A Developers Journey From Oracle EBS Forms To Oracle EBS ADF Pages

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 Collaborate 2013 Denver, Colorado

## Abstract

This presentation will explain my journey from an Oracle EBS Forms Developer to an Oracle ADF Pages Developer. My journey starts as an experienced Oracle Forms Developer that is starting a new Oracle ADF application with no knowledge of Oracle ADF "skill set" and ends with supporting an ADF application. Prior to starting this new Oracle ADF application, I had attended some formal Java education training classes, but that was ten years ago so I only remember concepts.

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 finally deploying that application in a WebLogic /Oracle EBS environment.

## 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. How to deploy an application in a Weblogic / Oracle EBS environment?
- 6. 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 finally 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 frame work of Model, View, Controller. The ADF framework separate 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 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 understand 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 & enterprise manager screens.

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

The most difficult and frustrating part of my journey was 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

## A Developers Journey from Oracle EBS Forms to Oracle EBS ADF Pages

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 validation routines, setting a database sequence on a table, used to build error messages, referencing built-in calls like setting current date & 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	

#### 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?







## 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 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 & 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.

<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 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-
	demo/faces/index.jspx

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<br/>Oracle White Paper April 2011)<a href="http://www.oracle.com/technetwork/developer-tools/jdev/adf-task-flow-design-132904.pdf">http://www.oracle.com/technetwork/developer-</a><br/>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.

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

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.

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.			
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.

## 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 1.4http://docs.oracle.com/javaee/1.4/tutorial/doc/JSPIntro7.htmlTutorial

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		

	Note: Add custom code to set a value after it is retrieved from the database and before it's 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			
	<ul><li>Set Where By Clause</li><li>Set Order By Clause</li></ul>			
	For Form Developers, this is where your block triggers 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			
	<ul><li>Get Attribute</li><li>Set Attribute</li></ul>			
	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.			
	<ul> <li>Initialize code for your session</li> <li>Call methods defined in the application module</li> <li>Conditionally Rendering</li> <li>Set variables</li> </ul>			
	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).

👔 🍓 Search 🛛 🕅 Rui	n/Debug/Profile		
⊕ - Project Source Paths         ∪ Us           ⊕ - ADF Model         ⊙ Us           ⊕ ADF View         ⊕ Ant           ⊕ - Ant         ₽ Business Components	e <u>C</u> ustom Settings e Project Settings Configurations: sult		Customize Settings
Compiler     Dependencies     Dependencies     Dependencies     Dependencies     Dependencie     Devendencie     Devendencie     Devendencie			New Delete Restore Defaults
Java EE Application J3P Tag Libraries J3P Tag Libraries Libraries and Classpath Resource Bundle Run Drobbug Profile Technology Scope	Edit Run Configuration "Defaul     Edit Run Configuration "Defaul     Edit Run Configuration "Defaul     Pl/SQL     Pl/SQL     Pl/SQL     Pl/SQL     Pl/SQL     Pofiler     Profiler     Profiler     Profiler     Peucopy     Remote     Pobugger     Remote	It*  Launch Settings Default Run Target:  Attempt to Run Active File Before Default  Virtual Machine: Java Options:  Cent Program Arguments:  Bun Directory:  Rgmote Debugging	Browse
Hep           bb 21, 2013 4:53:49 PM MST> <errc< td="">           bb 21, 2013 4:53:49 PM MST&gt; <errc< td=""></errc<></errc<></errc<></errc<></errc<></errc<></errc<></errc<></errc<></errc<></errc<></errc<></errc<></errc<></errc<></errc<></errc<></errc<></errc<>	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

_	-	_	
I	T	I	
-	~	-	-

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

## 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

Name	
🔄 jdevstudio11122inst	all.exe

Click On jdevstudio11122install.exe

Oracle Installer		
	ORACLE <sup>.</sup>	
	Preparing the installer	
	Cancel	

🕢 Oracle Installer - Oracle JDeveloper 11g (11.1.2.2.0)		
Welcome This installer will guide you through the insta Oracle JDeveloper 11g Release 2 (11.1.2.2.	ilation of ORACLE"	
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 Next	

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

-		
S Oracle Installer - Oracle JDeveloper 11g (11.1.2.2.0)		
Choose Middleware Home Directory Specify the Middleware Home where you wish to install Orade Products.	ORACLE	
Middleware Home Type     O Use an existing Middleware Home     O Create a new Middleware Home		
Middleware Home Directory		
Dk\JEV_11_1_1_6		
Browse Reset		
Egit Previous Next		
🗟 Oracle Installer - Oracle JDeveloper 11g (11.1.2.2.0)		
Choose Install Type Select the type of installation you wish to perform.		
Typical     Install the following product(s) and component(s):         Developer and ADF         WebLogic Server		
Custom     Choose software products and components to install and perform optional     configuration.		

Exit

Previous Next

## A Developers Journey from Oracle EBS Forms to Oracle EBS ADF Pages

👧 Oracle Installer - Oracle JDeveloper 11g (11	1.1.2.2.0)	
Choose Product Installation Directo Provide the directories where you wish to install Oracle Products.	ories	ORACLE
		💍 Discard Changes
Middleware Home Directory		
D:\JEV_11_1_1_6		
Product Installation Directories		
The Product Home might contain shared utilitie	s and any products or components for w	which unique directories
JDeveloper and ADF:		
D:\JEV_11_1_6\jdeveloper		
WebLogic Server: D:\JEV 11 1 1 6\w/server 10.3		
Exit		Previous <u>N</u> ext
🔄 Oracle Installer - Oracle IDeveloper 11g (11	11220)	
Choose Shortcut Location The installer creates shortcuts to Oracle componer user with administrative privileges, you can specify	nts, samples, and tools. As a v where these shortcuts are created.	ORACLE
Select the Start Menu folder in which you want to create Oracle shortcuts:		
I Users" Start Me	enu folder (recommended)	
For some installations, this setting may limit the automatic creation of server shortcuts for users without administrative privileges. Refer to the documentation for more information.		
O <b>Local user's Start Menu folder</b> Select this option if you need to ensure that other profiles registered on this machine will not have access to these shortcuts.		
E⊻it		Previous Next
Oracle Installer - Oracle JDeveloper 11g (1)	1.1.2.2.0)	
Installation Summary The following Products and JDKs will be installed.		ORACLE
Developer and ADF	Description	
JDeveloper Studio	Oracle JDeveloper and ADF is a compl	ete IDE for
Application Development Framework Ru     WebLogic Server	Service-Oriented Architecture (SOA) a is ranked best among major Java ven	and Java development that dors in Forrester
Core Application Server	"hot-pluggable" with Oracle and non-O	n Middleware, JDeveloper is Dracle environments,
Administration Console     Onfiguration Wizard and Upgrade Fram	supporting all major J2EE application s	ervers and databases.
WebLogic SCA		
WebLogic JDBC Drivers		
UDDI and Xquery Support		
JDKs idk160 24	Approximate Installed Size	
	Highlighted item:	990.6 MB
	Total of all selected items:	1,702.3 MB
E <u>x</u> it		Previous Next

## A Developers Journey from Oracle EBS Forms to Oracle EBS ADF Pages

🛃 Oracle Installer - Oracle JDeveloper 11g	(11.1.2.2.0)
Complete Develop	pment Environment
<ul> <li>Java SE</li> <li>Java EE</li> <li>Web Development</li> <li>Web Services</li> <li>Database</li> <li>XML</li> </ul>	Oracle JDeveloper
Installingorade.jdeveloper.json.jar Egit	ORACLE' 15% Previous Mext
oracle Installer - Oracle JDeveloper 11	1g (11.1.2.2.0)
Installation Complete Click the Done button to exit the installer.	ORACLE
ORACLE	Message Congratulations! Installation is complete.
3	
	V kun Quickstart
Exit	Previous

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, I would 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
Ctrl + -	Go To Class
Ctrl + Alt + -	Go To File
Alt + Home	Locate file in app navigator
App Navigator	Find as you type (Comparable to Oracle Forms)
Ctrl + F	Find
Ctrl + Shift + F	Find In Files
Ctrl + Alt + H	Highlight Fields

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	
<b>*</b> -	
🕞 FormsVsAdf Overview 🗴 👪 FileLibraryUtils.java 🗴 🖌	
(🔞 - Find 🕹 🖓 🖄 🌽 🛤 🛟	54 🗟 🔩 I 🕽 🐘 🕪 🍋 I 🗃 🖼 -

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.

Select roles	ect Role	×
Role:		
0	Default Role	•
1	Enables all technologies.	
0	BPM Process Analyst	
	Role for BPM Business Analyst.	-
0	Customization Developer	
	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.	-
	ways prompt for role selection on startup	
	OK Cance	•

## Create a new application:

💩 Create Fusion Web Appl	ication (ADF) - Step 1 of 5
Name your applicatio	
	Application Name:
Application Name	FormsVsAdf
Project 1 Name	Directory:
<ul> <li>Project 1 Java Settings</li> </ul>	C:\JDeveloper\1Test\FormsVsAdf Browse
Project 2 Name	Application Package Prefix:
<ul> <li>Project 2 Java Settings</li> </ul>	com.xx_myproject
	Application Template:
	BPM Application     Creates a BPM application. The application consists of one BPM project. This     project has also SOA technology
	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 Business 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     JavaReams technologies.     ✓
Help	< Back Next > Einish Cancel

Α	Develop	pers J	Journey	from	Oracle	EBS	Forms to	Oracle	EBS	ADF	Pages
	201010		· · · · · · · · · · · · · · · · · · ·		010010	~	1 01110 00	010010			

Name your project	
Application Name Project 1 Name	Broject Name:         MyProjectModel           Directory:         C:\Developer\lTest\Forms\sAdf\MyProjectModel         Browse
Project 1 Java Settings Project 2 Name Project 2 Java Settings	Project Technologies     Generated Components     Associated Ubraries       Available:     Selected:       ADF Proces     ADF Proces       ADF Node Proveer     ADF Proge Frow       ADF Proge Frow     Image: Component Selected:       ADF Swing     Image: Component Selected:       BPM     Image: Component Selected:       Internet Selected:     Image: Component Selected:
Help	ADF Business Components is the business services API provided by the Orade Application betweetoment Framework (Orade ADF). ADF Business Components governs interaction between the rest of the application and the data stored in the < <u>Back</u>

💩 Create Fusion Web Appl	ication (ADF) - Step 3 of 5	×
Configure Java settin	gs	
Application Name     Project 1 Name     Project 1 Java Settir     Project 2 Name	Your new project starts with a default package, a source root director directory. Default Package: com xx-mysroject.model have Source Dath:	ry, and an output
Ü Project 2 Java Settings	C: UDeveloper (ITest Forms VsAdf/WyProjectModel arc Qutput Directory: C: UDeveloper (ITest Forms VsAdf/WyProjectModel dasses	Broyse Browse
<	< Back Next >	Einish Cancel

🚖 Create Fusion Web Appl	ication (ADF) - Step 4 of 5
Name your project	
Application Name	Project Name: MyProjectUI Directory: C:\Developer\ITest\FormsVsAdfMyProjectUI Browse
Project 1 Java Settings	Project Technologies Generated Components Associated Libraries
Project 2 Java Settings	Avalance: ADF Backs ADF Desktop Integration ADF Desktop Integration ADF Desktop Integration ADF Desktop Integration ADF Page Flow HTML Bava Database (Offine) ESP and Servlets Mult Database (Offine) ESP and Servlets Mult ADF Paces features induct: file upload support, data balances, ADF Faces features induct: file upload support, direct side validation, partial rendering of a page (AIX-AVE), data tables,
Help	< Back Next > Enish Cancel

## A Developers Journey from Oracle EBS Forms to Oracle EBS ADF Pages

💩 Create Fusion Web Appl	ication (ADF) - Step 5 of 5	×
Configure Java settin	gs	F
Application Name Project 1 Name Project 1 Java Settings	Your new project starts with a default package, a source root directory, and directory.	an output
Project 2 Name Project 2 Java Settir	Java Source Path: C: UDeveloper (ITest\FormsVsAdf\MyProjectUI[src 0.thm): Developer.	Browse
	C:UDeveloper\1Test\FormsVsAdf\MyProjectUT\classes	Browge
<	< Back Mext > Einist	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

	a series and a series a series	served a served as served as	
Computer ► OS (C:) ► JDeveloper ► 1Ter	st ▶ FormsVsAdf ▶		
File Edit View Tools Help			
Organize  Include in library  Share with  B	urn New folder		
> like dev	A Name	Date modified	Type
Documents and Settings		butembuiled	type
Drivers	🎍 .adf	2/23/2013 7:50 AM	File folder
FileZilla 3.5.3 win32	🕌 MyProjectModel	2/23/2013 7:50 AM	File folder
EileZilla-2.5.2	🍰 MyProjectUI	2/23/2013 7:50 AM	File folder
Identity	🗼 src	2/23/2013 7:50 AM	File folder
Intel	FormsVsAdf.jws	2/23/2013 7:50 AM	JWS File
> inter			
Jove_class			
> Joeveroper			
b and the d			
• (), Trest			
A B CommeN/cA 46			
• Gr FormsvsAdr	=======================================		
P add			
a 👔 MyProjectiviodel			
P III Classes			
4 🏭 MyProjectUI			
classes			
public_html			
4 🕌 src			
META-INF			

EFormsVsAdf Overview	x	
Checklist Java Files Page Flows Neb Pages	Fusion Web Application Quick Start Checklist Create the application by following step-by-step instructions describing how to build Fusion Web Applications according best practice recommendations.	to Orade
lusiness Components Iinding Files	1 > The Plan Your Application	Started
Offline Databases	2 > Connect to a Database	Started
	3 Duld Business Services	Started
	4   Compared to the second sec	Started
	5 > Design Pages	Started
	6 F 🙀 Add Common Components (Lookups, Search and Menus)	Started
	7      Implement Business Logic	Started
	8 > Error Secure Your Application	Started
	9      Internationalize Your Application	Started
	10 > Debug and Test Your Application	Started
	11  Package and Deploy Your Application Not	Started
	12 > SOA-Enable Your Application	Started

Quick Start Guide

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 concern, with a database connection to a single database and schema.

## A Developers Journey from Oracle EBS Forms to Oracle EBS ADF Pages

JDeveloper	Oracle Forms Developer
Create Database Connection 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. Create Connection In:  Application Resources Cognection Name: FestDatabase Connection Bassword: FestDatabase Connection SuccessI Fest Connection SuccessI Felp Cot Cancel Cot Cot Cancel Cot Cancel Cot Cancel Cot	Connect       Image: Conne

## **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.

This list is filtered ac		10logies
This list is nitered ac		
	cording to the curren	t project's <u>selected technologies</u> .
Gro Search		
Categories:		Items: Show All Descriptions
		Accoriation ^
Applications	i	
Connections	3	Business Components Deployment Profiles
Deployment	Descriptors	Business Components Diagram
Diagrams	Profiles	Confer M Date Madel Commences
External Ap	olications	Derault Data Houer Components
Java		and Domain
Projects		Chity Business Logic Unit
BPM Tier		E transford
DE Rusiness Tier	a Componente	Launches the Create Entity Object wizard, which allows you to create an
Data Contro	s components	entity object. Use entity objects to represent tables or UML entities and to
Security		implement business rules.
SOA Tier		To enable this option, you must select a project in the Application
All Items		Navigator. Before you can finish creating the new entity object, you will be
		prompted to select (or create) a database connection.
Help		OK Cancel
	ISH DUBURUS	~
Initialize Business (	Components Proj	ect 🔁
is project has not yet	t been initialized for	r Business Components. After specifying the following information for your
siness Components P	roject (jpx file), yo	ou will be prompted to create your business Component(s).
ecify the database co	onnection that lets	you create Business Components from existing database objects.
nnection: TestDat	abase	- 🔶 🖌 🤹
User Name:	apps	
Driver:	oracle.jdbc.Oracle	Driver
Connect Strings	idhcuaraclauthinu@	//acaobadh a panduit com: 1521/ERCA
connect string.	Japanon aciana in tra	//or debable a pandale com. 1551/EbbA
	flavor and type m	ap that fits your application.
oose the proper SQL		
oose the proper SQL		-
oose the proper SQL		•
pe Map: Oracle		•
oose the proper SQL L Flavor: Oracle pe Map: Oracle		•
oose the proper SQL QL Flavor: Oracle pe Map: Oracle		•
oose the proper SQL 9L Flavor: Orade pe Map: Orade		
oose the proper SQL L Flavor: Oracle pe Map: Oracle Help		
oose the proper SQL PL Flavor: Orade pe Map: Orade		- OK Cancel
L Flavor: Orade pe Map: Orade Help		Cancel
Lefavor: Orade pe Map: Orade Help		CK Cancel
Dese the proper SQL QL Flavor: Orade pe Map: Orade Help Create Entity Object - Sb	ep 1 of 6	CK Cancel
UL Flavor: Oracle pe Map: Oracle Help	ep 1 of 6	OK Cancel
oose the proper SQL L. Flavor: Orade pe Map: Orade Help	ep 1 of 6	OK Cancel
oose the proper SQL L Flavor: Oracle pe Map: Oracle Help create Entity Object - Ste me	sp 1 of 6	CK Cancel
oose the proper SQL I. Flavor: Orade pe Map: Orade Help	ep 1 of 6 Entity objects are use	CK Cancel      Cancel      CK      Cancel      C      Concel      C      Concel      C
oose the proper SQL L Flavor: Oracle pe Map: Oracle ireate Entity Object - So me Name	ep 1 of 6 Entity objects are use	CK Cancel
oose the proper SQL L Flavor: Oracle pe Map: Oracle Help create Entity Object - Ste me Name Attributes	ep 1 of 6 Entity objects are use Package: cor	
oose the proper SQL L Flavor: Oracle De Map: Oracle Help Ireate Entity Object - St me Name Attributes Attribute Settings	ep 1 of 6 Entity objects are use Package: cor Name: cor	CK Cancel CANCEL CANCEL CANCEL CONTRACTORS CONTRACTOR
oose the proper SQL L Flavor: Oracle pe Map: Oracle Cracle Entity Object - St me Name Attribute Settings Java	ep 1 of 6 Entity objects are use Package: co Nege: co Evidence Entity	CK Cancel CK Cancel CK Cancel CK Cancel CC CANCEL CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
oose the proper SQL L Flavor: Oracle Drace Per Map: Oracle Help Create Entity Object - St me Name Attribute Settings Java Generate	ep 1 of 6 Entity objects are use Package: co Name: E Egtends Entity:	OK Cancel  OK Cancel  OK  Cancel  Comparison  Compari
Coase the proper SQL L Flavor: Oracle Drace Drace Drace Drace Lelp Craste Entity Object - Sta Craste Status Status Craste Status St	ep 1 of 6 Entity objects are use Package: cor Narge: Estends Entity: Estends Entity: Estends Entity:	CK Cancel CK CA
oose the proper SQL I, Flavor: Orade De Map: Orade Eleip Greate Entity Object - St me Name Attrobutes Attrobutes Attrobutes Java Generate Sumary	ep 1 of 6 Entity objects are use Package: cor Name: S Estends Entity: Property Set: d Data Source	CK Cancel
oose the proper SQL L Flavor: Oracle Drace Elelp Crace Help Crace Name Name Attrobute Settropute Settro	ep 1 of 6 Entity objects are use Package:	CK Cancel CK Cancel CK Cancel CC CANCE Cancel CCC CCCC CCCC CCCCCCCCCCCCCCCCCCCCCC
oose the proper SQL L Flavor: Oracle De Map: Oracle Ltelp reate Entity Object - So me Name Attributes Attributes Attributes Attributes Attributes Summary	ep 1 of 6 Entity objects are use Package: or Nage: Egtends Entity: Data Source Decide whether to b	CK Cancel CK Cancel CK Cancel CK Cancel CC CANCEL CCC CANCEL CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
oose the proper SQL L Flavor: Oracle Drace PMap: Oracle Lelp Create Entity Object - Sb me Name Attribute Settings Java Constrate Summary	ep 1 of 6 Entity objects are use Package: Egtends Entity: Property Set: Data Source Decke whether to 1 Decke whether to 1 0 Database Scherr	CK Cancel  CK CK CANCEL  CK C
Lelp Create Entity Object - SD Create Entity	ep 1 of 6 Entity objects are use Package: cor Narge: cor Estends Entity: Property Set: Deta Source Deta Source 0 Dgtabase Schem Database Schem	CK Cancel CK CA
oose the proper SQL IL Flavor: Orade pe Map: Orade ireate Entity Object - St me Name Attribute Summary Summary	ep 1 of 6 Entity objects are use Package:	CK Cancel  CK Cancel  CK Cancel  CK Cancel  CK Cancel  C C C Cancel  C C C C C C C C C C C C C C C C C C C
In the proper SQL, U, Flavor: Oracle pe Map: Oracle Dracle Entity Object - SS me Name Attrobute Settings Java Generate Settings Java Summary Sum	ep 1 of 6 Entity objects are use Package:	CK Cancel CK CA
oose the proper SQL I, Flavor: Oracle De Map: Oracle Elelp Create Entity Object - Sta me Name Attrobutes Attrobutes Attrobutes Sumary	ep 1 of 6 Entity objects are use Package: or Egtends Entity: Data Source Data Source Database Schem Schema Object:	CK Cancel
oose the proper SQL L Flavor: Oracle Drace PMap: Oracle Etelp Create Entity Object - St me Name Attribute Settings Java Constrate Summary	ep 1 of 6 Entity objects are use Package: Extends Entity: Broperty Set: Data Source: Decke whether to 1 0: Database Schem Schema Object:	CK Cancel
oose the proper SQL I, Flavor: Oracle pe Map: Oracle reate Entity Object - St me Name Attrobutes Attrobutes Java Generate Summary	ep 1 of 6 Entity objects are use Package: cor Name: cor Estends Entity: Property Set: Deta Source Deta Source Deta Source Deta Source Deta Source Deta Source Schema Object:	CK Cancel

## A Developers Journey from Oracle EBS Forms to Oracle EBS ADF Pages

🔆 Create Entity Object - :	Step 2 of 6
Attributes	
Mame_	Click New to create new attributes. Click New from Table to add attributes for unmapped columns.
Attributes	Entity Attributes:
Adva     Summary	Headerid     Image: Comparison of the second s
Help	<pre></pre>

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

tribute Settings					
Name	Select Attribute	HeaderId			
Attributes	Attribute	HeaderId		Updatable	
Attribute Settings	Turner	Number			
Java	Droperty Set	Number		Always	
Generate	Value Turner	Vivone>		O While New	
Summary	Value:	C giteral C gapression		Edit	
	Persiste     Mandatr	.nt	Primary Key	-Refer After	
	Change	indicator	✓ Queryable	Keiresn Aner	
	Derived	from SQL Expression	Effective Date		
	Precision	n Rule	O Start O End	Update	
	Disgrimin	Nator	History Column	Insert	
	Database Colur	mn			
	Name: HEAD	ER ID	Type: NUMBER		
		-			
Help			< Back Next >	Finish Cancel	
eate Entity Object - S	tep 4 of 6			×	ח
а				010101	
	1		Of the second seco		4
Name	Optionally genera	ate Java classes to add custom beha	vior.		
	Generate Ent	tity Object Class			
\ttributes	-Indude-				
Attributes Attribute Settings	Include −	sors	Create Method		
Attributes Attribute Settings Java	- Include - ✓ Acces □ Data	isors Manipulation Methods	Crea <u>t</u> e Method		
ttributes ttribute Settings ava enerate	☐ Data	isors Manipulation Methods	Create Method		

Generate Entity Definition Class

Help

Classes Extend...

< Back Next > Einish Cancel

A Developers Journey from Oracle EBS Forms to Oracle EBS ADF Pages

Create Entity Object -	step 5 of 6	n le	2
Generate		evelopianereren Anthere	÷
Ó Nama	Select the checkbox to cre	eate a default view object, containing an attribute for each of the entity object attributes.	
Attributes	Generate Default View	w Object	
Attribute Settings	Package: com.xx.myproj	ject.model	ow <u>s</u> e
Java	Ngme: oeOrderHeader	rsAlVVO	
Generate	-		
Summary	Select the checkbox to ad	dd an instance of the view object to the specified application module. If the application module	does
Juliana y	not exist, it will be created	d.	
	Application Module		
	Package: com.xx.myproj	ject.model	o <u>w</u> se.
	Name: AppModule	Bg	owse.
	_		
Help		< Back Next > Einish C	ance
4	a		
Create Entity Object -	inish		~
ummary		01	2
	1		
O Nama	You have finished the	Create Entity Object Wizard	





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).



## A Developers Journey from Oracle EBS Forms to Oracle EBS ADF Pages







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.

Here are the steps to create an association link







ssociation Properties		
- Fame - Entity Obertia - Association Property - Edit Association Durty - Samery	Source Accessor Entity Object: anD derHeaderskil Cologies in: seCreterskil Accessor Name: peOrderHeaderskil DerDetablese Bry Constraints Secondaria	Destrution Accessor Entity Object: aeCriderLivesAll C Bugete in ceOrderLandersAll Accessor Name: peOrderLivesAl
	Ordigen fit Exclusion Causale Orders     Dyslement Causale Orders     Ordersky Exclusion Exclusion     Order State State     Order State     Order Orderskin     Order Orderskin     Order Orderskin     Order Orderskin     Order Orderskin	) (adi Tinp level Contarier





Here are the steps to create a view link:

5	N		u Kon versigen
	Application Navigator		
	Formel/sAdf		
	* Projects	8	M 7.5.
	MyProjectModel     Application Sourn     Or com.xx.mpp     Or com.xx.mpp     Or com.xx.mpp     Or com.xx.mpp     Or com.xx.mpp     Or code     Or	ces roject.model.assoc roject.model.eo HeadersAl LinesAl Chidolal.jax	
	B 🕼 conuxx.mp B 🐏 oeOrde B 🐏 OeOrde B 🔄 MyProjectUI	Mew     Exclude Project Content     Delete	Ctri-N
	* Application Resources	New Entity Object     New Dgmain     New Jassociation     New View Object     New View Unity     New View Unity     New View Inity     New View Inity	
	Connections     Descriptors     Deta Controls	New Business Components from Tables New Default Data Model Components New Business Components Diagram	
		Create Databage Objects Synchronize with Database	
	Recently Opened Files	Make Reguild	Ctrl+Shitt/F9 Alt+Shitt/F9
	Stuchare ×	Refactor Beformat Organize Imports	Chi+Alb-L Chi+Alb-O
		Compare With	

> 0	veate View Link - Step	1 of 7	-		
Na	me				-21
Ģ	Name	Ven Links are t	used for coordinating hierarchies of master -	detal view objects.	
÷	New Objects	Package:	com.xx.myproject.model.vp.link		Brogsen
	View Link Properties	Naget	HeadersToLinkLink		
	EBI Source Query	Display Name:	Headers To Link Link		9
		Egtends			Bouse
	Belo			(ja) (jat) (ja	Cancel





## A Developers Journey from Oracle EBS Forms to Oracle EBS ADF Pages



Edit Destination Que	ry		
	Destination		
	Attributes	Bind Variables	
	oeOrderLinesAll.HEADER_ID	:Bind_HeaderId	
View Link Properties			
Edit Source Query			
Edit Destination Que			
Application Module	Outry Career		
	Where clause for Destination to Source		
	-Bod Headerld - orOrdenseadersal HEA	OFR ID	
	-		

Create View Link - Step	6 of 7	- ×
Application Module		1
V Rank View Clarcts View Link Properties	Select the checkbox to add an instance of this view link to an application module. If the specified application to add and the specified application bookle.	son module does not
Edit Destination Overy		
Application Hodule		
Sumary .		
Refo	< Back Bent >	Enish 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 & 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 our displayed below:

1. Create a read-only view

» Create View Object - ! Name	rp 1 of 9		-31
Kanne     Konne     Konne     Konne     Konne     Arstelander     Arstelander     Arstelander     Arstelander     Anne     Anne     Anne     Anne	tien rägerla an för somen, föreng, menschar, A Pedage: Inn som menschart och var and Neget: Införstansitansion Dader hene: Mit States States i States Begarhet: Begarhet földt: Öffense Steller för delta soven tiper spå verket för delta för Ogen späkaler som för spå verket för delta för Begarhet som första första delta för Begarhet som första delta för delta för Begarhet som första delta för delta för Begarhet som första delta för delta för Begarhet som första delta för Begarhet som första delta för delta för Begarhet som första delta för delta för Begarhet som första delta första första för Begarhet som första för delta första första för Begarhet som första första första första första första första för Begarhet som första första första första första första första första första Begarhet som första första första första första första första första första Begarhet som första första Begarhet som första fö	nd serving used hadress darks for the generic world of a given application lask. ork a bases for the view object. A a garry	Brupe
1340		- chek Berry Da	in Caro

2. Add your SQL statement with a parameter

Qu	ery	
Ţ	Name.	Enter your custom SELECT statement and click Test to check its syntax. Provide the ORDER BY clause separately,
	Bind Variables Attribute Mappings Attributes Attribute Settings Javia Application Module Summary	milet magnetic 1000. mil.gorout.int from mil.gorout.item where organization_id = :pdrpmiletionId
		Carry Classe Date for Lines (Bit- Bits) (Date for Lines (Bits)) Rode State (Date Form) (Bits) 20, Bolds (Date To and

3. Define your parameter

» C	reate View Object - St	ep 3 of 9		-
3in	d Variables			- <u>1</u>
	Name	Variables:	Nd .	
Ļ.	Query.			
¢.	Bind Variables			i.
÷	Attribute Mappings			
	Attributes	New Reno	un l	
	Attribute Settings	( and ) ( and		
	Java	Variable	Custom Properties Control Hints	
	Application Module	Name:	pOrganizationId	
	Summary	Type:	String	· Browse
		Value Type:	Uteral      Expression	Test
		Yalue:	102	Edt
			Updatable	
	htelp.		disk Nexts Disk	
	Clerk		d Rays Dawn by Dawn	

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

Attribute Settings				· • • • • •
L. Name	Seject Attribute:	InventoryStenId		Updatable
Query	Name:	InventoryItemId		
Bind Variables	Type:	Number	• 8	owse
<ul> <li>Attribute Mappings</li> </ul>	Property Set:	<tool></tool>	•	
<u>Attributes</u>	Value Type:	Literal      Expression		Above
Attribute Settings	<u>V</u> alue:			Edit
Java	Mapped	to Golumn or SQL	Key Attribute	Never
Application Module	Selected	l in Query	Quegyable	0.1420
Summary	Disgrimin	ator: View OEntity	Effective Date	
	Bassivat	ie		
	Query Column			
	Alog: (	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.

## A Developers Journey from Oracle EBS Forms to Oracle EBS ADF Pages



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.

	ues
ist of Values Name:	LOV_InventoryItemId
Configuration	II 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:	mtSystemItems 👻 💠
List Attribute:	InventoryItemId
List Return Values	
Map any suppleme attribute for which	ntal values that your list returns to the base view object (it always returns a value to the the list is defined).
View Attribute	List Attribute
InventorvItemId	InventorvitemId
Help	OK Cancel
Help Create List of Va	OK Cance
<u>Help</u> Create List of Va	OK Cancel
Help Create List of Va st of Values Name: Configuration	Cov_InventoryItendd
Help Create List of Va st of Values Name: Configuration	Ues Cancel CV Jivenbryltenid Tribu Teck with List of Yalues
Help Create List of Va st of Values Name: Configuration Default List Type: Display Attributes	Lees
Help Create List of Va st of Values Name: Configuration Default List Type: Display Attributes Select display attri multiple values art	
Help Create List of Va st of Values Name: Configuration Default List Type: Display Attributes Select display attr (multiple values ar Avgalable:	

	Show in Con	ngo Box: All 💌	
List Search Include Search Region: All Que	ryable Attributes		•
Query List Automatically			
Choice List Options			
Query Limit:		10 🔺	
Query Limit: Most Recently Used Count:		10 🜩	
Query Limit: Most Recently Used Count: Elter Combo Box Using:	<pre></pre>		

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

Formers Add On	erview × 0000/ProjectAppModule.v	ni a pelonor	dert mesAllV00 sml ×			
ieneral						
nthy Objects	Attributes					Set Source Orde
ittributes	Vew object attributes can be mapp	ed to entity attri	sutes, calculated or SQL-derived.			
)uery	(*		8.60			
eva .	( Rate					
en Accessors						+·/>
st UII Hints	Name	Type	Alias Name	Entity Usage	Info	
	EnventoryDenid	Nather	DWENTORY_THEN_D	petroprinesAl	1 72	
	TaxDete	Date	TAX_DATE	oeOrderLinesAl		
	TaxCode	String	TAX_CODE	oeOrderLinesAl		
	TaxRate	Number	TAX_RATE	oeOrderLinesAl		
	InvoiceInterfaceStatusCode	String	INVOICE_INTERFACE_STATUS_CODE	oeOrderLinesAl		
	DemandClassCode	String	DEMAND_CLASS_CODE	orOrdeninesAl		
	PriceListId	Nunber	PRICE_UST_ID	oeOrderLinesAl		
	PricingDate	Date	PRJCING_DATE	oeOrderLinesAl		
	Stipment/Kunber	Number	SHIPMENT_NUMBER	oeOrderLinesAll		
	Agreement2d	Number	AGREEMENT JD	oeOrderLinesAl		
	ShipmentPriorityCode	String	SHIPHENT_PRIORITY_CODE	oeOrderLinesAl		
	🗉 Custom Properties: Invento	ryltemid				4-/3
	E List of Values: InventoryIte	mid				+/>
	Enable this attribute to display a lo	t of values to use	in the user interface.			
	the states					
	LISTS OF VIRUNES					

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.



Dbject Navigator	Property Palette		
ITEMLOV	浅泉  加佐   岡和	Π 📌	Find:
B-Forms	LOV: ITEMLOV		
B Triggers	= General		
Alerts	# Name	ITEM	LCN
<ul> <li>Attached Libraries</li> </ul>	B. Schelass Information	1.00	
B Data Blocks	<ul> <li>Subcass an ormation</li> </ul>	104	
E- ORDER HEADERS	<ul> <li>Comments</li> </ul>		
C CNLPOR II ATE OFT	= Functional		
	and Title	Item I	List Of Values
E Bems	+ List Type	Recor	ed Group
Relations	# Record Group	ITEM	RG
CE_ORDER_HEADE	R_OE_ORDER_LI  + Column Mapping Properties		More
CE_ORDER_LINES	<ul> <li>Elter Before Display</li> </ul>	No	
E 78 CALENDAR	In Action of States	100	
BLOOKNAME	<ul> <li>Automatic Usplay</li> </ul>	NO	LOV Column Mapping
E TO PROGRESS_INDICATOR	Automatic Refresh	Yes	Column Names
E THE DETAILBLOOK	Automatic Select	Yes	(country)
E Canvases	Se Automatic Skip	Yes	- SEGRENTI -
E-LOVs	Automatic Position	Yes	INVENTORY_ITEM_ID
TEM.OV	Se Automatic Column Width	Yes	1
TIMEZONES_AND_DATETIN	E Bhusical		
TIMEZONES	= Physical	-	
APPCORE_200M	» X Position	0	
APP_ACCESS	+ Y Position	0	
APPCORE_ALT_REGIONS	⊯ Width	3.75	· · · · · · · · · · · · · · · · · · ·
ENABLE LIST LAMP	<ul> <li>Height</li> </ul>	4	Refurn Them
Object Groups	34		OF OPDER UNES TOPE
Parameters	Description of 2012 total		- locoused mentance
Popup Menus	Propercy 6363 hink		Display Width
E Dt ARE CLISTORI (Rackage Str	*1		0
R APP CLISTOM (Package Box	N)		
- CHECK PACKAGE FAILURE	* (Drocenture Bock/)		Column ]]Be
CITAR ALL MASTER DETA	11 S* (Broceck an Body)		
CHERY MASTER DETAILS	(Procedure Body)		and and the second second
Property Classes	(conservation and f)		QK <u>Cancel</u> <u>Help</u>
Record Groups			

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

DATATOR	( TIEM ID	6			1
- L		Property Palette			
2	B B ORG D	海島  加払   田和	n 🗴	Find:	N.
	8-60 LINE_TYPE_ID	Item: INVENTORY_ITEM_ID			
	⊕ GEI LINE_NUMBER.	= General			
	E GORDERED_ITEM	# Name	INVENTORY ITEM ID		
6	E-BUREQUEST_DATE	Ren Type	Text Item		
	R-GO SCHEDULE SHIP DATE	<ul> <li>Subclass Information</li> </ul>			
	E SE ORDER OUANTITY UOM	» Comments			
1	8- EEI PRICING_QUANITITY	<ul> <li>Help Book Topic</li> </ul>			
	E- E PRICING_QUANTITY_UOM	# Euroctional			
	E GANCELLED_QUANTITY	# Navination			
5	IE- BEE SHEPPED_QUANTITY	= Data			
	B- GD BLARRED_QUANTITY	* Data			
	E B FOP LED QUANTITY	* Calculation			
	E E SHIPPING CUANTITY LICH	* Records			
	DELIVERY LEAD TIME	* Database			
	E E TAX_DEMPT_FLAG	= List of Values (LOV)			
	E TAX_DEMPT_NUMBER	List of Values	TEMLOV		
•	E M TAX_EXEMPT_REASON_CODE	<ul> <li>List X Position</li> </ul>	0		
B	B-BEISHP_FROM_ORG_ID	<ul> <li>List Y Position</li> </ul>	0		
	E- III SLEDWENTORY	<ul> <li>Validate from List</li> </ul>	No		
	E E SHP_TO_ORG_D	= Editor			
	E E DE MER TO ORG TO	<ul> <li>Editor</li> </ul>	<11ul>		
	IE SHIP TO CONTACT ID	<ul> <li>Editor X Position</li> </ul>	0		
	. See DELIVER TO CONTACT ID	<ul> <li>Editor Y Position</li> </ul>	0		
	E ED INVOICE_TO_CONTACT_ID	= Physical			
	Image: Sold_TO_ORG_DD	<ul> <li>Visible</li> </ul>	Yes		
	E GLICUST_PO_NUMEER	<ul> <li>Canvas</li> </ul>	<11.1>		
	BE SHIP_TOLERANCE_ABOVE	+ Tab Page	(3.4)		
	E BIOEMAD BLOCK DECOM	* X Position	0		
	E E DE VED DOLL THE COL	a V Bushino	-		
	E E CUSTOMER DOCK CODE	8 145-85	1 (01)		
	8- EEI CUSTOMER_306	a teach	100		
	8- CUSTOMER_FRODUCTION_LINE	- mage	.100		
	E GUST_MODEL_SERIAL_NUMBER.	* Devel	Lundreu		
	Image: Book of the second s	* Kendered	1705		

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:

This list is filtered according to the cu	irrent project's <u>selected technologies</u> .	
Categories:	Items:	Show All Descriptions
General	Business Components from Tables	^
- Appin-autors - Connections - Deployment Profiles - Diagrams - External Applications - Java - Projects - BPM Tier	Application Module Launches the Cheat Application Module witzerd, an application module. Use application modules to view objects, to handle transactions, and to pro methods. To enable this option, you must select a project Navigator. Before you can finish creating the ne will be promoted to select (or create) a database	which allows you to create to assemble and organize vide business service in the Application ex application module, you e connection.
<ul> <li>Business Tier</li> </ul>	🐴 Association	
ADF Business Components	Business Components Deployment Profiles	
Security	Business Components Diagram	
SOA Tier All Items	in Default Data Model Components	
	-180 Dompio	~

Create Application Mo	dule - Step 1 of 5		×
Name			2
🤪 Name	Application mod	ules are for assembling, packaging, and deploying the view objects and business services of an ap	plication.
Data Model	Package:	com.xx.myproject.model.am	Browse
Application Modules	Name:	MyProjectAppModule	]
Java	Display Name:	My Project App Module	9
) Summary	Extends:		Browse
	Property Set:	<none></none>	]
			<i>,</i>
11ala			Grand
Geb		< gack Ensh	Cancel

Once you shuffle the Header and Line, click finish.



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.

0n	scie Forms Builder - C:\FORMVSADF.fmb			
	hiert Navieator	-		
	Negative and the second s	Property Palette		
	Prisale ·	34 22 34 0	) 🔗 Finds	84
	E-20 FORMISADE	Form Module: FORM/SADF		
6	S Triggers	= General		
-	Attached Libraries	■ Name	FORMVSADF	
- U	Data Blocks	Subclass Information	MODULE	
	R G OF OPER LINES	<ul> <li>Comments</li> </ul>		
100	R- CALENDAR	<ul> <li>Help Ecoli, Title</li> </ul>		
	E SLOCKNAME	* Functional		
×	PROGRESS_INDICATOR	= Menu Security		
R	E - M DETAILBLOCK	Menu Role		
10	B Canvases	= Navigation		
	8-LOVs	Mouse Navigation Limit	Form	
	Object Groups	First Navigation Data Block	OF ORDER HEADERS	
×	Popup Menus	* Records		_
	Program Units	# Database		
10	(8) PP_CLISTOM (Package Spec)	# Physical		
	B - APP_CUSTOM (Package Body)	R Coordinate System		
(6)	CHECK, PWLIAKE, PALLARE* (Procedure Body)	h Her 20 Centrals	Nex	
1	OLERY MASTER DETAILS* (Procedure Body)	Green Machine Cartaine	103	
-10	8 Property Classes	<ul> <li>Form Horizontal Toolbar Carryas</li> <li>Form Marking Toolbar Carryas</li> </ul>	- the second sec	
	Record Groups	- Form Percical Tooldar Carries	5007	
	Visual Attributes	= International	P. ()	
	Windows	* Direction	Derauk	
	- Menus	= Compatibility		
	Object Libraries	Runtime Compatibility Mode	4.5	
	≥ fundt en Peckaget ≌ Database Bijecto	z Biok to which Form Builder should navi	ngate at form startup and CLEAR_FORM.	2

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:

PROCEDURE when new form instance

```
IS
     win_msg VARCHAR2(2000);
l_org_id NUMBER := TO_NUMBER(FND_PROFILE.VALUE('ORG_ID'));
l_user_id NUMBER := TO_NUMBER(FND_PROFILE.VALUE('USER_ID'));
l_percision NUMBER := NULL;
l_name hr_operating_units.name%TYPE;
BEGIN
|| Set First Navigation Block
  FIRST NAVIGATION BLOCK ('OE ORDER HEADERS');
/*
|| Assign Valued To Control Block
  :CONTROL.USER ID := 1 user id;
|| Set Operating Unit Name
* /
 SELECT hou.name
  INTO l_name
  FROM hr_operating_units hou
WHERE hou.organization_id = l_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 Daveloper 11g necesse 1 - romsviskatijas i MyMoj lie _fdit _View Application Refactor _Search _Navi	ectULjpr igate <u>B</u> uild <u>B</u> un Versigni •   ▲ ≌ ﷺ <b>●</b> •   ▶ •	ng <u>T</u> ools <u>W</u> ind	dow <u>H</u> elp	
Application Navigator ×		E FormsVsAdf O	verview ×   MyProjectAppModule.xml	× MoorderLines48/VO.xm/ ×
E FormsVsAdf	+ 🔁 •			
* Projects	🖸 🍓 🏹 • 🐲 • 👘	General		
MyProjectModeBundle.properties	💩 New Gallery	State-Saperte	and date	×
Bedry Stratego Module     MyProjectAppModule.xml     MyProjectAppModule.xml     Bedry ToLinestInk     Bedry ToLinestInk     Bedry ToLinestInk     Bedry ToLinestInk	All Technologies Cu This list is filtered acco	rent Project Techno	logies project's <u>exilicated technologies.</u>	
US ceOrderHeadersAl	Caterorier		Tiama	Show All Descriptions
	Connections Deployment De Deployment Pr Diagrams External Applic	scriptors ofiles ations	ADF Task Flow Creates a task flow source file whos flow or part of the web application's To enable this option, you must sele Application Navigator.	e contents define either a bounded task unbounded task flow. ct a project or a file within a project in the
🗄 👯 HeadersToLinkLink	-Java Projects		aDF Task Flow Based on Human Tas	k
<ul> <li></li></ul>	NML BDM Tier		ADF Task Flow Template	
B- D MyProjectUI	-SOA Tier	1	B JSF Declarative Component	
Web Content     Web 3N <sup>4</sup>	Web Tier     Anniet		JSF Page	
🖻 🧰 Page Flows	HTML		ISF Page Flow and Configuration (fe	ices-config.xml)
- Ambratan Past star	- 57		35F Page Fragment	
Data Controls     Deta Controls     Deta Controls     Deta Control	-Jor Serviets -All Items	~	S JSF Page Template	
Recently Opened Files	Help			OK Cancel

Efine either a bounded task flow or part of the web application's pages or page fragments, but not both. You can also this time.  public_html/WEB-INF  Browse
pages or page fragments, but not both. You can also this time. public_html\WEB-INF  Browse
publc_html\WEB-INF  Browse
public_html\WEB-INF Browse
publc_html\WEB-INF  Browse
public_html/WEB-INF Bcowse
late Changes
OK Cancel



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.

Oracle JDeveloper 11g Release 1 - FormsVsAdfijvs : MyProjectModeljpr     File Edit View Application Refactor Search Navigate Build Run Versioning Tools Window Help	- Search
<u>File Edit View Application Refactor Search Navigate Build Run Versioning Tools Window H</u> elp	Search
	🔗 - Search
【 2 回 目 目   ◎ ( ○ ) 从 目 目   ○ - ○ -   品 -   品 益 益 道 本 -   ▶ - 孝 -	
🕞 🚱 Application Navigator X 💦 MyProject AppModule.xml X 🖄 OeOrderLines AllVVO.xml X 🕼 Sales Orders-btf.xml X 🕑 🐨 🛗 Component F	Palette × 🕼 Resource Pale
🔮 🖻 FormsVsAdf 🔹 🔹 🔍 🔍 🔍 100% 🗣   Show ។ । 🐧 🐴 🦷 । 💥 🏭   💽 🚯 । 🔞 💥 💥   🗿 🦳 💥 ADF Task Flow	
Projects Q № V + SE + Bounded Task Flow	
D     D-     MyProjectHodel       B     D-     Application Sources	ents
Constant Select Method Call	l on
Router     Router	Restore call keturn v Case votrof Flow Rule

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

🐯 method-call - Ini	tializeSalesOrderMethod - Property Inspector ×	_
🖳 I 📌 📴 I 🖉	Find	<b>₽</b> ₽) <b>?</b>
🖃 General		^
• Activity ID *: I	nitializeSalesOrderMethod	
Method *: #	t{pageFlowScope.OrderSalesPfs.initalizeScopeValues}	~
···· Outcome *·····		
• Fixed Outcome:	SalesOrderJsff	• ~
toString():	<default> (false)</default>	• ~
Description		
Display Name:		~
Description:		~
Small Icon:		~
Large Icon:		~ L
Parameters		
		+ ×
Class	Value *	
Return Value:		× •

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.



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.

Gracle IDeveloper 11g Release 1	- FormsVsAdf1.jws : MyProjectM
Elle Edit View Application	Refactor Search Navigate
3 2 8 6 19 (* 1 X B	B 0 . 0 . 8 4
Application Navigator X	
FormsVsAdf1	+ 🖪 ·
* Projects	3 97.2.
Constant Sectors Sectors     Constant Sectors Sectors     Constant Sectors Sectors     Constant Sectors Sectors     Constant Sectors	di ana Ana bangan ang ang ang ang ang ang ang ang an
a a cardardandara bi	
e Derations	

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



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

Oracle JDeveloper 11g Release 1 - FormsVsAdf1.jws : MyProjectUi.j	* Chicago and a constant of the second state			
<u>File Edit View Application Refactor Search Navigate</u>	Build <u>R</u> un Versi <u>o</u> ning <u>T</u> ools <u>W</u> indow <u>H</u> elp			
🕑 🖻 🗐 🗊 🔍 🔍 🐘 🛍 🛯 🗶 • ( ) 🛦 盐 🛎 🛎 👘 • ( ) • 💩 • ( )				
@Application Navigator ×	🗟 OrderSalesPageFlowScopeBean.java 🗴 👔 MyProjectAppModule.xml 🗴 🔯 OrderSales-btf.xml 🗴 🕎 OrderSales_btf_OrderSales_btf_InitializeSalesOrderMethodPageDef.xml 🗴			
🔁 FormsVsAdf1 🔹 🔁 🔹	Page Data Binding Definition			
Trojects 💽 🗞 🍸 - 💯 - 🖉 -	This shows the Oracle ADF data bindings defined for your page. Select a binding to see its relationship to the underlying Data Control.			
ctModel	Date Technology and the feature of the technology of the Technology and			
ication Sources	Data binding Registry: <u>com/xx/myproject/view/Databindings.cpx</u>			
com.xx.myproject.model	Bindings and Executables Contextual Events Parameters			
com.xx.myproject.model.am				
MyProjectAppModule	Model			
MyProjectAppModule.xml				
MyProjectAppModuleClient.java	Bindings db / X Executables db / X Data Control			
MyProjectAppModuleImpl.java	setAMSessionVariables			
com.xx.myproject.model.assoc				
com.xx.myproject.model.eo				

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.

ipr <u>B</u> uild <u>R</u> un Ver	si <u>o</u> ning <u>T</u> ools <u>W</u> indow	<u>H</u> elp	Contraction (set a	Country Made	Monat No.	
i 🎎 🕌 🛥 🔹 i 🖡	> - 🎄 - 📕-					
OrderSales-btf.	aml ×					V
						🗱 I 🕐 🗖
General Description	🕼 Input Parameter D	efinitions				÷×
Control Flows	Name * 🔺	Class	Value	Req	uired	
Managed Beans	pWarehouse	java.lang.Strir	ng #{pageFlowSc	cope.pWarehouse}	<b>V</b>	11.
Parameters Behavior	🕼 Return Value Defin	itions				+×
	Name * 🔺	c	lass	Value *		

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

Dracle Forms Builder - C:\FORMVSADF.fmb				
File Edit Proper	ty Program Tools Window Help			
· · · · · · · · · · · · · · · · · · ·	Object Navigator			
		S 84		
	Find:	1		
	Forms	A		
	Formvsadf			
	Alerts			
-	Attached Libraries			
	B Data Blocks			
	Editors			
	E LOVs			
	Dect Groups     Parameters			
	🔀 🚽 📈 PWAREHOUSE	🐺 Property Palette		
	APPCORE_PROFILE_NOT_DEFINED		A	
	APPCORE_FUNCTION_PROCESSED	-4 #   2 2   3 0   1	Find: % %	
	APPCORE_NONE_AVAILABLE	Parameter: PWAREHOUSE		
1	APPCOPE ZOOM VALUE	= General	A	
	APPCORE ALT REGIONS	Name	PWAREHOUSE	
-	FOLDER_BLOCKS	Subclass Information	TEXT LITEM	
	QUERY_ONLY	* Comments	THE PARTY AND A DESCRIPTION OF	
	G_QUERY_FIND	= Data		
	HR_MODE	Parameter Data Type	Number	
	B Program Units	Maximum Length	30	
	Property Classes	Parameter Toitial Value		
	H Record Groups	- Fordinator and a Faule		
	Visual Attributes			
	D-PL/SQL Libraries		<b>v</b>	
	Object Libraries	<u> </u>	Þ	
	Built-in Packages	Source object and source module inform	nation for referenced object.	
	the Database Objects			

## **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>

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 was overwhelming, time consuming, always challenging, sometimes frustrating; however, the most important thing to remember is that it is possible. My journey started with no knowledge of ADF. It continues with spending hours exploring the internet to find documentation and videos to assist in my understanding of ADF, countless attempts to build a single application that would work from end-to-end, to attending ADF design meetings where I would write down five to ten ADF concepts or terms that I need to research after each meeting, to building my first application where I need ask ten "how to" questions of day, to building an application on my own and concluding with supporting that application in a production environment.

I wrote write this paper because I know there are other Oracle Forms Developers that still need to take on this journey and I want to provide a high level how-to guide. Secondly, I wanted to share some of the great references that helped me gain my understanding of ADF. My understanding of ADF grows daily and it has been nine months since I started

my journey and I have to say that I just scratched the surface of ADF's capabilities, so be patient and take that first step into the world of ADF.