

Oracle ADF & JDeveloper for Forms Developers



Presented by: John Jay King
King Training Resources - john@kingtraining.com

Download this paper from: <http://www.kingtraining.com>



- Be aware of Oracle's Application Development Framework (ADF) standards-based applications
- Understand ADF development using advanced graphical editing and declarative techniques
- Know ADF's Java and XML underpinnings
- See how JDeveloper creates ADF applications
- Grasp ADF BC's reusable data source support
- Learn how to build UIs graphically with drag and drop components, and declarative properties



- John King – Partner, King Training Resources
- Providing training to Oracle and IT community for over 20 years – <http://www.kingtraining.com>
- “Techie” who knows Oracle, SQL, Java, and PL/SQL pretty well (along with many other topics)
- Leader in Service Oriented Architecture (SOA) design and implementation
- Member of ODTUG (Oracle Development Tools User Group) Board of Directors
- Moved to Scottsdale, Arizona last year (we miss Colorado!)
- Active member of Rocky Mountain Oracle Users Group (RMOUG)



- Forms Developer
- 4GL Developer
- Java Developer
- All of the above
- None of the above



- 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
- Java EE is one of the most widespread application environments today
- ADF's 4GL features make application development much easier than normal Java "coding"
- Oracle is rewriting their ERP stack as "Fusion Applications" using ADF; the already rich toolset gets richer every day



- Probably not well
 - Someone with very basic Java and Web Skills can easily create applications with ADF (much the same as someone with basic PL/SQL could create very basic Oracle Forms)
 - Someone on your team needs to know Java very well
 - Someone on your team needs to know Groovy
- Someone on your team needs to understand ADF and its available components very well

Is Forms Going Away?



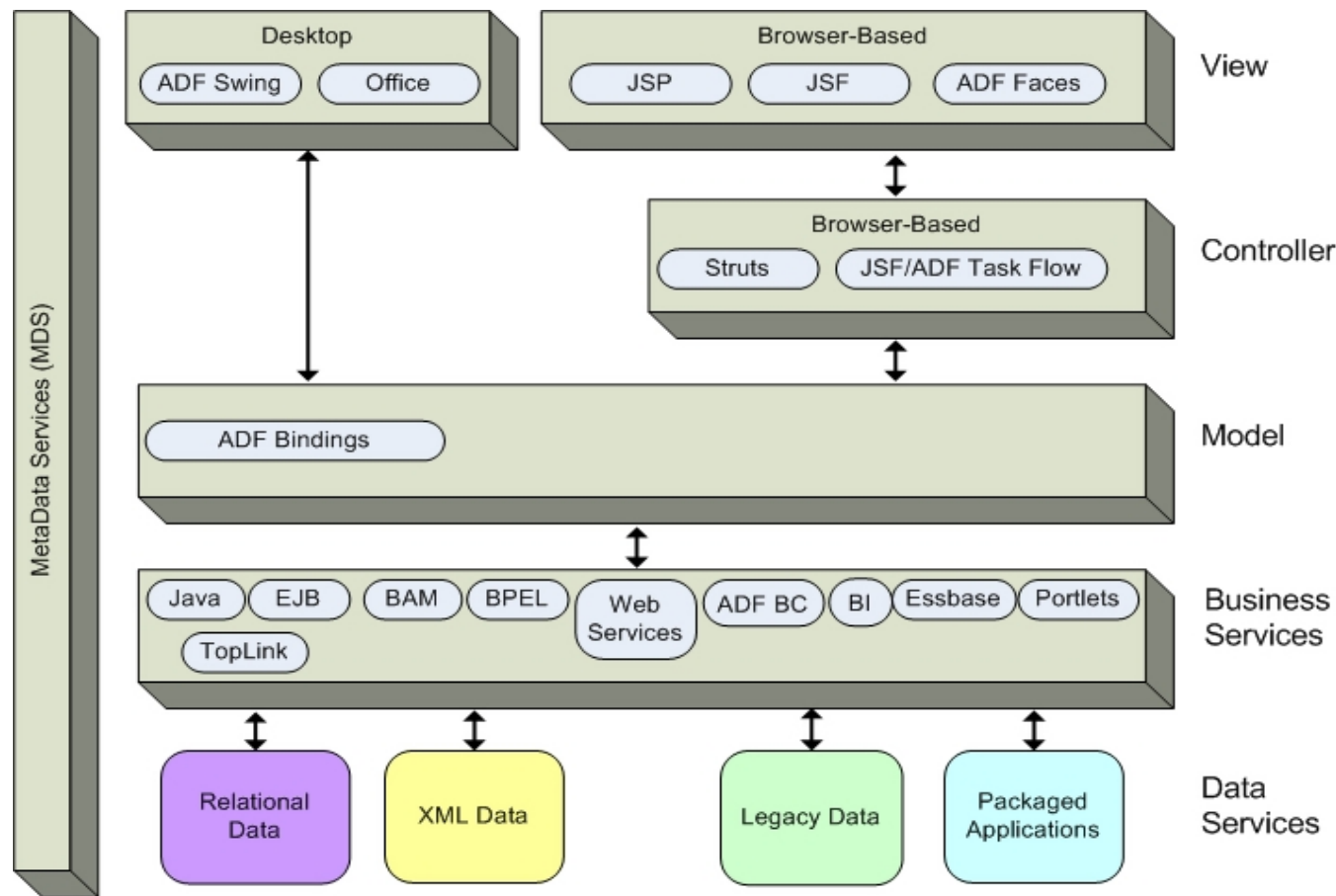
- NO, NO, NO, NO, NO
- Oracle is committed to supporting Oracle Forms for many years to come
- A new version of Oracle Forms (12c) is on the way!



- ADF is a “meta-Framework” interacting with a variety of underlying software components (including Frameworks) to provide:
 - Database connectivity and transfer
 - Mapping of application views to data sources
 - Database interaction: constraints, keys, data types, master/detail, null handling
 - Data caching via entity objects
 - Transaction management (locks, commit, rollback, etc...)
 - Declarative validation
 - Business logic and event handling
 - User Interface (UI) logic, flow, look & feel
 - Data-bound UI Components
 - UI properties including: formatting, colors, defaults, visual components, LOVs, etc...



- ADF Technology simplifies interaction with “Java” EE and Oracle’s Fusion Middleware





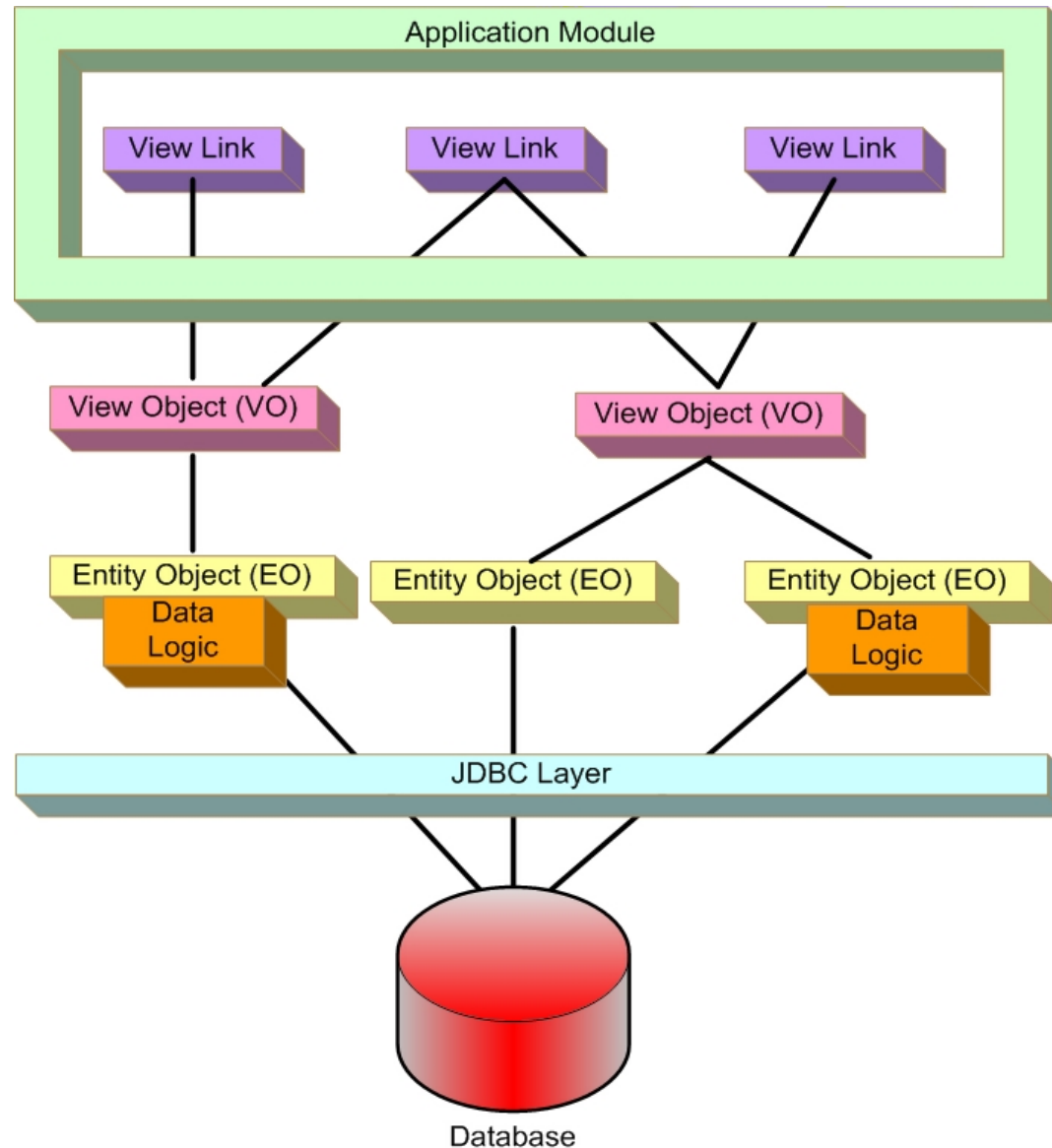
- ADF has many parts but two are central to creating applications
 - ADF BC Business Components (data)
 - ADF Faces Graphical User Interface



- ADF Business Components (ADF BC) is a framework that simplifies developing Java EE business services
- ADF BC is part of the ADF Business Services layer and is used to provide:
 - Persistence and data retrieval with SQL using data views
 - Object-Relational Mapping (ORM) between Java classes and database data
 - Simplified data access, validation, and business logic
 - Transactional infrastructure
 - Implementation of best practices



- ADF BC is implemented using a variety of objects to:
 - Define Insert-Update-Delete views to perform queries and data manipulation
 - Define query views (read-only)
 - Define links between queries





- ADF BC uses a variety of object types to represent data:
 - Database tables and views Application Base Data
 - Entity Objects Business rules, validations, defaults for a table (or view)
 - View Objects SQL output to query, filter, join, modify, or sequence data
 - Application Modules Use View Objects to access/modify data acting as a back-end data service
 - Appl. Module Data Model Describes actual View Object uses
- Objects may be reused in multiple Application Modules



- After identifying Entity Objects and View Objects two additional ADF Data Model components are used
 - Data Controls Java objects used to abstract View Object Business Services
 - Binding Containers Java object; provides data access to a single ADF application page, fragment, or activity



- Java Server Faces (JSF) is a Web-tier framework of JSP technology and JSP Tag libraries to create and use User Interface components
- JSF is extended by components of Oracle ADF Faces
- JSF includes:
 - Runtime architecture
 - Library of JSF components
 - JSF “Life Cycle”
 - Many JSF-Oriented Files



- Even though JSF sought to simplify user interface; it is often felt to be too complex
- Oracle has extended JSF as “ADF Faces” providing a set of libraries and tags that include enhanced UI components and easier use
- Oracle has presented ADF Faces to the Open Source community where it is part of the Apache Foundation Trinidad MyFaces project

<http://myfaces.apache.org/trinidad/index.html>



- Using ADF Faces is simple using JDeveloper:
 - Add Application layout containers to describe user interface
 - Add ADF Faces components to layout containers
 - All UI is done with ADF Faces; no HTML coding
- Features added by ADF Faces:
 - Pop-ups and Dialog boxes
 - Data Visualization Tools: Charts, graphics, etc...
 - Declarative AJAX support
 - More...



- The ADF Controller extends the JSF controller and controls ADF's MVC (Model-View-Controller) in ADF
- ADF Controller features include:
 - Sequence of page displays (may be conditional)
 - Allows partial-page processing in the same way as full page processing; only the necessary part of a page is rendered, the rest is unchanged
 - Allows reuse of page parts
 - Provides conditional control of page flow



- ADF Faces is designed to create “rich-client” (RC) interfaces; full-featured and declarative including:
 - Complete JDeveloper support graphic development (screen-painter) and property palettes
 - Visual Editor
 - Property Inspector
 - Changeable “skins” to easily alter look-and-feel
 - Modifiable look-and-feel properties (declarative)
 - Layout control



- JDeveloper provides a world-class, easy to use IDE
- Oracle has extended JDeveloper beyond Java to include:
 - Oracle ADF modeling, business services, and GUI design
 - XML edit including Syntax Checking & Schema Validation
 - SQL development including debugging of stored PL/SQL
 - UML Modeling and MDA (Model Driven Architecture)
 - Web Services development
 - ESB design
 - BPEL design
 - Portlets



- JDeveloper is Free!
- To learn more about JDeveloper, see Oracle's website:

<http://www.oracle.com/technetwork/developer-tools/jdev/overview/index.html>



- Oracle WebLogic Server is Oracle's preferred platform to provide both a standard Java EE environment and an environment specifically tailored to Oracle Fusion Middleware; providing:
 - Complete Java EE 5 compatibility
 - Complete Java SE 6 compatibility
 - Web Services support
 - Integration with Oracle's Fusion Middleware tools



- Oracle WebLogic Server is the replacement for Oracle Application Server (OAS) and OC4J
- OAS and OC4J are still supported and may be used instead of WebLogic if desired but ADF requires at least Java 1.5 / Java 5 (needed for ADF)
- To learn more about Oracle WebLogic Server see Oracle's website:

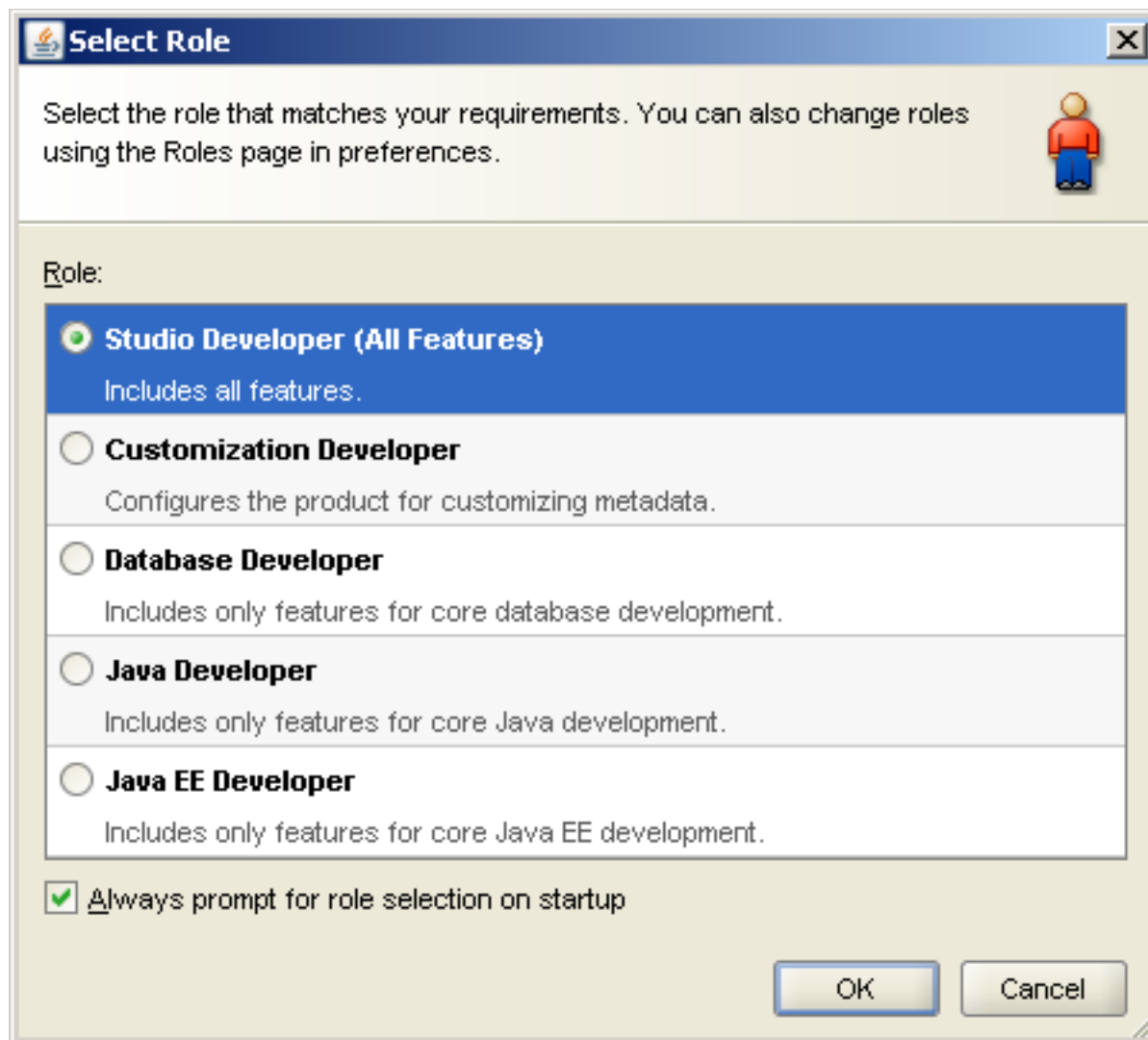
<http://www.oracle.com/us/products/middleware/application-server/index.htm>

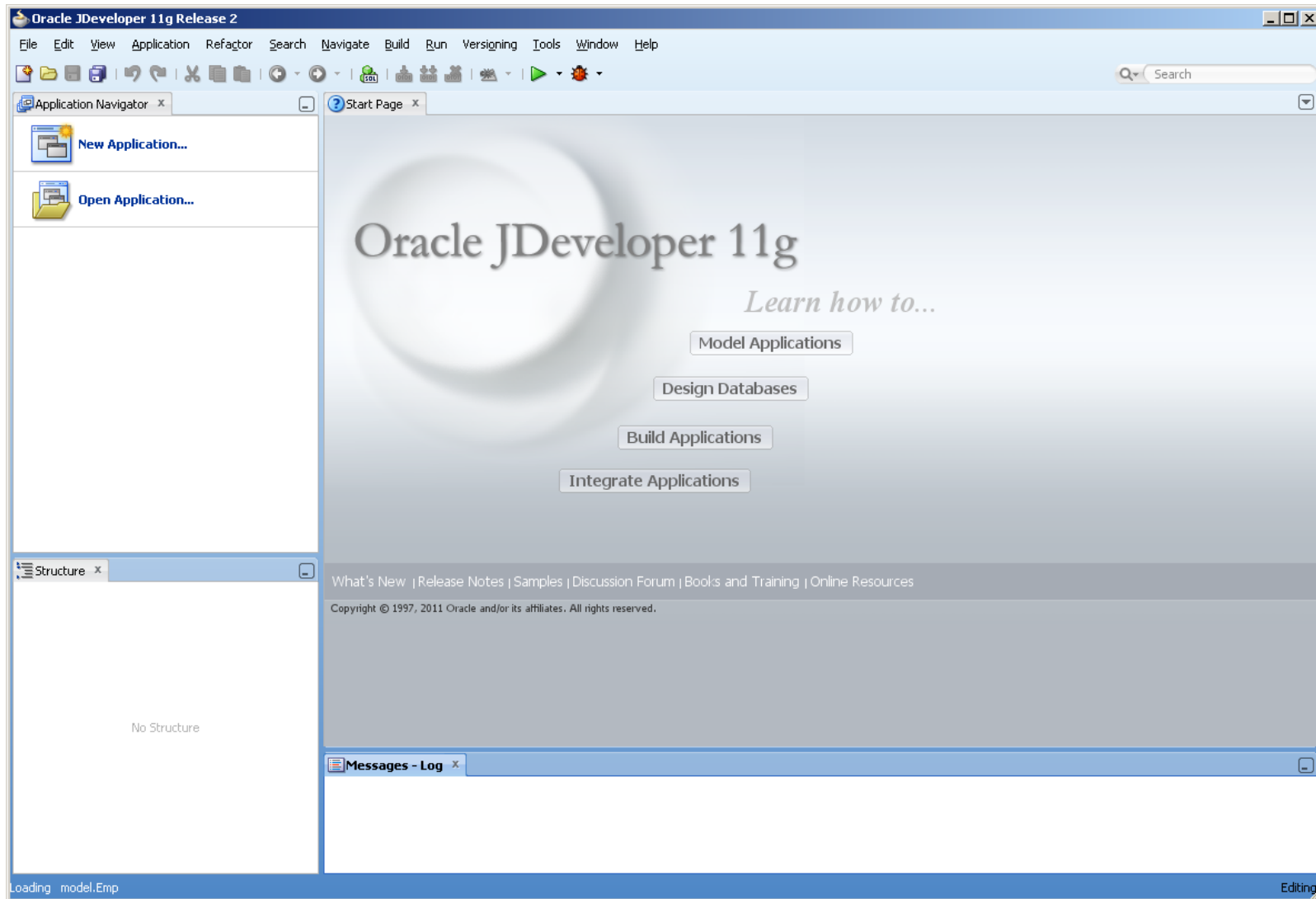


Oracle JDeveloper **11g**
11.1.2.1.0
Productivity with Choice

ORACLE

Copyright © 1997, 2011 Oracle and/or its affiliates. All rights reserved.







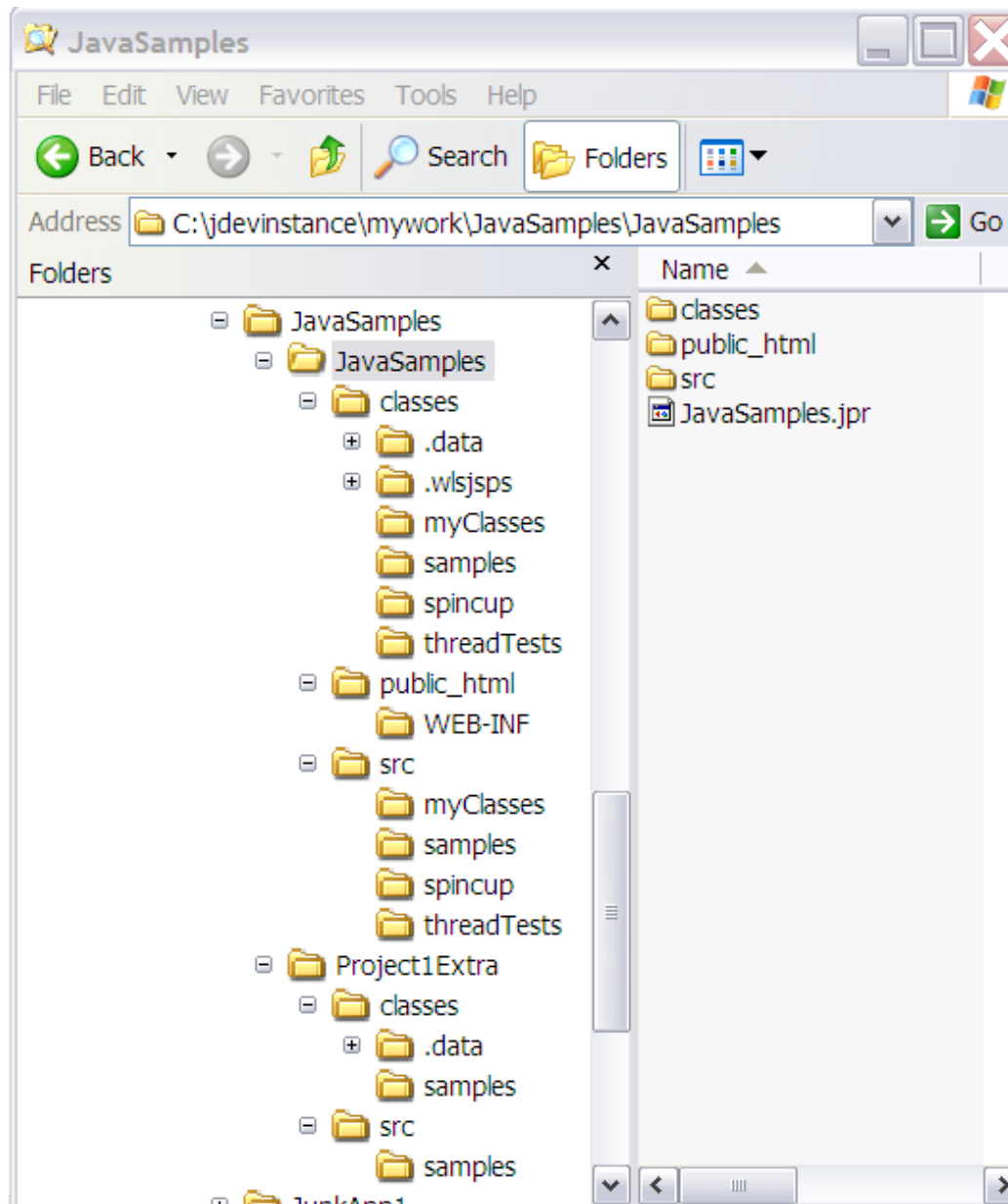
The screenshot shows the Oracle JDeveloper 11g IDE interface. The main editor area displays the source code of a JSP file named 'WelcomeJSP.jsp'. The code includes HTML tags for a head section with a title and a body section with a loop that prints 'Hello World!'. Annotations with blue arrows point to various parts of the IDE:

- Navigator(s)**: Points to the 'Application Navigator' on the left, which shows the project structure.
- Structure Displays**: Points to the 'WelcomeJSP.jsp - Structure' view at the bottom left, which shows a tree view of the document's structure.
- Editor Area and Message Area**: Points to the central code editor and the status bar at the bottom.
- Menu**: Points to the 'File' menu in the top menu bar.
- Search Text**: Points to the search field in the top toolbar.
- Toolbar(s)**: Points to the top toolbar containing various icons for file operations and development.
- Palette(s)**: Points to the 'Component Palette' on the right, which lists various JSP components like Attribute, Body, Declaration, etc.
- Property Inspector**: Points to the 'Head - Property Inspector' on the right, which shows properties for the selected 'head' element.



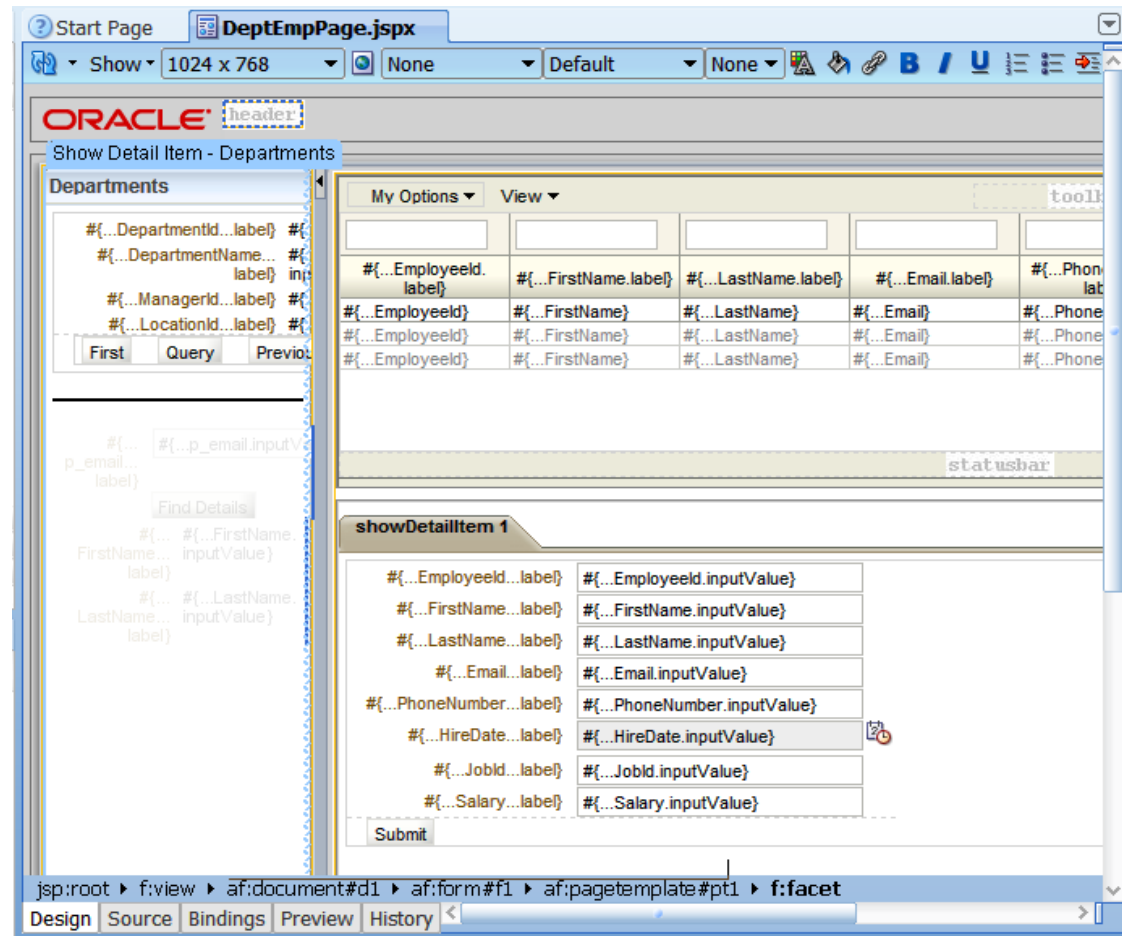
- JDeveloper uses a non-standard, Oracle-specific “Application” to group a collection of “Projects” (similar to how it is done by other IDEs)
- All files representing an “Application” share a common root directory (folder) on a disk
- Many Applications may be open at once in JDeveloper; but only one at a time will be visible in the Application Navigator

JDeveloper Directory Structure





- JDeveloper has many Code Editors & Visual Editors including: Java, XML, HTML, JSP, JSF/ADF Faces, BPEL, & more





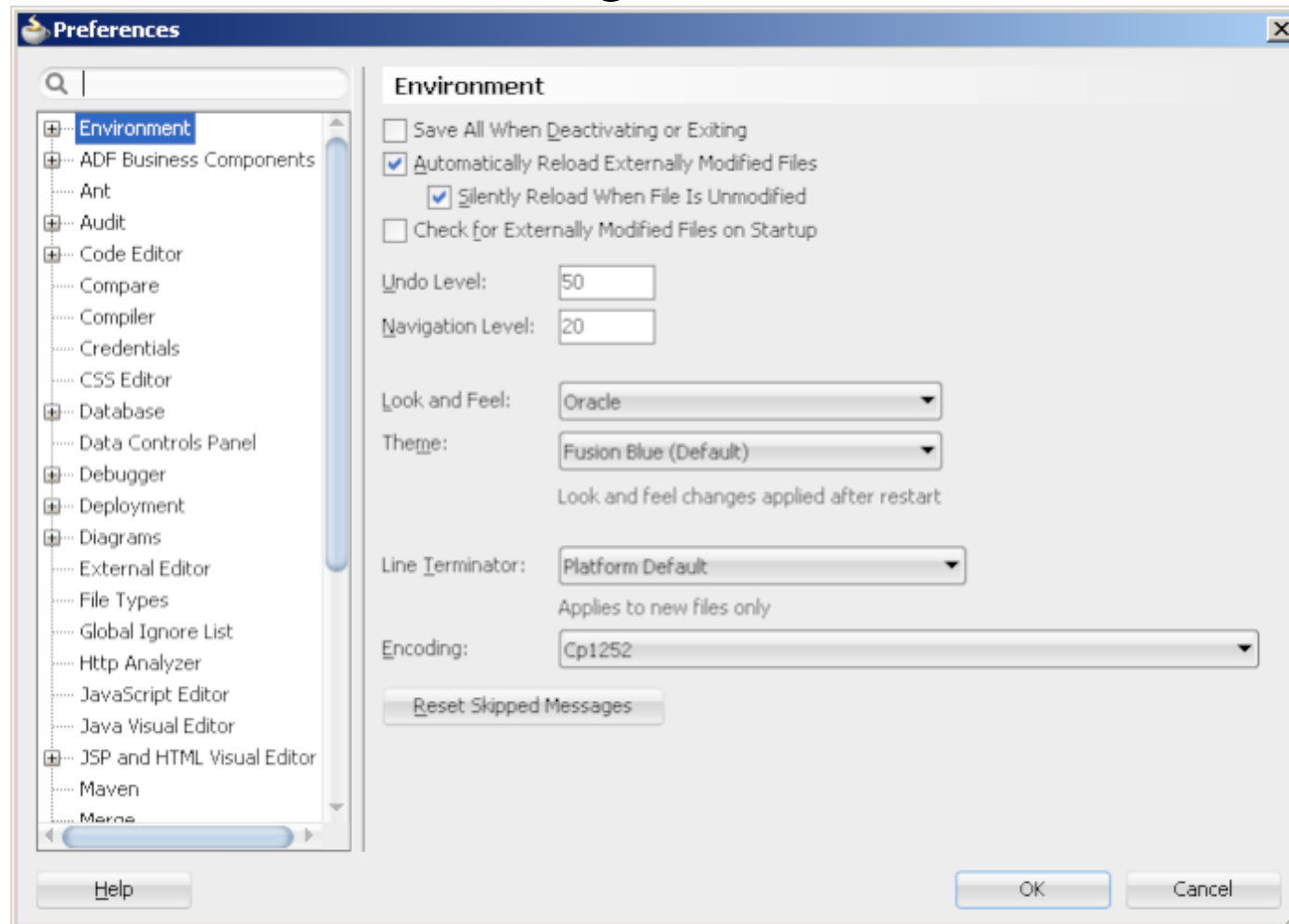
- JDeveloper allows local and remote debugging

The screenshot shows the Oracle JDeveloper 11g IDE. The main window displays a web page in design view with a table of employee data. A dialog box titled "ADF Lifecycle Breakpoints" is open, listing various lifecycle phases such as "JSF Restore View", "Initialize Context", and "Prepare Model". The "Before JSF Restore View" breakpoint is selected. The bottom status bar shows the current page flow path: "panelaccordion#pa1 > af:showdetailitem#scd1 > af:panelformlayout#pfl1 > af:panellabelandmessage#plam1".

DepartmentId	DepartmentName	ManagerId	LocationId	EmployeeId	FirstName	LastName	Email	Phone
1	Accounting	102	1	101	John	Deere	john.deere@oracle.com	515.121.20.34
2	Marketing	103	2	104	Neena	Kochhar	neena.kochhar@oracle.com	515.121.20.34
3	IT	104	3	105	Lex	Deane	lex.deane@oracle.com	515.121.20.34
4	Finance	105	4	106	Anna	Hartstein	anna.hartstein@oracle.com	515.121.20.34
5	Operations	106	5	107	Parto	Serpell	parto.serpell@oracle.com	515.121.20.34

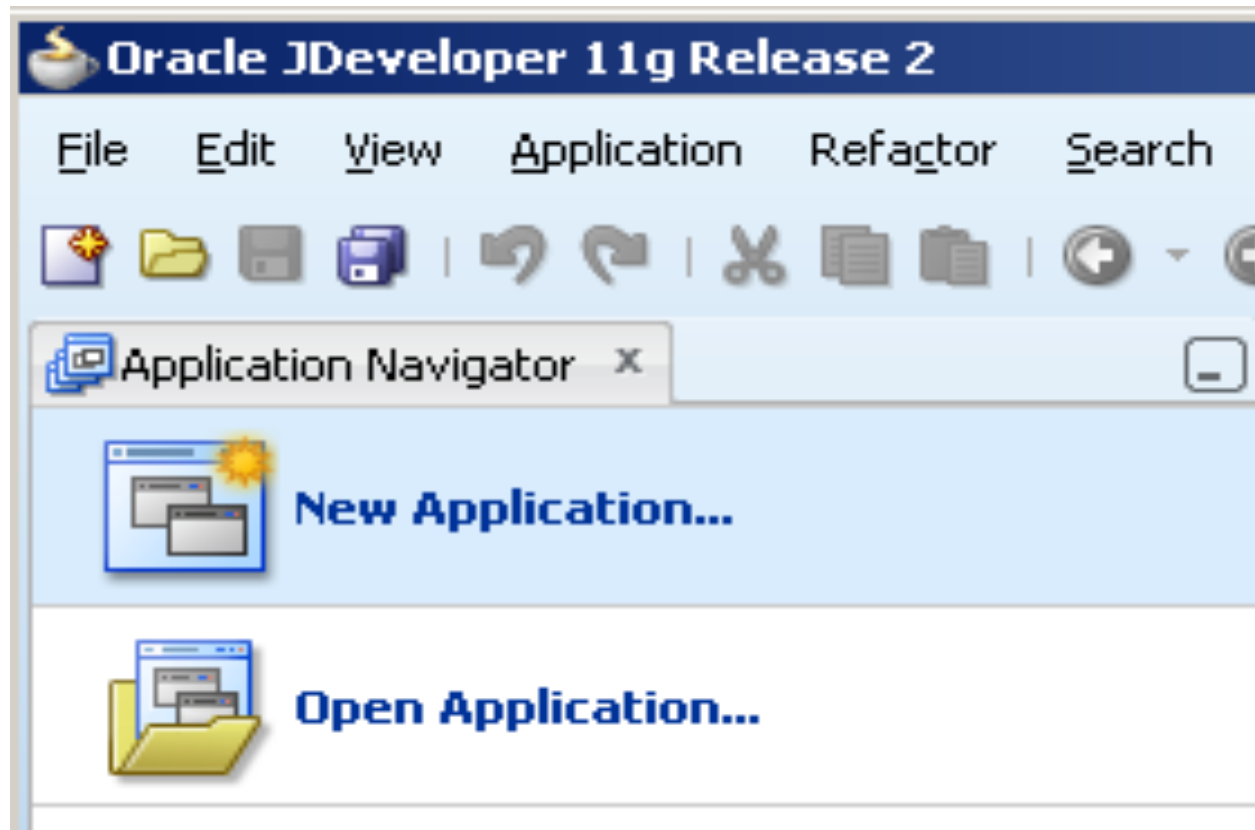


- JDeveloper is customizable; preferences may be viewed/modified using Tools->Preferences





- To create a new application use the JDeveloper menu's File->New->General->Applications option, or click Application Navigator -> New Application





Available Items

Search

Categories:

- General
 - Applications**
 - Connections
 - Projects
- Business Tier
 - Web Services
- Database Tier
- All Items

Items: Show All Descriptions

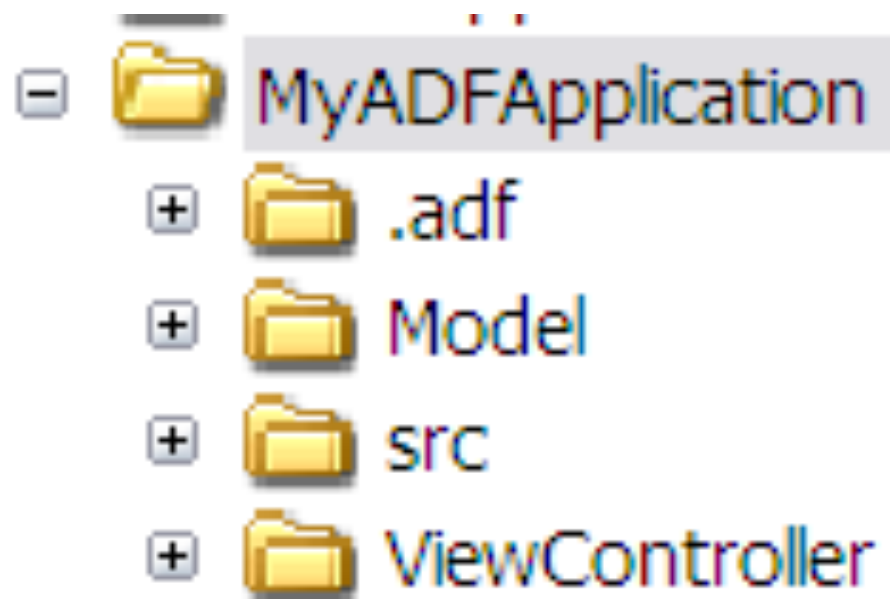
- Java Desktop Application
- Application from EAR File
- Application Template
- Custom Application
- Database Application
- Fusion Web Application (ADF)**
Creates a databound ADF web application. The application consists of one project for the view and controller components (ADF Faces and ADF Task Flows), and another project for the data model (ADF Business Components).
- Java Desktop Application (ADF)
- Java EE Web Application
- Maven Application
- UML Application

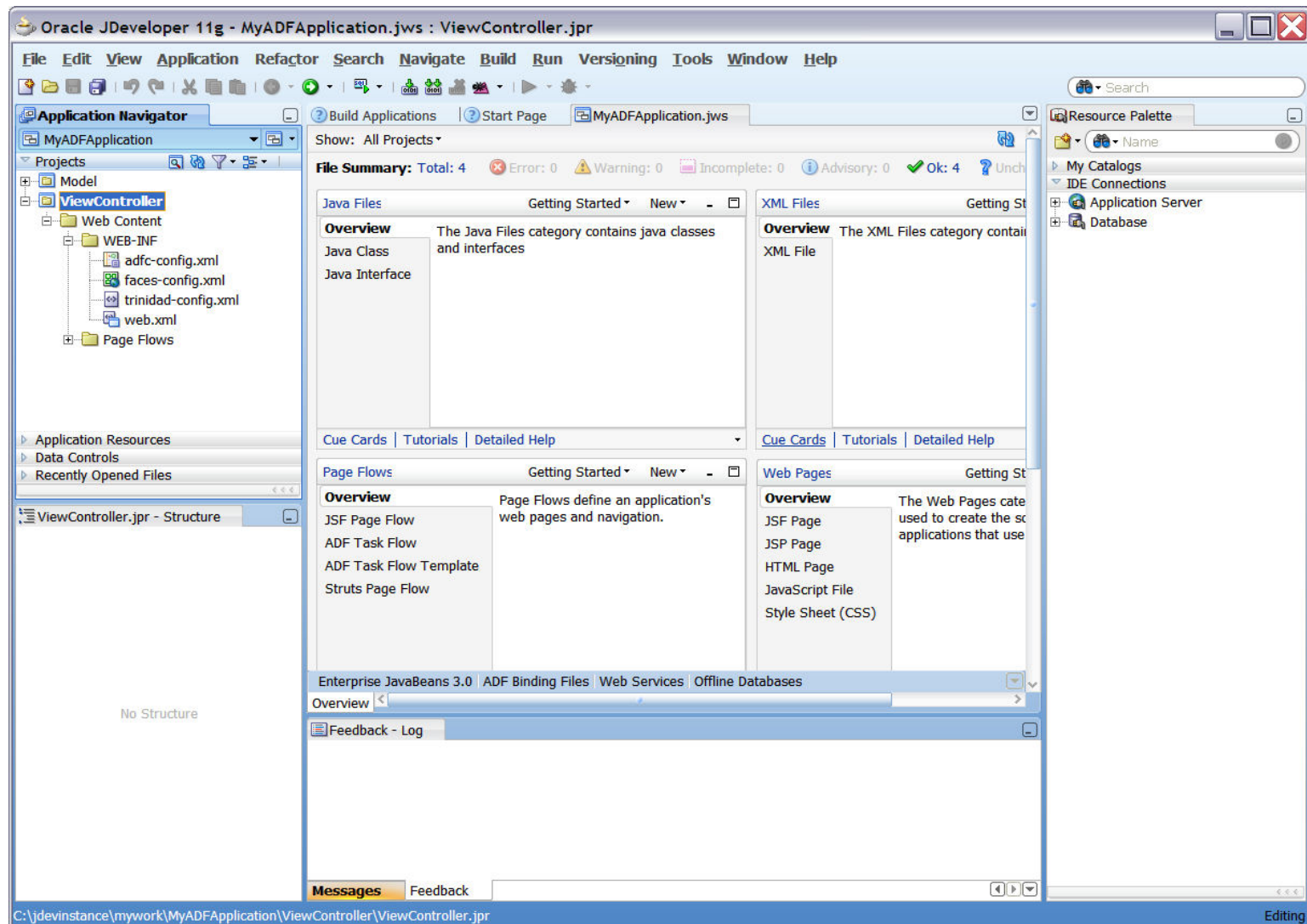


- When a JDeveloper ADF Web Application is created ADF uses the MVC (Model-View Controller) pattern
- JDeveloper creates two subordinate projects
 - Model Data and Business Rules
 - ViewController User Interface
- ADF provides the “Controller”



- Review the directory structure created to support the application and the associated projects







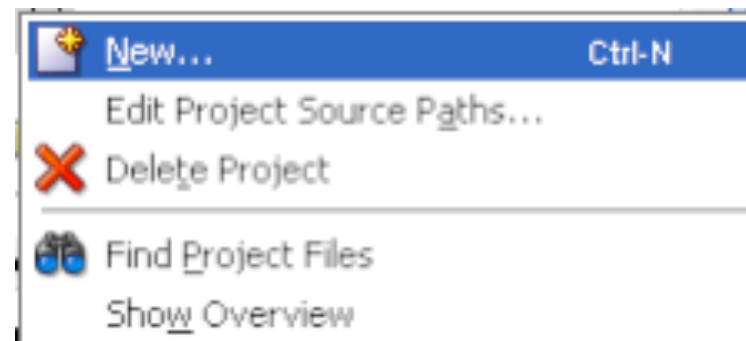
- The following pages show how to create ADF BC objects using the Wizards provided by JDeveloper
- Each object created may be created individually using JDeveloper's features or by coding them manually rather than using the Wizards
- JDeveloper's database modeling capabilities are shown to good effect by the use of Database Connections and Wizards



- The “Create Business Objects from Tables” Wizard follows a few simple steps:
 - Create Business Component, select type of Business Component to be built
 - Select Database Connection to be used (may create Database Connection via Wizard)
 - Build Entity Objects using database Tables/Views
 - Build Updateable View Objects (if desired)
 - Build Read-Only View Objects (if desired)
 - Save Application Module

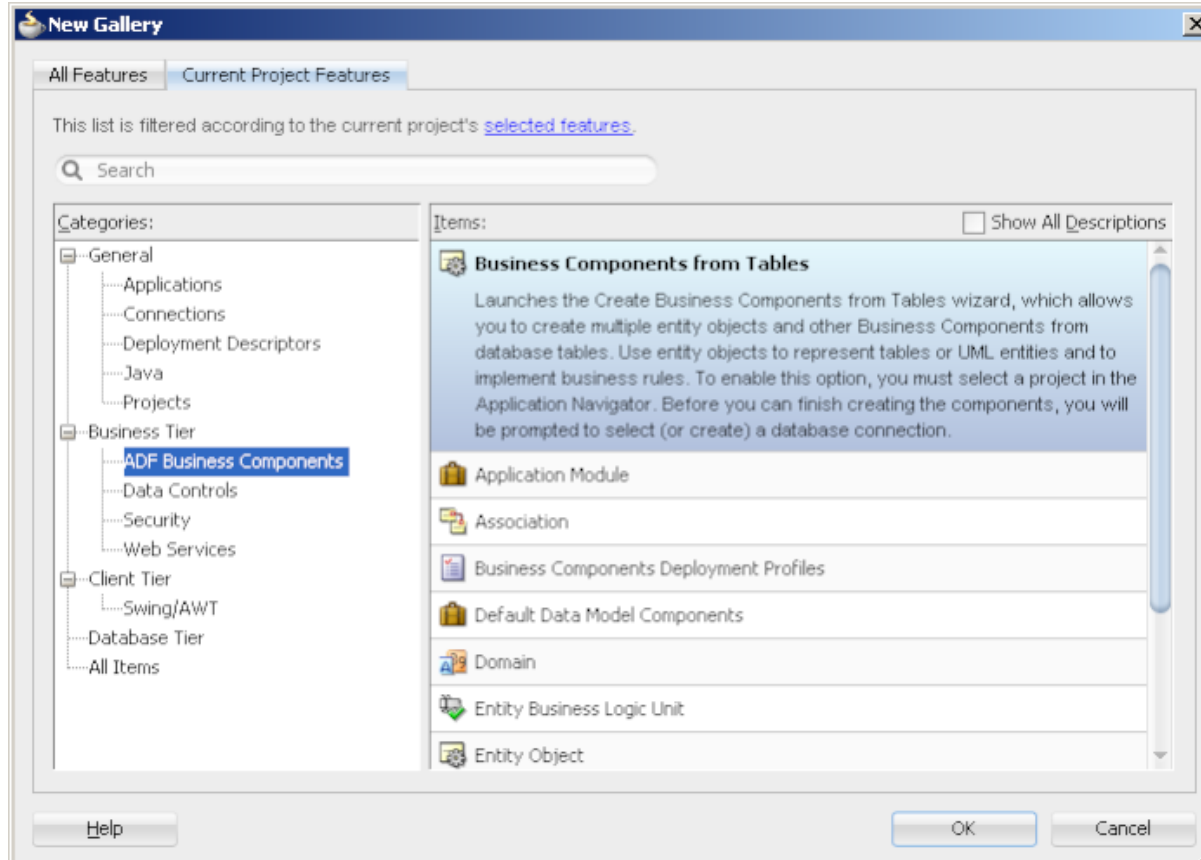


- Start building new components as follows:
- Right-click on the application's “Model” project and choose “New”





- Choose **Business Tier -> ADF Business Components -> Business Components from Tables** from the “New Gallery”








- Choose an existing Database Connection from the drop-down list or build a new one by clicking the green plus sign (Oracle client and tnsnames not required!)

Initialize Business Components Project

Select the database connection to use while developing your business components.

Connection:   

User Name: john

Driver: oracle.jdbc.OracleDriver

Connect String: jdbc:oracle:thin:@localhost:1521:orcl

Select the SQL platform and data type map for your application. Note that the data type map cannot be changed once the project is initialized.

SQL Platform:

Data Type Map:



Create Database Connection [X]

Configure a new database connection and add it to the current application (MyADFApplication).

Create Connection In: Application Resources IDE Connections

Connection Name:

Connection Type:

Username: Role:

Password: Save Password

- Oracle (JDBC) Settings -

Enter Custom JDBC URL

Driver:

Host Name: JDBC Port:

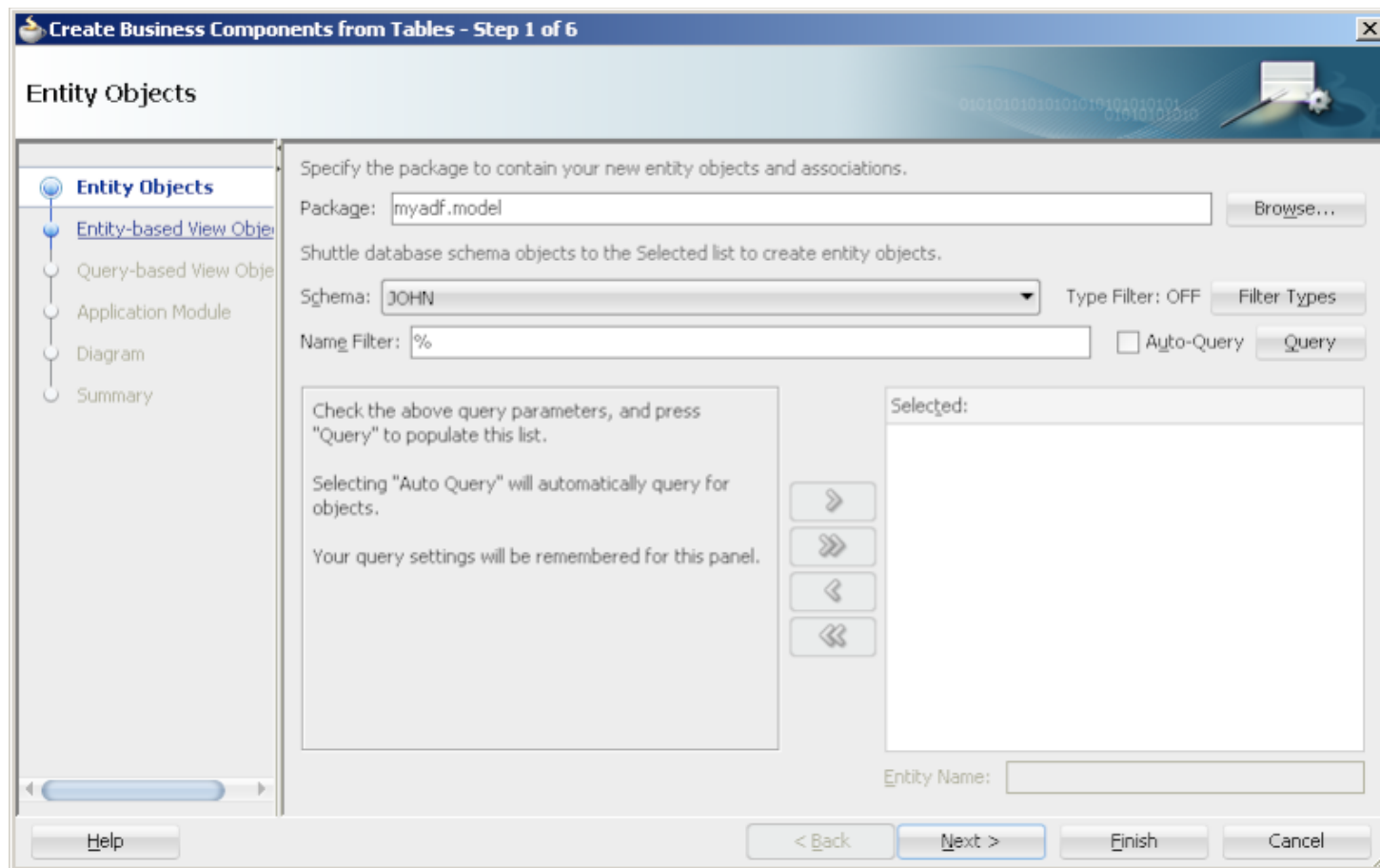
SID:

Service Name:

Success!

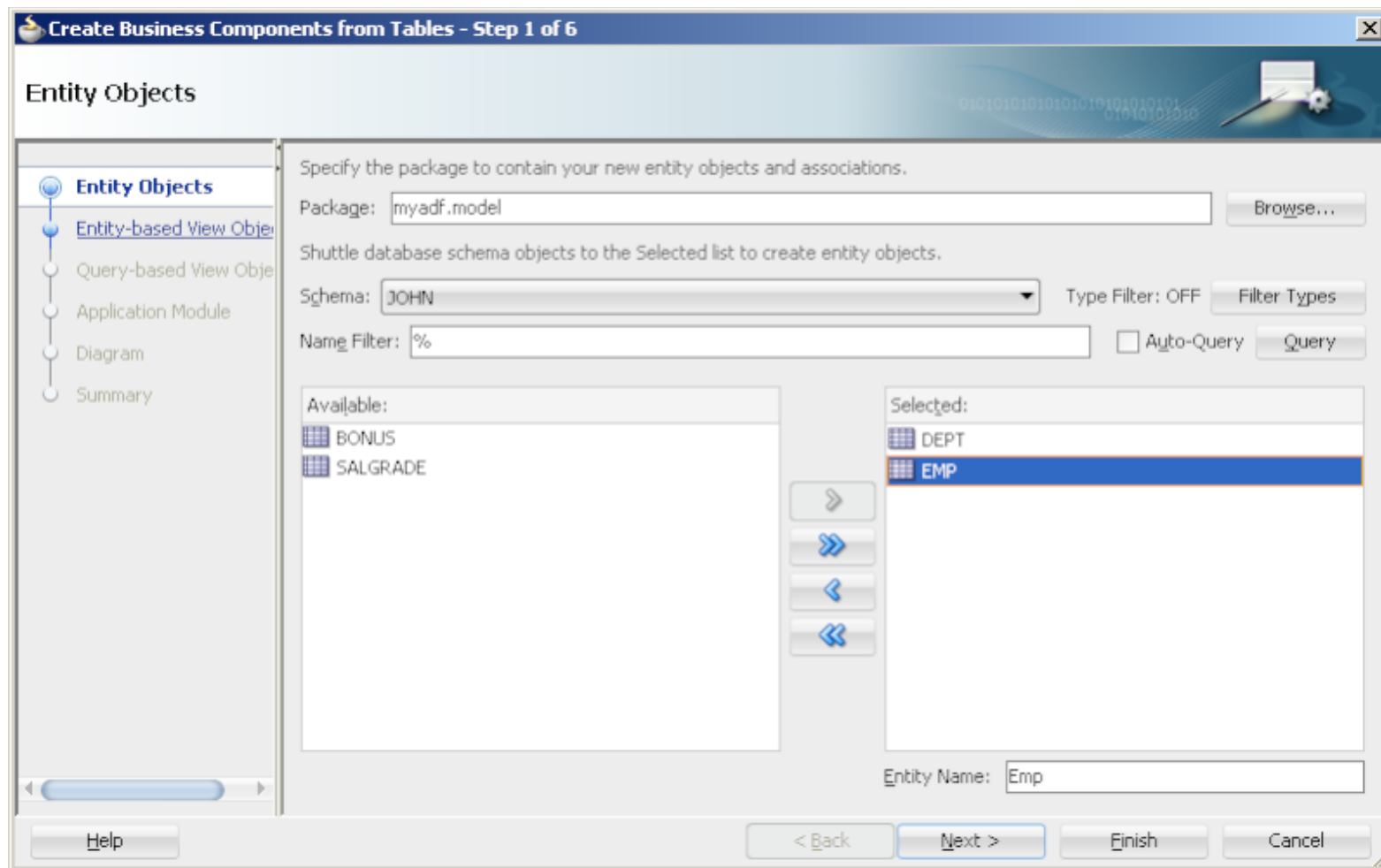


- Add, verify, or alter package name as desired; verify Schema to be used; modify filter (if desired) using SQL “LIKE” wild cards; click “Query” to view accessible database objects



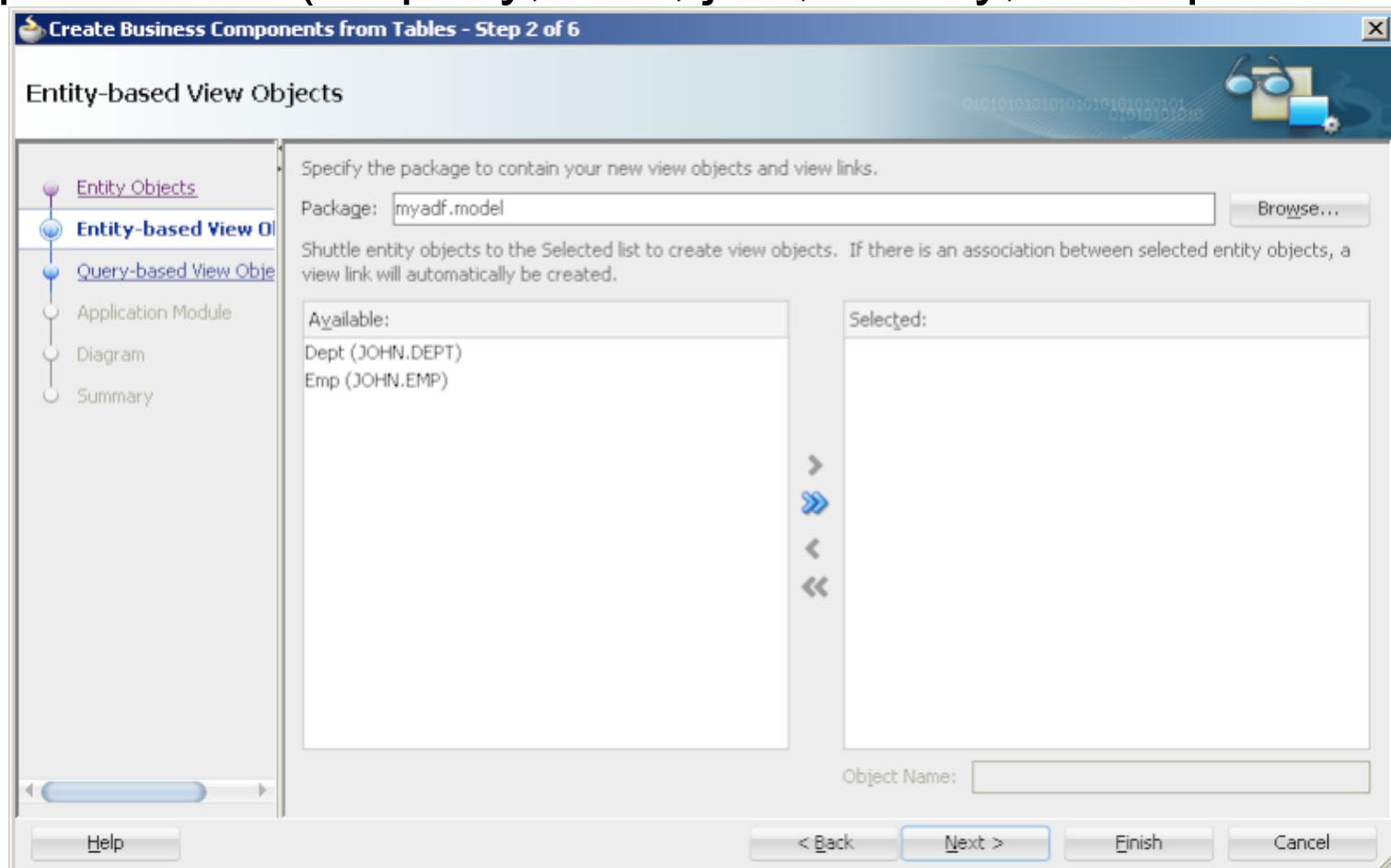


- Choose the tables and/or views to be part of the Entity Object and move them to the “Selected” side of the wizard



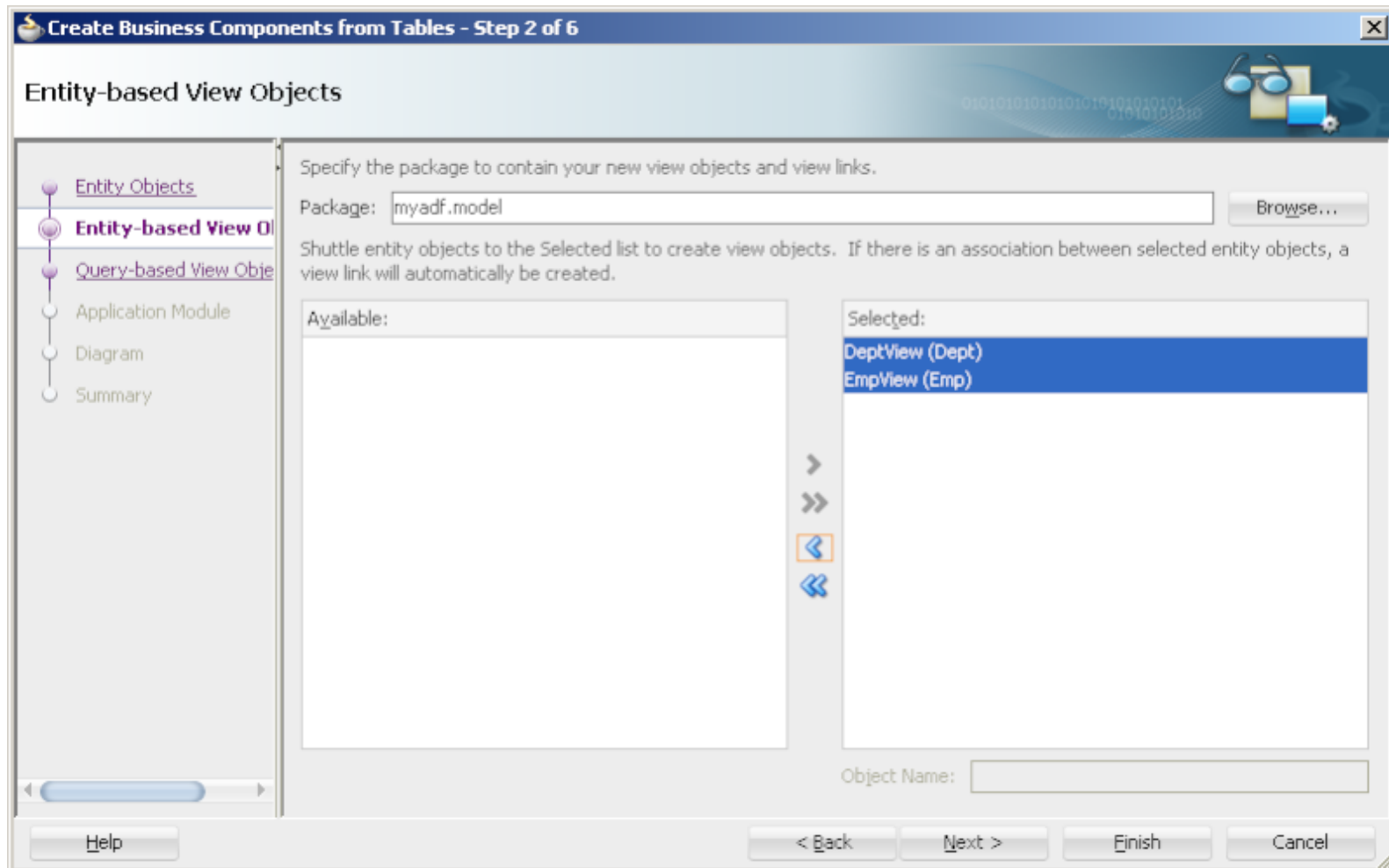


- After creating Entity Objects; the wizard offers to create Updateable Entity-based View Objects representing the output of SQL (to query, filter, join, modify, or sequence data)



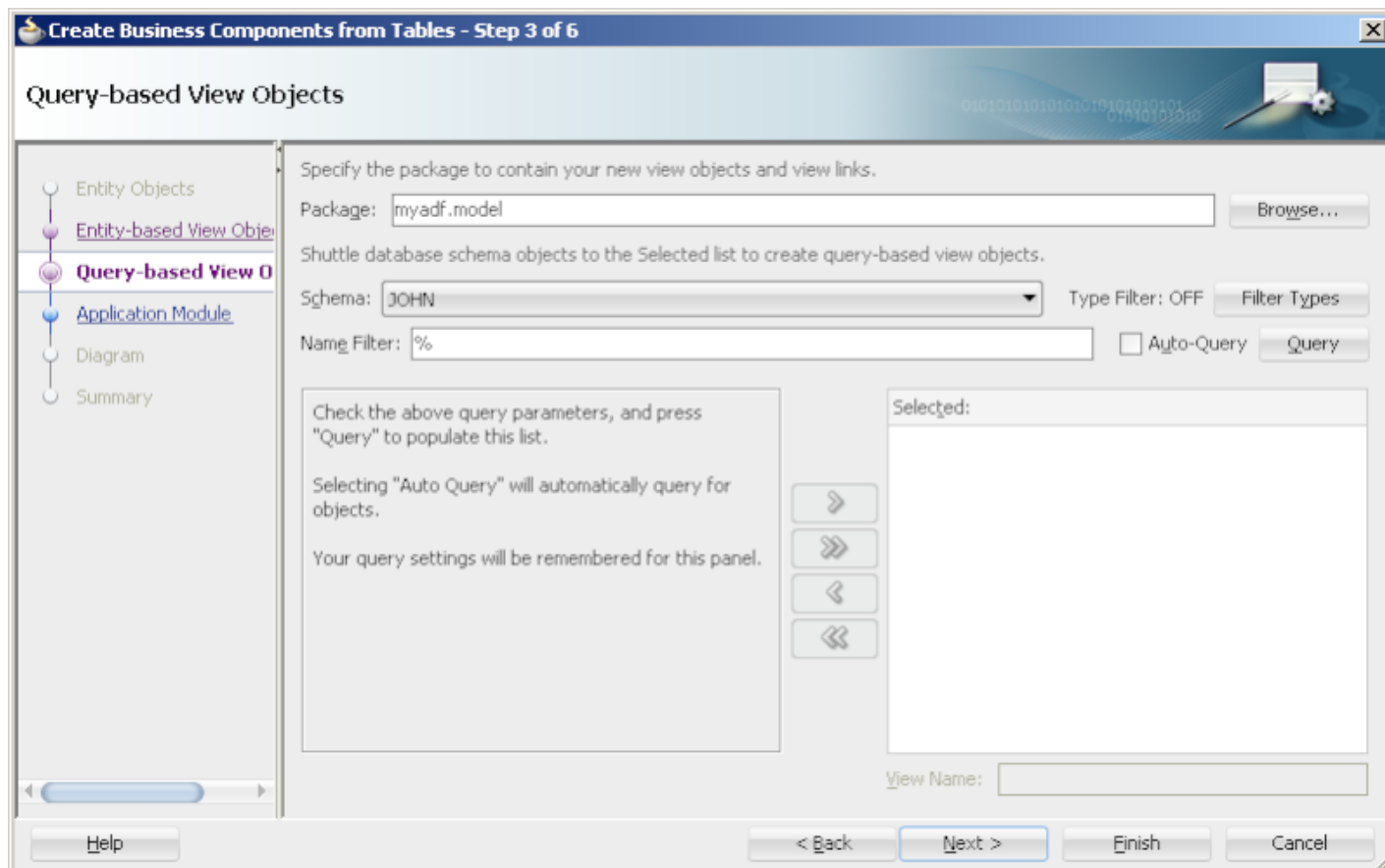


- Select Entity Objects to be used by the view being created; move them to the “Selected” side of panel



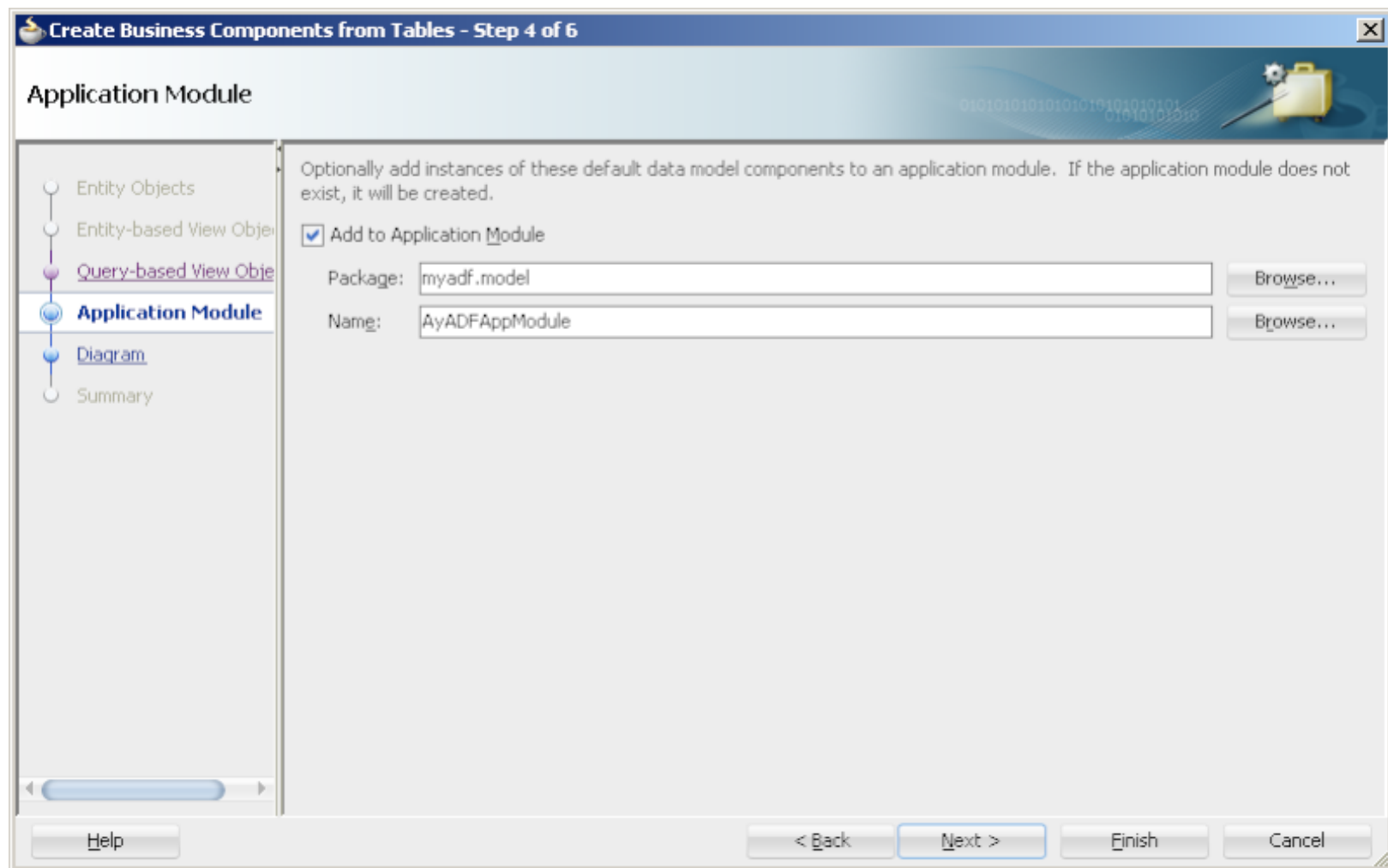


- After creating Updateable Entity-Based View Objects; the wizard goes on to create Read-Only Query-based View Objects (used for List-of-Values)



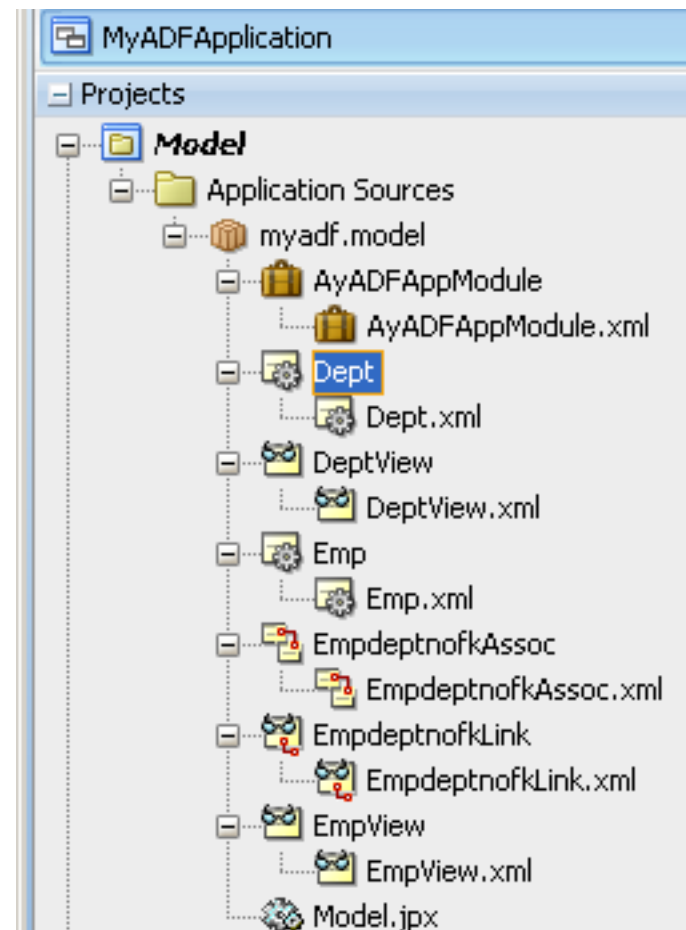
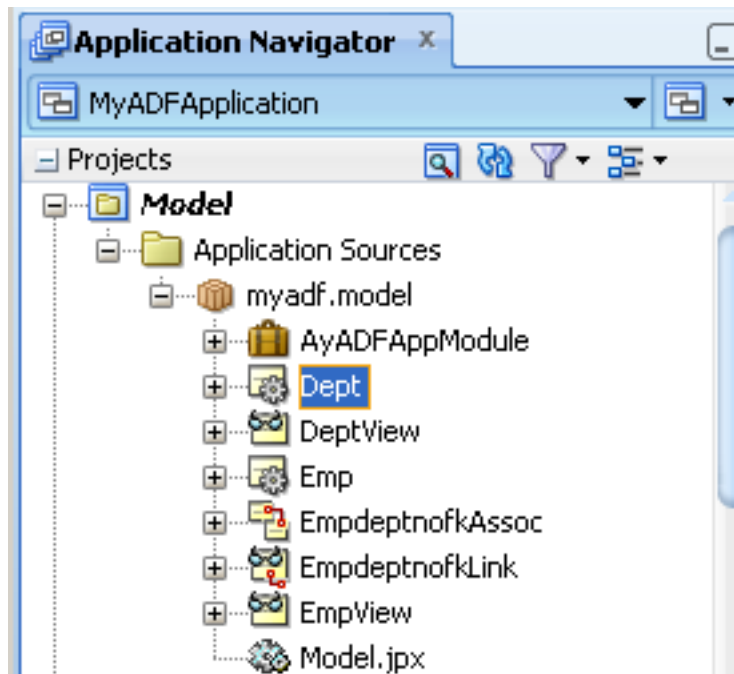


- Name the Application Module and click Finish



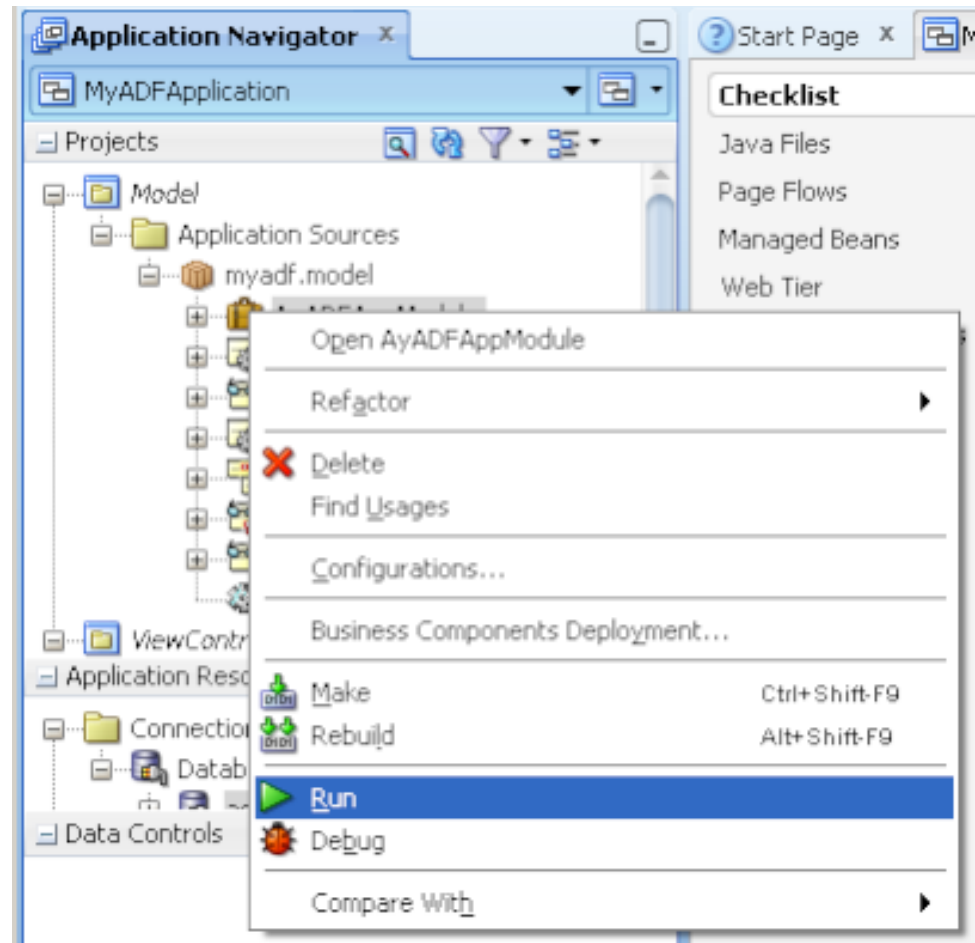


- Note the use of XML to declaratively support ADF BC



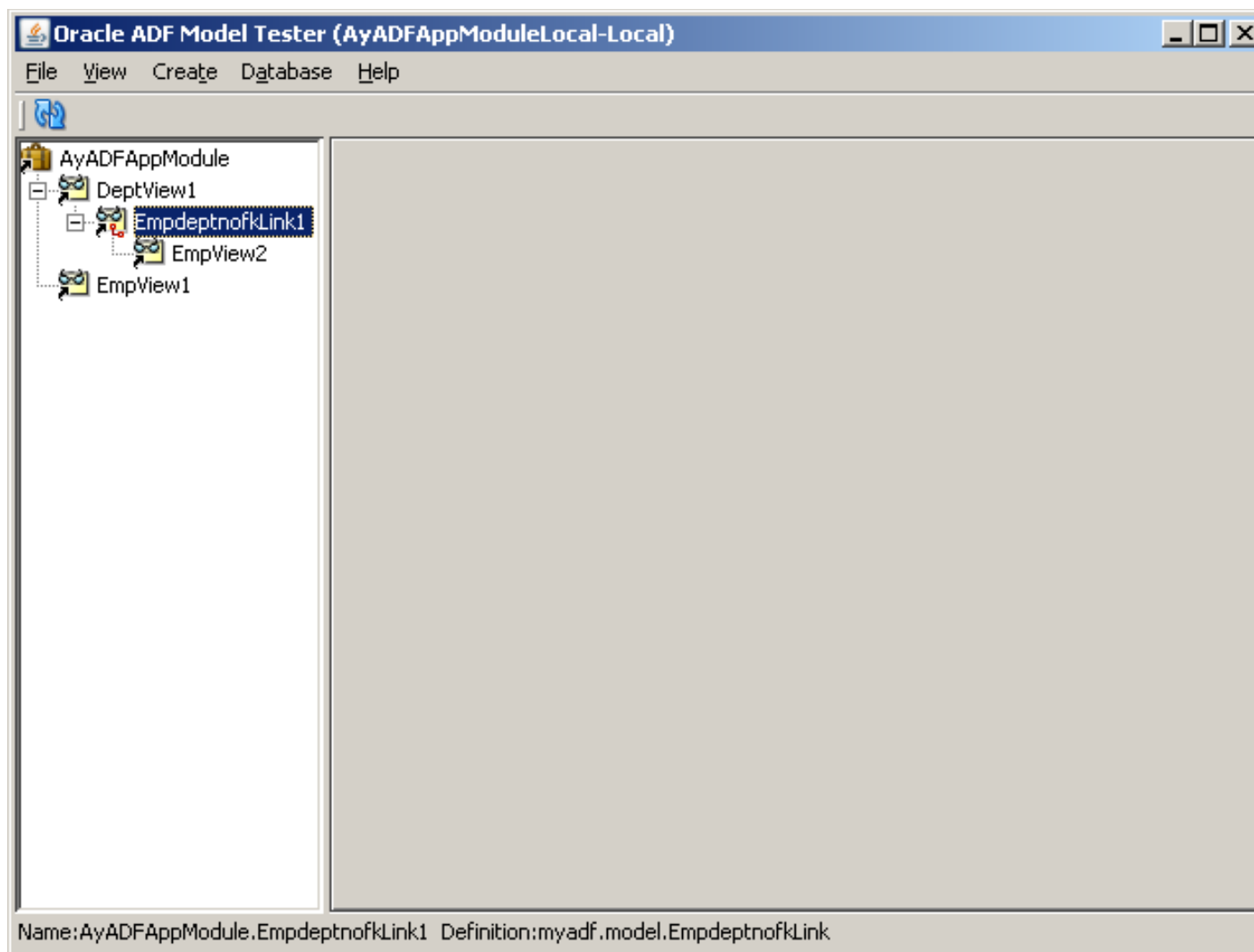


- JDeveloper provides a tool to “browse” ADF BC Application Module objects graphically; using the Application Navigator, find the Application Module to be viewed; right-click and choose “Run” to start





- Choose the Business Component to be tested





- Oracle's Business Component Browser displays data from the underlying database objects (screen should look familiar to Oracle Forms users)
- If referential keys are defined in the database (Primary Keys and Foreign Keys) the ADF BC Wizard automatically arranges the tables into a Master-Detail relationship



Oracle ADF Model Tester (AyADFAppModuleLocal-Local)

File View Create Database Help

AyADFAppModule

- DeptView1
 - EmpdeptnofkLink1**
 - EmpView2
- EmpView1

EmpdeptnofkLink1

Deptno: 10
Dname: ACCOUNTING
Loc: NEW YORK

Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
7782	CLARK	MANAGER	7839	1981-06-09 ...	2450		10
7839	KING	PRESIDENT		1981-11-17 ...	5000		10
7934	MILLER	CLERK	7782	1982-01-23 ...	1300		10

Name: AyADFAppModule.EmpdeptnofkLink1 Definition: myadf.model.EmpdeptnofkLink



- Use the “Specify View Criteria” (Binocular) icon to Search



The screenshot shows the Oracle ADF Model Tester interface. The left pane displays a tree view with 'AyADFAppModule' expanded to show 'DeptView1', 'EmpdeptnofkLink1', 'EmpView2', and 'EmpView1'. The main area shows the 'EmpdeptnofkLink1' view with search criteria: Deptno: 20, Dname: RESEARCH, and Loc: DALLAS. Below the criteria is a table of results.

Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
7369	SMITH	CLERK	7902	1980-12-17 ...	800		20
7566	JONES	MANAGER	7839	1981-04-02 ...	2975		20
7788	SCOTT	ANALYST	7566	1987-04-19 ...	3000		20
7876	ADAMS	CLERK	7788	1987-05-23 ...	1100		20
7902	FORD	ANALYST	7566	1981-12-03 ...	3000		20

Name: AyADFAppModule.EmpdeptnofkLink1 Definition: myadf.model.EmpdeptnofkLink



- Enter Search criteria and click “Find”

View Criteria Select predefined criteria, or define ad hoc criteria

Predefined criteria:

Available: Selected:

Ad hoc criteria:

Criteria

Enter an operator followed by a value:

Deptno:

Dname:

Loc:

EmpView:

Find OR>> Remove Remove All Cancel Help



Oracle ADF Model Tester (AyADFAppModuleLocal-Local)

File View Create Database Help

AyADFAppModule

- DeptView1
 - EmpdeptnofkLink1
 - EmpView2
- EmpView1

EmpdeptnofkLink1

Deptno: 20
Dname: RESEARCH
Loc: DALLAS

Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
7369	SMITH	CLERK	7902	1980-12-17 ...	800		20
7566	JONES	MANAGER	7839	1981-04-02 ...	2975		20
7788	SCOTT	ANALYST	7566	1987-04-19 ...	3000		20
7876	ADAMS	CLERK	7788	1987-05-23 ...	1100		20
7902	FORD	ANALYST	7566	1981-12-03 ...	3000		20

Name: AyADFAppModule.EmpdeptnofkLink1 Definition: myadf.model.EmpdeptnofkLink



- JDeveloper's Database Navigator allows browsing of database objects
(parts of Oracle's SQL Developer tool have been incorporated into JDeveloper)

The screenshot shows the JDeveloper interface. On the left, the Database Navigator tree view shows the following structure:

- IDE Connections
 - MyADFApplication
 - adfsample
 - Tables (Filtered)
 - BONUS
 - DEPT
 - EMP
 - EMPNO
 - ENAME
 - JOB
 - MGR

On the right, the table view for the EMP table is displayed with the following columns:

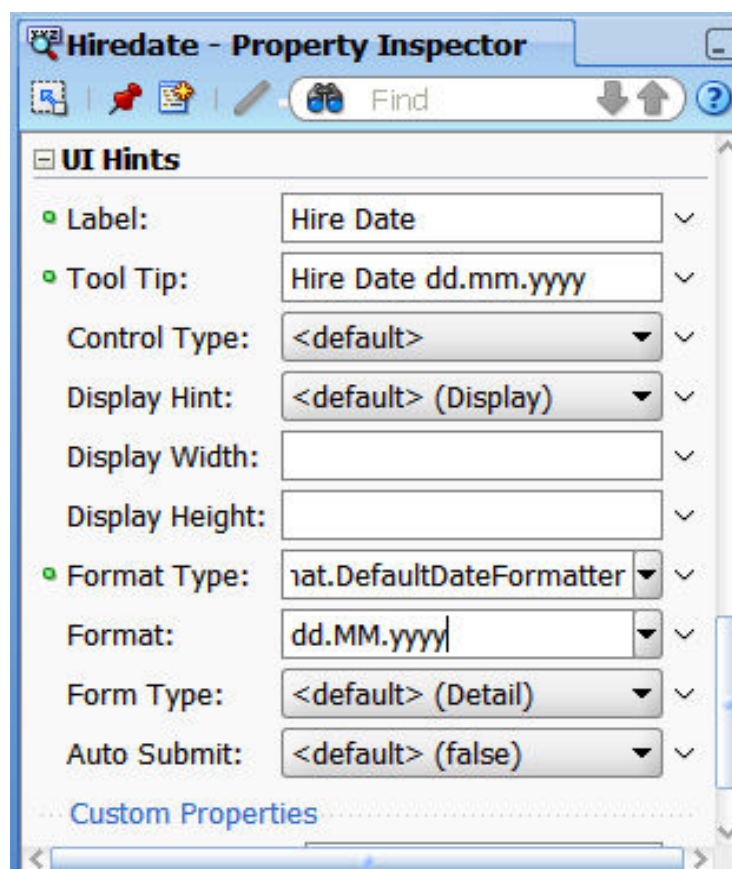
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
EMPNO	NUMBER(4,0)	No	(null)	1 (null)	
ENAME	VARCHAR2(10 BYTE)	Yes	(null)	2 (null)	
JOB	VARCHAR2(9 BYTE)	Yes	(null)	3 (null)	
MGR	NUMBER(4,0)	Yes	(null)	4 (null)	
HIREDATE	DATE	Yes	(null)	5 (null)	
SAL	NUMBER(7,2)	Yes	(null)	6 (null)	
COMM	NUMBER(7,2)	Yes	(null)	7 (null)	
DEPTNO	NUMBER(2,0)	Yes	(null)	8 (null)	



- Once the initial Business Components are created in the application, it might be useful to:
 - Set default values
 - Define formatting
 - Validate data



- Like Oracle Forms (and other 4GLs) properties are listed



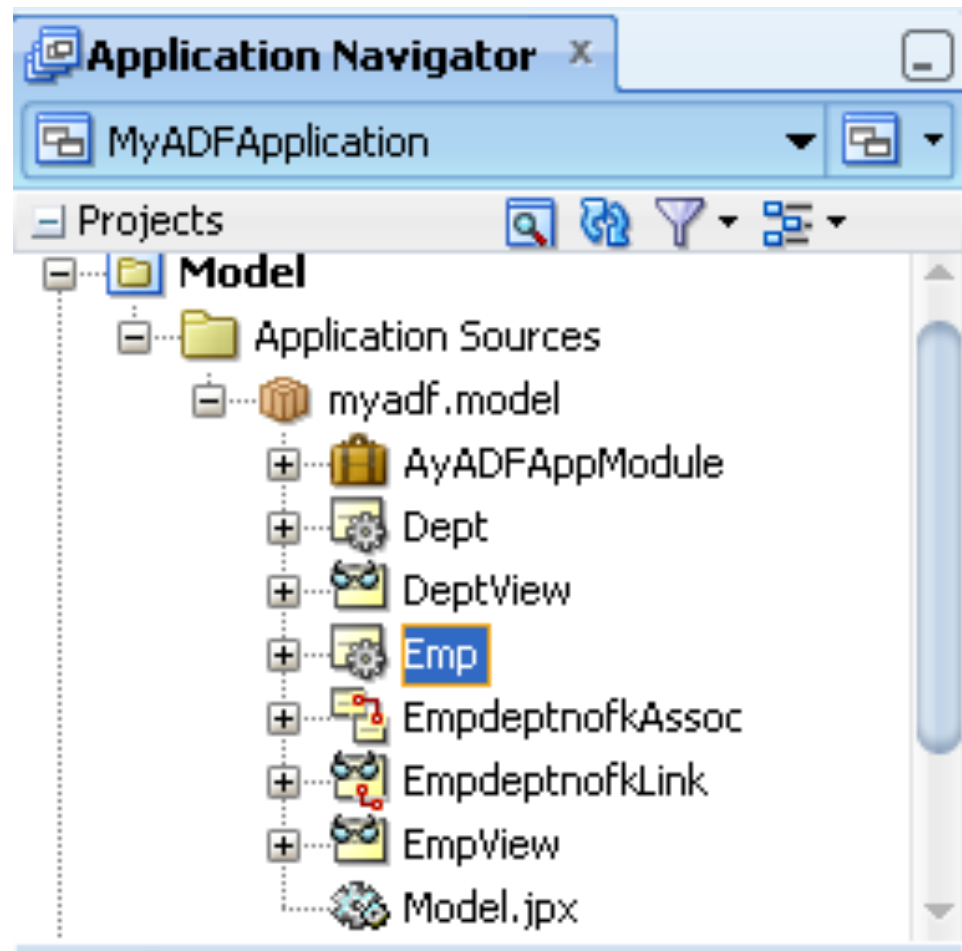


- ADF uses XML files to store declared definitions

```
70 <Attribute
71   Name="Hiredate"
72   ColumnName="HIREDATE"
73   SQLType="TIMESTAMP"
74   Type="oracle.jbo.domain.Date"
75   ColumnType="DATE"
76   TableName="EMP">
77   <TransientExpression><![CDATA[adf.currentDate]]></TransientExpression>
78   <DesignTime>
79     <Attr Name="_DisplaySize" Value="7"/>
80   </DesignTime>
81   <Properties>
82     <SchemaBasedProperties>
83       <LABEL
84         ResId="myadf.model.Emp.Hiredate_LABEL"/>
85       <TOOLTIP
86         ResId="myadf.model.Emp.Hiredate_TOOLTIP"/>
87       <FMT_FORMATTER
88         ResId="myadf.model.Emp.Hiredate_FMT_FORMATTER"/>
89       <FMT_FORMAT
90         ResId="myadf.model.Emp.Hiredate_FMT_FORMAT"/>
91     </SchemaBasedProperties>
92   </Properties>
93 </Attribute>
94 <Attribute
95   Name="Sal"
```



- Use JDeveloper to modify appearance of database column values by double-clicking an Entity Object



Entity Object Edit Panel



Oracle JDeveloper 11g Release 2 - MyADFApplication.jws : Model.jpr : C:\JDeveloper\mywork\MyADFApplication\Model\src\myadf\model\Emp.xml

File Edit View Application Refactor Search Navigate Build Run Versioning Tools Window Help

Application Navigator x Start Page x MyADFApplication Overview x Emp.xml x

Structure

- MyADFApplication
 - Projects
 - Model
 - Application Sources
 - myadf.model
 - AyADFAppModule
 - Dept
 - DeptView
 - Emp
 - EmpdeptnofkAssoc
 - EmpdeptnofkLink
 - EmpView
 - Model.jpj
 - Application Resources
 - Procedures
 - Public Database Links
 - Public Synonyms
 - Schemas
 - Sequences
 - Synonyms
 - Tables
 - BONUS
 - DEPT
 - EMP
 - SALGRADE
 - Tables Groups
 - Triggers
 - Type Bodies
 - Tunes
 - Data Controls
 - AyADFAppModuleDataControl
 - Recently Opened Files

General

Entity Object

Attributes
Business Rules
Java
Business Events
View Accessors

Entity objects are used to encapsulate the business logic and database storage details for your business entities.

Name: Emp

Package: myadf.model

Extends: <None>

Property Set: <None>

Schema Object: EMP

Business Components Project: Model

Alternate Keys + - ✕

Tuning

Custom Properties + - ✕

Security: Emp

Business Logic Groups + - ✕

Overview Diagram Source History

Running: Model.jpr - Log x

```
[100] ##### QueryCollection.fini no RowFilter
[161] ##### QueryCollection.fini no RowFilter
[162] Fail during disconnect.
Feb 7, 2012 7:32:22 PM oracle.jbo.jbotester.MainFrame exit
INFO: BC4J Tester exit code(0)
Process exited with exit code 0.
```

Messages Running: Model.jpr x

Loading myadf.model.Emp | Selected: Emp | Editing

Entity Object Attributes



The screenshot shows the Oracle JDeveloper IDE with the 'Emp.xml' file open. The 'Attributes' tab is selected, displaying a table of entity attributes. The 'Empno' attribute is highlighted as the primary key.

Name	Type	Column Name	Column Type	Extends
Empno	Integer	EMPNO	NUMBER(4, 0)	
Ename	String	ENAME	VARCHAR2(10)	
Job	String	JOB	VARCHAR2(9)	
Mgr	Integer	MGR	NUMBER(4, 0)	
Hiredate	Timestamp	HIREDATE	DATE	
Sal	BigDecimal	SAL	NUMBER(7, 2)	
Comm	BigDecimal	COMM	NUMBER(7, 2)	
Deptno	Integer	DEPTNO	NUMBER(2, 0)	

Below the table, the 'Details' tab shows the configuration for the 'Empno' attribute:

- Name: Empno
- Display Name: Empno
- Description:
- Type: Integer
- Property Set: <None>
- Default Value:
- Updatable: Always
- Persistent: Persistent, Transient
- Mandatory: Mandatory, Refresh on Insert
- Primary Key: Primary Key, Refresh on Update
- Queryable: Queryable, Change Indicator
- Precision Rule: Precision Rule

The bottom of the window shows a 'Messages' pane with the following log output:

```
[100] ##### QueryCollection.fini no RowFilter
[161] ##### QueryCollection.fini no RowFilter
[162] Fail during disconnect.
Feb 7, 2012 7:32:22 PM oracle.jbo.jbotester.MainFrame exit
INFO: BC4J Tester exit code(0)
Process exited with exit code 0.
```



The screenshot displays the Oracle JDeveloper 11g Release 2 IDE. The main window shows the Business Rules editor for the entity 'Emp'. The Business Rules tree on the right lists various attributes and their database constraints:

- Emp
 - Entity Validators
 - Attributes
 - Empno: Database Constraint - Mandatory, Database Constraint - Precision : (4,0)
 - Ename: Database Constraint - Precision : (10)
 - Job: Database Constraint - Precision : (9)
 - Mgr: Database Constraint - Precision : (4,0)
 - Hiredate
 - Sal: Database Constraint - Precision : (7,2)
 - Comm: Database Constraint - Precision : (7,2)
 - Deptno: Database Constraint - Precision : (2,0)

The Messages window at the bottom shows the following log output:

```
[160] ##### QueryCollection.finl no RowFilter
[161] ##### QueryCollection.finl no RowFilter
[162] Fail during disconnect.
Feb 7, 2012 7:32:22 PM oracle.jbo.jbotester.MainFrame exit
INFO: BC4J Tester exit code(0)
Process exited with exit code 0.
```



- Validations and Business Logic may be added including:
 - Client-side validation
 - Format masks
 - Default Values
 - Declarative Range (and other) Validation
 - CSS (Visual Attributes)
 - List of Values
 - Calculated field
 - Code Validation
 - Extensible for complex application validation
 - Transactional Triggers



Add Validation Rule for: Sal

Define the Validation you want to perform with this rule and configure the Validation Failure response.

Rule Type: Range

Rule Definition Validation Execution Failure Handling

Attribute: Sal

Operator: Between

Range

Minimum Value: 500

Maximum Value: 6000

Hint: Enter valid values for the selected attribute's type.

Help OK Cancel

Validation Error Messages



Add Validation Rule for: Sal

Define the Validation you want to perform with this rule and configure the Validation Failure response.

Rule Type: Range

Rule Definition Validation Execution **Failure Handling**

Validation Failure Severity Error Informational Warning

Failure Message

Enter text for the translatable validation failure messages.

Message Text:

Salary should be between 500 and 6000

Token Message Expressions:

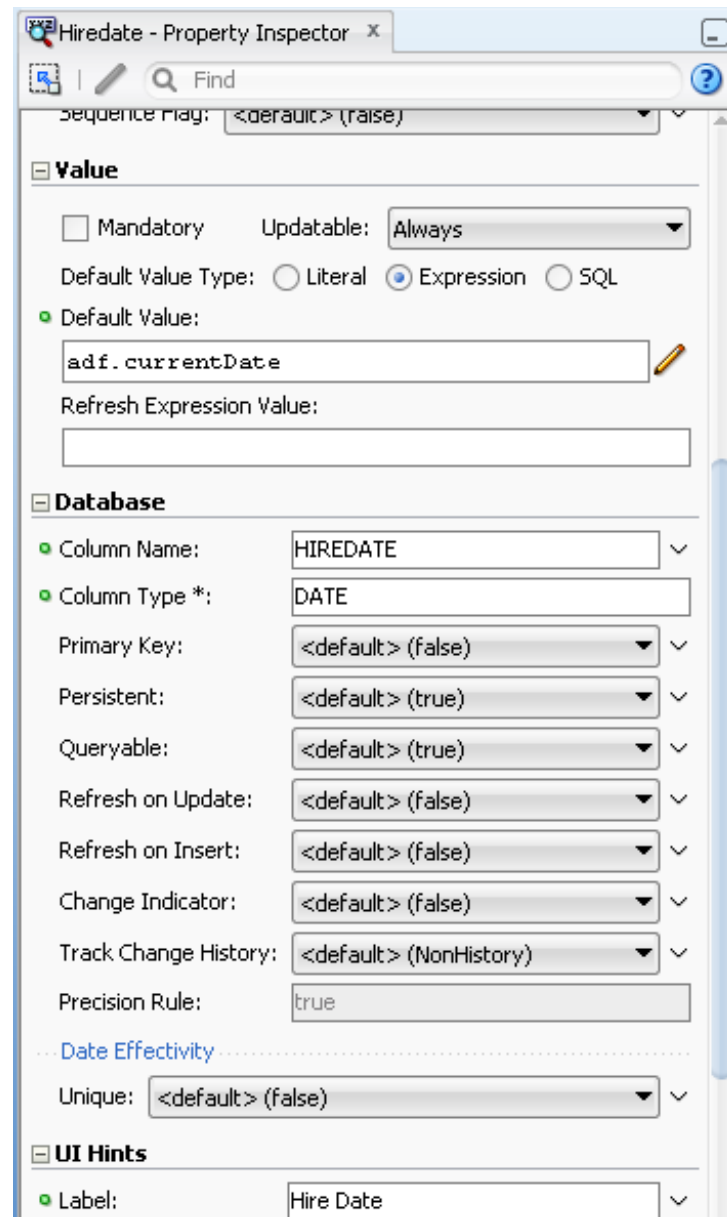
Message Token	Expression

Hint: to add a token message expression, include a token of the form {0} in your error message string.

Help OK Cancel



- Using the Property Inspector, open the “Value” properties and set the default value (in this case “adf.currentDate” using ADF’s “Groovy” support)





- Use an Attribute's Property Palette “UI Hints” section to control formatting, label, tool tip, etc... (note this formatting uses Java SimpleDateFormat options)

The screenshot shows the 'Hiredate - Property Inspector' window. The 'UI Hints' section is expanded, displaying the following properties:

Property	Value
Label:	Hire Date
Display Name (Plural):	Hire Dates
Tool Tip:	Hire Date yyyy-MM-dd
Control Type:	<default>
Display Hint:	<default> (Display)
Display Width:	
Display Height:	
Format Type:	Simple Date
Format:	yyyy-MM-dd
Form Type:	<default> (Detail)
Auto Submit:	<default> (false)



```
<Attribute
  Name="Hiredate"
  ColumnName="HIREDATE"
  SQLType="DATE"
  Type="oracle.jbo.domain.Timestamp"
  ColumnType="DATE"
  TableName="EMP"
  DefaultValue="adf.currentDate">
  <Properties>
    <SchemaBasedProperties>
      <LABEL
        ResId="myadf.model.Emp.Hiredate_LABEL"/>
      <LABEL_PLURAL
        ResId="myadf.model.Emp.Hiredate_LABEL_PLURAL"/>
      <TOOLTIP
        ResId="myadf.model.Emp.Hiredate_TOOLTIP"/>
      <FMT_FORMATTER
        ResId="myadf.model.Emp.Hiredate_FMT_FORMATTER"/>
      <FMT_FORMAT
        ResId="myadf.model.Emp.Hiredate_FMT_FORMAT"/>
    </SchemaBasedProperties>
  </Properties>
</Attribute>
```




```
ModelBundle.properties
1 #
2 myadf.model.Emp.Sal_Rule_0=Salary should be between 500 and 6000
3 myadf.model.Emp.Hiredate_LABEL=Hire Date
4 myadf.model.Emp.Hiredate_TOOLTIP=Hire Date dd.mm.yyyy
5 myadf.model.Emp.Hiredate_FMT_FORMATTER=oracle.jbo.format.DefaultDateFormat
6 myadf.model.Emp.Hiredate_FMT_FORMAT=dd.MM.yyyy
```



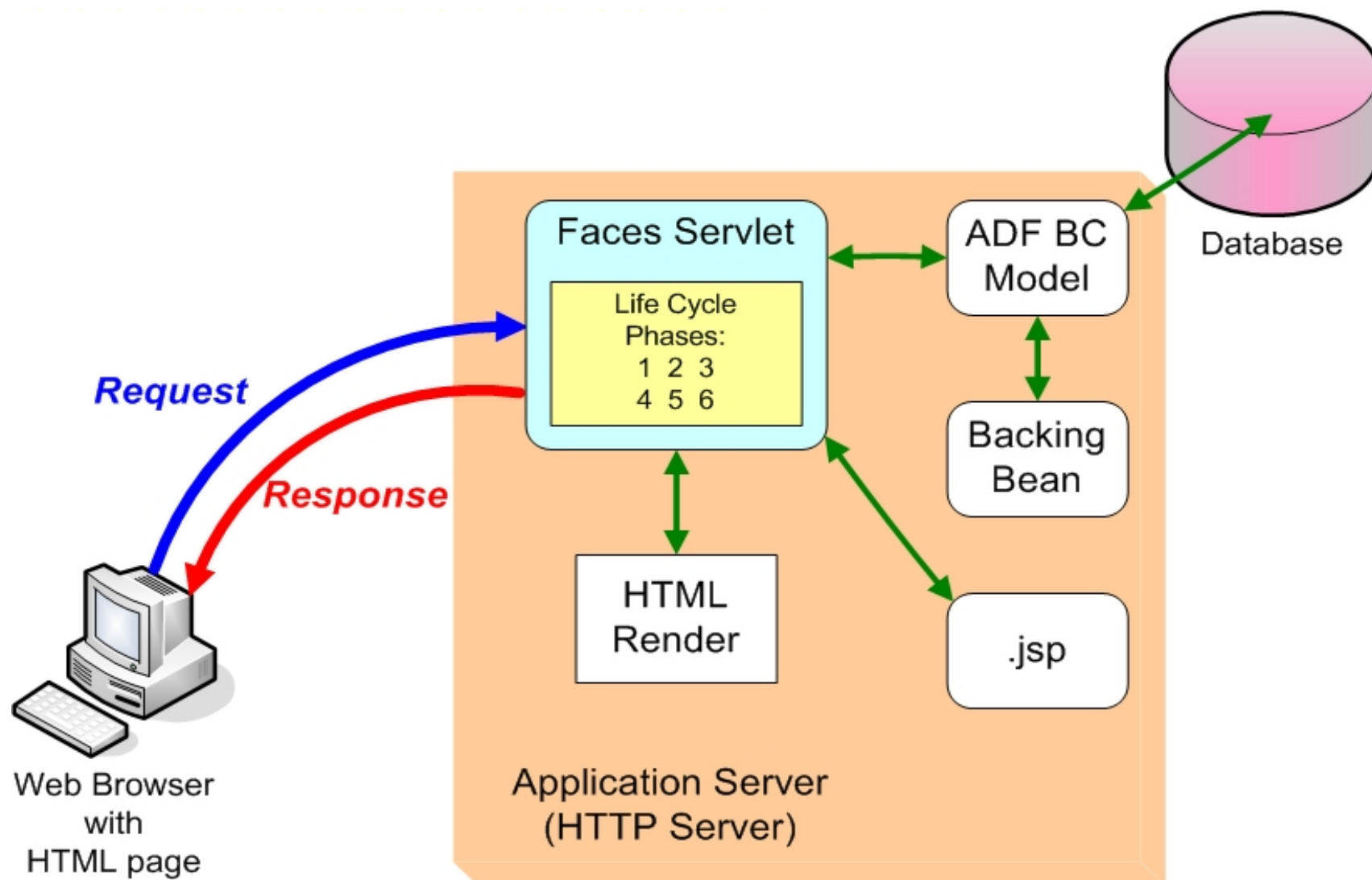
- In Oracle Forms we defined “data blocks” that represented tables and views that would be used in our forms
- ADF BC components do that and more, plus they may be shared by many applications
- In Oracle Forms once the “data block” was created we would then use it to create the presentation
- With ADF we use ADF Faces to accomplish the same thing and more
(again creating components that may be reused by other applications)



- Most 4GLs offer some type of “Data Object” or “Data Access Object” capability
 - Usually include wizard-based development
 - Usually work with relational database; do not usually support procedure-based data
 - Sometimes provide ability to find and link data objects using database dictionary
 - Sometimes provide stand-alone reusable data objects
 - Sometimes linked to GUI development via “drag-and-drop” capability



- Oracle's Business Component Browser is impressive, but hardly a customer-facing interface
- ADF Faces extends the Java Server Faces (JSF) framework using XML tags to describe the UI
- ADF Faces provides a Rich-Client Interface that uses JavaScript and AJAX components; users must have a reasonably up-to-date browser (Internet Explorer 7.0 or higher, Mozilla Firefox 2.0 or higher, Safari 3.0 or higher) to use all of its features
- ADF Faces is designed to make creation of "rich-client" (RC) interfaces full-featured and declarative where possible





- Even though the ultimate page delivered to the Client Browser is HTML; with JDeveloper's Visual Editor and the combination of ADF Faces and JSF Faces it uses to create .jspx pages there is little need for ADF Developers to code HTML or CSS



- Yield to JDeveloper's declarative mechanism and refrain from coding



- The ADF Controller extends the standard JSF controller and controls the MVC in ADF
- ADF Controller features include:
 - Sequence of page displays (may be conditional)
 - Allows partial-page processing in the same way as full page processing; only the necessary part of a page is rendered, the rest is unchanged (makes page processing faster)
 - Allows reuse of page parts
 - Provides conditional control of page flow



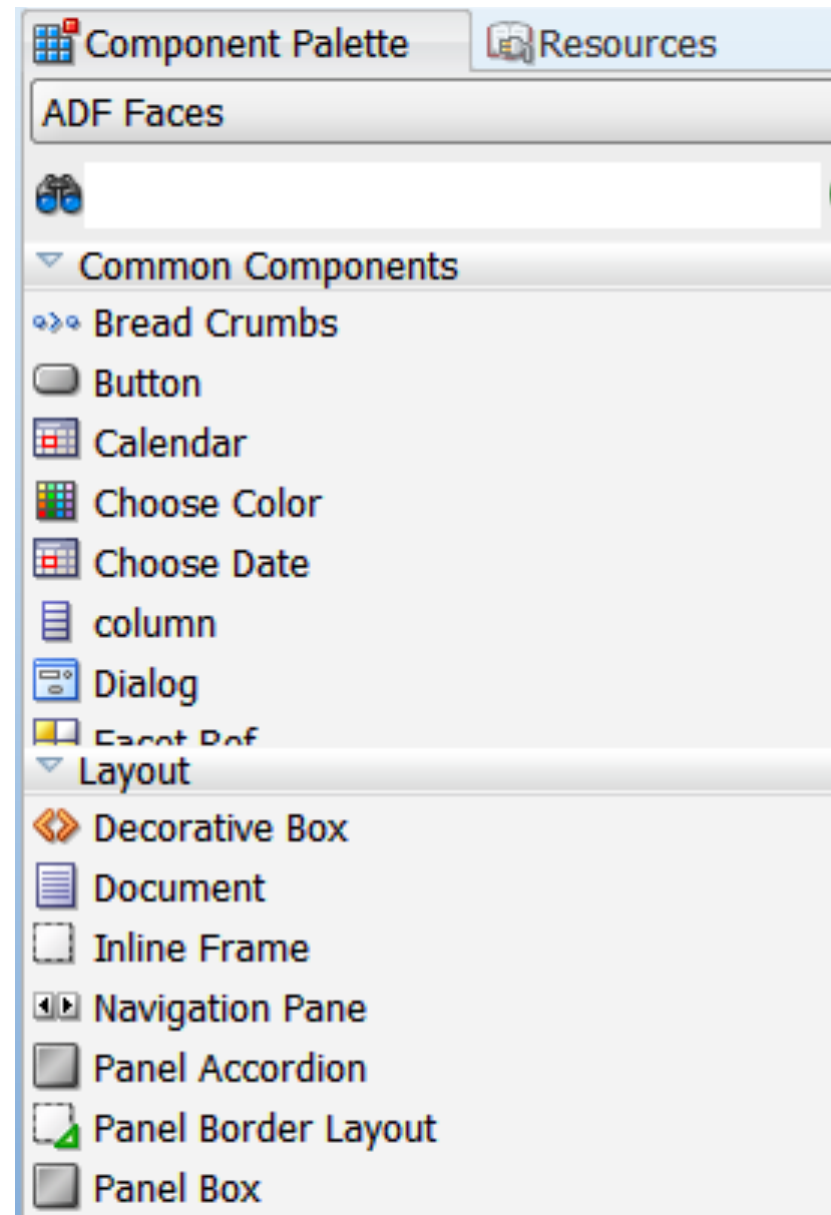
- JSF (and ADF Faces) follows a predictable cycle:
 1. Restore Components
 2. Apply Request Values
 3. Process Validations
 4. Update Model Values
 5. Invoke Application
 6. Render Response
- This Life Cycle is normally transparent; however, it is useful to understand it when debugging



- JDeveloper's Visual Designer may be used to “paint” a User Interface using the Component Palette
- The JDeveloper Visual Designer is intended to be WYSIWYG (What You See Is What You Get); however the nature of the web and HTML is that it's really WYSIKOWYG (What You See Is Kind-Of What You Get; thank you Peter Koletzke...)

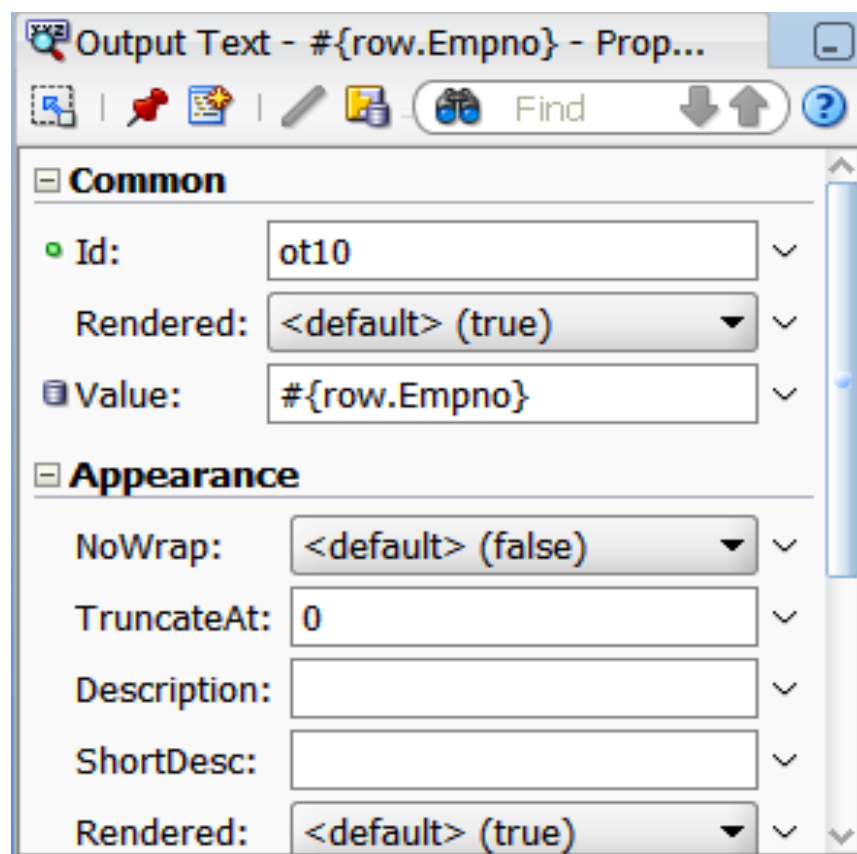


- The ADF Faces Component Palette includes icons representing various User Interface objects
- Drag-and-drop desired components into the position desired



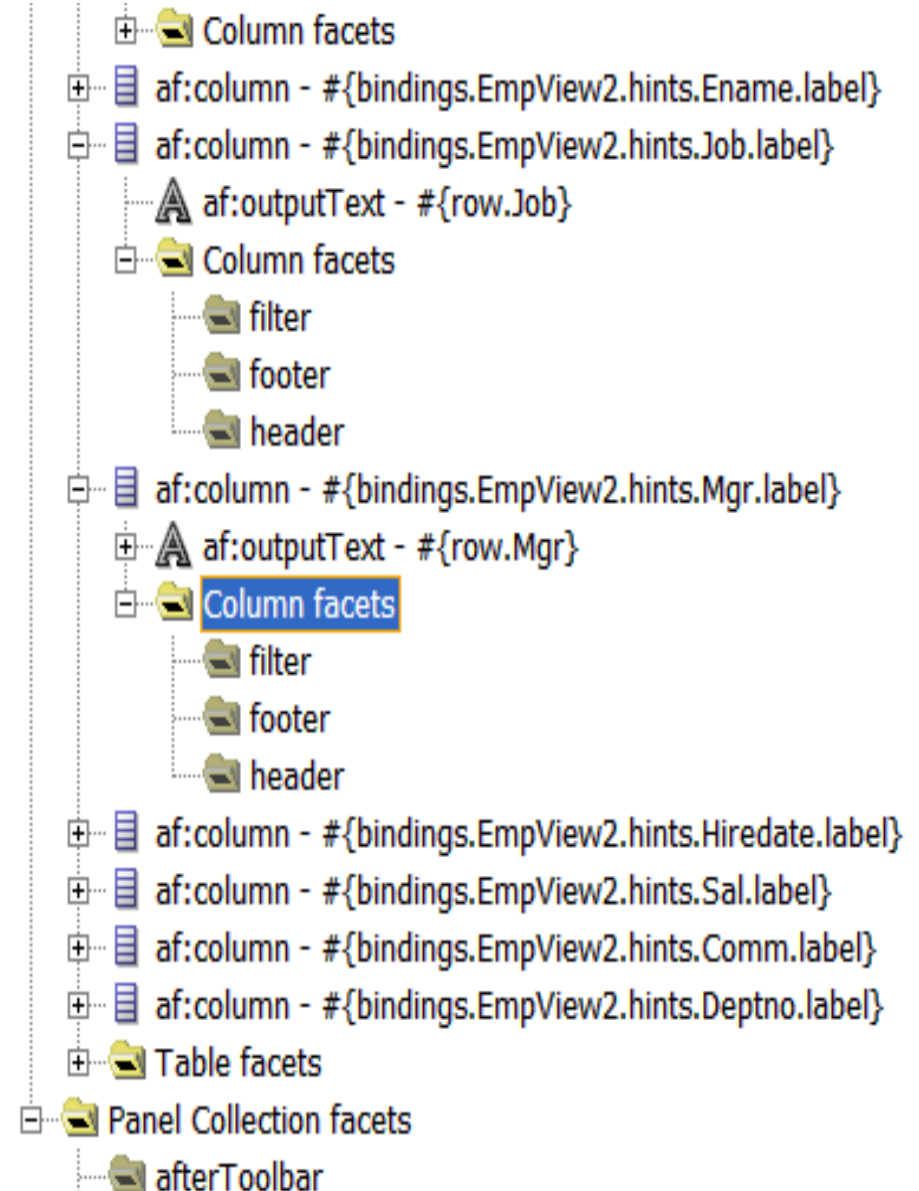


- When editing Web Pages, the Property Inspector shows properties for the various “facets” and components displayed upon the page





- The “facets” are components that are used to contain groups of other components
- JDeveloper’s “Structure” Window lists facets in the current page





- Pages in ADF are sometimes divided by Panels; pre-existing templates exist to help create the number of desired Panels
- Each Panel in turn may be divided into smaller areas using a Panel Splitter
 - By default Panel Splitters split an area horizontally
 - Panel Splitters have an “Orientation” property that allow the split to be vertical



- ADF provides many “container” objects specifically designed to hold data objects including:
 - Panel Collections are facets that contain other objects
 - Panel Accordions are facets that contain other objects but shrink-and-grow depending upon mouse movement
 - Tabbed Panels are facets that allow components to be placed into a tabbed structure



- UI Components provided by ADF Faces include:
 - Buttons
 - Calendars
 - Choose Color
 - Forms
 - Input Text
 - Output Text
 - Panel Collection
 - Submit
 - Tables
 - more...



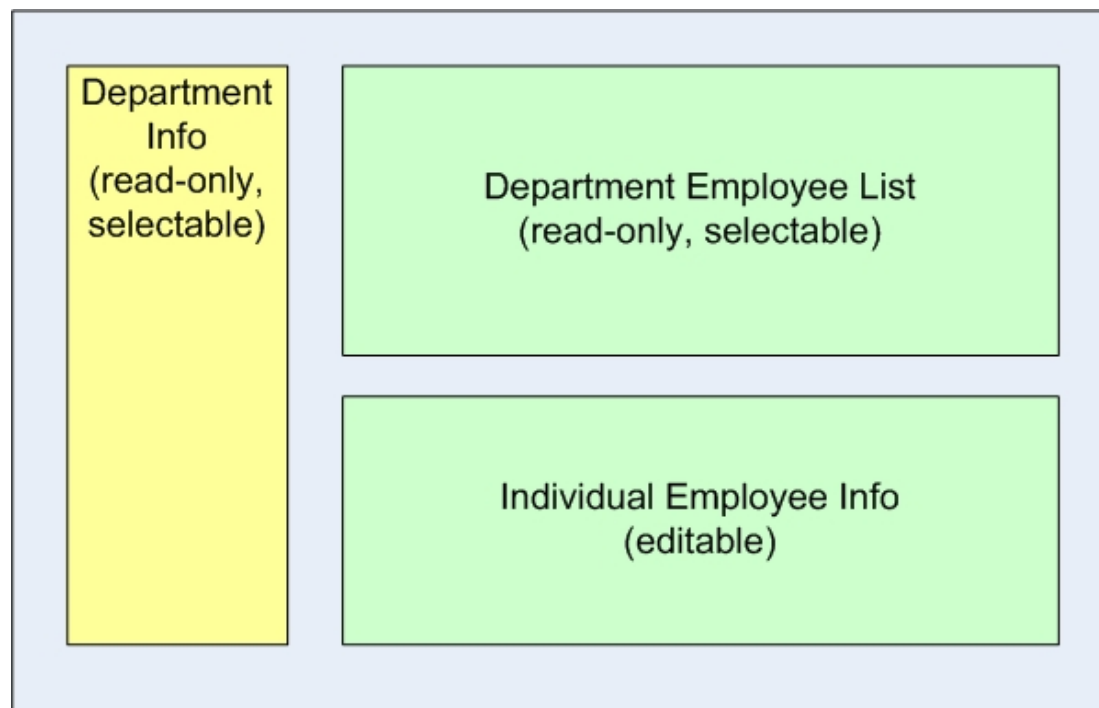
- JDeveloper's interface allows the creation of web components using drag-and-drop processing
 - Drag-and-drop may also be used to associate View Objects with UI Components
 - This has the effect of “binding” the data to the data control object



- The following pages walk through the creation of a simple Web Application using ADF Faces and ADF BC objects as follows:
 1. Design Web Page
 2. Create new JSF Page using JDeveloper
 3. Add Visual Components to JSF Page
 4. Bind Visual Components to ADF BC Objects

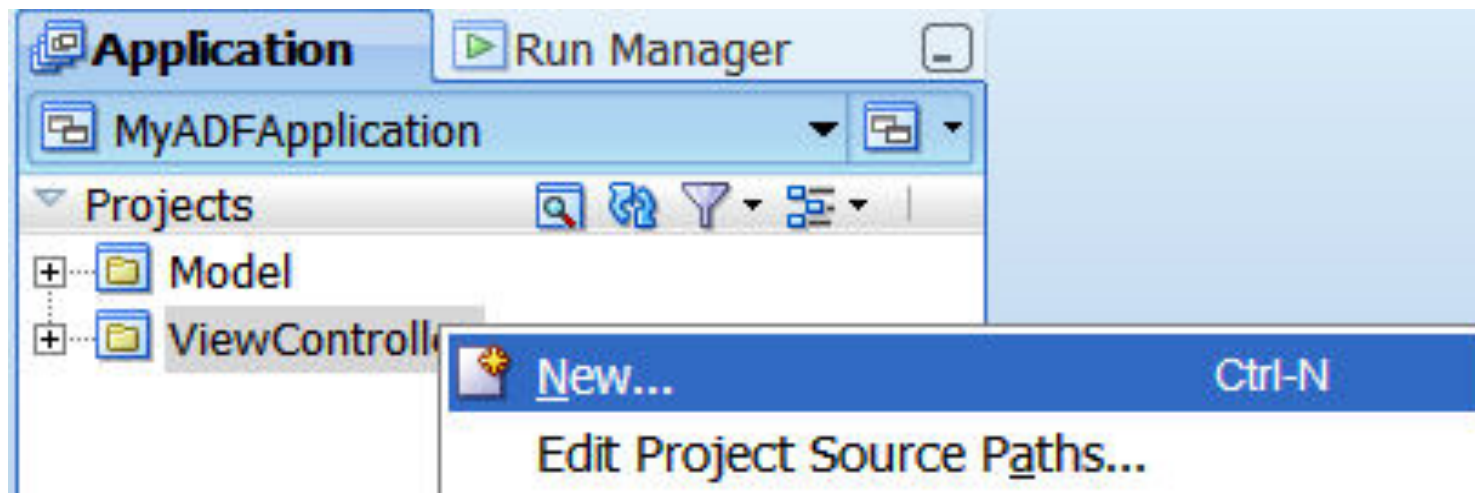


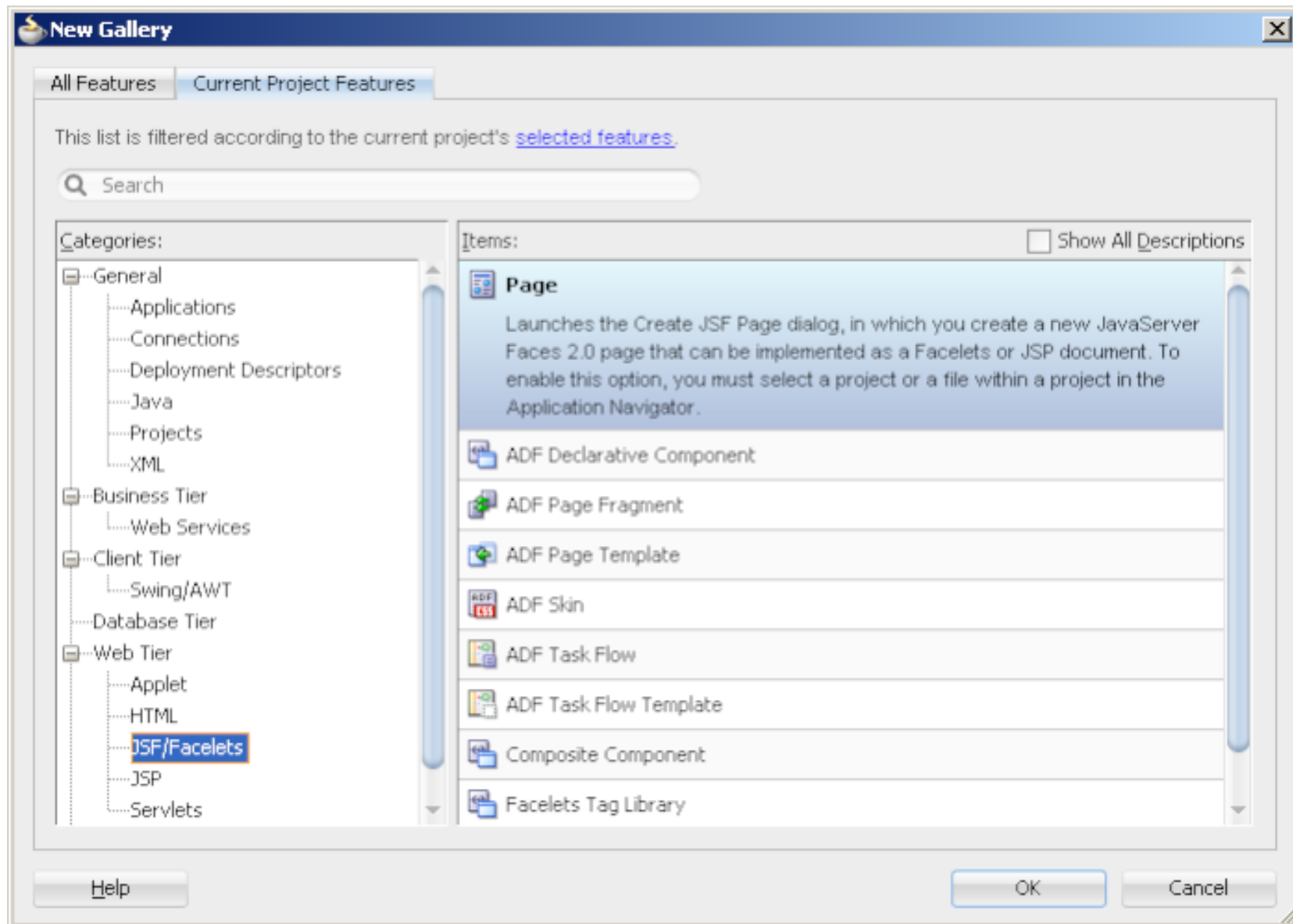
- Rough design: Department info on the left, list of Department Employees (for selected department) in the upper-right, and the information for a single employee on the lower-right (selected from list)





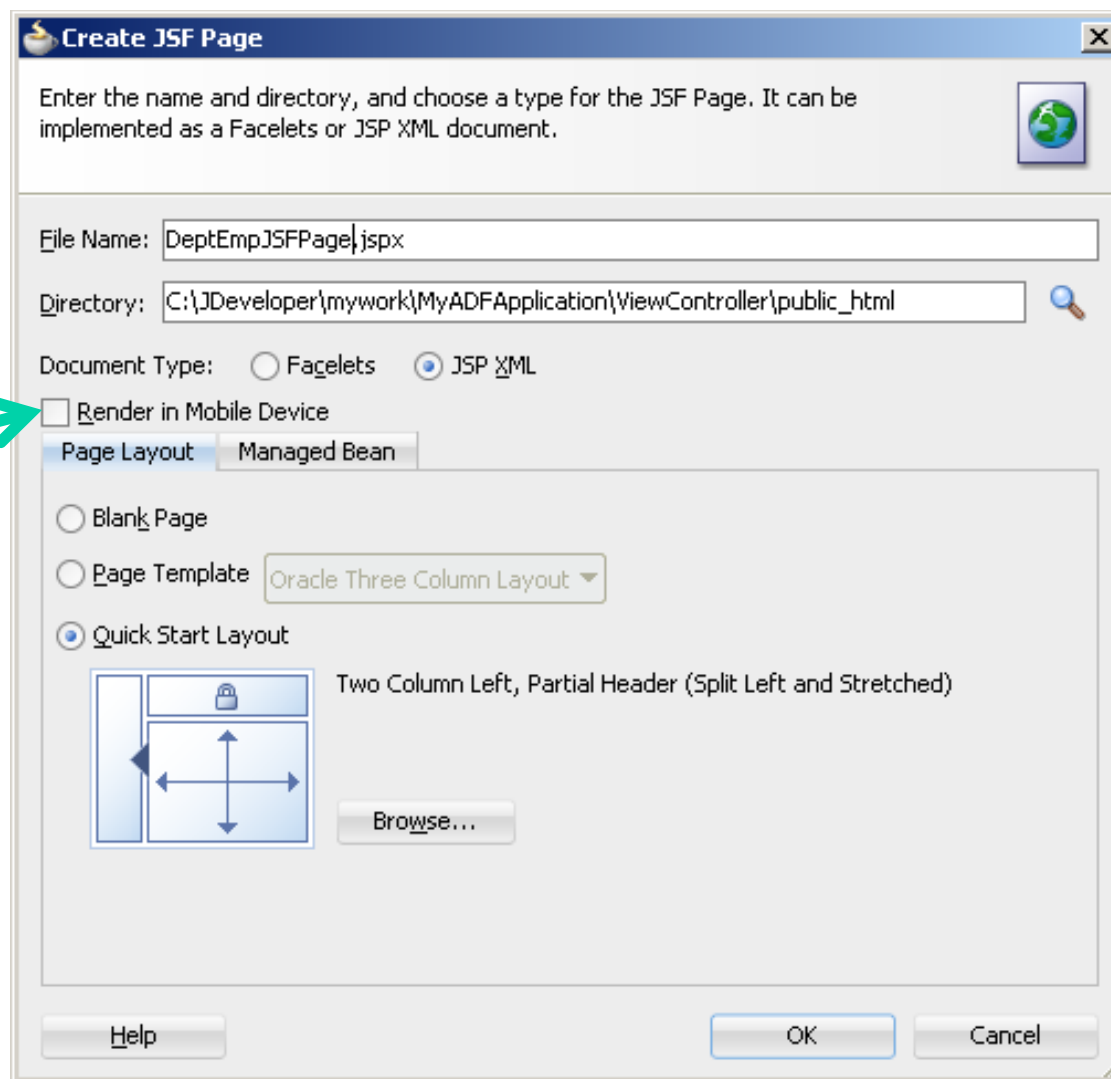
- To create an ADF Faces page, right-click on an Application's ViewController Project and choose “New” to display the “New Gallery” dialog







Not used in demo but pretty cool!



– Note the “JSP XML” Document Type

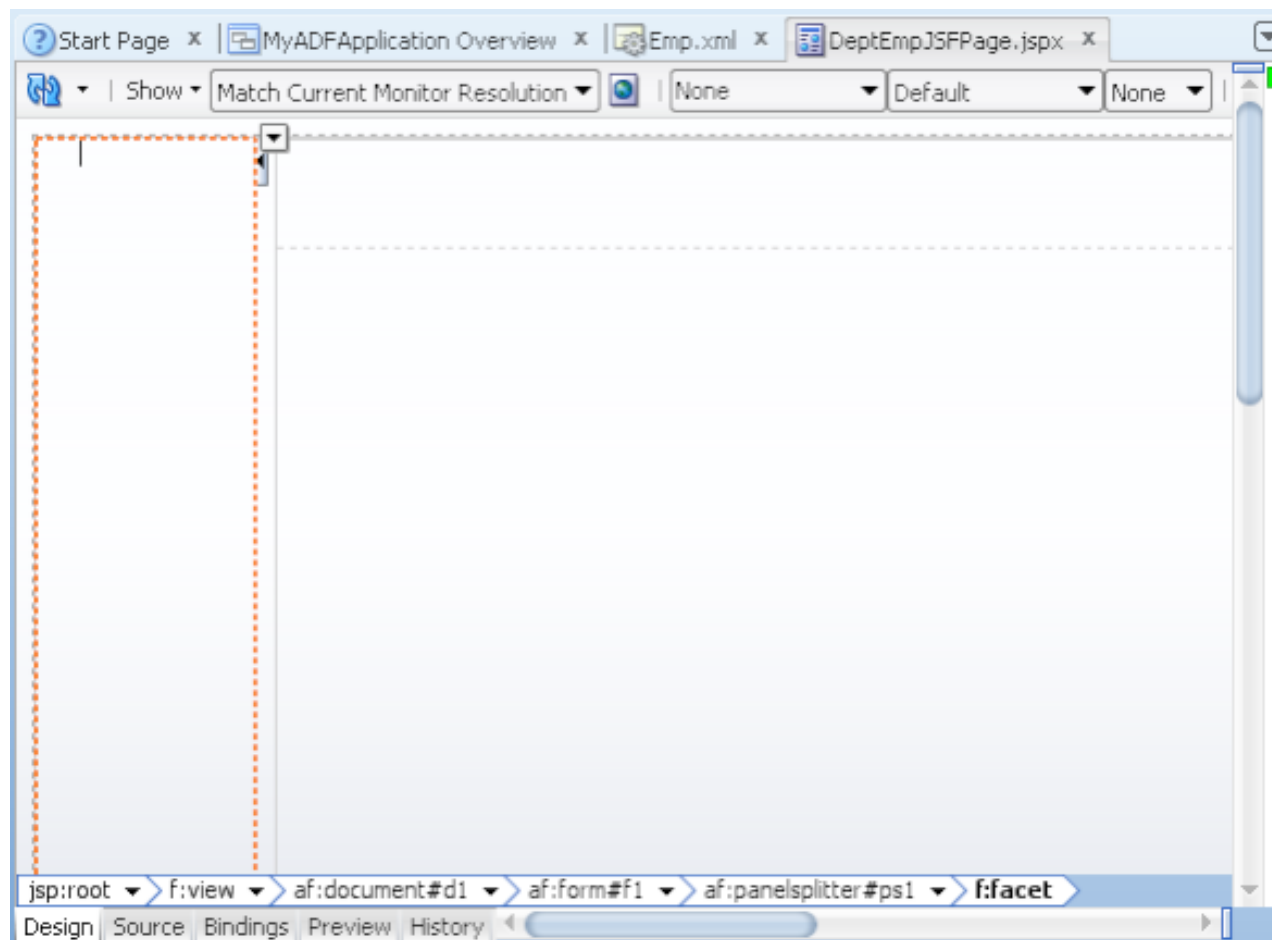


The screenshot shows the Oracle JDeveloper 11g Release 2 IDE. The main window is titled "Oracle JDeveloper 11g Release 2 - MyADFApplication.jws : ViewController.jsp". The interface includes a menu bar (File, Edit, View, Application, Refactor, Search, Navigate, Build, Run, Versioning, Tools, Window, Help), a toolbar, and several toolbars. The left pane contains the "Application Navigator" showing the project structure for "MyADFApplication", including "Projects", "Application Sources", "Web Content", "WEB-INF", "Application Resources", and "Data Controls". Below it is the "Structure" view for "DeptEmpJSFPage.jsp", showing a tree of components: "jsp:root", "jsp:directive.page", "f:view", "af:document - DeptEmpJSFPage.jsp", "af:form", "af:panelSplitter - horizontal", "f:facet - first", and "f:facet - second". The main design area shows a JSP page with a dashed orange box around a facet. The right pane contains the "Component Palette" and the "Facet - first - Property Inspector" showing the "Name" property set to "first". The bottom pane shows a "Messages" window with the following log output:

```
Running: Model.jpr - Log x
[160] ##### QueryCollection.find no RowFilter
[161] ##### QueryCollection.find no RowFilter
[162] Fail during disconnect.
Feb 7, 2012 7:32:22 PM oracle.jbo.jbotester.MainFrame exit
INFO: BC4J Tester exit code(0)
Process exited with exit code 0.
```

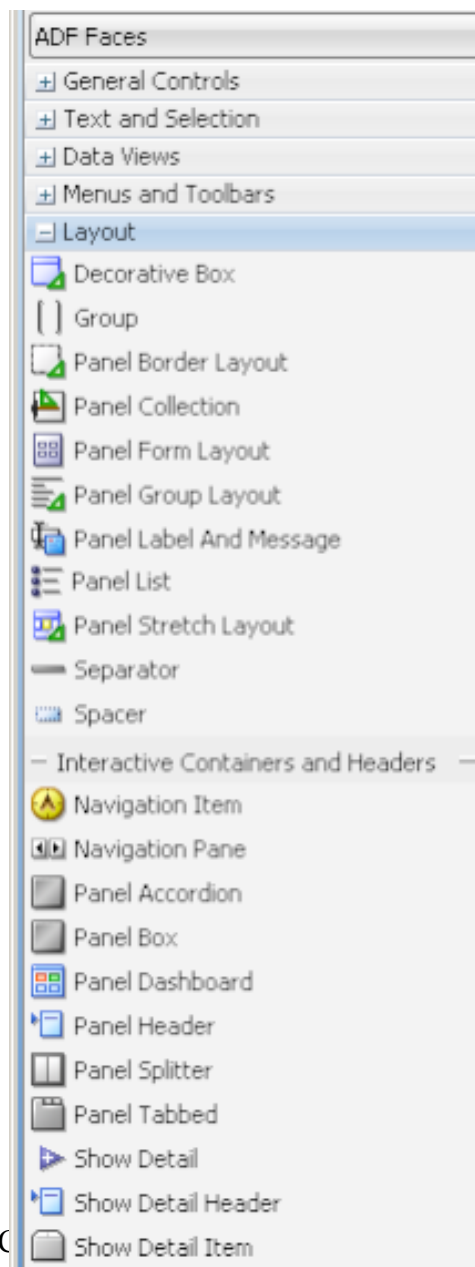


- The supplied quick start layout is ready to have objects dropped into it (resizing might be needed)



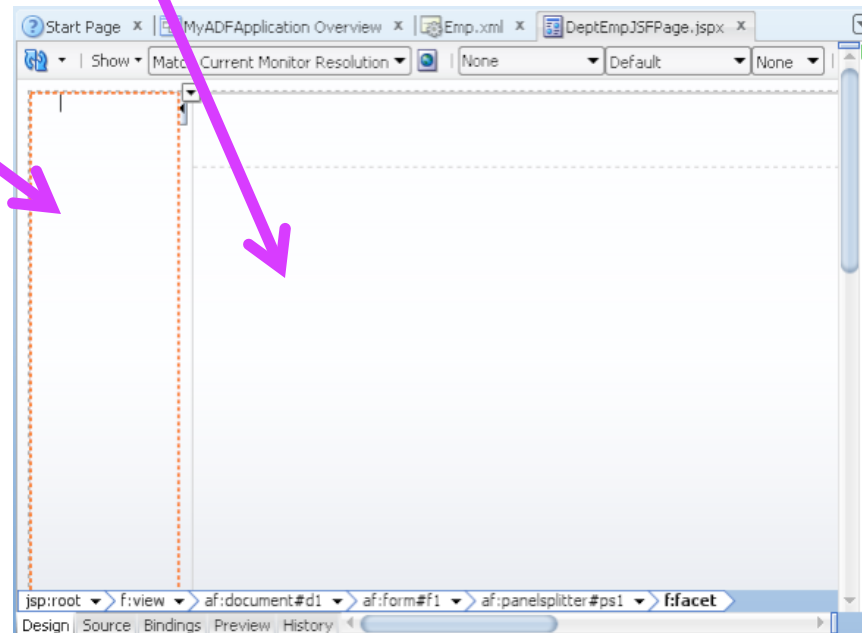
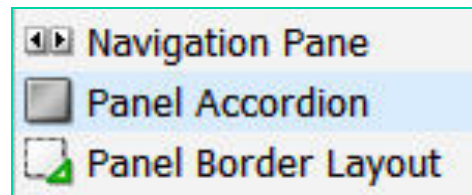


- ADF provides several “container” objects to hold data including:
 - Panel Accordion
 - Panel Tabbed
 - Panel Collection
 - Panel Splitter
 - more ...



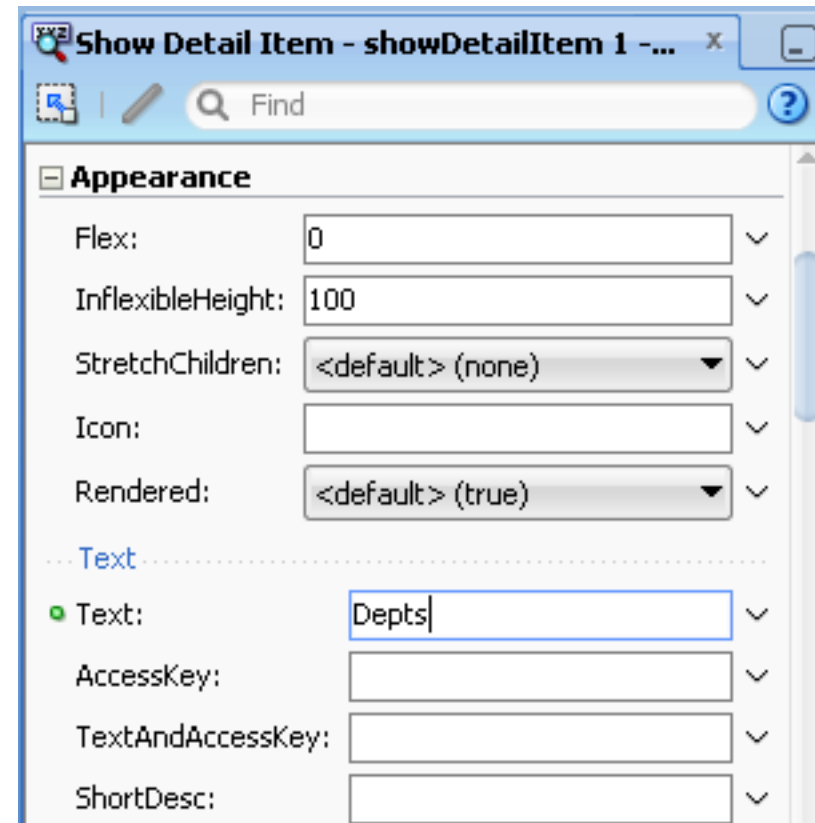


- To add an Accordion Component to the web page where the individual Dept values will appear and where the individual Emp values will appear (left-column and lower-right)



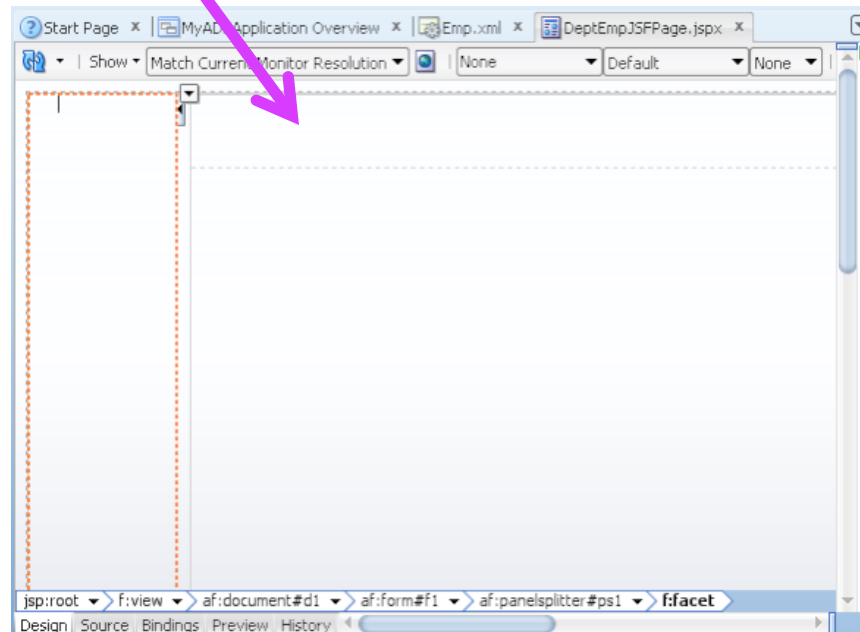
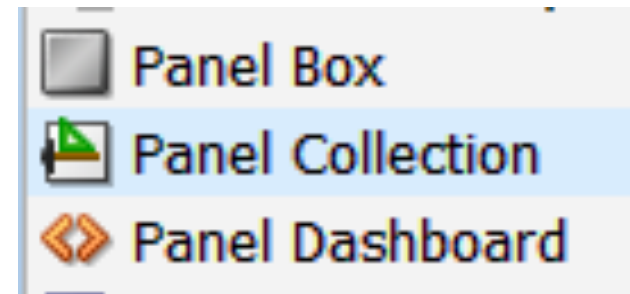


- To alter the Accordion's title, click on the Accordion and modify its Property Inspector Text item (changed to “Depts”)



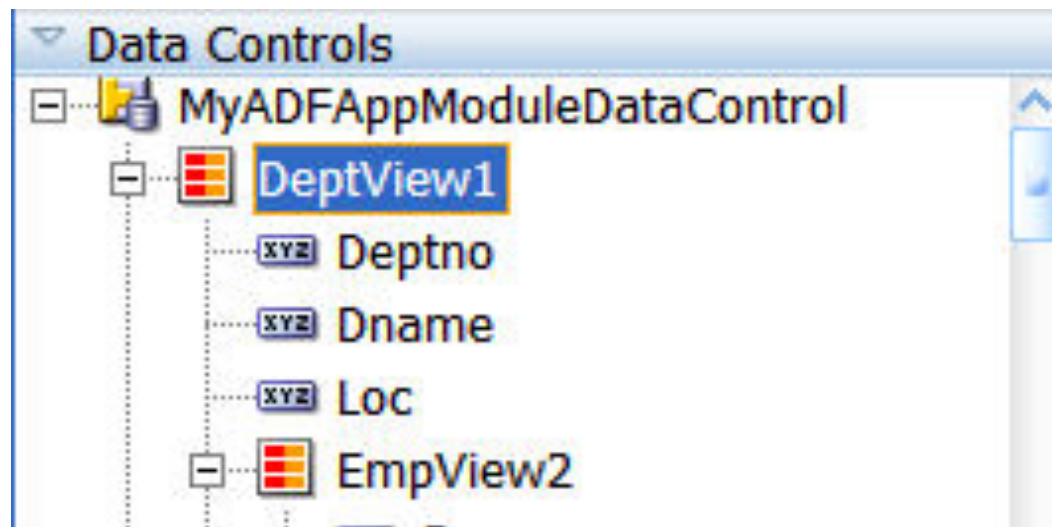


- Find the “Panel Collection” component in the Layout components and drag it to the “first” (top) part of the Splitter area



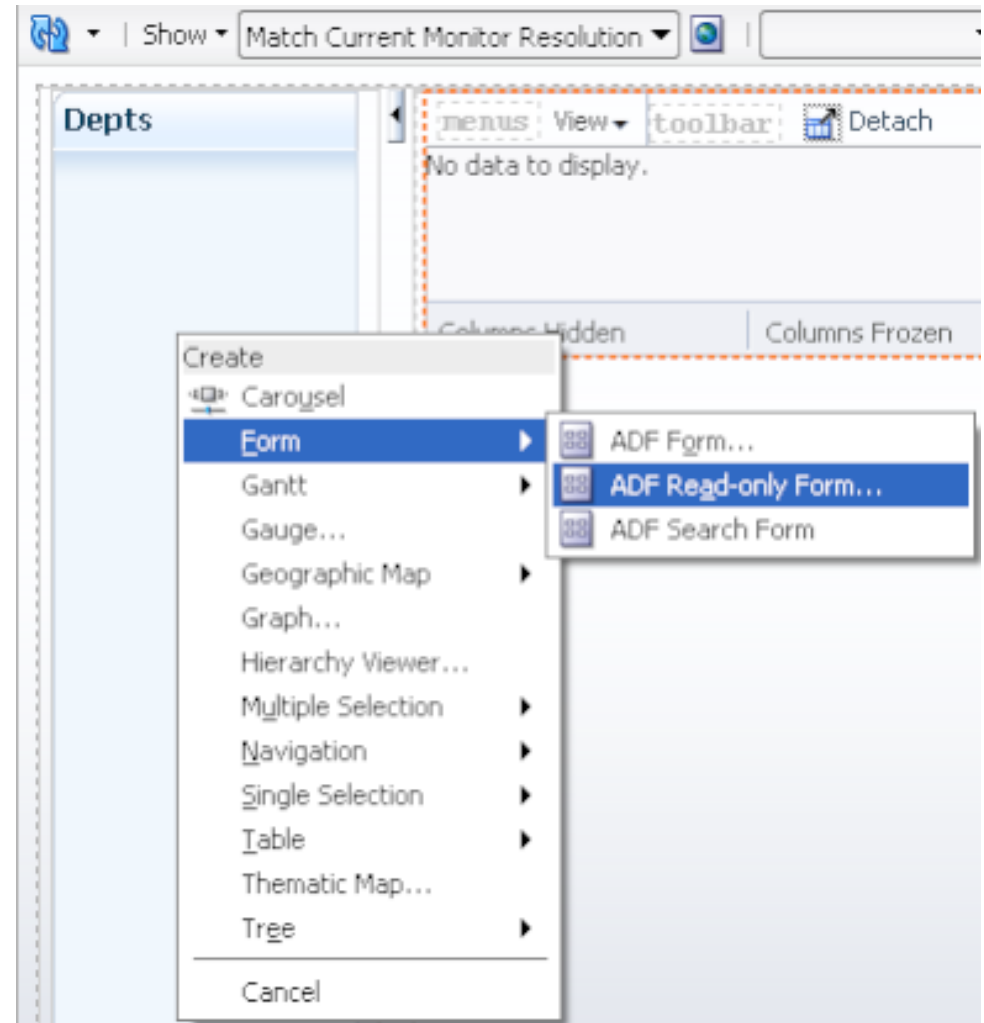


- To “bind” data to web page components, simply drag ADF BC data objects to the Visual Editor
- Open the “Application Navigator” and expand the “Data Controls” accordion to see the ADF BC components created earlier then drag “DeptView1” to the “Depts” accordion



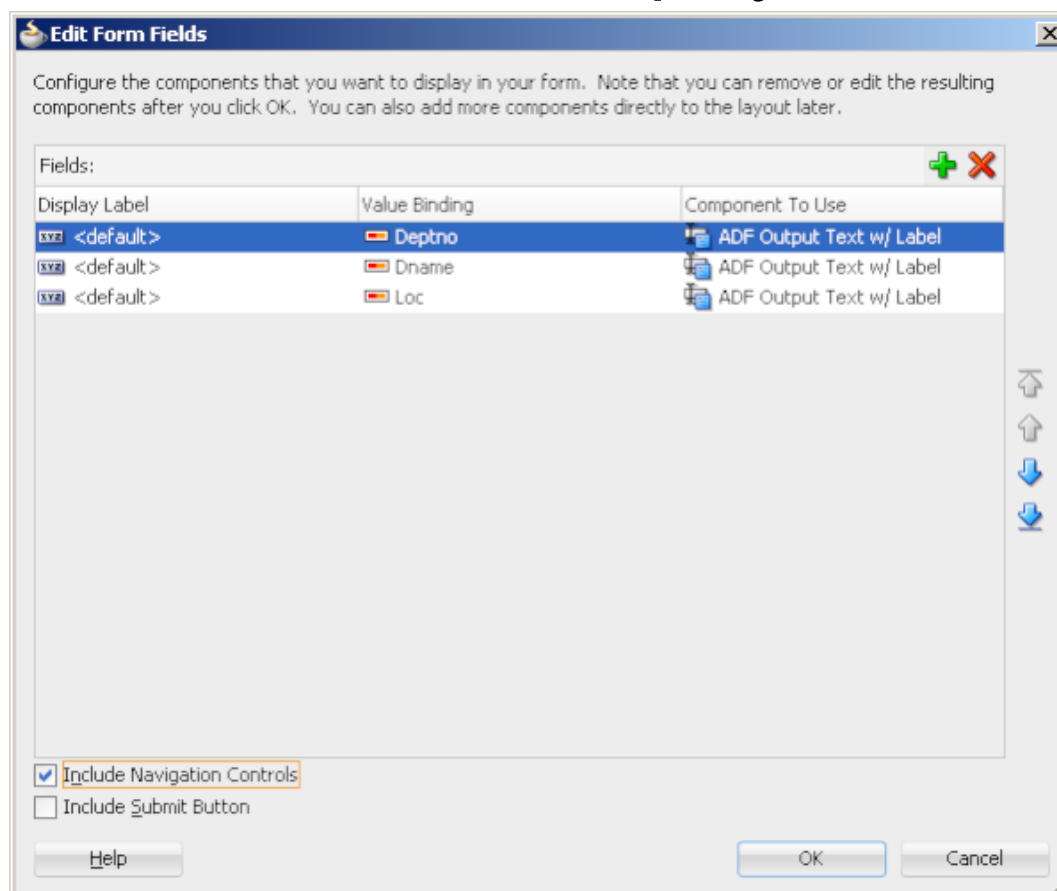


- When prompted; choose “Create Forms -> ADF Read-Only Form” to populate the Department data display



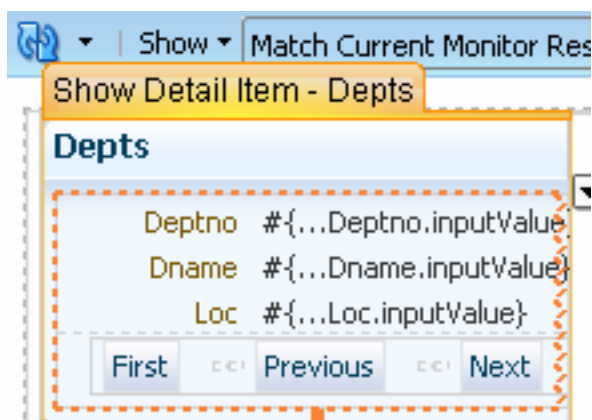


- Check the “Include Navigation Controls” box
- You may also modify display labels and add, delete, or reorganize the values displayed



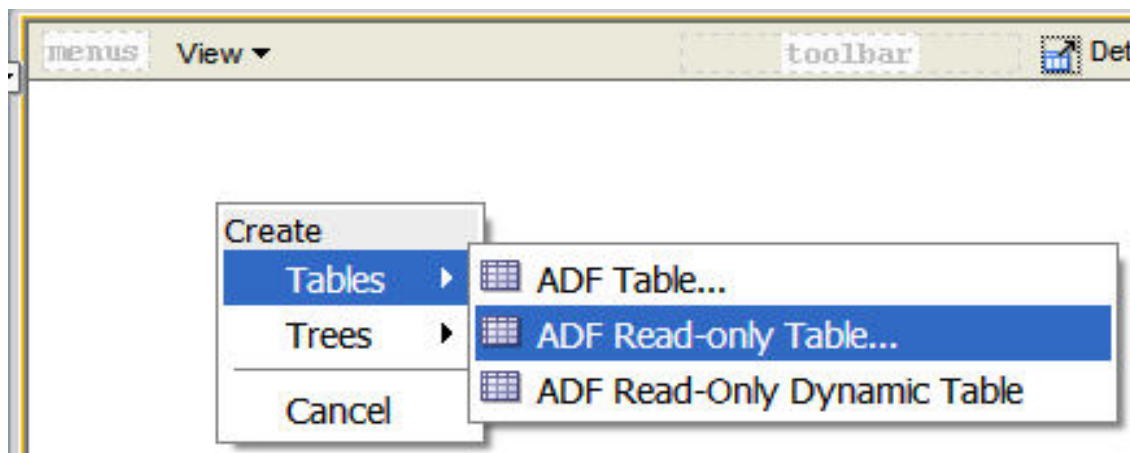


- After adding the Department information; the “Depts” accordion should look like the following

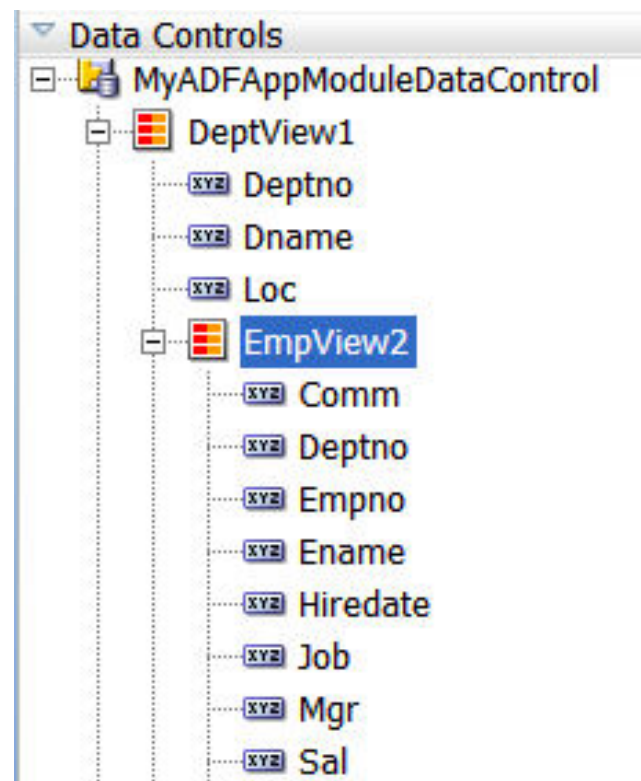




- Next, to add Department Employees to the page, drag the EmpView2 data control

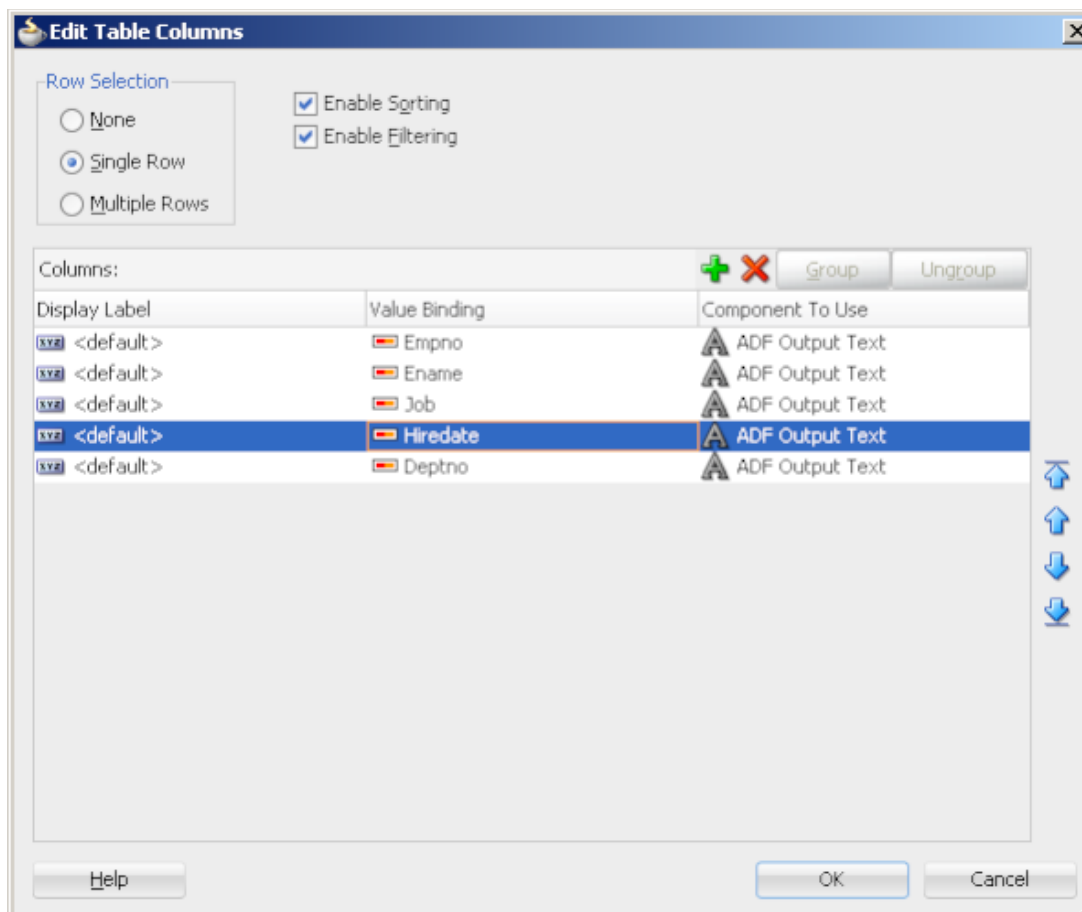


“Create Tables -> ADF Read-Only Table”





- Check all three navigation controls: Row Selection, Sorting, and Filtering; as before columns may be relabeled, added, deleted, reorganized





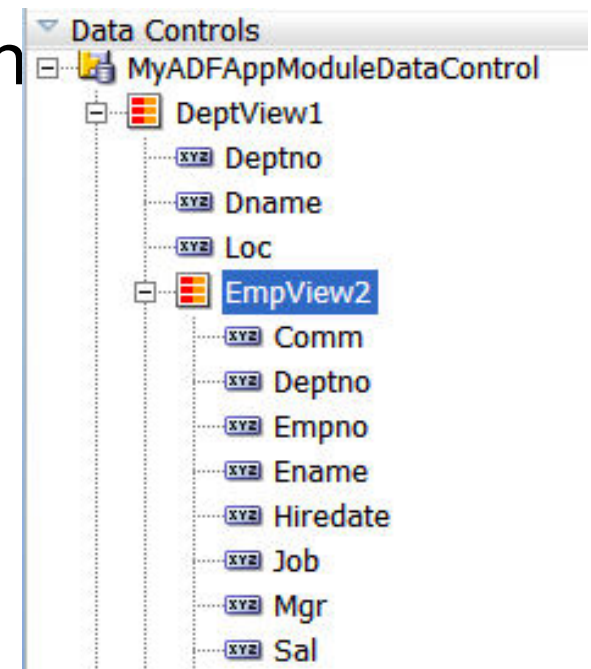
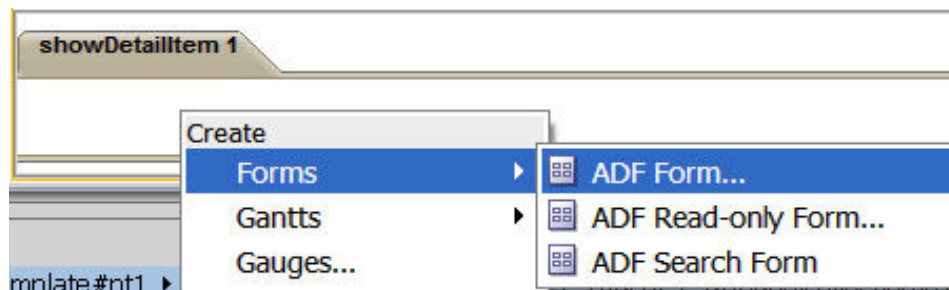
menus View toolbar Detach

Empno	Ename	Job	Hire Date	D
#{...Empno}	#{...Ename}	#{...Job}	#{...Hiredate}	#
#{...Empno}	#{...Ename}	#{...Job}	#{...Hiredate}	#
#{...Empno}	#{...Ename}	#{...Job}	#{...Hiredate}	#

Facet center - center | Columns Frozen



- Finally, add the individual Employee display to the Accordion area at the bottom of the page



- When prompted, choose “Create Forms -> ADF Form” to select the display format (this part of the form will be editable)



- Delete the DEPTNO from the display (highlight & click X); check “Include Submit Button”

Edit Form Fields

Configure the components that you want to display in your form. Note that you can remove or edit the resulting components after you click OK. You can also add more components directly to the layout later.

Display Label	Value Binding	Component To Use
<default>	Empno	ADF Input Text w/ Label
<default>	Ename	ADF Input Text w/ Label
<default>	Job	ADF Input Text w/ Label
<default>	Mgr	ADF Input Text w/ Label
<default>	Hiredate	ADF Input Date w/ Label
<default>	Sal	ADF Input Text w/ Label
<default>	Comm	ADF Input Text w/ Label

Include Navigation Controls
 Include Submit Button

Buttons: Help, OK, Cancel



Depts
menus View toolbar Detach

Empno	Ename	Job	Hire Date	D
#{...Empno}	#{...Ename}	#{...Job}	#{...Hiredate}	#
#{...Empno}	#{...Ename}	#{...Job}	#{...Hiredate}	#
#{...Empno}	#{...Ename}	#{...Job}	#{...Hiredate}	#

Columns Hidden
Columns Frozen

Empno

Ename

Job

Mgr

Hire Date

Sal

Comm

Depts

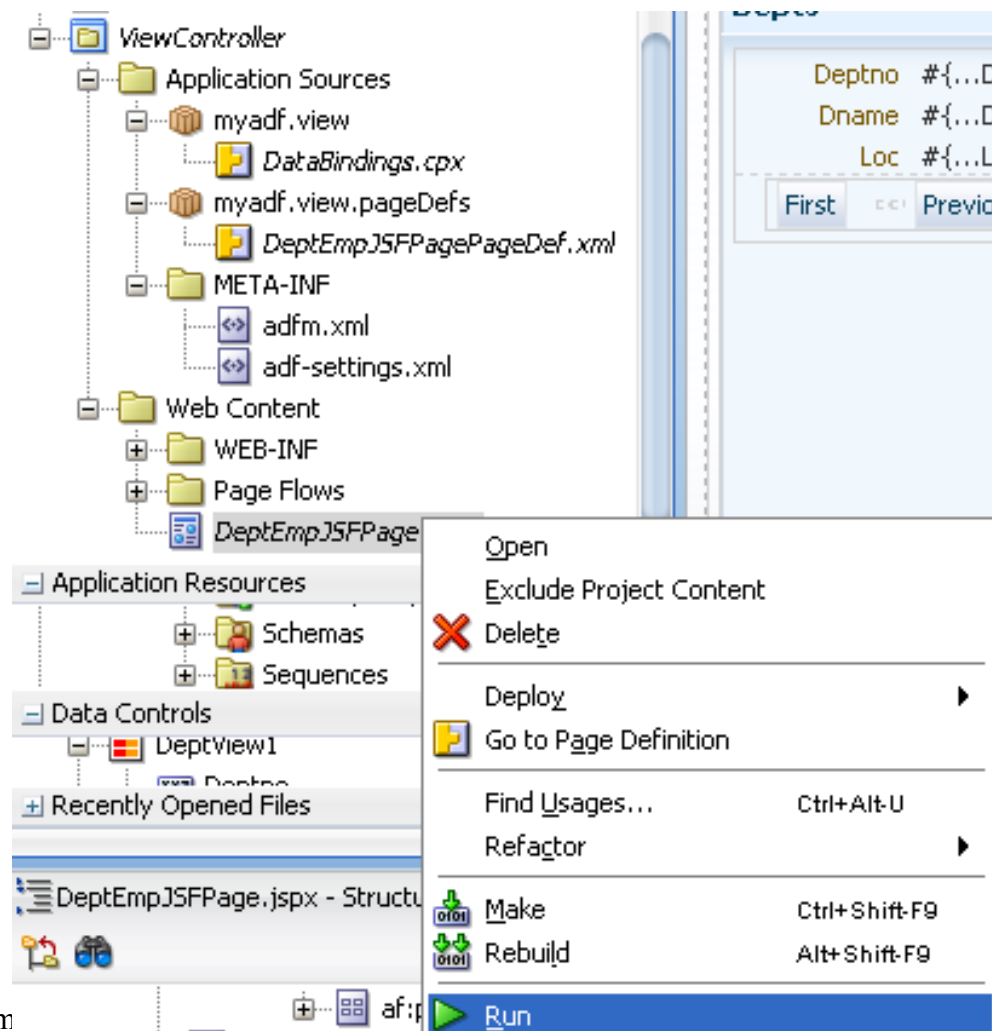
Deptno #{...Deptno.inputValue}

Dname #{...Dname.inputValue}

Loc #{...Loc.inputValue}



- To test the Web Application; right-click the “.jspx” file in the ViewController project and choose “Run”





- The first time you execute a Web application JDeveloper starts its built-in WebLogic Application Server; this takes a while
- You can track the progress of the Server's startup in JDeveloper's DefaultServer Log

```
Running: DefaultServer - Log
<Oct 25, 2009 1:07:52 AM EDT> <Info> <Management> <BEA-141107> <Version: WebLogi
<Oct 25, 2009 1:07:54 AM EDT> <Notice> <WebLogicServer> <BEA-000365> <Server sta
<Oct 25, 2009 1:07:54 AM EDT> <Info> <WebLogicServer> <BEA-002900> <Initializing se
<Oct 25, 2009 1:07:55 AM EDT> <Notice> <LoggingService> <BEA-320400> <The log fi
<Oct 25, 2009 1:07:55 AM EDT> <Notice> <LoggingService> <BEA-320401> <The log fi
<Oct 25, 2009 1:07:55 AM EDT> <Notice> <Log Management> <BEA-170019> <The server
```

- Once the Server is “up” your web page should be displayed in a browser (again, please be patient!)



DeptEmpJSFPPage.aspx

Depts

Deptno 20
Dname RESEARCH
Loc DALLAS

First Previous Next Last

View Detach

Empno	Ename	Job	Hire Date	Deptno
7369	SMITH	CLERK	1980-12-17	20
7566	JONES	MANAGER	1981-04-02	20
7788	SCOTT	ANALYST	1987-04-19	20
7876	ADAMS	CLERK	1987-05-23	20
7902	FORD	ANALYST	1981-12-03	20

*Empno

Ename

Job

Mgr

Hire Date

Sal

Comm

Submit



- Several files make up the typical ADF Web Application
 - A .jspx file is used to define each web page
 - Web pages reference a page definition XML file (.xml)
 - Bindings are described in another XML file (.cpx)



- ADF defines a web page using an XML .jspx file

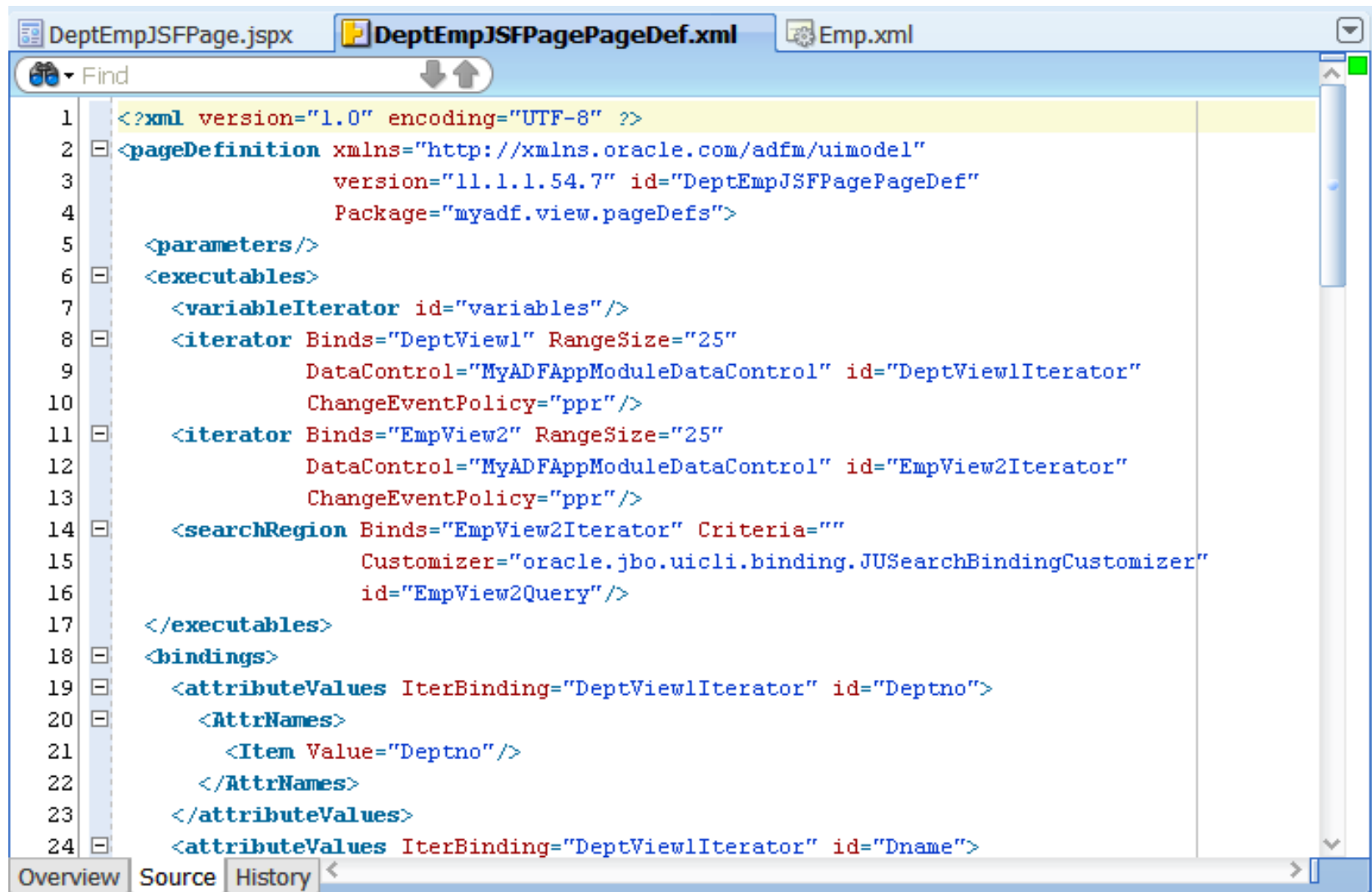
```

7  <f:view>
8  <af:document id="d1">
9    <af:messages id="m1"/>
10 <af:form id="f1">
11   <af:pageTemplate viewId="/oracle/templates/threeColumnTemplate.jsx"
12     id="pt1">
13     <f:facet name="center">
14       <af:panelSplitter id="ps1" orientation="vertical">
15         <f:facet name="first">
16           <af:panelCollection id="pcl">
17             <f:facet name="menus"/>
18             <f:facet name="toolbar"/>
19             <f:facet name="statusbar"/>
20           <af:table value="#{bindings.EmpView2.collectionModel}"
21             var="row" rows="#{bindings.EmpView2.rangeSize}"
22             emptyText="#{bindings.EmpView2.viewable ? 'No data to display'"
23             fetchSize="#{bindings.EmpView2.rangeSize}"
24             rowBandingInterval="0"
25             filterModel="#{bindings.EmpView2Query.queryDescriptor}"
26             queryListener="#{bindings.EmpView2Query.processQuery}"
27             filterVisible="true" varStatus="vs"
28             selectedRowKeys="#{bindings.EmpView2.collectionModel.selected"
29             selectionListener="#{bindings.EmpView2.collectionModel.makeCu

```

document#d1 ▶ af:form#f1 ▶ af:pagetemplate#pt1 ▶ f:facet ▶ af:panelaccordion#pa1 ▶ af:showdetailitem#sdi1 ▶

Design Source Bindings Preview History



The screenshot shows an IDE window with three tabs: DeptEmpJSFPage.jspx, DeptEmpJSFPagePageDef.xml (selected), and Emp.xml. The XML content is displayed in a text editor with a line number column on the left. The XML defines a page with parameters, executables (including iterators and a search region), and bindings for attribute values.

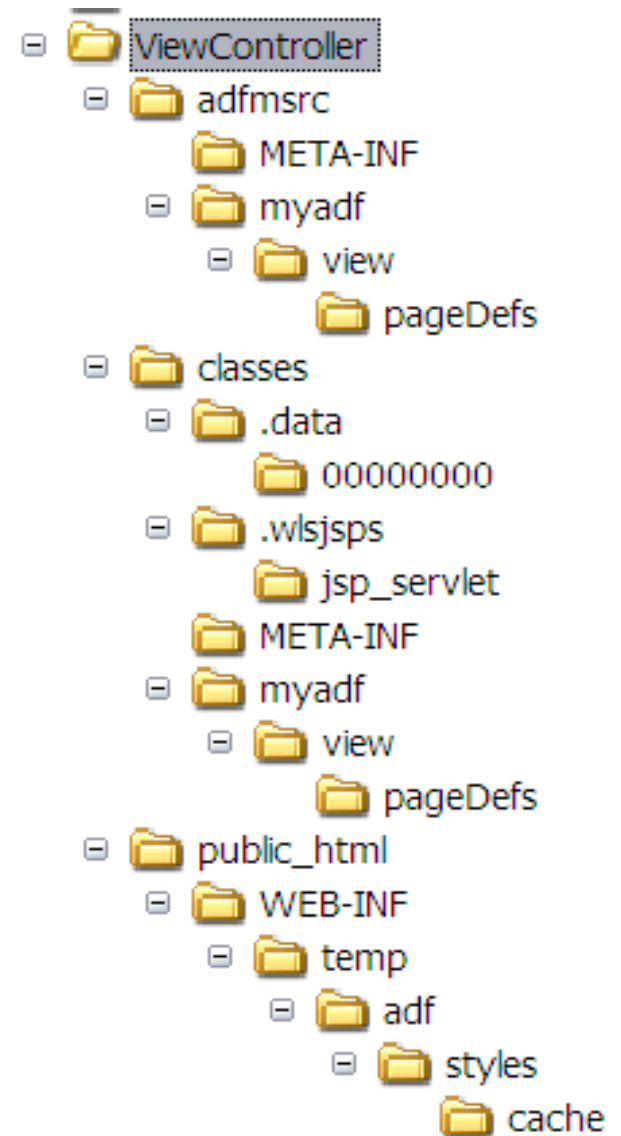
```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <pageDefinition xmlns="http://xmlns.oracle.com/adfm/uimodel"
3                 version="11.1.1.54.7" id="DeptEmpJSFPagePageDef"
4                 Package="myadf.view.pageDefs">
5     <parameters/>
6     <executables>
7         <variableIterator id="variables"/>
8         <iterator Binds="DeptView1" RangeSize="25"
9                 DataControl="MyADFAppModuleDataControl" id="DeptView1Iterator"
10                ChangeEventPolicy="ppr"/>
11        <iterator Binds="EmpView2" RangeSize="25"
12                DataControl="MyADFAppModuleDataControl" id="EmpView2Iterator"
13                ChangeEventPolicy="ppr"/>
14        <searchRegion Binds="EmpView2Iterator" Criteria=""
15                    Customizer="oracle.jbo.uicli.binding.JUSearchBindingCustomizer"
16                    id="EmpView2Query"/>
17    </executables>
18    <bindings>
19        <attributeValues IterBinding="DeptView1Iterator" id="Deptno">
20            <AttrNames>
21                <Item Value="Deptno"/>
22            </AttrNames>
23        </attributeValues>
24        <attributeValues IterBinding="DeptView1Iterator" id="Dname">
```



```
DeptEmpJSFPage.jspx | DeptEmpJSFPagePageDef.xml | DataBindings.cpx | Emp.xml
Find
1 | <?xml version="1.0" encoding="UTF-8" ?>
2 | <Application xmlns="http://xmlns.oracle.com/adfm/application"
3 |           version="11.1.1.54.7" id="DataBindings" SeparateXMLFiles="false"
4 |           Package="myadf.view" ClientType="Generic">
5 |   <pageMap>
6 |     <page path="/DeptEmpJSFPage.jspx"
7 |           usageId="myadf_view_DeptEmpJSFPagePageDef"/>
8 |   </pageMap>
9 |   <pageDefinitionUsages>
10 |     <page id="myadf_view_DeptEmpJSFPagePageDef"
11 |           path="myadf.view.pageDefs.DeptEmpJSFPagePageDef"/>
12 |   </pageDefinitionUsages>
13 |   <dataControlUsages>
14 |     <BC4JDataControl id="MyADFAppModuleDataControl" Package="myadf.model"
15 |                     FactoryClass="oracle.adf.model.bc4j.DataControlFactoryImpl
16 |                     SupportsTransactions="true" SupportsFindMode="true"
17 |                     SupportsRangeSize="true" SupportsResetState="true"
18 |                     SupportsSortCollection="true"
19 |                     Configuration="MyADFAppModuleLocal" syncMode="Immediate"
20 |                     xmlns="http://xmlns.oracle.com/adfm/datacontrol"/>
21 |   </dataControlUsages>
22 | </Application>
```



- The XML files representing the ViewController project are distributed using a directory structure





Feature	Forms	ADF
Declarative database access	Yes	Yes
Reuse of database access	Some	Yes
Declarative user interface development	Yes	Yes
Automatic screen generation	Yes	Some
Reuse of user interface	Some	Yes
Web Deployment	Yes	Yes
Client-Server Deployment	No	Yes
Fusion Applications development tool	No	Yes
Customizable	Yes	Yes
Built with open standards	No	Yes



- Quick Start Guide to Oracle Fusion Development
 - Grant Ronald
 - Oracle Press
- Oracle JDeveloper 11g Handbook
 - Duncan Mills, Peter Koletzke, Dr. Avrom Roy-Federman
 - Oracle Press
- Oracle Fusion Developer's Guide
 - Frank Nimphius, Lynn Munsinger
 - Oracle Press



- Oracle Forms is not going anywhere; it is not necessary to “convert” things to ADF (third-party tools are available to help if you insist)
- Oracle’s design emphasis and new features will support the Java-based ADF mechanism; Oracle Forms probably won’t see much in the way of new functionality
- JDeveloper and ADF allow me to create simple forms almost as easily as in forms except:
 - ADF BC (data) creates reusable components
 - ADF Faces (view) creates reusable components
- I did not write a single line of Java in this demo!



Training Days 2013

<http://www.rmoug.org>

Watch for the Date!



MARK YOUR CALENDARS!

COLLABORATE 12
April 22-26, 2012
Mandalay Bay Convention Center
Las Vegas, Nevada

REGISTER TODAY!

JW MARRIOTT.
SAN ANTONIO HILL COUNTRY



ODTUG
Kscope12



SAN ANTONIO, TEXAS * JUNE 24-28

Application Express * Database * Developer's Toolbox
Business Intelligence * Essbase * Hyperion Applications
Hyperion Business Content * Fusion Middleware

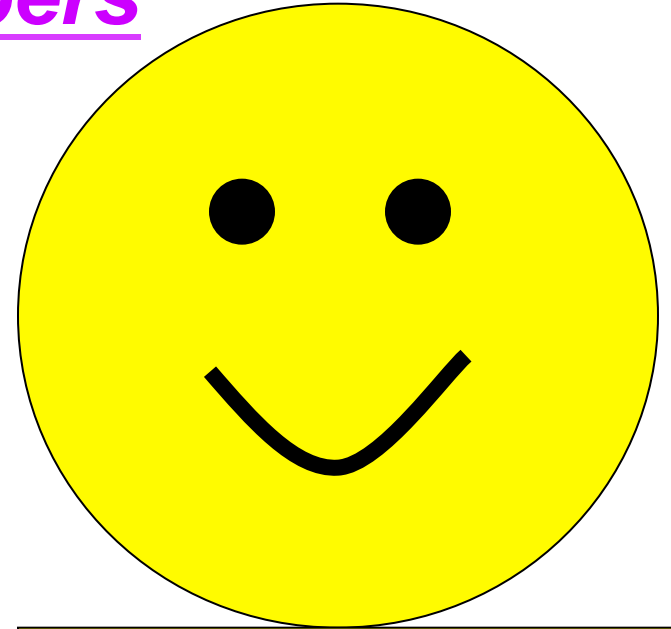
www.kscope12.com

Please Fill Out Session Evaluations

Oracle ADF for Forms Developers

To contact the author:

John King
King Training Resources
PO Box 1780
Scottsdale, AZ 85252 USA
1.800.252.0652 - 1.303.798.5727
Email: john@kingtraining.com



Thanks for your attention!

Today's slides and examples are on the web:

<http://www.kingtraining.com>