

# ADF On-Ramp: What You Need to Know to Use ADF

Peter Koletzke  
Technical Director &  
Principal Instructor



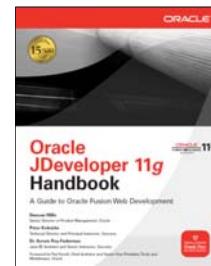
quoverta



## Agenda

- What is ADF?
- ADF core technologies
- Required languages

Slides and award-winning white paper will be available on the NYOUG and Quoverta websites.



quoverta

## Survey

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



quoverta

2

## 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*

3

quoverta

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
  - Available starting in JDeveloper 10g
  - Provides a consistent developer experience
- Pre-ADF available in OAF
  - Oracle Application Framework (UIX/MVC)
- Based on Model-View-Controller Java EE design pattern



QUOVERA

5

## ADF Essentials

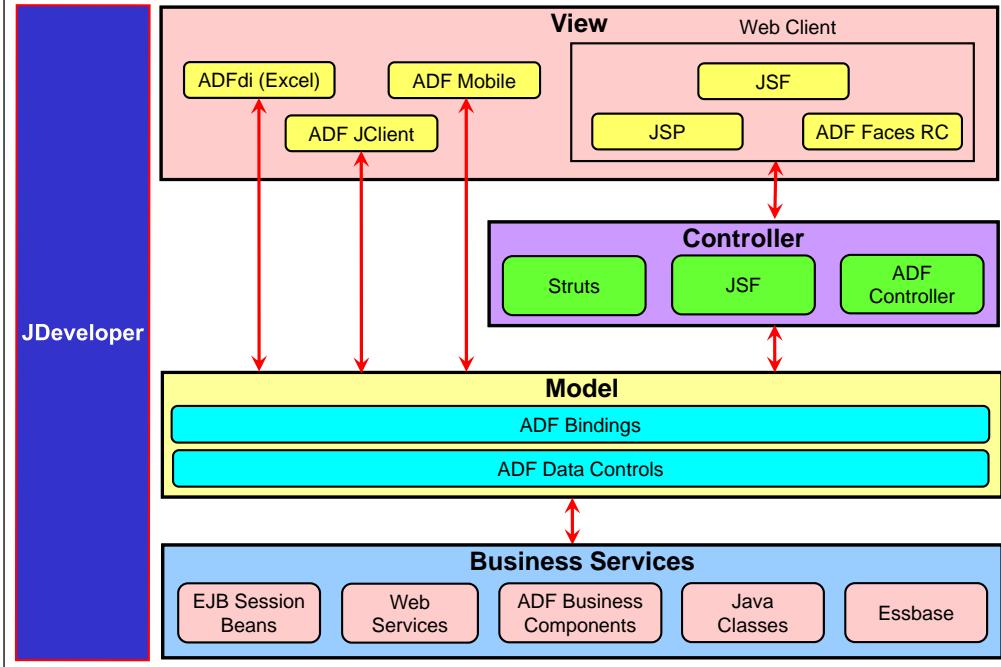
- No-license-fee version of ADF
  - Runs on the public domain app server, Glassfish, not WebLogic Server
- Works in JDeveloper
- Works in Eclipse
  - Through Oracle Enterprise Pack for Eclipse
- More information on OTN
  - <http://www.oracle.com/technetwork/developer-tools/adf/overview/adfessentialsfaq-1837249.pdf>



QUOVERA

7

## ADF Architecture



## Which ADF Technologies to Use?

- *Fusion Applications*: Oracle application 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, at the architectural level

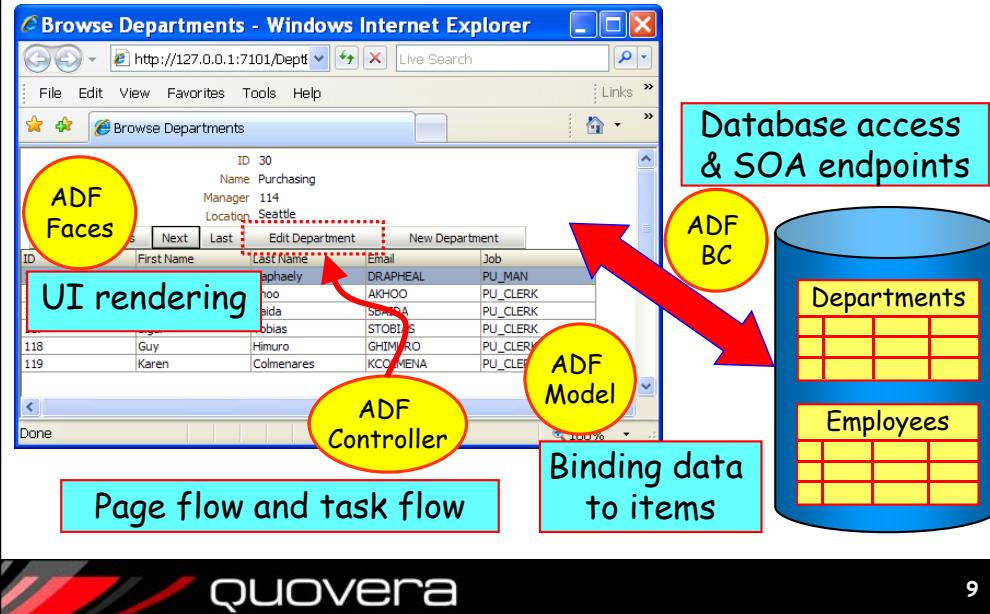
but OOS



QUOVERA

8

# Where Do The Core Technologies End Up?



## 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*

## Agenda

- What is ADF?

- ADF core technologies

ADF BC  
ADF Faces  
ADF Model  
ADF Controller

- Required languages



10

## 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 thingies 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
  - 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



QUOVERA

13

## View Object Code

View Object = SELECT statement

```
<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">
  ...
</ViewObject>
```

View Attribute = Column in query

```
<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"
  ...
</Attribute>
```

QUOVERA

15

# Sample ADF BC Development

Entity object editors

QUOVERA

## Agenda

- What is ADF?

- ADF core technologies

ADF BC  
ADF Faces  
ADF Model  
ADF Controller

- Required languages



QUOVERA

16

## ADF Faces Rich Client Overview

- Fits into the View layer of ADF
- Evolution:
  - 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



QUOVERA

17

## Some Components

The screenshot shows a JDeveloper 11g interface with several ADF Faces components highlighted:

- af:inputText**: A text input field with the value "198".
- af:inputListOfValues**: A list of values component with the value "50".
- Help**: A help button.
- af:menuItem**: A menu item labeled "Using the application" and "About TUHRA".
- af:inputDate**: A date input field showing "6/21/1999".
- af:commandButton**: A command button labeled "Last".
- af:commandImageLink**: A command image link labeled "Logon".
- af:selectOneChoice**: A select one choice component showing a list of job titles. The "Shipping Clerk" option is selected.
- af:inputText**: Another text input field with the value "I Love JDev 11g".
- af:selectBooleanCheckbox**: A checkbox labeled "I Love JDev 11g".
- af:inputText**: A third text input field with the value "af:inputText".
- af:inputList**: A list component showing a grid of cereal boxes, specifically Chex cereal.

QUOVERA

19

## ADF Faces RC Features

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



QUOVERA

18

## AJAX in ADF Faces RC

- Asynchronous JavaScript and XML
- **Partial Page Rendering (PPR)** in ADF Faces
  - “Declarative AJAX”
- Much AJAX in ADF Faces is transparent
  - Built into the components
  - Nothing special needs to be done
- You can setup non-default AJAX behavior using properties
  - *partialSubmit* – used by command items
  - *autoSubmit* – used by input items/lists, etc.
  - *partialTriggers* – all components, sets up the “viewer” (listener)



AJAX provides a cleaner user interface!



QUOVERA

20

# Sample ADF Faces Development

The screenshot shows the Oracle JDeveloper IDE interface. The top bar includes tabs for 'deptBrowse.jspx', 'Show', 'Full Screen Size', 'None', 'Default', and 'None'. Below the toolbar is a code editor window containing JavaServer Pages (JSP) code. The code includes EL expressions like `\${...DepartmentId...label}` and `\${bindings.DepartmentId.inputValue}`. To the right of the code editor is the 'Property Inspector' panel, which displays properties for a selected 'Output Text' component. The 'Common' tab is selected, showing fields for 'Id' (set to 'o3'), 'Value' (set to `\${bindings.DepartmentId.inputValue}`), and 'Style'. Other tabs include 'Appearance', 'Behavior', and 'Advanced'. At the bottom of the IDE are tabs for 'Design', 'Source', 'Bindings', 'Preview', and 'History'. The status bar at the bottom shows 'jsp:root > f:view >' followed by navigation links for 'Design', 'Source', 'Bindings', 'Preview', and 'History'.

QUOVERA

21

## Agenda

- What is ADF?

### • ADF core technologies

ADF BC  
ADF Faces  
ADF Model  
ADF Controller

- Required languages



QUOVERA

23

## 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="Logout" 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>
        </af:panelBorderLayout>
    </f:facet>
</af:panelStretchLayout>
```

QUOVERA

22

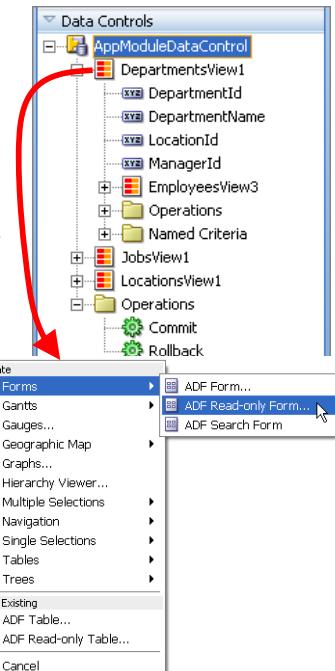
## ADF Model

### • 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



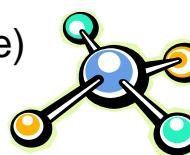
QUOVERA

24

# 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



## Drop As Examples: Form and Table

deptBrowse.jspx

```
#...DepartmentId..label) #...DepartmentId.inputValue
#...DepartmentName..label) #...DepartmentName.inputValue
#...ManagerId..label) #...ManagerId.inputValue
#...LocationId1..label) #...LocationId1.inputValue
```

Note the binding expressions

First Previous Next Last Edit Department New Department

|                       |                      |                     |                  |                  |
|-----------------------|----------------------|---------------------|------------------|------------------|
| #...EmployeeId..label | #...FirstName..label | #...LastName..label | #...Email..label | #...JobId..label |
| #...EmployeeId        | #...FirstName        | #...LastName        | #...Email        | #...JobId        |
| #...EmployeeId        | #...FirstName        | #...LastName        | #...Email        | #...JobId        |
| #...EmployeeId        | #...FirstName        | #...LastName        | #...Email        | #...JobId        |

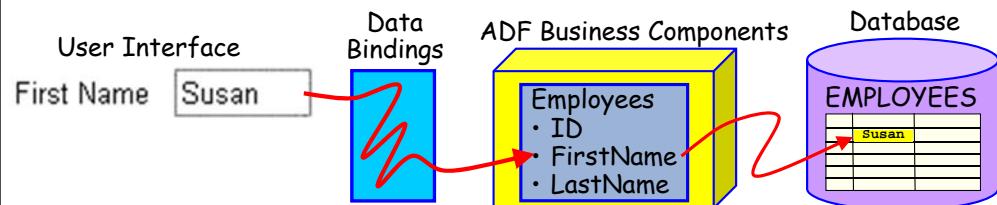
contextMenu

jsp:root > f:view >

Design Source Bindings Preview History < >

# 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



## Editing Bindings

searchEmp.jsff

Find

Page Data Binding Definition

This shows the Oracle ADF data bindings defined for your page. Select a binding to see its relationship to the underlying Data Control.

Page Definition File: fragments/searchEmpPageDef.xml

Link to PageDef file

Bindings and Executables Contextual Events Parameters

Bindings

- EmployeeSearch
- EmployeeId
- FirstName
- LastName
- Email
- PhoneNumber
- DepartmentName
- JobTitle

Executables

- variables
- EmployeeSearchItera
- BasicEmployeeSearch

Data Control

- TuhraServiceDataControl
- DepartmentEdit
- EditEmployee
- EmployeeSearch

# Binding Code

## In the JSF page file

```
<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>
```

## In the bindings PageDef file

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

# 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
  - Like JSF but more components available



# Agenda

- What is ADF?

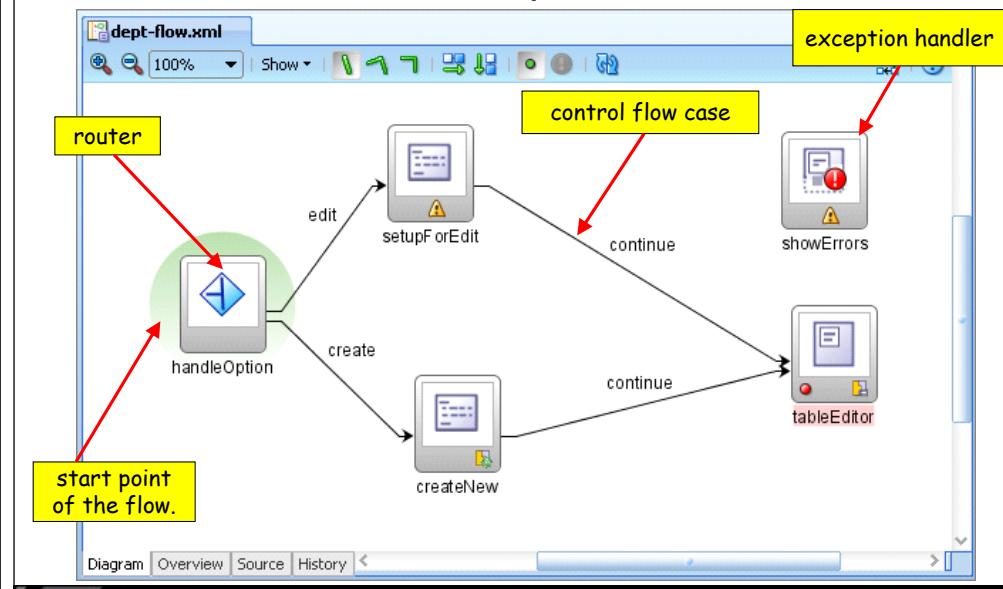
- ADF core technologies

ADF BC  
ADF Faces  
ADF Model  
ADF Controller

- Required languages



# Sample ADF Controller Development

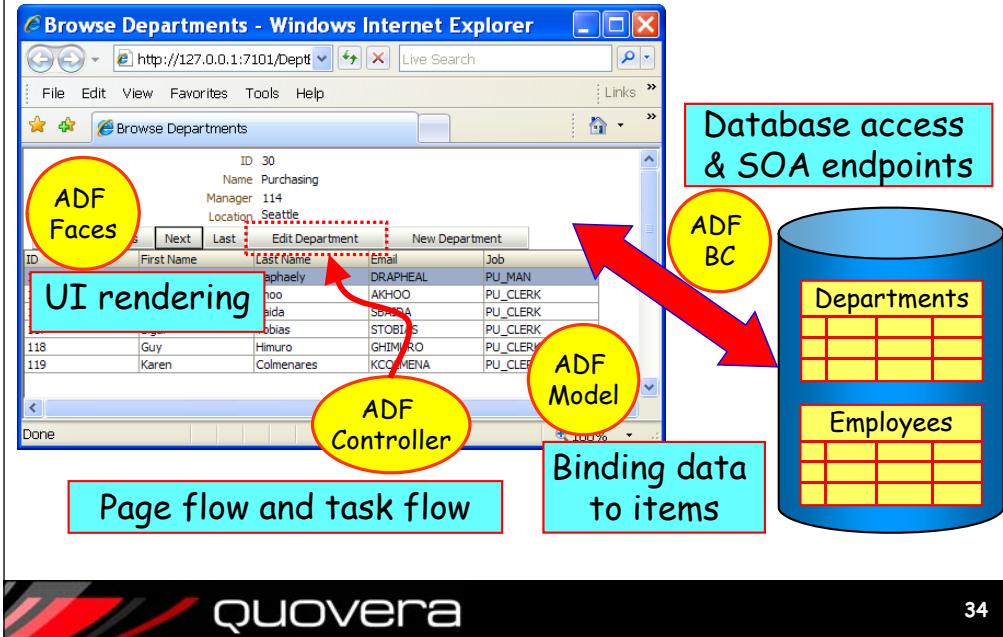


# Sample ADF Controller Code

```
<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>
```



# Summary: ADF Core Technologies



34

## Agenda

- What is ADF?
- ADF core technologies
- Required languages



## Which Languages Do You Use?

1. Java
  - All important programmatic code
  - Think “trigger code” as in 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<br>(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<br>(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         | For ensuring 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  |

## 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



## 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

38



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

Quovera

[www.quovera.com](http://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

Quovera

39

Quovera

40