

ADF On-Ramp: Developing With the Core ADF Stack

Peter Koletzke

Technical Director & Principal Instructor



quovera



Survey

- Job responsibilities?
 - DBA, developer
- Languages?
 - PL/SQL, Java
 - Other
- Tools?
 - APEX
 - Developer Forms/Reports
 - JDeveloper
 - Eclipse, NetBeans
 - Other




quovera

2

Agenda

- What is ADF?
- ADF Business Services
- ADF Model, View, and Controller

Slides and white paper will be available on the UTOUG and Quovera websites.

Screenshots are 11g unless marked as 12c 



quovera

3

On the Positive Side...

If we do not find
anything pleasant, at least
we shall find something new.

Si nous ne trouvons pas des choses agréables, nous trouverons du moins des choses nouvelles.

—Voltaire (1694-1778), *Candide*

quovera

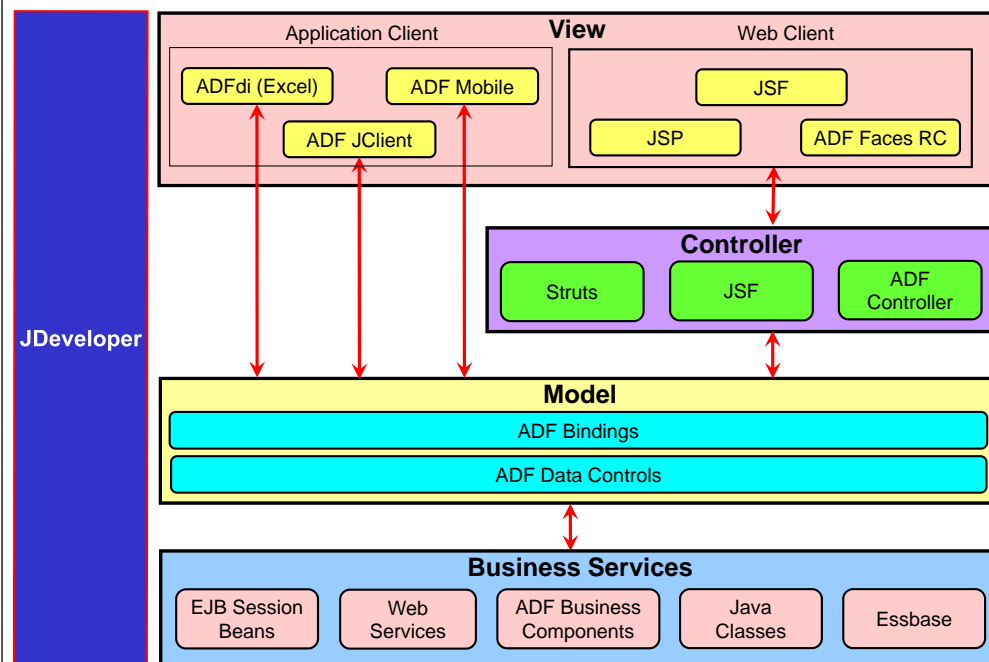
4

Oracle Application Development Framework (ADF)

- A *framework* is a prebuilt service for solving a particular problem – like access to the database
 - Code libraries and standards support the framework
 - Implements code reuse and best practices
 - An architecture with code libraries
- ADF is a *meta-framework*
 - A wrapper for other frameworks, JDev 10g and later
 - Provides a consistent developer experience
- Pre-ADF available in OAF
 - Oracle Application Framework (UIX/MVC)
- Based on Model-View-Controller design pattern (SmallTalk, Java EE)



ADF Architecture



Which ADF Technologies to Use?

- An example: Fusion Applications (Oracle applications suite)
- *Core technology stack* used for Fusion Apps is:
 - ADF Business Components
 - ADF Faces Rich Client
 - ADF Model
 - ADF Controller
- Other *high-level technologies* or strategies also used
 - SOA, ESB, Business Rules, WebCenter, BPM, BPA, BAM
 - Need to consider those, too, Out of scope



ADF Essentials

- No-license-fee version of ADF
 - Runs on the public domain app server, Glassfish, not WebLogic Server
- Works in JDeveloper
- Works in Eclipse
 - Through Oracle Enterprise Pack for Eclipse
- More information on OTN

<http://www.oracle.com/technetwork/developer-tools/adf/overview/adfessentials-1719844.html>

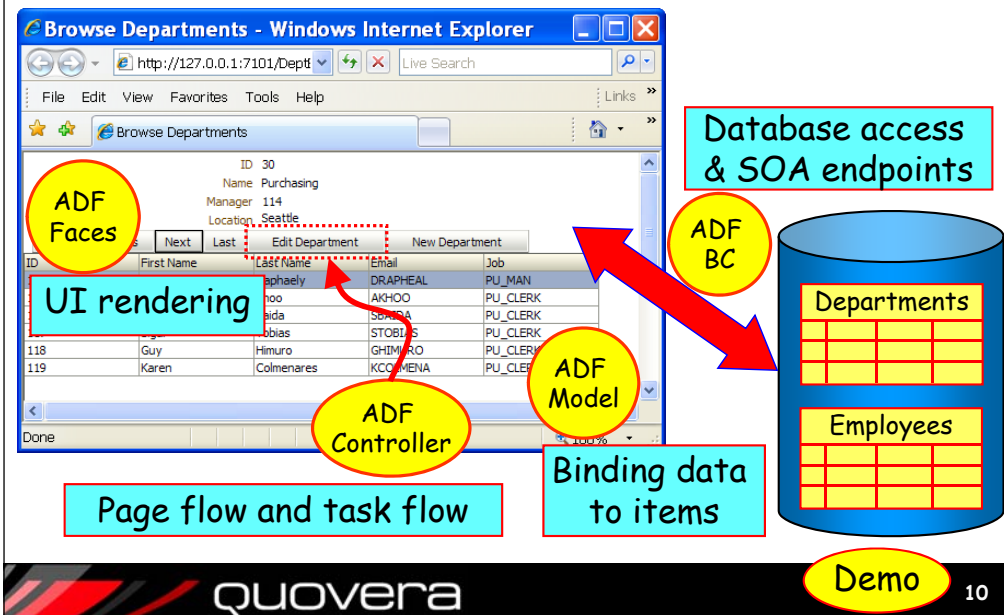


ADF Mobile

- Similar ADF principles and development methods
 - Business services
 - Pages with components (amx)
 - Task flows
- Supports iOS and Android
- Use local services like camera & contacts
- OTN demo video and ADF Academy video
www.oracle.com/technetwork/developer-tools/adf-mobile



Where Do The Core Technologies End Up?



Which Languages Do You Use?

1. Java
 - All important programmatic code
 - Think “trigger code” as in Oracle Forms
2. XML
 - The components rely on XML
 - Property editors create it for you
3. JavaScript and Cascading Style Sheets
 - Add functionality to HTML pages
 - Usually the components do this work for you
4. Expression Language
 - Used in JSF binding properties
5. Groovy
 - ADF BC scripting



How Much of Each Do You Use?

Language (Use)	Level Needed	Primary Use
Java (frameworks such as ADF Faces)	Basic	Business components code for validation and special handling of model objects, as well as coding conditional page flow.
Java (extending framework features)	Expert	Supplementing or replacing functionality supplied by the framework. This requires research into the framework's capabilities and architecture.
XML	Basic	The JSF tags and the HTML renderer take care of the HTML for you. XML is used for JSF JSP files
JavaScript	Basic/None	Providing customized user interaction functionality, for example, special handling of a checkbox selection.
Cascading Style Sheets	Basic/None	Developing skins for a consistent look and feel. If you use prebuilt look-and-feel templates, no CSS coding is needed.
Expression Language	Basic/Intermediate	Supplying data to components from properties or methods in the application.
Groovy	Basic	Expressions for ADF Business Components

Agenda

- What is ADF?
- ADF Business Services
- ADF Model, View, and Controller



The World View

In this best of all possible worlds ...
everything is for the best.

Dans ce meilleur des mondes possibles ...
tout est au mieux.

—Voltaire (1694-1778), *Candide*

ADF Business Components

- ADF BC: an option in the Business Services layer of ADF
- Persistence: storing data in a database
- O/R mapping: Translates relational database things to object-oriented (Java) whatsits
- Handles JDBC mechanics
 - Creates SQL and handles results
- Primarily declarative
 - XML source code to define the use of framework classes



More About ADF BC

- Various component types, e.g.,
 - View objects: define queries
 - Entity objects: define insert-update-delete (“DML”)
 - View links: view object relationships
 - Associations: entity object links
 - Application modules: Define the data models and the database transaction
- It does not create user interfaces



Sample ADF BC Development

Entity object editors

Demo

View Object Code

View Object = SELECT statement

View Attribute = Column in query

```
<ViewObject
xmlns="http://xmlns.oracle.com/bc4j"
Name="AllEmployees"
Version="11.1.1.53.41"
SelectList="Employees.EMPLOYEE_ID,
Employees.FIRST_NAME,
Employees.LAST_NAME,
Employees.JOB_ID,
Employees.EMAIL,
Employees.HIRE_DATE,
Departments.DEPARTMENT_NAME,
Departments.DEPARTMENT_ID,
Departments.LOCATION_ID"
FromList="DEPARTMENTS Departments,
EMPLOYEES Employees"
Where="Departments.MANAGER_ID =
Employees.EMPLOYEE_ID"
BindingStyle="OracleName"
CustomQuery="false"
PageIterMode="Full"
UseGlueCode="false">
...
```

```
<Attribute
Name="EmployeeId"
IsNotNull="true"
Precision="6"
Scale="0"
ColumnName="EMPLOYEE_ID"
SQLType="NUMERIC"
Type="oracle.jbo.domain.Number"
ColumnType="NUMBER"
TableName="EMPLOYEES"
PrimaryKey="true">
<DesignTime>
<Attr Name="_DisplaySize"
Value="22"/>
</DesignTime>
</Attribute>
<Attribute
Name="FirstName"
Precision="20"
ColumnName="FIRST_NAME"
...
```

List-of-Values (LOV)

- Defined for an view object attribute
- Associate a query (view object) with the attribute
- Set up is declarative
 - Start in the view object editor
- UI code will display this attribute specially
 - Pulldown list item loaded with values
 - Popup list-of-values dialog



VO Editor – LOV Subtab

List of Values:

Name	List Data Source	List Attribute
LOV_JobId	JobsLOVAccessor	JobId

Create List of Values

The image shows two screenshots related to creating a list of values. The top screenshot is the 'Create List of Values' dialog box. It has a 'List of Values Name' field containing 'LOW_JobId'. Under 'Configuration', 'UI Hints' is selected. The 'List Data Source' is 'JobsLOVAccessor' and the 'List Attribute' is 'JobId'. The 'Display Attributes' section shows 'Available' attributes: JobId, MaxSalary, and MinSalary. The 'Selected' attribute is 'JobTitle'. The 'List Search' section has 'Include Search Region' set to 'All Queryable Attributes'. The bottom screenshot shows a table titled 'EmployeeJobHistoryWL1' with a yellow 'At runtime' label. The table contains columns for 'Last Name/Surname', 'Email Address', 'Phone No.', 'Current Job', and 'Empl...'. The 'Current Job' column is highlighted, showing a list of job titles including 'Sales Representative', 'Public Relations Representative', 'Purchasing Clerk', 'Purchasing Manager', 'Sales Manager', 'Shipping Clerk', 'Stock Clerk', and 'Stock Manager'.

View Criteria

- Used for queries
- View Object Editor
 - 11g Query page
 - 12c VC page
- View Criteria dialog
 - Defines the WHERE clause declaratively
 - Bind variables are optional

The image shows the 'Edit View Criteria' dialog box for 'BasicEmployeeSearchCriteria'. The 'Criteria Definition' tab is active, showing a tree view of criteria: 'BasicEmployeeSearchCriteria' containing a 'Group' with three criteria: 'First Name CONTAINS :queryFirstName', 'AND LastName CONTAINS :queryLastName', and 'AND Email CONTAINS :queryEmail'. The 'View Object Where Clause' field contains a complex SQL WHERE clause. The 'Criteria Item' section shows 'Conjunction' set to 'AND', 'Attribute' set to 'Email', 'Operator' set to 'Contains', and 'Operand' set to 'Bind Variable' with 'Parameter' set to 'queryEmail'. Other options include 'Ignore Case', 'Ignore Null Values', and 'Validation' set to 'Optional'.

View Criteria in the DCP

- Data Controls panel lists the defined criteria
- You can drag and drop a view criteria onto a JSF page

The image shows a 'Create' menu with options: 'ADF Filtered Table...', 'Query', and 'Cancel'. The 'Query' option is expanded, showing sub-options: 'ADF Query Panel', 'ADF Query Panel with Table...', and 'ADF Query Panel with Tree Table...'.

The image shows the 'Data Controls' panel. It contains a tree view with 'EmployeeSearch' expanded, showing attributes: 'DepartmentId', 'DepartmentName', 'Email', 'EmployeeId', and 'FirstName'. Below this is a 'Named Criteria' folder containing 'All Queryable Attributes' and 'BasicEmployeeSearchCriteria'. A red arrow points from the 'BasicEmployeeSearchCriteria' item to the 'ADF Query Panel with Tree Table...' option in the 'Create' menu.

- Components are added for each attribute in the view criteria

Declarative Validation Rules

- Simple rules written on the entity object
- Method 1
 - Attributes tab
 - Validation Rules subtab
- Method 2
 - Business Rules tab
 - Attribute or entity level

The image shows the 'Employees.xml' entity editor. The 'Attributes' tab is active, showing a table of attributes with columns 'Name', 'Type', and 'Column Name'. The 'Validation Rules' subtab is selected, showing a table with columns 'Validation Rule' and 'Type'. A red arrow points from the 'Validation Rules' subtab to the 'Validation Rule' table.

Name	Type	Column Name
HireDate	Timestamp	HIRE_DATE
JobId	String	JOB_ID
Salary	BigDecimal	SALARY
CommissionPct	BigDecimal	COMMISSION_PCT
ManagerId	Integer	MANAGER_ID
DepartmentId	Integer	DEPARTMENT_ID
CreatedBy	String	CREATED_BY
CreatedDate	Timestamp	CREATED_DATE
ModifiedBy	String	MODIFIED_BY
ModifiedDate	Timestamp	MODIFIED_DATE

Validation Rule	Type
Precision : (2,2)	Database Constraints
'Must qualify newValue <= new java.lang.Flo... Expression	

Declarative Validation Benefits

- Quick validation on the app server side
- No Java coding!
- Add a friendlier message for ADF BC-level errors (e.g., length)
- Messages stored in a message bundle file that you can internationalize
- All UIs built from the entity object will contain the same validation
 - Like a trigger in the database
- You need to decide where to place business rules code



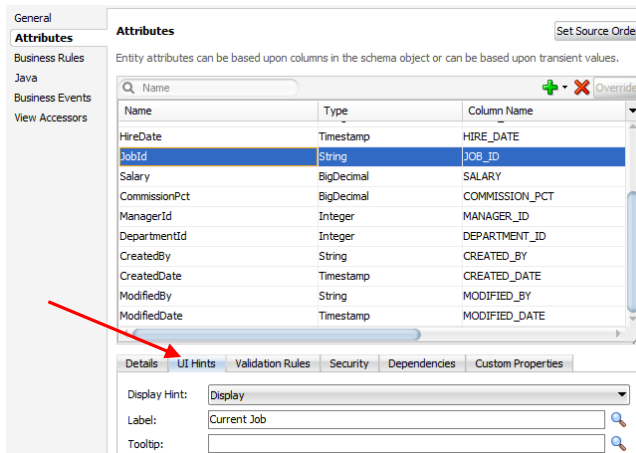
UI Hints

- A.k.a., *control hints*
- UI definitions stored in the entity object or view object definitions
 - Universal to all UIs created from those EOs or VOs
 - Provides consistency
 - “Set it and forget it”
- Can put the hints on either EO or VO
 - Rule of thumb: use the entity object UI hints whenever possible
 - Override on view object level if needed



Setting UI Hints

- In Entity Object editor, select attribute and click the UI Hints tab in the Property Inspector
- **OR** View Object editor Attributes tab, UI Hints subtab



Agenda

- What is ADF?
- ADF Business Services
- ADF **View**, Model, and Controller



ADF Faces Rich Client Overview

- Fits into the View layer of ADF
- History:
 - ADF UIX → ADF Faces → Apache Trinidad
 - ADF Faces → ADF Faces RC
- Built on top of JSF APIs
- Deployable on any 1.2 implementation of JSF
- Support for pop-ups and dialogs
- ADF model support out-of-the-box
- Data Visualization Tools (DVT) components
 - Charts, Gantt, Pivot, Maps, Hierarchy

Really rich!



ADF Faces RC Features

- Changeable “skins”
 - Common look-and-feel characteristics
 - Skin editor in JDev 11.1.2 and beyond
- Layout management features
- Extensive set of properties
 - Declarative access to application metadata
 - Properties can reference dynamic values using Expression Language
- Template support
- Declarative AJAX
 - Property-driven



Some Components

* ID 198 Last
af:inputText af:button
 Department 50 af:link (with icon)
af:inputListOfValues * Job Shipping Clerk
af:commandMenuItem af:selectOneChoice
 * Hire Date: 6/21/1999 af:inputDate
af:selectBooleanCheckbox I Love JDev
af:inputDate af:selectBooleanCheckbox

Sample ADF Faces Development

deptBrowse.jspx
 # {...DepartmentId...label} #{...DepartmentId.in
 #{...DepartmentName...label} #{...DepartmentNan
 #{...ManagerId...label} #{...ManagerId.inp
 #{...LocationId1.label} #{...LocationId1.inp
 First Previous Next Last Edit Department
 # {...EmployeeId...label} # {...FirstName.label} # {...LastName.label}
 # {...EmployeeId} # {...FirstName} # {...LastName}
 # {...EmployeeId} # {...FirstName} # {...LastName}
 # {...EmployeeId} # {...FirstName} # {...LastName}
 contextMenu:
 jsp:root > f:view >
 Design Source Bindings Preview History <

Property Inspector: Output Text - #{bindings.DepartmentId.inputValue} - Property Inspector
 Common
 Id: ot3
 Rendered: <default> (true)
 Value: #{bindings.DepartmentId.inputValue}
 Appearance
 NoWrap: <default> (false)
 TruncateAt: 0
 Description:
 ShortDesc:
 Rendered: <default> (true)
 Style
 Behavior
 Escape: <default> (true)
 Converter: <default>
 PartialTriggers:
 Advanced
 Customization

Demo

ADF Faces JSF Snippet

```
<jsp:root xmlns:jsp="http://java.sun.com/JSP/Page" version="2.0"
  xmlns:f="http://java.sun.com/jsf/core"
  xmlns:af="http://xmlns.oracle.com/adf/faces/rich">
...
<af:panelStretchLayout styleClass="AFVisualRoot" topHeight="105px"
  bottomHeight="20px">
  <f:facet name="top">
    <af:panelBorderLayout>
      <f:facet name="start">
        <af:image source="/images/tuhra.gif" shortDesc="TUHRA Logo"/>
      </f:facet>
    </f:facet name="end">
      <af:panelGroupLayout layout="horizontal" halign="right"
        valign="bottom">
        <af:commandImageLink text="Logon" shortDesc="Logout from TUHRA"
          depressedIcon="/images/groupdisconnect_dwn.png"
          disabledIcon="/images/groupdisconnect_dis.png"
          hoverIcon="/images/groupdisconnect_ovr.png"
          icon="/images/groupdisconnect_ena.png"
          disabled="true"
          rendered="#{attrs.anonymous}"/>
        <af:commandImageLink text="Logoff" shortDesc="Logout from TUHRA"
          depressedIcon="/images/groupdisconnect_dwn.png"
          disabledIcon="/images/groupdisconnect_dis.png"
          hoverIcon="/images/groupdisconnect_ovr.png"
          icon="/images/groupdisconnect_ena.png"
          disabled="true"
          rendered="#{!attrs.anonymous}"/>
      </af:panelGroupLayout>
    </f:facet name="end">
  </af:panelStretchLayout>
</jsp:root>
```

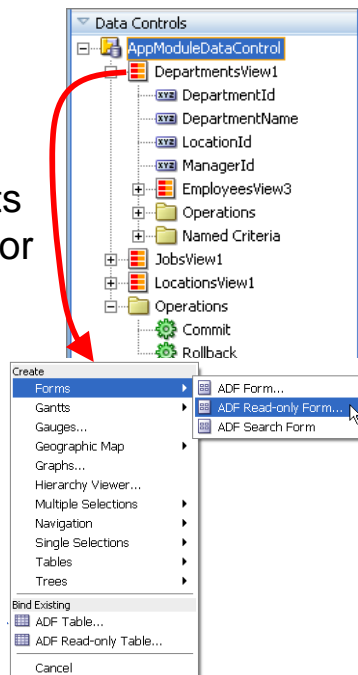
Agenda

- What is ADF?
- ADF Business Services
- ADF View, **Model**, and Controller



ADF Model

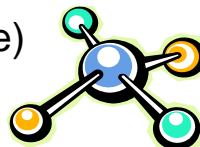
- ADF Data Controls
 - Provides list of components or groups of components for a node in the data model
 - “Drop as” options
- ADF Bindings
 - Prebuilt connection from the ADF BC to the UI
 - Drag and drop action above does the work



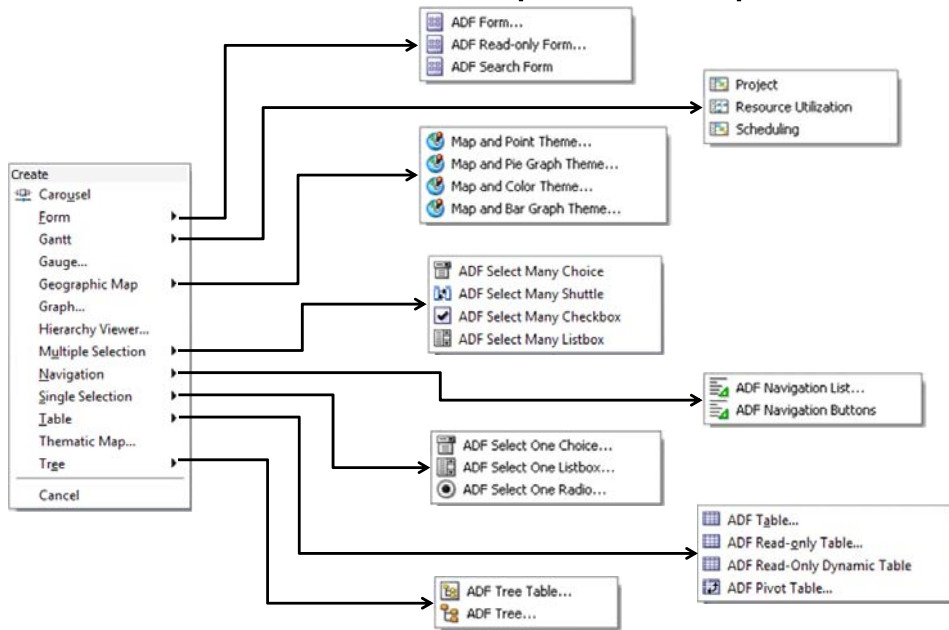
ADF Data Controls

- Business Services abstraction
 - Makes Model components available to ViewController
 - Automatically created with ADF BC
 - Can be created for other business services
 - For non-ADF BC, defined in DataControls.cpx
- Provide list of “Drop as” options that create pre-bound components
 - Collection level (view object instance)
 - Attribute level (view attribute)

The Good News:
You don't normally write data controls

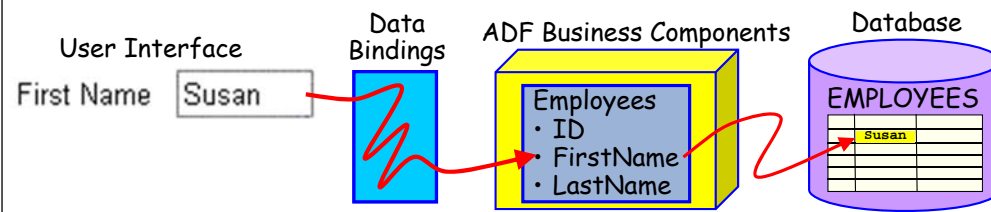


Collection "Drop As" Options



ADF Bindings

- Association of a business service data element or action with a UI component
 - Relatively automatic in Oracle Forms
 - Definitely not automatic in native Java EE
- Binding normally takes a lot of coding
 - One-off solution is not the answer
 - Need a framework to assist



Drop As Examples: Form and Table

Editing Bindings

Binding Code

JSF Page Snippet

```
<af:inputText value="#{bindings.DepartmentId.inputValue}"
  label="#{bindings.DepartmentId.hints.label}"
  required="#{bindings.DepartmentId.hints.mandatory}"
  columns="#{bindings.DepartmentId.hints.displayWidth}"
  maximumLength="#{bindings.DepartmentId.hints.precision}"
  shortDesc="#{bindings.DepartmentId.hints.tooltip}"
  id="it1">
</af:inputText>
```

Bindings (PageDef) Snippet

```
<bindings>
  <attributeValues IterBinding="DepartmentsView1Iterator"
    id="DepartmentId">
    <AttrNames>
      <Item Value="DepartmentId"/>
    </AttrNames>
  </attributeValues>
```

Agenda

- What is ADF?
- ADF Business Services
- ADF View, Model, and **Controller**

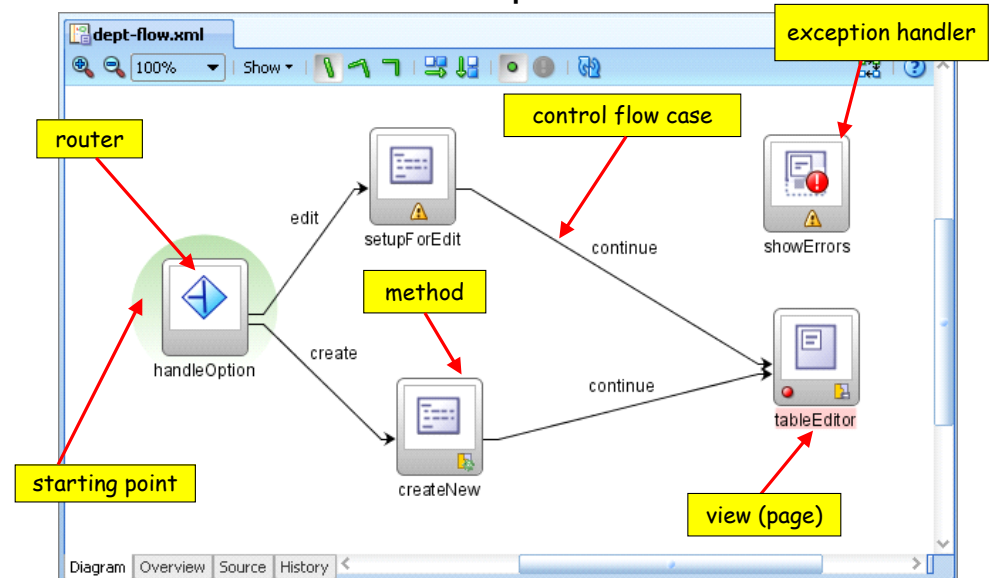


ADF Controller (ADFc)

- Extension to standard JSF Controller functionality
- Defines *task flows*
 - Logic and page fragment components
 - Embedded on the page in a region component
- Benefits
 - Page fragment re-use
 - Executing code in a logic-defined flow
 - “Task flow” not “page flow”
 - Security
 - Exception handling and transaction management
- Defined in a diagram



Sample ADF Controller Development



Sample ADF Controller Code

```

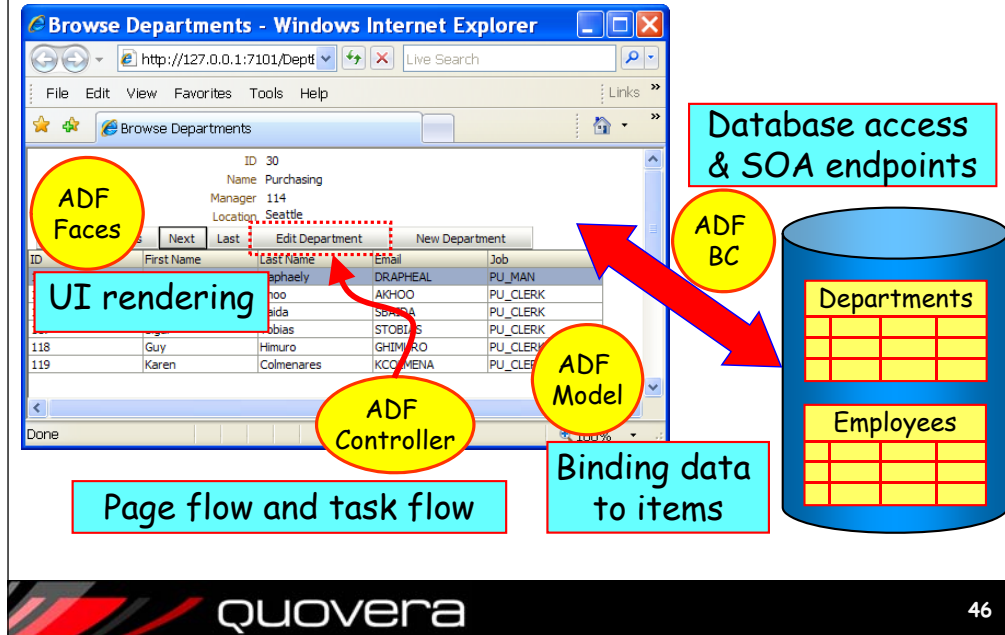
<task-flow-definition id="dept-flow">
<default-activity>deptBrowse</default-activity>
<view id="deptBrowse">
  <page>/deptBrowse.jspx</page>
</view>
<view id="deptEdit">
  <page>/deptEdit.jspx</page>
</view>
<control-flow-rule>
  <from-activity-id>deptBrowse</from-activity-id>
  <control-flow-case>
    <from-outcome>toEdit</from-outcome>
    <to-activity-id>deptEdit</to-activity-id>
  </control-flow-case>
</control-flow-rule>
<router id="checkForExplicitID">
  <case id="__6">
    <expression>#{!empty pageFlowScope.employeeId}
    </expression>
    <outcome>byId</outcome>
  </case>
  <default-outcome>currentUser</default-outcome>
</router>
<method-call id="queryEmployeeById">
  <method>#{bindings.queryEmployeeById.execute}</method>
  <outcome>
    <fixed-outcome>queryEmployeeById</fixed-outcome>
  </outcome>
</method-call>

```

From
To

Demo

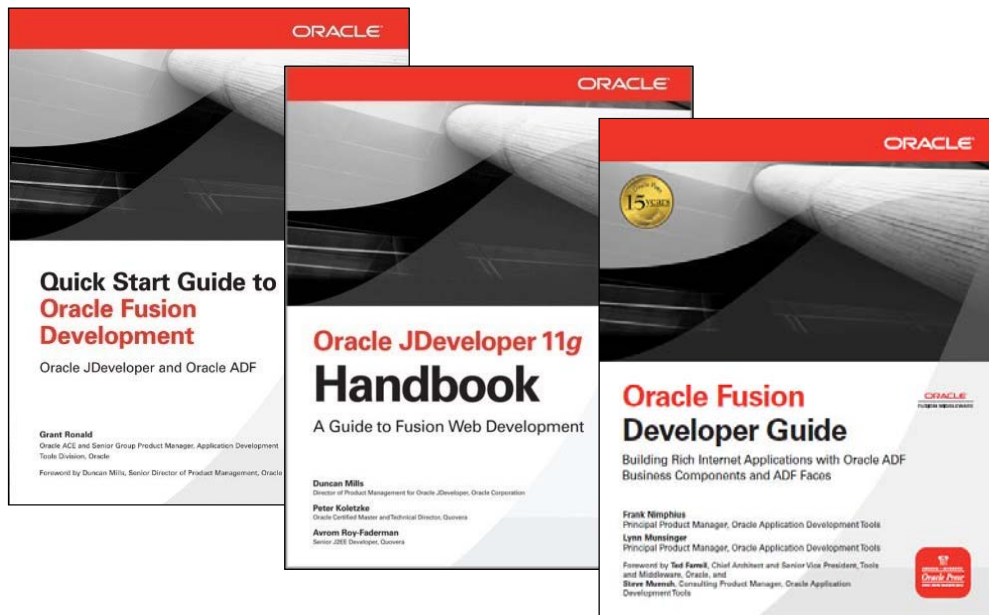
Summary: ADF Core Technologies



QUOVERA

46

Shameless Book Plugs



QUOVERA

47

Final Voltaire Wisdom

The secret of being a bore is to tell everything.

Le secret d'ennuyer est celui de tout dire.

—Voltaire (1694-1778), *Sept Discours en Vers sur l'Homme*

QUOVERA

48

Summary

- Oracle is building Fusion Applications with an ADF core technology stack
- ADF offers a consistent developer experience regardless of the technologies
- ADF Business Components provide access to the database and other data sources
- ADF Model connects ADF BC to ADF Faces
- ADF Faces provide 150+, feature-rich item and container components for JSF pages
- ADF Controller manages page flow and task flow



- Books co-authored with Dr. Paul Dorsey, Avrom Roy-Faderman, & Duncan Mills



www.quovera.com

- Founded in 1995 as Millennia Vision Corp.
- Profitable without outside funding
- Consultants each have 10+ years industry experience
- Strong High-Tech industry background
- 200+ clients/300+ projects
- JDeveloper Partner
- More technical white papers and presentations on the web site