle Developer Forms both require a database connection to access a database schema. ADF also has the ability to connect to multiple databases as well as to other sources of data (for example, BAM, BPM MDS, Content Repository, External Application, RIDC, SOA-MDS, URL, Worklist, WSIL). For this presentation, we are only concerned, with a database connection to a single database and schema.

JDeveloper Oracle Forms Developer

😓 Create Database Connection		
Choose Application Resources to create a database connection owned by and deployed with the	Connect	X
Choose Application Resources to create a database connection owned by and deployed with the current application (FormsVsAdf). Choose IDE Connections to create a connection that can be added to any application.		
	User Name: apps	-
Create Connection In: Application Resources IDE Connections 		
Cognection Name: TestDatabase	Page 44	-
Connection Type: Orade (JDBC)	Password: ****	
Username: apps Role:		_
Password: Save Password	Database: ebsa.world	
- Oracle (JDBC) Settings	- ,	
Enter Custom JDBC URL	Connect Cancel Help	
Driver:		
Host Name: JDBC Port: 1531		
○ sīD:		
Service Name: EBSA		
Test Connection		
Success!		
Help OK Cancel		

Build Business Services (Entity Objects / Views / Links / Application Module)

ADF and Oracle Form applications are typically built to assist in managing data that resides in a database. Both ADF and Oracle Forms framework support, create, update, read and delete (CRUD) operations; however, there is one major difference. ADF automatically handles CRUD operations; whereas, Oracle Forms uses the Forms record manager to store records retrieved from the database but it is the responsibility of the Developer to manage, create, update and delete (CUD) operations. In Oracle Forms, CUD operations are executed using Data Block triggers and PL/SQL code.

In ADF, CRUD operations are performed on entity objects or database tables. This is the reason why ADF requires a separate updatable view objects to be used by the UI. Entity objects are used by the ADF framework to automatically handle CRUD operations and view objects are the mechanism to maintain the data in your user interface. There are two types of view objects: Updatable and Non Updatable. If the view object is updatable, then there will be an entity object definition defined in that view object. Any view object that needs to be consumed by UI (pages) will be packaged into an application module so it can be exposed to your UI (pages).

In terms of Oracle Forms, it can be argued that each Oracle Form application contains a virtual application module because it is a collection of view objects that are implicitly exposed to the user interface or canvases.

Creating Business Service Objects

In order to assist you in creating Entity and View objects, JDeveloper has created various wizards to reduce the time to build your application. When you need to create an Entity object, place your cursor on the Model, or "MyProjectModel" and right click and select

"New". A pop-up gallery window will then appear. Select Business Tier \rightarrow ADF Business Components \rightarrow "Create Entity Object Wizard" to launch the entity wizard.

💏 Search		
Categories:	Items:	Show All Description
General Applications Applications Connections Deployment Profiles Degrams External Applications Projects Business Ter Deta Controls Security Scalarity Scal	implement business rules. To enable this option, you must selec	izard, which allows you to create an expresent tables or URL entities and to a project in the Application thing the new entity object, you will be

🍌 Initialize I	Business Components Project
	has not yet been initialized for Business Components. After specifying the following information for your ponents Project (jpx file), you will be prompted to create your Business Component(s).
Specify the d	database connection that lets you create Business Components from existing database objects.
Connection:	TestDatabase 🔹 🕹 🦯 🔍
User N	lame: apps
Driver:	: oracle.jdbc.OracleDriver
Conne	ct String: jdbc:oracle:thin:@//oraebsdb-a.panduit.com:1531/EBSA
Choose the p	proper SQL flavor and type map that fits your application.
SQL Flavor:	Oracle 🗸
Type Map:	Oracle 💌
Help	OK Cancel

Van	eate Entity Object -		
<u>ب</u>	Name	Entity objects are used	to encapsulate the business logic and database storage details for your business entities.
	Attributes Attribute Settings Java Senerate Summary	Neme: 200 Estends Entity: 200 Property Set: 400 Data Source - Decide whether to ba (*) Database Schema Qatabase Schema:	se your entity object on a database schema object, or a Business Components service interface.
	elp		< Back Next > Finish Cance

ttributes		100
Name	Click New to create new attributes. Click New from Table to add attributes for unmapped columns.	
Attributes	Entity Attributes:	
Attribute Settings	HeaderId	^
Java	OrgId OrderTypeId	1
Generate	OrderNumber	
	VersionNumber	
Summary	ExpirationDate	
	OrderSourceId	
	SourceDocumentTypeId OrigSysDocumentRef	
	SourceDocumentId	
	OrderedDate	
	RequestDate	
	PricingDate	
	ShipmentPriorityCode	
	DemandClassCode Pricel istId	
	TaxExemptFlag	
		~
	New New from Table Remove	

Since a primary key is not defined in the database, select primary key.

ibute Settings				PARABACILA
Name Attributes	Select Attribute			Updatable
Attribute Settings	Name: Iype: Property Set: Value Type: Value:	HeaderId Number <none> © Literal O Expression</none>	•	Browse Algays O While New Edit
	Persister Mandatg Change Derived Precision Disgrimin Default \ Chatabase Colum	87Y Indicator from SQL Expression a Rule aktor alue:	Primary Key Urigue Queryable Effective Date Start End History Column	_Refresh After ☐ Update ☐ Insert
Help	Name: HEADR		Type: NUMBER	
Create Entity Object - S	tep 4 of 6		0101010101010101010	
lava				
Attributes	Generate Ent			

Help

<Back Next > Einish Cancel

enerate	
Name	Select the checkbox to create a default view object, containing an attribute for each of the entity object attributes.
Attributes	Generate Default View Object
Attribute Settings	Package: com.xx.myproject.model Brow
Java	Name: oeOrderHeadersAllVVO
Summary	Select the chedbox to add an instance of the view object to the specified application module. If the application module do not exist, it will be created. Application Module Padage: Construmerorect.model Brown
9 <u>Summary</u>	not exist, it will be created.





Perform the same steps for oe_order_lines_all

When you are done, your project should look like



In Oracle Forms, database view objects perform roughly the same function as an ADF view object. It is the mechanism to maintain the data in your user interface. As with ADF framework, Oracle Forms framework has wizards to create your Data Block (view objects):

Create data blocks (entity & view objects).









Perform the same steps for oe_order_lines and when you are done the Forms navigator screen should look like



In ADF, the next step is to create a master-child relationship between the headers and lines table. The relationship is connected by a database column or header_id. ADF supports two different styles of master-detail relationship:

- Association This is a master-detail relationship defined at the entity level and since it is defined at the entity level, each entity object can reference each other in CRUD operations. As a general rule, create an association link when you need to enforce business rules, create or delete records in a master detail relationship. Note: If you have primary-foreign key relationship defined in your database, then an association link will automatically be created for you.
- <u>View Link</u> This is a master-detail relationship defined at the view level and is mainly used to sync up blocks of data in your application. If you are creating a master-detail relationship screen in your application, then you should create a view link because it supports automatic master-detail synchronization. If you have association, then use the association to link the two objects in the UI.

Here are the steps to create an association link






reate Association - Step	3 of 5		
sociation Properties	5		1
Tome Entry Objects Association Property Edit Association Query Summery	Source Accesso Entity Object: ecOrderHeadersAll © Egopoe in: ecOrderLinesAll Accessor Nome: peOrderHeadersAll	Cestnation Accessor Entry Object: ceOrderU C Exgose in: ceOrderH Accesso; Name: peOrderUnesAl	
	Christen för Calabater Casada Deleter Deplement Casada Deleter Casada Egdets för Antsolan Lögde Tork Antsolan Gingen Oland Cardjaner Olan Effective Entert Association	h Taplevel Cantalous	
denne bladen			
reate Association - Step t Association Quer	,		
	Source Attributes Bind Variables	Destination Attributes	Bind Variables
t Association Quer	Seurce	Destination	Ind Vandes Sted Jasserid
t Association Quer Name Britty Objects Association Properties Edit Association Que	Source Attributes Bind Variables	Destination Attributes	
t Association Quer Name Britty Objects Association Properties Edit Association Que	Sorre Jackson Britanski odolmanimalmakrid drei padrid	Destination Attributes	

	Where clause for Destination to Source	Test	Explain Plan
	:Bind_HeaderId = oeOrderHeadersAILHEADER_JD		
		Test	Explain Plan
Help	< §adt	Next >	Enish Cano



Here are the steps to create a view link.



Note: I am using the association in the view link

ew Objects		
Name	Select each pair of source and destination view object at	tributes that do ne the view link, then click Add.
View Objects	Cardinality: 1 to *	
View Link Propert Edit Source Quer Edit Destination C Application Modul Summary	uery estimates and a second se	Select Destination Attribute: Generation Attributes Generative and CoheAttributes Generative and CoheAttributes Generati
	Source Attribute(s) HeaderToLinesLink	Bestmation Attribute(s) Bestmation Attribute(s) HeaderToLinesLink

ew Link Properties	
Versen in Verse State Anseeser Verse Lad Anse	ett DeOrderLeesARVO

dit Source Query		
	Source	
Name	Attributes	Bind Variables
Ven Link Properties	seOrderHeadersAILHEADER_JD	Bind_Header1d
Edit Source Query		
Edit Destination Query		
Application Markale	Outry Clauses	
Same	Where dause for Source to Destination	
	Bind_Headerid + ceOrderLinesAl HEADER_30	
		Ited Epiler Plan

> Create View Link - Step	5 of 7	
Edit Destination Que	ny	
	Destination	
Vero Objects Vero LER Properties	Attributes peCroferLinesALHEADER_ID	Bind Variables dbind_Header3d
Edit Source Overy Edit Destination Que		
Aufkator Hodde	Query Classes Where glasse for Destination to Source (Bind Josedari d = ceOrdomeaderski/EADEL_D	
		Tegt) Englien Han
Bib		< gad. gent > Brish Cancel





When you are done defining the links, your project should look as follows:



In Oracle Forms, we do not have a distinction between table-based and view link masterdetail relationship. We have a single relationship, which is a combination of an association and view link. It supports auto-query of the detail block and enforces cascade deletes.

Create a relationship between headers and lines (association and view links).



Once the relationship is created, Oracle Forms will automatically add code to enforce the relationship.



Now that we have our business relationships defined, we need to apply a common business functionality to help our Users understand the data. One such business component is a list of values (LOV) component, which translates database Ids to readable values.

Here is an ADF example of creating a list of values related to the inventory_item_id on the sales order line table. The database id should be translated to a SKU# or product code. ADF provides a wizard to create a read-only view object. The wizard steps are displayed below:

1. Create a read-only view

Name			-21
R Name	Ven skjects ø	re for joining, filtering, projecting, and surfing your business data for the specific needs	of a given application task.
Entity Objects	Package:	com.xx.myproject.model.vo.readonly	Brogse.
Attractor	Neget	mdSystemDensVO	
Alternative Settings	Display Name:	Md System Items Vo	9
Query .	Egtends		Bipvise.
in Bed Variables	Experty Set:	diane>	•
		aled programmatically, not based on a query lajed at design time (Static Lief)	

2. Add your SQL statement with a parameter

Que	sry	
示	Name	Enter your custom SELECT statement and click Test to check its syntax. Provide the ORDER BY dause separately.
÷	Query	(Query Statement
	<u>Bind Variables</u> Attribute Mappings Attributes Attribute Settings Java Application Module Summary	neise seguenti ine, i energenzazi ine, i energenzazi energenzezi e
		Qary Class

3. Define your parameter – Note: In this case, we hard coded the organization_id but in a real application I would derive the value and set a session value.

Bir	d Variables				- 21
	Name	Xariables:			
	Querx	pOrganization	10		
3	Bind Variables				(† (†
ţ	Attribute Mappings				
Ŷ	Attributes	Neg Reng			
	Attribute Settings	(mail) (mail			
	Java	Variable	Custom Properties Control Hints		
	Application Module	Name:	pOrganizationId		
	Summary	[ype:	String		· Prowse
	Summary	Type: Value Type:	String • Literal _ Expression		Test
	Summary				
	Summary	Value Type:	Uteral Expression	✓ Bequired	Test

Here is an example of passing a session value.

Versioning Too	als Window Help						
- i 🕨 - 👼 -	n Hannen Teh						
	Address of the second	ter firms and x	Silon data data data data data data data dat	ni X GoeOrderHeadersALxmi X	Destantion	formelle and the	6
(1) Myrrojeco-ppro	iouicantor.java × Canabys	NETULETIS.KTV A	Estodenceordereattencov.x	is A Togloeordemeaderskalters A Ti	g orde salesrage los	iscopebeart.java ×	
General Entity Objects	Query						
Attributes							
Query	Data for this view object v	ill be retrieved from	n the datasource using the follow	ing SQL query.			
Java	select segmentl it	en.					
View Accessors	inventory_i						
List UI Hints	from mtl_system_it						
	where msi.organiza ORDER BY item	tion_id = :po	rganizationid				
	Bind Variables					+ 🖊 🗙	Override
	Named bind variables can	be used in the SQL	query of this view object.				
				2			
	Name		Type	Value	Info		
	pOrganizationId		String	return adf.userSession.user			
	View Criteria	📥 Bind Variab	ie .		×	n -	F/x
	Vew criteria are named ex	-					
		Variable	Custom Properties Control Hin	8			
		Name:	pOrganizationId				
		Lype:	String		· Browse		
		Value Type:	Uteral Expression		Test		
		Yalue:		.userData.organizationId;	Edit		
Overview Source	Materia (Tone.	Lecols add. usersession	.userbaca.organisacionid,	Fac		
Messages - Log		1				-	
	plete connection real						6
	:01:35 FM oracle.se		 Updatable 	Required			
	d to validate the x					complete. One of	
	s.oracle.com/oracle. 1:01:35 PM oracle.set					£a.	
	d to validate the x					complete. One of	
'{"http://xmln	s.oracle.com/oracle					La.	
	1:01:55 PM oracle.ad						
	ensure that a Defaul						
Memory leaks a	nd/or unexpected be			OK	Cancel		
This message m	may be avoided by per						

Value	return adf.userSession.userData.organizationId;
-------	---

This organization_id session value is set in the AM Module. The organizationId is determined by another processes and in Oracle EBS it would be determine by calling a database function "fnd_global.apps_initialize"

```
private String getOrganizationId() {
    return (String)getSession().getUserData().get("organizationId");
}
private void setOrganizationId(String organizationId) {
    getSession().getUserData().put("organizationId",organizationId);
}
```

4. Identify what column in your view is the primary key.

attribute Settings				. • 2
, Name	Seject Attributes	InventoryItemId		Updatable
Query	Name:	InventoryItemId		
Bind Variables	Type:	Number	• Brow	58
 Attribute Mappings 	Property Set:	(tione>	-	
Attributes	Value Type:	Uteral ○ Expression		Algers
Attribute Settings	Yakue:		Edi	While Ne
Application Module Summary	Selected	ator: O View O Entity	Key Attribute Quegrable Effective Date Start End	 Neger
	Query Column			
	Alag:	INVENTORY_ITEM_ID	Type: NUMBER	
	Expression:	INVENTORY_ITEM_ID		

Click Finish

5. This next step needs to associate the list-of-values view to the attribute or database column defined on the sales order line view object or database table. In terms of ADF, this is called creating a view accessor to your sales order line or view object. The definition of a view accessor is the mechanism that lets you

obtain the full list of possible values from the row-set of the data source view object.



Note: You could also set the Vo parameter via the edit.... Option

Click On "Plus" Sign	Edit
v Versigning Iools Window Help	
) 🖶 MyProjectAppModuleImpl.java × 💆OeOrderLinesAllVVO.xml × 🔯 oeOrderHeadersAll.xml × 🛗 OrderS	alesPageFlowScopeBean.java ×
Sedit List of Values	View Accessors
List of Values Name: LOV_InventoryItemId	Select a view object or shared view instance and shuttle it to the selected list to create a view accessor.
Configuration UI Hints	Select a view doject or shared view instance and shuttle it to the selected list to create a view accestor.
Select a view accessor for the list data source, and then choose the list attribute that maps to the current viet object attribute.	Available Vew Objects: Vew Accessors: Edit @vernde
List Data Source: mtlSystemItemsLov	
List Attribute: InventoryItemId	er un contactingo gettinoder volledorny
List Return Values	
Map any supplemental values that your list returns to the base view object (it always returns a value to the attribute for which the list is defined).	
View Attribute	
InventoryItemId InventoryItemId	
	Name: Accessor Name: mtlSystemItemsLov
	Definition: com.xx.myproject.model.vo.readonly
Help OK Cancel	Help OK Cancel
Produkte Male and Analysis for Features Field of and and be descent in the descent	Dioday Name
Sedit View Accessor: mtlSystemItemsLov	
View Object	Number
View Object Configure the view object query for this accessor.	<none></none>
View Definition: com.xx.myproject.model.vo.readonly.mtlSystemIter	ms Diview O Entit
- View Criteria	
Select the view criteria that you want to apply to this view object. If	you select multiple view criteria, they will be combined with an AND operator.
Agalable:	Selected:
	>
G Gind Parameter Values	Uiteral O Ex
Provide values for any bind parameters defined for this query.	
Parameter Type	Value
	eng.String
9	<default> (true)</default>
T Order By:	Edit <default> (true)</default>
, N Help	Apply OK Cancel
	j -dektien and na marka meno
	Enter Value

Once the view accessor is added to the line view object, it will be available in the "List Data Source" drop down list. Select the list data source and complete the mapping.

st of Values <u>N</u> ame:	LOV_InventoryItemId	
Configuration	JI Hints	
Select a view acce object attribute.	ssor for the list data source, and then	choose the list attribute that maps to the current view
List Data Source:	mtlSystemItems	
List Attribute:	InventorvItemId	
List Return Values		
List Return Values Map any suppleme attribute for which	ntal values that your list returns to the the list is defined).	base view object (it always returns a value to the
List Return Values Map any suppleme attribute for which	ntal values that your list returns to the the list is defined).	List Attribute
List Return Values Map any suppleme attribute for which	ntal values that your list returns to the the list is defined).	

	toryItemId
Configuration UI Hints	
Defaul <u>t</u> List Type: Input Text	with List of Values
Display Attributes	
	list of values and combo box. Optionally show a subset in the combo box
inultiple values are separated l Available:	sy write space). Selected:
InventoryItemId	Item 1
ist Search	Show in Combo Box: Al V
	Show in Compo Box: All V OG
nclude Search Region: All Qu	
nclude Search Region: All Qu Query List Automatically	
ndude Search Region: All Qu Query List Automatically Choice List Options	eryable Attributes
ndude Search Region: All Qu Query List Automatically Choice List Options	eryable Attributes

After the list of value is attached, you will see the list-of-values in the attribute window.

FormslisAdf Ov	erview × MyProjectAppModule.x		derLinesAllVV0.xml ×			
ieneral						
ntity Objects	Attributes					Set Source Order
Attributes	Vew object attributes can be mapp	ed to entity attrib	sutes, calculated or SOL-derived.			
juery .	(Anne		·			
ava lew Accessors	(W THEFT		V 1)			+·/X
ist UE Hints	Name	Type	Alas Name	Entity Usage	anfo	
	InventoryTeenad	Number	INVESTORY TRAVED	seOrderLinesAl		2
	TaxDate	Date	TAX DATE	peOrderi pesAl		
	TayCode	String	TAX CODE	orOrderLinesAl		
	TaxRate	Number	TAX RATE	oeOrderLinesAl		
	InvoiceInterfaceStatusCode	String	INVOICE INTERFACE STATUS CODE	oeOrderLinesAl		
	DemandClassCode	String	DEMAND CLASS CODE	peOrderLinesAl		
	PriceListEd	Number	PRICE LIST ID	oeOrderLinesAl		
	PricingDate	Date	PRICING DATE	peOrderLinesAl		
	ShipmentNumber	Number	SHOPMENT_NUMBER	peOrderLinesAl		
	Agreementid	Number	AGREEMENT_ID	peOrderLinesAl		
	ShipmentPriorityCode	String	SHIPMENT_PRIORITY_CODE	oeOrderLinesAll		
	Custom Properties: Invento	ryItemId				+-/x
	Elist of Values: InventoryIte	mId				*/×
	Enable this attribute to display a list	t of values to use	in the user interface.			
	Lists of Values:					
	Name LOV Inventors/tentid		List Data Source mtDVstemiterrs		ist Attribute	

This is very similar to creating a list of values in Oracle Forms. First, a record group (view object) needs to be created. Then we need to create a list of values object, which is an explicit mapping of the record group result set to a data block definition. Lastly, we need to link the list of values definition to a specific field in the data block.

For my Oracle Forms example, I did not use the list-of-values wizard because I want to show the same type of steps as in ADF.

1. Create a record group which is equivalent to a read-only view object in ADF.



2. Create a list of values (LOV) mapping between the definition of the result set and the Data Block field.

Use the LOV <u>Wi</u> zard Build a new LOV <u>manually</u> <u>OK</u> <u>Cance</u>	l <u>H</u> elp		
Build a new LOV manually	I <u>H</u> elp		
Build a new LOV manually	Help		
	Help		
<u>OK</u> <u>C</u> ance	Help		
	a <u>neib</u>		
cle Forms Builder - C:\FORMVSADF.fmb			
at Property Program Tools Window Help			
bject Navigator	(m)		
	Property Palette		
MLOV 💌	- 二二	n 🖌 🗌	Find:
©-Forms	LOV: ITEMLOW		
B- M FORMVSADE	= General		
Triggers Alerts	* General	ITEM	4+1
Attached Libraries		LOV	0v
Data Blocks	Subclass Information	LOY	
G. ORDER_HEADERS D. Triggers	Connects Functional		
ON-POPULATE-DETAILS	# Functional	Dec. 1	ist Of Values
ON-CHECK-DELETE-MASTER	■ TRie = List Type		d Group
Relations	Record Group	ITEM	
SOE_ORDER_HEADER_OE_ORDER	Column Mapping Properties	11274	Nore
In the operation of the second sec	Couries Happing Properties Filter Before Display	No	P0010
B 2 CALENDAR	Pater berore Display Automatic Display	No	
B - B BLOOMAME B - B PROGRESS_INDECATOR	Automatic Display Automatic Refresh	Yes	LOV Column Mapping
E DETAILBLOCK	Automatic Select	Yes	Column Names
Canvases	Automatic Skip	Yes	SEGMENTI
Editors ⊡-LOVs	 Automatic Position 	Yes	INVENTORY_ITEM_ID
TENLOW	Automatic Column Width	Yes	
TIMEZONES_AND_DATETIME	# Physical	res	
TIMEZONES	+ X Position	0	
APPCORE_ZOOM	+ Y Position	0	
APP_ACCESS	a wath	3.75	
ENABLE LIST LAMP	h Height	4	Return Item
Object Groups	2 Tage	4	OE_ORDER_LINES.IM/EP Browse
Parameters Popup Menus	Property 6363 Hint		Concerned in the second
Program Units	Property constraint.		Display Width
B Q APP_CUSTOM (Package Spec)			0
B APP_CUSTOM (Package Body)			Column I tie
OHECK_PACKAGE_FAILURE* (Procedure Bo OHECK_PACKAGE_FAILURE* (Procedure Bo OHECK_PACKAGE_FAILURE* (Procedure Bo OHECK_PACKAGE_FAILURE*)			
OFECK_PACKAGE_FALLRE* (Procedure Bo OLEAR_ALL_PASTER_DETAILS* (Procedure Bo OLEAR_ALL_PASTER_DETAILS* (Procedure Bo Property Classes	h/)		QKGancelB

3. Link the list of values to a data block field

IMENTOR		6		
Interior.	at Marth	Property Palette		
נ	8-50 ORG_ID	3.2 2.5 3.4	∩ 📌 Find:	8. 4
2	B-BEI HEADER_ID	Item: IMENTORY_ITEM_ID		
3	8- GELLINE_TYPE_3D 8- GELLINE_NUMBER			
-	8 WE CROERED ITEM	= General		
8	R-GO REQUEST DATE	None	INVENTORY_ITEM_ID	
8	B- HE PROMISE DATE	Item Type	Text Item	
2	8 GE SCHEDULE_SHIP_DATE	 Subclass Information 		
-	B GED ORDER_QUANTITY_UOM	 Comments 		
<	IEEE PRICENG_QUANTITY	+ Help Book Topic		
	8- E PRICING_QUANTITY_UOM	* Functional		
	B-BO CANCELLED_QUANTITY	# Navigation		
	Generation Generation	* Data		
	8 GO PLUFILLED_QUANTITY	* Calculation		
	8- CO SHOPPING_QUANTITY	* Carcuston		
	B B SHIPPING QUANTITY_UOM	* Records		
	IN SEE DELIVERY LEAD, TIME			
	8 GE TAX_DENPT_FLAG	= List of Values (LOV)		
	E TAX_EXEMPT_NUMBER	List of Values	ITEM, OV	•
	E TAX_DEMPT_REASON_CODE	 List X Position 	0	
È i	8-80 SHP_FROM_ORG_ID	 List Y Position 	0	
	B G SUBWENTORY	 Validate from List 	No	
	R GE INVOICE TO ORG ID	= Editor		
	8 G DELIVER_TO_ORG_D	 Editor 	 dute 	
	B BI SHP_TO_CONTACT_ID	 Editor X Position 	0	
	- MI DELIVER TO CONTACT ID	 Editor Y Position 	0	
	IS HE INVOLCE_TO_CONTACT_ED	= Physical		
	Image: Sold_TO_ORG_DD	 Visible 	Yes	
	B-BD CUST_PO_NUMBER	 Canvas 	(94)	
	8: EE SHEP_TOLERANCE_ABOVE	+ Tab Page	<na></na>	
	B-B SHP_TOLERANCE_BELOW	+ X Position	0	
	B DEMMD_BUCKET_TYPE_CODE B B RLA_SCHEDULE_TYPE_CODE	Y Position	0	
	8- GEI CUSTOMER_DOCK_CODE	Whith		
	B-BBI CUSTOMER_308		1.031	
	B GUSTOMER_PRODUCTION_LINE	Height	.100	
	8 - EED CUST_MODEL_SERIAL_NUMBER	+ Devel	Lowered	
	8- GEI PROJECT_ID	 Rendered 	Yes	
	8 GE TASK_ID	The lat of values (LOV) to attach to		
	IN THE INTENTORY ITEM ID	The lat of values (LOV) to attach t	is the text item.	•

Once we have all our business services (entity objects, view objects, links and list of values objects) completed, our next step is to group the view objects together into application module so they can be exposed to the View layer. Here are the steps to create an application module in ADF:

🔀 Search	current project's selected technologies.
Categories:	Items: Show All Description
General General Applications Connections Connections Depolyment Depropries Depolyment Depropries Depolyment Depropries Depolyment Depropries Depolyment Depropries Depolyment Depropries Depolyment Depolyment Depolyment Depolyment Depol	Business Components from Tables Application Module Nazard, which allows you to create an application module. Use application modules to assemble and organize methods. To enable this option, you must select a project in the Application Novigator. Before you can finish creating the new application module, you will be promoted to assess the application module. You will be promoted to assess the application module is a select (or create) a database connection. Business Components Deployment Profiles Business Components Deployment Profiles Default Data Model Components Deployment
Help	OK Cancel
eate Application Module - Step 1 of	s
	5
Application ms	S dules are for assembling, packaging, and deploying the view objects and business services of an application.
Name Application mo	
Name Application mo Data Madel Package:	dules are for assembling, padcaging, and deploying the view objects and business services of an application.
Re Application mo Data Model Package: http://cation.Modules Name:	dules are fir assembling, packaging, and deploying the view objects and business services of an application. com.xx.myproject.model.am Broggee MyProject/opModule
Re Application mo bata Model Package: Application Modules Name: Java Dippley Name:	dules are fir assembling, packaging, and deploying the view objects and business services of an application. com.xx.myproject.model.am Broggee MyProject/opModule
Name Data Model Package: Application Modules Name: Java Display Name: Summary	dules are fir assembling, packaging, and deploying the view objects and business services of an application. con.xx.nyproject.model.am Broggee My Project.app Module Vy Project.app Module

Once you shuffle "click Arrow" the Header and Line, click finish.

< Back Next > Einish Cancel

ata Model		\backslash				1
Kane Data Hodel Data Hodel Actication Modules Java Summary	data model tree, and circ Ayailable View Objects: Com.xx.myproject.m (a) (a) com.xx.myproje (a) (b) com.xx.myproje (c) (c) (c) (c) (c) (c) (c) (c) (c) (c)	ct.model.vo dersAlVVO rLinesAlVVO via HeadersToLinkl isAlVVO	of the view o		model. odule idersAll	be its parent in th
Help	New View Instance: New View Link Instance:	OeOrderLinesAllVVO1 HeadersToLinkLink2	Vie	e <u>w</u> Instance: ew Lin <u>k</u> Instance: ew Definition: ew Link Definition: < Back	com.xx.myproje	nk1 ct.model.vo.OeO

Help

Note: Once you bind the objects in your UI Layer, the application module representation is visual in the View Layer Bindings (see below).



When you are done defining the application module, refresh the "Data Controls" window and your project should look as follows:



For Oracle Forms, there is no action required to expose the Data Blocks to the user interface because the application module is implicitly exposed to the user interface or canvas.

Initialize Application Session Values

For either Oracle Forms or ADF applications, we need to initialize the session when the application is launched. The purpose of the initializing is to prepare the application for the User who is about to enter into the application. In Oracle Forms, this is performed by the "When-New-Form" trigger which sets session specify values like inventory warehouse, operating unit, the correct operating unit, initialize descriptive flexfields or set the initial visual canvas. In ADF, the same actions can be performed by calling a

managed bean method or an application module method. As a refresher, a method is a collection of Java code with a sole purpose to carry out a task.

In a real-life ADF application, we would set session specific variables within the initialization method. For example, based on a given users responsibility, we could set visible and edit privileges within the application, specify an operating unit, determine what UI page should be rendered and place scope specific variables into Java Map so they can be referenced in other parts of the application.

Initializing your application is tightly coupled with controlling the flow of your application, so I will explain how you initialize your application in the following section (Design Application Flow).

Design Application Flow

This next section compares how you control the flow of your application in Oracle Forms versus ADF. In Oracle Forms, we initially control the flow of the application by setting the "First Navigation Data Block" in project properties pallet.

	Constant Frigure Frig	Property Public Property Public Property Public Property Public Property Public Property Public Public	PORMSADE HOOME	
	Triggers Marks Marks Marks Data Blocks B ⊕ CC, CARDE JEACERS B ⊕ CC, CARDE JEACERS	Form Module: FORMSADF # General # Name # Studies Priormation - Connerts - Help boot Tals # Franctional # Menos Security M Meru: Role	FORMISADE	P. 1
	Triggers Aktos Attached Ubraies B ⊕ CC_ROEP, JEADENS B ⊕ CC_ROEP, JEADENS B ⊕ CC_ROEP, JANS B ⊕ CC_RO	Ceneral Anne Subclass Information Comments High Book Tale Franctional Hous Security Menu Role		
	Alerts Matched Ibraries Data Blocks Big Oct, ORDER, JALEES Big Oct, ORDER, JALEES Big Oct, ORDER, JALEES Big Oct ALERCONARC Big Optimized Status Big Optimized Status Converses Intere	Nome Subclass Information Comments Insurerts Indu Dock Table Functional M House Security Menu Role		
	Attached libraries Data Miccia Data Miccia	Nome Subclass Information Comments Insurerts Indu Dock Table Functional M House Security Menu Role		
	⊕ ⊕	Subclass Information Comments Help Book Title Franctional Menu Security Heru Role		
	 ⊕ of_order_lines ⊕ of_order_lines ⊕ order_lines ⊕ orderst_index ⊕ process_index for ⊕ process_index for ⊕ process_index for ⊕ orderst_index for ⊕ ordex for ⊕ orderst_	Connerts Halp Book Title Functional Menu Security Menu Role		
	B - SB CALENDAR B - SB ELOCANAME B - SB PROGRESS_INDICATOR B - SB DETAILELOCK Canvases falters Lavs	Help Book Title Frunctional Menu Security Menu Role		
	Bacomane Brogress_INDICATOR By Progress_INDICATOR Greases Iddors Lotys	# Functional # Menu Security & Menu Role		
	8 9 PROGRESS_INDICATOR 9 90 DETAILBLOOK Canvases Iddors LOVs	= Menu Security Menu Role		
	E B DETAILELOCK Canvases Editors LOVs	Menu Role		
	Canvases Editors LOVs			
	Editors LOVs	# Navigation		
		Mouse Nevigation Limit	Form	
	Object Groups Parameters	First Navigation Data Block.	OE ORDER HEADERS	
8	Popup Menus	* Records		
в	Program Units	* Database		
	B APP_CUSTOM (Package Spec)	= Physical		
	APP_OUSTOM (Package Body)			
÷1	- CHECK_PACKAGE_FAILURE* (Proces		Yes	
2	 DETAILS* (Procedule) OUERY_MASTER_DETAILS* (Procedule) 			
8	Property Classes		TOOLBAR	
	Record Groups	 Form Vertical Toolbar Canvas 	dute	
	Reports	= International		
	Visual Attributes Windows	 Direction 	Default	
O-Menus		= Compatibility		
	Libraries	Runtime Compatibility Mode	4.5	
O Object	Libraries Packages		1	

The initialization of the Oracle Forms session and the option to set the initial canvas is set by the "Pre-Form" or the "When-New-Form" trigger. Here is an example of some PL/SQL code that is called by the form level trigger WHEN-NEW-FORM-INSTANCE:

```
|| Assign Valued To Control Block
  :CONTROL.USER_ID := l_user_id;
|| Set Operating Unit Name
*/
 SELECT hou.name
  INTO l_name
FROM hr_operating_units hou
  WHERE hou.organization id = 1 org id;
|| Set Window Titles
*/
win_msg := 'Order Maintenance' || ' (' || l_name || ')';
set_window_property('MAIN_WIN', TITLE, win_msg);
 || Get and Set Percision
 */
  SELECT fc.precision
   INTO l_percision
FROM hr_operating_units hou,
  gl_sets_of_books gsob,
fnd_currencies fc
WHERE hou.organization_id = l_org_id
   AND hou.set_of_books_id = gsob.set of books id
   AND gsob.currency_code = fc.currency_code;
   :PARAMETER.SOB PERCISION := NVL(l percision, 2);
/*
|| Set DFFs
*/
   FND DESCR FLEX.DEFINE
     block => 'OE_ORDERS_HEADERS',
field => 'DF',
   (
     appl_short_name => 'ONT',
      desc flex name => '<some value>',
      title => '<some value>'
   );
END when new form instance;
```

For Oracle Forms, the Developer can use hot-keys, GO-BLOCK and CALL-FORM options to control the flow of your application. The navigation coding is buried within forms triggers and PL/SQL code and sometimes can be difficult to debug as compared to Oracle ADF, which has a tool to visually control your flow of the application.

In ADF, you control the flow of your application by using a task flow. A task flow is a visual way to control the flow of your application. When you initially create your bounded task flow object, the task flow will be completely empty and it is your responsibility to visually define your flow. Once you launch the task flow wizard, you will see the following:

Oracle JDeveloper 11g Release 1 - FormsVsAdLjws : MyProject	ULjpr		
File Edit View Application Refactor Search Navigat	e <u>B</u> uild <u>B</u> un Versi <u>o</u> ning <u>I</u> ools	Window Help	
🕒 🔤 🛃 i 🤊 🤍 i 💥 🛍 🛍 i 🔾 • 🔘 • i 🏤 • i	👍 \$\$ 🚑 🛳 이 🕨 이 🌞 이		
2. Application Navigator ×	 EFormsVsJ 	Adf Overview X I 🎰 MyProjectAppModule.xml X 💆 OeOro	lerLinesAlVVO.xml ×
FormeVsAdf	- 🖪 -		
E Projects	🖪 🗞 🍸 * 🎏 * 👘 General		
Projects Projects Projects Projects Projects Projects Projects Projectmodel.an Projectmodel.an Projectmodel.an	📥 New Gallery	and development of the second se	- ×
	All Technologes Current Project Te This list is filtered according to the cur		
a la la becroemesersAl ≥ la la cal ceorderLinesAl	Categories:	Items:	Show All Descriptions
Constant of the second se	Connections Deployment Descriptors Deployment Profiles Diagrams External Applications Java	ADF Taskk Flow Creates a task flow source fle whose contents def flow or part of the web application's unbounded ta To enable this option, you must select a project or Application Navigator.	sk flow.
HeadersToLinkLink Gr. xx.myproject.model.vo.readonly	Projects	ADF Task Flow Based on Human Task	
com.xx.myproject.model.vo.readony indef.vo.readony indef.vo.readony	-XML - BPM Tay	ADF Task Flow Template	
⊖ O MyProjectUI	-SOA Tier	- ISF Declarative Component	
Web Content WEB-2NF	-Web Tier	35F Page	
Page Flows	-HTML	35F Page Flow and Configuration (faces-config.am	
Application Resources	- 25F	15F Page Fragment	
Application Resources Data Controls	JSP Serviets		
H	All Items	SF Page Template	
Recently Opened Files	. Help	(OK Cancel

Create Task Flow		×
unbounded task flow. A bounded task flow can re	ie whose contents define either a bounded task flow or part of the web application fer specifically to JSP pages or page fragments, but not both. You can also flow to be a train at this time.	on's
Eile Name:		
SalesOrders-btf.xml		
Directory:		
	sVsAdf/MyProjectUI\public_html\WEB-INF	Browse
Create as Bounded T	ask Flow	
-		
Create with Page Fr	agments	
Create Train		
Base on Template:		-
✓ Update the Task	Flow when the Template Changes	
Help	ОК	Cancel
_	Subsequences, half in Country, Sont Month Ind	
ld <u>B</u> un Versigning <u>T</u> ools <u>Window</u> <u>H</u>	de la constance	

ld Bun Versigning Lools Window Help	
1.4 m · () · · · · ·	
Semaniskel Overview × MMyProject/cpModule.aml × MOOrOrderLinesAMVIO.aml × CasesOrders-bdf.aml ×	() ()
<a>4 10% • 200 • 11 • 7 1 27 kg • 0 18g	20
Sounded Test Prov	
Drop content onto this blank diagram from the Component Palette,	
Resource Palette, or Application Navigator.	
Dagram Overview Source History <	~ > •

In ADF, we initialize our application by adding a "Method Call" activity in the task flow and we set this activity as the default. The Method Call activity will reference a managed bean method or application module method. In this example, I am referencing a managed bean method.

_			I TO THE MALE AND AND A
٨	Oracle JDeveloper 11g Release 1 - FormsVsAdf	jws : MyProjectModel.jpr	
Ð	le <u>E</u> dit <u>V</u> iew <u>Application</u> Refactor <u>S</u> e	earch <u>N</u> avigate <u>B</u> uild <u>R</u> un Versi <u>o</u> ning <u>T</u> ools <u>W</u> indow <u>H</u> elp	
	- 🔾 i 🎒 🐰 i 🦻 🍋 i 🕄	🖸 - । 🏯 - । 📥 🕍 🚢 🗮 - । 🕨 - 🌞 -	💏 - Search
	Application Navigator ×	MyProjectAppModule.xml × Model Alive.xml × SalesOrders-btf.xml ×	
Application Server	🔁 FormsVsAdf 🔹 🔁 👻	🍳 🔍 100% 👻 i Show * i 🚺 🐴 🦷 i 🚟 👫 i 💿 🔂 i 🔞 💥 🕄 🏹	ADF Task Flow
cation	▼ Projects 💽 🕅 🏹 ▼ 📴 ▼ 🗌	Bounded Task Flow	ee
I Ser	MyProjectModel Application Sources		Source Elements
ĕ	com.xx.myproject		Components
	B-m model		- Activities
		Select Method Call	Method Call
5	t∎ 🍿 am	Scient Michild Call	Rarent Action
H R	B - @ assoc		Router
l e			• · · · · · · · ·
8	i⊞@ vo		😨 Save Point Restore
Database Navigator	MyProjectModelBundle.pr		Task Flow Call
igi	😑 🛅 MyProjectUI		🕼 Task Flow Return
đ			I URL View
	🖻 🔚 Web Content		Se View
	🔅 🔚 WEB-INF		4
	🖻 👘 Page Flows		- Control Flow
≣₹	adfc-config		- Control Flow Case
Thumbnai	SalesOrders-btf	InitializeSalesOrderMethod	🔆 Wildcard Control Flow Rule

In this example, I named the "Method Call" activity InitializeSalesOrderMethod and it references a managed bean called initalizeScopeValues.

🐯 method-call - I	nitializeSalesOrderMethod - Prope	erty Inspector X	_
强 i 📌 📴 i	/	馣 Find	₽ ₽ ?
🗆 General			^
Activity ID *:	InitializeSalesOrderMethod		
Method *:	#{pageFlowScope.OrderSalesPfs	s.initalizeScopeValues}	~
···· Outcome *····			
 Fixed Outcome 	e: SalesOrderJsff		• ~
toString():	<default> (false)</default>		• ~ .
Description			
Display Name:			~
Description:			~
Small Icon:			~
Large Icon:			~ _
Parameters			
			+ X
Class		Value *	
Return Value:			~ v
		Pa	age Flow Editing

Here is a sample code that would be contained in the managed bean method. In this case, we are referencing an application parameter, which is defined in your task flow. In addition, it is calling an Application Module (AM) method that will initialize the EBS database session variables.

```
public void initalizeScopeValues() {
   System.out.println("Start - initalizeScopeValues");
   Map<String,Object> params = new HashMap<String,Object>();
   Map pfMap = AdfFacesContext.getCurrentInstance().getPageFlowScope();
   // These are task flow parameters which are similar to Oracle Form Parameters
   params.put("Warehouse", pfMap.get("pWarehouse"));
   System.out.println("Parameter - Warehouse: " + params.get("Warehouse"));
   // Get Session Values Like User Name
   SecurityContext context = ADFContext.getCurrent().getSecurityContext();
   String username = context.getUserName().toLowerCase();
   System.out.println("Variable - username: " + username);
   //Call AM Method To Initalize Session Variables In The Model
```

```
BindingContainer bc = BindingContext.getCurrent().getCurrentBindingsEntry();
bc.getOperationBinding("setAMSessionVariables").execute();
System.out.println("End - initalizeScopeValues");
```

In order to reference an AM method, you first need to create an Application Module Java class, add that method to the Application Module so it can be exposed via a Data Controller, which in turn allows you to consume and bind that AM method to Method Call Activity or UI Page. Here are the steps to create an AM Java Class that will hold your setAMSessionVariables method:



Open the newly created Java Class and write your new method.



Note: When using session variables, you should test your application for activation and passivation because the userData HashMap is not persisted by default when activation / passivation occurs for application modules that have Application Module pooling enabled, which means that custom session data may be lost between requests. What we need to do is to passivate the session user data together with the other state data stored by the framework and load it back when the AM is requested the next time (when it gets activated again). To do this we have to overwrite two methods in the ApplicationModuleImpl class.

```
@Override
protected void activateState(Element aElement) {
    super.activateState(aElement);
    mLogger.info("+++++++ activateState");
    Hashtable lData = getSession().getUserData();
    if (aElement != null) {
        // 1. Search the element for any <PrivData> elements
        NodeList nl =
    aElement.getElementsByTagName(PRIVATEDATA);
    }
}
```

```
if (nl != null) {
            // 2. If any found, loop over the nodes found
            for (int i = 0, length = nl.getLength(); i <</pre>
length; i++) {
                // 3. Get first child node of the <PrivData>
element.
                Node child = nl.item(i).getFirstChild();
                if (child != null) {
                    // 4. Set the data value to the user data
hashmap
                    String lDataString =
child.getNodeValue();
                    String[] lSplitkeyval =
lDataString.split(";");
                    for (int ii = 0; ii <
lSplitkeyval.length; ii++) {
                        mLogger.fine("..."+lSplitkeyval[ii]);
                        String[] lSplit =
lSplitkeyval[ii].split("=");
                        lData.put(lSplit[0], lSplit[1]);
                    }
                    break;
                }
            }
        }
    }
}
ROverride
protected void passivateState(Document aDocument, Element
aElement) {
    super.passivateState(aDocument, aElement);
    mLogger.info("----- passivateState");
    // 1. Retrieve the value of the user data to save and
build a string representation
    Session lSession = getSession();
    Hashtable lData = lSession.getUserData();
    String lDataString = "";
    Set<String> keyset = lData.keySet();
    if (!keyset.isEmpty()) {
        Iterator<String> keys = keyset.iterator();
        while (keys.hasNext()) {
           String key = keys.next();
            mLogger.fine("..."+key + "=" + lData.get(key));
            IDataString += key + "=" + IData.get(key) + ";";
        }
    }
    // 2. Create an XML element to contain the value
    Node node = aDocument.createElement(PRIVATEDATA);
    // 3. Create an XML text node to represent the value
    Node cNode = aDocument.createTextNode(lDataString);
    // 4. Append the text node as a child of the element
    node.appendChild(cNode);
    // 5. Append the element to the parent element passed in
    aElement.appendChild(node);
}
```

OTN Harvest - April 2012 <u>http://www.oracle.com/technetwork/developer-</u> tools/adf/learnmore/april2012-otn-harvest-1609383.pdf Add the new method to the Client Interface.



Refresh your Data Control Panel and you will see that the method is now exposed and can be consumed by the Method Call Activity in the View Layer.



The last set is to bind the exposed AM method or "setAMSessionVariables" to your Method Call Activity or InitializeSalesOrderMethod.

Oracle JDeveloper 11g Release 1 - FormsVsA		
ile Edit View Application Refactor :	Search Navigate Build Bun Versigning Iools Window Help	
🔁 🖬 🗐 🤎 🗮 🗋 🛍 🔘 •	• O - 1 急 - 1 急 益 道 無 • 1 ▶ • 尊 • 〓•	
Application Navigator	OrderSales/PageFlowScopeBean.ava * Mr/rujectAcpModule.xml * CoderSales of Xxwl * Mr/ProjectAcpModuleImpl.ava *	
Forms//sAdf1	- C + 4 4 100% - 900- 1 4 1 - 0 0	3 0 7
		Dig w
	Su 02: 3 - 02 - Bounded Task Pow	
tModel cation Sources		
cation sources con.xx.myproject.model		
com, xx.myproject.model.am		
MyProject/apModule		
MyProjectAppModule.xml		17
MyProjectAppModuleClient.java		
MyProjectAppModule.java		
MyProjectAppModuletmpl.java		
com.xx.myproject.model.assoc		
am.xx.myproject.model.eo		
am.xx.myproject.model.vo		
£4		
cation Sources	SalesOrder E	
com, xx.myproject, view	suescieer >	
com.xx.myproject.view.bean com.xx.myproject.view.pageDefs		
com, xx, myproject, view, pagebiens PageFragements	InitializeSaleeOrderMethod SalesOrderPage	
TaskFlows		
OrderSales bif OrderSales bif InitializeSale	ter Christe Manifester Christelle and Christelle Christ	
META-DIF		
Content		
PageFragements	I Drog and Drop Mathod	
WEB-DVF	Drag and Drop Method	
WEB-DIFT/TaskFlows		
Page Flows		
Taskflows	into the Icon.	
adfc-config	into the reon.	
DrderSalesTestPage.jspx		
Application Resources		
* Data Controls		
MyProjectAppModuleDataControl	Degran Overvei Source History <	
E E oeOrderHeadersAl		
Operations	Running: IntegratedWebLogi:Server - Log X	
E TetAMSession//ariables()	(a) (b) (c) (c) (c) (c) (c) (c) (c) (c) (c) (c	Actions •
	[01:09:02 FMI Application Redeployed Successfully.	

Note: Once you drag and drop the exposed method onto the Method Call Activity, the ADF framework will automatically create the binding for you.



Once you initialize your ADF application, the next step is to navigate to your first page. In this example, I only have one method and one UI page so this is a very simplistic way to demonstrate the initialization and navigation of an ADF application.



In a real life ADF application, the task flow design can quite complex.



Application Parameters

In the previous section or Design Application Flow, I mentioned that ADF applications can accept parameters. Parameters are defined as part of a task flow. In this example, I created one required parameter that accepts the warehouse Id. The parameter value can be retrieved by getting the value from the pageflowscope hash map object as demonstrated in the previous session.

	si <u>o</u> ning <u>T</u> ools <u>W</u> indov > • ∰ • <mark>■</mark> •	v <u>H</u> elp				
OrderSales-btf.						
						242 ?
General Description Activities	🕼 Input Parameter D	efinitions				+ ×
Control Flows	Name * 🔺	Class	Value	Requi	red	
Managed Beans	pWarehouse	java.lang.String	#{pageFlowSco	pe.pWarehouse}	✓	11.
Parameters						
Behavior	😡 Return Value Defin	itions				+ ×
	Name * 🔺	Class		Value *		
						11.

In Oracle Form, the application parameter is defined under the application and the parameter value can be referenced by prefixing the variable with the key word "parameter" or :PARAMETER.PWAREHOUSE

Cracle Forms Builder - C:\FORMVSADF.fmb File Edit Property Program Tools Window Help		
	• • ×	
V3 E Attached Uharates E Data Blocks E Canvases E Canvases E Edworks E Domenters F Proprietors F Proprietors	<mark>第 Property Palette</mark> 電動 急強 国际 ①	X Find
PI HELP_TARGET PI APPCORE_ZOOM_VALUE PI APPCORE_ALT_REGIONS	Parameter: PWAREHOUSE	PWAREHOUSE
	Subclass Information Comments	TEXT_ITEM
	= Data	
🙂 🗄 Popup Menus	Parameter Data Type	Number
🖶 🕀 Program Units	 Maximum Length 	30
Property Classes B Record Groups	 Parameter Initial Value 	
Cector anotype Cector anotype Fisual Attributes By Windows PArenus PArenus PArenus		•
Doject Libraries Built-in Packages Database Objects	Source object and source module inform	ation for referenced object.

Designing Your UI Application

Translating most of the commonly used Oracle Forms UI components, such as modal and non modal windows, canvas, stacked canvases, frames, input text items, labels, buttons, tabs, current record indicator and multi-row block "Table" is straight forward. In Oracle Forms, we have a very limited set of components as compared to ADF, so the challenge is trying to understand all the new components and features available with ADF. In order to help the Developer understand each component, Oracle provides a tool called "ADF Faces Rich Client Demos", or <u>http://jdevadf.oracle.com/adf-richclient-demo/faces/index.jspx</u>.

In addition to hundreds of new UI components, ADF introduces the concept of reusable pages or page fragments. Page fragments allow you to create parts of a page and a page can be made up of one or more page fragments. When building your application, determine if you have any common tasks that could take advantage of page fragments. Another reason to use page fragments is to break down very large and complex pages so it is easier to maintain.

In order to transition your current understanding of Oracle Form components, build a simple master-detail project and practice using these ADF components and discovering any new features by reviewing "ADF Faces Rich Client Demos".

For additional reference, review an Oracle video that compares the differences between Oracle Forms and ADF. This video is called "Redeveloping Oracle Forms in ADF" ADF Insider Essentials.

Redeveloping Oracle Forms in ADF <u>http://www.youtube.com/watch?v=TiqbW1CAMMc</u>

Common Layout Components

ADF and Oracle Forms are similar in that they provide a declarative way of adding components to pages. With Oracle Forms, the Developer will declaratively add field components as well as any layout features to the page using the property pallet and or canvas layout components. The Developer is also responsible for the exact position of each item on the page.

Item Property Pallet		Canvas			
= Physical		G FORMVSADF: FORM_CANVAS (OE_ORDER_HEADERS)			
♥ Visible	Yes	🗅 😂 🖬 👒 鱶 🖗 🔭 🕅 🛍 🧶 🞯 🛣 🖉 Canvas: FORM_CANVAS 💽 Block: OE_ORDER_HEADERS			
Canvas	FORM_CANVAS	TrAnal ▼ 10 ▼ B Z U (金) 등 ‡ 등 ∰ ⊕ 山 (巻 📥) =			
Tab Page	<null></null>	× m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1			
X Position	1.396				
Y Position	.365	Order Number: ORDER_NUMBER			
Width	2				
♥ Height	.25				
➡ Bevel	Lowered				
➡ Rendered	Yes				
Show Vertical Scroll Bar	Yes				
		o 1650			

Oracle Forms

🐻 Property Palette		-		
[명희] 12 22 [코 14] 0. 🖉 Find: 🔤				
Canvas: FORM_CANVAS				
= General				
Name	FORM_CANVAS			
Ganvas Type	* Content			
Subclass Information	CANVAS			
 Comments 				
 Help Book Topic 				

One of major differences between Oracle Forms and ADF is the Developers ability to control the exact location of the components on a page. Oracle Forms requires the Developer to specify the exact location of the component on the page; whereas, ADF can only control the relative position of a component on the page. The exact position of the components will be determined when the page is rendered to the User.



Comparison of the property palette components between Oracle Forms and ADF:



ADF – Sample of components

Another major difference between Oracle Forms and ADF is that ADF has layout features that surround the components on the page. This is very similar to Oracle Reports whereby we have frames and anchors to control the flow of the page. The ADF layout features control the flow of the page as well as how the components are displayed inside the page. As part of these new layout features, there are two new concepts to understand.

One new concept is called a facet. A facet is a layout component that contains other layout components and it is responsible for maintaining the relative positioning of all of its contents. For example, a panelBorderLayout contains a top, bottom, start, end and center facet and any items placed into each area or facet will retain their relative position on the page; therefore, a company logo will be placed in the top facet and any copyrights info will be placed in the bottom facet. This will ensure the "picture frame" will remain consistent regardless of how the center of the page is displayed.

The second new concept is called geometry management and component stretching. Geometry management is the process by which the parent components, and child components negotiate the actual sizes and locations of the components in an application or page. At the core of geometry management, solution is a resize notification mechanism that allows components that support geometry management to be notified of browser resize activity. In other words, geometry management will make sure the application is displayed correctly based on the size of the screen.

In this next section, I will explain some of the common layout components and their purpose.

• af:panelForm – Mostly used for input fields on a page. The layout will automatically right justify the prompts and left justify the fields.

Order Number	#{OrderNumber.inputValue}	PaymentTypeCode	#{PaymentTypeCode.inputValue}
OrderedDate	#{OrderedDate.inputValue}	SalesDocumentName	SalesDocumentName.inputValue}

Note: Some UI design call for left justification of labels and this can be accomplished by modifying the Cascading Style Sheet (CSS), which is called a Skinning in ADF.

- af:panelHeaderLayout This component provides a title for the group of items contained within the layout. In Oracle Forms, this would be comparable to a Frame.
- af:panelGroupLayout The layout allows you to layout components in a row or column. For simple ADF pages without panelsplitter or panelStretchLayouts, the

Component Palette × Resource Palette ADF Faces 8 ADF Faces Common Components ▽ Layout Decorative Box Document Inline Frame I Navigation Pane Panel Accordion Panel Border Layout Panel Box Panel Collection 😁 Panel Dashboard B Panel Form Layout Panel Group Layout Panel Header ≣ Panel List Panel Splitter 🛃 Panel Stretch Layout Panel Tabbed 📴 Panel Window — Separator Chow Detail Header 📖 Spacer I Toolbox

outermost layout will be a panelGroupLayout with a setting of vertical which will cause all remaining components/layouts to flow vertically on the page.

0	Vertical
	Field1
	Pulldown1
	Button1
0	Horizontal
	Field 1 Pulldown 1 Button 1

- Note: You can nest another layout within the outermost layout and that nested layout's components will retain its individual layout properties such as horizontal positioning but the layout itself or container will be arranged vertically within the page.
- af:panelBorderLayout The components dropped into each facet will maintain their relative position at all times. This layout is used when you need a header section for a logo and information in footer section of the page.

	mental Ven-		toulbar	🛃 Detach	[7]	
	Lineld	LineNumber	OrderedEtem			
	#[Lineid.inputVi	#(LineNumber.k	#(Ordereditem)			
	#[Linet5.inputVi	#[.LiteNumber in	#L.Ordensdtam.(
	#C. Linekt inputV-	#(.LineNumber 3	#(. Orderedten i			
art						red

- af:panelStretchLayout Expands contained components to fill the width of the page. This is mostly used if another layout feature cuts off the display of a layout component.
- af:panelSplitter –This component will divide an area into two horizontal or vertical panels. The Developer also has the option to allow the User to adjust the size of each panel. I use this feature with a nested accordion layout to provide a quick reference menu or links on the left hand side of the page.
- af:panelAccordian Looks and acts like several email software layouts.

	leader 2	~~~~~~~~~	necnecnec
abel 1.			

For Oracle Forms Developers, the ability of not controlling the exact position of a component will be frustrating until you learn how to control the components position by using the relative positioning layout options. In addition to Developers, the Application Architects who design these new pages should have an understanding of effort level required to build pages using relative positioning versus the ability to specify the exact location of a component on a page. For example, a page layout developed in ADF might take twice as long as Oracle Forms because the Developer has to find a way to nest layout components to achieve the desired page layout. I have provided a couple of references for designing the perfect layout.

Layout Best Practices	http://www.oracle.com/technetwork/developer-
	tools/adf/layoutbestpractices-084287.html
Achieving The Perfect Layout with	http://www.quovera.com/whitepapers/downloads/ko
ADF Faces RC	letzke_adf_faces_layout_paper.pdf

As part of mastering the layout components, its also important to understand that each layout component has stretching properies. The stretching properties are important because most ADF pages will have nested layout components so the nested layout componets must be compatiable to achive the correct layout. For example, a stretchable child component needs to be nested inside a stretching parent object otherwise the child component might not render properly on all browsers. Therefore, the correct combination of layout components is necessary to developing desired layout.

For additonal reference, I have provided a URL which contains the geometry properties for each component. For exmple, the panelGroupLayout geometry can be found using ADF Demo.

panelGroupLayout geometry properties	http://jdevadf.oracle.com/adf-richclient- demo/faces/components/panelGroupLayout.jspx
Table 8-1 ADF Faces Layout	http://docs.oracle.com/cd/E16764_01/web.1111/b31
Components	973/af_orgpage.htm

Deployment

Neither ADF nor Oracle Forms create a stand-alone executable; therefore, the compiled source code needs to run on some type of server that can interpret the compile code. In

the case of Oracle Forms, we typically use the Oracle's Enterprise Business Suite technology stack to run our compiled application. As for ADF applications, we need a Java EE-compliant server, such as a WebLogic Server (WLS) to run our compiled ADF application.

Deploying your application to a Java EE server requires two steps:

- 1. Prepare the deployment file. This step consists of creating a deployment file, which consists of collecting all the supporting libraries into a Java archive (JAR) file.
- 2. Copy the deployment file to the application server. In this step, you can use either JDeveloper to install your application on the WLS server or you can upload the JAR file to the WLS server and then use a WLS deployment tool in the WLS console to install your application.

Here are a few deployment references:

Deploying Applications To	http://www.quovera.com/whitepapers/downloads
WebLogic Server Using JDeveloper	/rmoug_2012_deployment_doc.pdf
and WLS Console	
FAQ for Integration of Oracle E-	https://metalink.oracle.com/metalink/plsql/show
Business Suite and Oracle	doc?db=NOT&id=1296491.1
Application Development	
Framework (ADF) Applications	

Conclusion

In conclusion, transitioning from Oracle Forms to ADF will be overwhelming, time consuming, always challenging, sometimes frustrating; however, the most important thing to remember is that it is possible. This paper was designed to provide a high level how-to guide for Developers transferring their skill sets from Oracle Forms to ADF pages. There are multiple references in the paper that should be reviewed to gain a better understanding of the basic concepts of ADF. Your true understanding of ADF development will be achieved when you start building your first couple of applications. Lastly, it is important to realize that it takes time to understand ADF, so be patient and take that first step in to the world of ADF.