ADF On-Ramp: What You Need to Know to Use the ADF Fusion Technology Stack

Peter Koletzke
Technical Director & Principal Instructor

Survey

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

Agenda

- What is ADF and Fusion?
- ADF Business Components
- JSF and ADF Faces
- ADF Model
- ADF Controller
- Required languages

Part 1

Part 2

Raffle at the End

Oracle JDeveloper 11g Handbook
A Guide to Fusion Web Development

Slides will be available on the NYOUG and Quovera websites. White paper will be published in the NYOUG Tech Journal.
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

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 the “plumbing”
• Pre-ADF available in OAF
• Based on Model-View-Controller Java EE design pattern

ADF Architecture

ADF Development Experience

• Visual
  • Visual editors (WYSIKOWYG)
  • UML diagrams
    – Class diagram
    – Activity diagram
    – Use Case diagram
    – Sequence diagram

ADF Bindings
ADF Data Controls
EJB Session Beans
Web Services
ADF Business Components
Java Classes
Portlets
ADF Development Experience

- **Declarative**
- **Property Inspector**
- **Structure window**
- **Expression Builder**
- **Binding editors**
- **Property editors**
  - For example, ADF BC

---

The Con-fusion

Welcome to the website of Fusion Restaurant, The Waterfront, Sausalito, Co. Cork.

FORD FUSION

The most seamless way to run Windows on your Mac.

SNL Shimmer commercial, 1/10/76

---

Where Do Fusion Developers Live?

140 Ellis St, San Francisco, CA

Photo courtesy of Sten Vesterli (vesterli.com)

---

What is Fusion?

- "It's a floor wax and a dessert topping!"
- **Fusion Applications**
  - Next version of E-Business Suite (Apps)
- **Fusion Middleware**
  - Tools for building and running the applications (and your custom apps)
- **Fusion Architecture**
  - How to assemble various technologies to build FA
  - How to connect FM pieces
Which Fusion Technologies to Use?

- Core technologies used to create Fusion Applications
  - ADF Business Components
  - ADF Faces Rich Client
  - ADF Controller

- Other high-level technologies or strategies also used
  - SOA, ESB, Business Rules, WebCenter
  - Need to look into those, too, at the architecture level

Sample App: The Ultimate Human Resources Application (TUHRA)
Hierarchy Viewer Page

Accounting
  Managed by: Shelley Rogers

Employee 205
Name: William Gritz
Email: kcg127.t7
Phone: 951.123.8181
Title: Public Accountant

Employee 206
Name: Shelley Rogers
Email: kcg127.t7
Phone: 951.123.8069
Title: Accounting Manager

Where Do The Fusion Technologies Show Up?

Sample Page Flow

Database access

UI rendering

Page and page fragment flow

ADF Faces

ADF Controller

ADF BC

Employees

Departments
**Agenda**

- What is ADF and Fusion?
  - ADF Business Components
- JSF and ADF Faces
- ADF Model
- ADF Controller
- Required languages

**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*

**Review: ADF Business Services**

- A layer of ADF
  - ADF Business Components is one option
  - Formerly known as Business Components for Java (BC4J)
- Provides access to database objects in an OO way
  - Uses JDBC to communicate with database
    - Java Database Connectivity
    - No SQL*Net needed

**ADF Business Components**

- ADF BC
- Persistence
  - Storing data in and retrieving data from a database
- O/R mapping
  - Translating relational database thingies to object-oriented (Java) whatsits
- Handles mechanics of creating SQL statements and of communicating with the database
More About ADF BC

- Bridges the gap between OO code and a relational database
- Objects that define queries
- Objects that define insert-update-delete
- Objects that link queries
- Objects that link “DML”
- ADF BC forms SQL statements
  - Uses XML definitions of specific database objects
- It does not create user interfaces

Sample ADF BC Development

Note: All screenshots show JDev 11g.

SELECT = View Objects

- Retrieve data from the database
- Manage caches of data
- **View object definitions**
  - Contain SQL queries
  - Act as templates for view object instances
- **View object instances**
  - The “thing” used in code
  - Technically, an object of the class

View Object Editor

Data for this view object will be retrieved from the datasource using the following SQL query:

```
SELECT 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
FROM Departments, Employees
WHERE Departments.DEPARTMENT_ID = Employees.DEPARTMENT_ID
```

Named bind variables can be used in the SQL query of this view object.
**View Object Code**

```xml
<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="true" PrecisionRule="true" EntityAttrName="EmployeeId" EntityUsage="Employees" AliasName="EMPLOYEE_ID" />
</ViewObject>
```

**ADF View Links**

- A view link definition adds an additional WHERE clause to the detail
  - Source is the master; destination is the detail
- For example, a department and its employees

```sql
SELECT D.DEPARTMENT_ID, D.DEPARTMENT_NAME, D.MANAGER_ID
FROM DEPARTMENTS D
WHERE D.MANAGER_ID = :1
```
```
SELECT E.EMPLOYEE_ID, E.LAST_NAME
FROM EMPLOYEES E
WHERE E.DEPARTMENT_ID = :1
```

**View Link Code**

```xml
<ViewLink>
  <xmlns=http://xmlns.oracle.com/bc4j" Name="EmpDeptFkLink" Version="11.1.1.53.41" EntityAssociation="hr.model.EmpDeptFkAssoc"
  DesignTime>
    <Attr Name="_isCodegen" Value="true"/>
  </DesignTime>
  <ViewLinkDefEnd Name="DepartmentsView" Cardinality="1" Source="true">
    <DesignTime>
      <Attr Name="finderName" Value="DepartmentsView"/>
      <Attr Name="isUpdateable" Value="true"/>
    </DesignTime>
    <AttrArray Name="Attributes">
      <Item Value="hr.model.DepartmentsView.DepartmentId"/>
    </AttrArray>
  </ViewLinkDefEnd>
  <ViewLinkDefEnd Name="EmployeesView" Cardinality="-1" Source="true">
    <DesignTime>
      <Attr Name="finderName" Value="EmployeesView"/>
      <Attr Name="isUpdateable" Value="true"/>
    </DesignTime>
    <AttrArray Name="Attributes">
      <Item Value="hr.model.EmployeesView.DepartmentId"/>
    </AttrArray>
  </ViewLinkDefEnd>
</ViewLink>
```

**Application Modules**

- Application modules represent the interface from business components to the application
  - The application always references the application module for data
  - Each instance (user) of the application will get its own application module instance
- The application module instance contains the data model
  - View object instances
  - View link instances for master-detail relationships
**INSERT, UPDATE, DELETE = Entity Objects**

- **Entity object definitions**
  - Correspond to database tables (or views)
  - Templates for *entity object instances*, which correspond to table rows
- **Entity object instances**
  - Write data to the database
  - Validate data before updates and inserts
  - Supply default values

---

**Why Declarative?**

- Arguably more productive
- Leverages native strengths of the frameworks
  - All frameworks can be extended
  - The less you extend the better
  - You can blame the framework for problems
- Version upgrades are potentially easier
  - You take advantage of framework bug fixes automatically
  - Same concept as database upgrades

---

**Entity Object Code**

```xml
<Entity xmlns="http://xmlns.oracle.com/bc4j"
  Name="Departments"
  Version="11.1.1.53.41"
  DBObjectType="table"
  DBObjectName="DEPARTMENTS"
  AliasName="Departments"
  BindingStyle="OracleName"
  UseGlueCode="false">
  <DesignTime>
    <Attr Name="_codeGenFlag2" Value="Access"/>
    <AttrArray Name="_publishEvents"/>
  </DesignTime>
  <Attribute Name="DepartmentId" IsNotNull="true"
             Precision="4"
             Scale="0"
             ColumnName="DEPARTMENT_ID"
             SQLType="NUMERIC"
             Type="oracle.jbo.domain.Number"/>
  <Properties>
    <SchemaBasedProperties>
      <LABEL ResId="hr.model.Departments.DepartmentId_LABEL"/>
    </SchemaBasedProperties>
  </Properties>
</Attribute>
```

---

**Agenda**

- What is ADF and Fusion?
- ADF Business Components
  - JSF and ADF Faces
- ADF Model
- ADF Controller
- Required languages
Now, a Word from Our Sponsor

The superfluous, a very necessary thing.

—Voltaire (1694-1778), *Le Mondian*

Review: ADF View

- A layer of ADF
  - Web client is one option
  - JSF is web client code
- The code that constructs the user interface
  - ADF Faces Rich Client is a set of JSF components used to render the UI
  - View layer receives data from and updates data to the Model layer, through the Controller layer

---

JSF Overview

- JavaServer Faces
  - Java EE standard
- Simplifies web development
  - No embedded Java inside tag language
  - High-level tags
- XML-like
  - Start and end tag required
  - Elements, commonly called “components”
  - Attributes

ADF Faces Rich Client Overview

- 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
- Rich feature set for customizing applications
- Fully declarative AJAX support (more later)
- 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!
JSF Snippet

```xml
<jsp:root xmlns:jsp="http://java.sun.com/JSP/Page" version="2.0"
  xmlns:af="http://xmlns.oracle.com/adf/faces/rich">
  <af:panelStretchLayout styleClass="AFVisualRoot" topHeight="105px"
    bottomHeight="20px">
    <f:facet name="top">
      <af:image source="/images/tuhra.gif" shortDesc="TUHRA Logo"/>
    </f:facet>
    <f:facet name="end">
      <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}"/>
    </f:facet>
  </f:facet>
</af:panelStretchLayout>
```

ADF Faces RC Features

- Solid development support in JDeveloper
- Changeable “skins”
  - Look-and-feel characteristics
  - Can be changed with one config property
- Layout management features
- Extensive set of properties
  - Supports declarative programming techniques
  - Property Inspector, Visual Editor

Some ADF Faces Components

- `af:inputText`
- `af:commandButton`
- `af:commandImageLink`
- `af:selectInputDate`
- `af:menuItem`
- `af:inputListOfValues`
- `af:selectOneChoice`
- `af:selectBooleanCheckbox`

AJAX

- Asynchronous JavaScript and XML
  - In ADF Faces: Partial Page Rendering (PPR)
- Update the page without refreshing the entire page
  - Smoother user experience
- Common Examples
  - Sorting a results table
  - Scrolling through a results table
  - Expanding a tree control

AJAX provides a cleaner user interface!
Using AJAX in ADF Faces RC

- Much AJAX in ADF Faces is transparent
  - Expanding a tree node / scrolling a table
  - Nothing special needs to be done
- Explicit AJAX attributes:
  - `partialSubmit` – Used by command items
  - `autoSubmit` – used by input items / lists etc.
  - `partialTriggers` – all components
- Example
  - Update line total when quantity or price change

AJAX Interactions – A Calculated Field

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price</td>
<td>$100</td>
</tr>
<tr>
<td>Value</td>
<td>#{bindings.listPrice.inputValue}</td>
</tr>
<tr>
<td>AutoSubmit</td>
<td>true</td>
</tr>
<tr>
<td>partialTriggers</td>
<td>Price</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tax</td>
<td>$17.50</td>
</tr>
<tr>
<td>Value</td>
<td>#{bindings.listPrice.inputValue * 0.175}</td>
</tr>
<tr>
<td>AutoSubmit</td>
<td>false</td>
</tr>
<tr>
<td>partialTriggers</td>
<td>Price</td>
</tr>
</tbody>
</table>

Sample ADF Faces Development

Agenda

- What is ADF and Fusion?
- ADF Business Components
- JSF and ADF Faces
  - ADF Model
- ADF Controller
- Required languages
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

Data Controls' "Drop As" Options

- Available options are offered for the data control type being dropped
  - Collections can include more than one attribute
  - Nested collections offer master-detail options (including master)
  - Attributes offer data UI items (text fields, list items, etc.)
  - Methods offer forms with a field for each parameter and a submit button
  - Operations offer buttons or links

Example: ADF Query Component

- Named Criteria "drop as" option
- Creates a component with Find mode capabilities
  - Allows pre-seeded and saved searches
  - Basic and advanced modes
  - Much more powerful than Forms QBE

Examples: Form and Table

Note the binding expressions
**Agenda**

- What is ADF and Fusion?
- ADF Business Components
- JSF and ADF Faces
- ADF Model
  - ADF Controller
- Required languages

---

**Review: Controller Layer**

- A layer of ADF
  - Used only for web client code
  - JSF controller is one option
  - ADF Task Flow Controller is another
- Determines which page or task loads next
- Transfers data from the View to the Model layer

---

**ADF Controller**

- Supplement to standard JSF Controller functionality
  - Declares which page to display next
    - Can be based on a condition
- Treats part of a page in the same way as a full page in normal JSF work
  - Only part of a page is rendered, the rest stays put
- Speeds up page processing
- Allows reuse of page parts
- Allows logic to be added to the flow
  - For example, conditional display of a page

---

**Bindings Code**

*In the JSF page file*

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

*In the bindings Page Def file*

```xml
<bindings>
  <attributeValues IterBinding="DepartmentsView1Iterator"
                   id="DepartmentId">
    <AttrNames>
      <Item Value="DepartmentId"/>
    </AttrNames>
  </attributeValues>
</bindings>
```
Task Flow Code

```xml
<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>
  <control-flow-rule>
    <from-activity-id>deptEdit</from-activity-id>
    <control-flow-case>
      <from-outcome>toBrowse</from-outcome>
      <to-activity-id>deptBrowse</to-activity-id>
    </control-flow-case>
  </control-flow-rule>
</task-flow-definition>
```

Sample ADF Controller Development

Agenda

- What is ADF and Fusion?
- ADF Business Components
- JSF and ADF Faces
- ADF Model
- ADF Controller
  - Required languages

The Laundry List

1. Java
   - All important programmatic code
   - Think “trigger code” as in Forms
2. HTML and XML
   - Tag languages
   - 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 Need?

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

### Summary

- Fusion is Oracle’s effort to merge application products and technologies
- Oracle is using the “Fusion Technology Stack” and ADF to build the next EBS – Fusion Applications
- ADF offers a consistent developer experience regardless of the technologies
- ADF Business Components provide access to the database and other data sources
- ADF Faces provide 150+, feature-rich item and container components for JSF JSP pages
- ADF Model connects ADF BC to ADF Faces
- ADF Controller manages page 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

http://www.quovera.com

- Founded in 1995 as Millennia Vision Corp.
- Profitable for 7+ years 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

Please fill out the evals
- Books co-authored with Dr. Paul Dorsey, Avrom Roy-Faderman, & Duncan Mills
- Personal web site: http://ourworld.compuserve.com/homepages/Peter_Koletzke