Oracle ADF Task Flow
Beyond the 10-Minute Demo

Presented by: John Jay King
Download this paper from: http://www.kingtraining.com
Session Objectives

• Learn how JDeveloper may be used to create ADF-based applications
• Use ADF Task Flow to control application execution
• Take advantage of ADF’s built-in application control (MVC)
Who Am I?

- John King – Partner, King Training Resources
- Oracle Ace ♠ & member Oak Table Network
- 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)
- Member of ODTUG (Oracle Development Tools User Group) Board of Directors
- Member of RMOUG (but I live in Arizona!)
Who Are You?

• Oracle ADF JDeveloper
  – New (< 1 yr)
  – 1-2 years
  – Over 2 years
• Oracle ADF Eclipse
• Oracle Forms Developer
• 4GL Developer (.NET, etc…)
• Java Developer
• All of the above
• None of the above
ADF Comes of Age

• Oracle Application Development Framework (ADF) is a Java-based development tool (much like Forms is a PL/SQL-based tool) designed to take full advantage of Java EE

• Java EE is one of the most widespread application environments today

• ADF’s 4GL features make application development easier than normal Java “coding”

• Oracle is rewriting their ERP stack as “Fusion Applications” using ADF; the already rich toolset gets richer every day
Oracle JDeveloper

• JDeveloper is a world-class, easy to use IDE
• JDeveloper goes beyond Java to include:
  – Oracle creation of mobile applications via checkbox
  – Oracle ADF modeling, business svcs, and GUI design
  – XML edit including Syntax Checking & Validation
  – SQL development with debugging of stored PL/SQL
  – UML Modeling and MDA (Model Driven Architecture)
  – Web Services development
  – ESB design
  – BPEL design
  – Portlets
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
JSF Life Cycle

- JSF (and ADF Faces) perform 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, when choosing where to store session variables and debugging it is useful to understand
• Even though JSF sought to simplify the API; 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

• Using ADF Faces is simple using JDeveloper:
  – Add ADF Faces components to layout containers
  – Add Application layout containers to describe user interface
  – Describe Task Flows
  – 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…
ADF Controller

- The ADF Controller extends the JSF controller and controls ADF’s MVC
- 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
What is MVC?

- The Model-View-Controller (MVC) pattern was first described by Glenn E. Krasner and Stephen T. Pope in the paper titled “A Description of the Model-View-Controller User Interface Paradigm in the Smalltalk-80 System” (1988)
MVC Pattern

• The MVC pattern separates functionality:
  – Model  Provides data services; changes to data management or business rules do not impact the view
  – View    Provides interface shown and/or delivered to the client; output changes don’t impact data
  – Controller Provides decision making and navigational control
ADF and MVC

• In ADF MVC is implemented via:
  – Model       ADF BC components
  – View        ADF Faces components
  – Controller  ADF/JSF Controller and Task Flows
JSF Task Flow

• JSF navigates declaratively (default file named “faces-config.xml”)
  – Navigation may be configured manually using XML
  – JSF Navigation may be configured graphically using JDeveloper's JSF Navigation diagrammer
ADF Task Flow

- ADF Task Flow (pageflow) navigation and transition is specified declaratively (default file named “adfc-config.xml”)
  - Navigation may be configured manually using XML
  - ADF Task Flow may be configured graphically using the JDeveloper ADF Task Flow diagrammer
Task Flow Diagrams

• Task Flow diagrams illustrate all or part of an application’s navigation
UnBounded vs Bounded

• ADF supports two types of Task Flows; Unbounded and Bounded

• Unbounded Task Flows
  – Multiple entry points and multiple exit points
  – No input parameters or return values
  – Page-oriented (initial page, help, menus, etc…)
  – Typically one per application; not reusable

• Bounded Task Flows
  – Single entry point and multiple exit points
  – Allows input parameters and return values
  – Process-oriented (checkout, calculate taxes, etc…)
  – Frequently many in an application; reusable objects
Using Task Flows

• Unbounded Task Flows
  – Top-level (bootstrap) flow
  – Navigation may begin anywhere in the flow using URL

• Bounded Task Flows
  – Process flow with defined boundary
  – Navigation must begin at defined point
  – Started via URL, other task flows, or Task Flow binding
  – When called from other task flows may use input parameters to pass data
  – When exiting may return values
  – Generally execute in an area of the parent page called a “Region”
ADF Region

- ADF Regions contain bounded task flows representing page fragments within a Page
Pages and Page Fragments

• Pages
  – Use .jsf or .jspx file suffix
  – May include many Page Fragments

• Page Fragments
  – Use .jsff suffix
  – Provide content for Pages (above)
ADF Task Flow Diagram

- Most applications use multiple web pages, the JDeveloper ADF Task Flow diagram is used to diagram movement through an application (sort of an electronic story-board)

- To diagram a “Page Flow”
  - Open the “adf-config” or “adfc-config” in an application's ViewController project (look under “Page Flows”)
  - Use “New ADF Task Flow”
Blank Task Flow

- Use the component palette to build flows by dragging existing pages or new components to diagram.
Component Palette

• The ADF Task Flow component palette contains tools used to create a task flow
<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Activity</td>
<td>Task Flow page</td>
</tr>
<tr>
<td>Control Flow Case</td>
<td>Navigation between two points in Task Flow</td>
</tr>
<tr>
<td>Method Call Activity</td>
<td>Call to method</td>
</tr>
<tr>
<td>Wildcard Control Flow Rule</td>
<td>Define activity access via known paths</td>
</tr>
<tr>
<td>Router Activity</td>
<td>Expression-based branching</td>
</tr>
<tr>
<td>Task Flow Return Activity</td>
<td>Pass control back to calling Task Flow</td>
</tr>
<tr>
<td>URL View Activity</td>
<td>Navigate to any page</td>
</tr>
</tbody>
</table>
Creating Task Flow

• Drag “View” components to represent pages
• Drag “Control Flow Rules” from the View where control begins to the View where control is to end
Component Review
View

- Displays one page (.jsf, .jspx, .jsff)
  - May have one or more Task Flow regions
  - Default view of Bounded Task Flow shaded
Control Flow

- Control flow describes flow between parts of a task flow
  - Control Flow properties describe actions to be taken
Passing Data

• Parameters and Return Values are generally passed using Expression Language (EL)
  #{controllerContext.currentViewPort.taskFlowContext.trainModel}

• ADF’s EL is similar to the JSP EL commonly used in Java
  – ADF’s EL usually uses the hash-mark/pound-sign (#) indicating a deferred reference
  – Deferred EL works within the JSF-cycle of ADF; immediate EL ($) may be evaluated prematurely
Input Parameter Example

• Here is an input parameter from a control flow
• Note the “pageFlowScope” in user for “username”
Servlet/JSF Memory Scopes

• Application Scope  Attributes/beans available globally to all users for life of application

• Session Scope  Attributes/beans available for life of browser session

• Request Scope  Attributes/beans available until application returns control to user (one page)
ADF Memory Scope

• Some memory scopes are ADF-specific

• View Scope
  Attributes/beans for single page for duration user works with page

• PageFlow Scope
  Attributes/beans live for duration of Task Flow (“pushed” if new Bounded Task Flow called; “popped” upon return)

• Backing Bean Scope
  Attributes/beans specific to specific component
ADF Variable Scope EL

#{applicationScope.varname}
#{sessionScope.varname}
#{requestScope.varname}
#{varname}
#{pageFlowScope.varname}
#{viewScope.varname}
#{backingBeanScope.varname}
Use Scope Wisely

• Use Application Scope for things that should live for the life of the server and that should be global
• Use Session Scope only when values should survive for the entire browser session duration
• Use Request Scope for data that is passed while processing a single page
• Use PageFlow Scope for data to be passed from screen-to-screen during a Task Flow
• Use View scope for value local to a given page
• Use Backing Bean scope for specific field data passed within a single page (like Request Scope)
Common Task Flow Practice

• Build Bounded Task Flows in separate applications to increase portability
• Use Task Flow within page fragments
• Use input parameters to pass data and provide conditional behavior
Task Flow Summary

• Unbounded Task Flows
  – Entry to application or where user might enter flow at any point
  – Menu pages

• Bounded Task Flows
  – Has single entry point
  – Part of a page within a Region
  – Used/Reused by other applications
  – Usually performs commit/rollback on exit
• The ADF EMG (Enterprise Methodology Group) has created a tool for testing ADF Task Flows
  http://java.net/projects/adf-task-flow-tester
Wrapping it all Up

- Oracle ADF provides a mostly-declarative capability to create applications easily using industry-standard technology “under the covers”
- Oracle ADF Task Flows improve on JSF Task Flows and allow declarative management of MVC application flow via GUI interface and XML files
- Bounded ADF Task Flows may be reused; processes that appear in several applications (e.g. “checkout”) are not repeated needlessly
- Bounded ADF Task Flows may use input parameters and produce result values; EL is most often used to simplify and increase reusability
Support on the Web

• OTN provides a great deal of support for ADF
  – See the Oracle ADF Architecture Square on Learn More tab - Architecture Square has links to suggested coding, project layout, and naming guidelines
  – Community tab has links to blogs and other community support

• ODTUG is the non-Oracle home for ADF developers
  http://www.odtug.com
  http://www.kscope13.com
Available Books

• 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 JDeveloper 11gR2 Cookbook
  – Nick Haralabidis
  – Packt Publishing
Training Days 2014

2 Days of inexpensive Oracle-related training in Denver!!

February 6-7

February 5: University day: More low-cost training!
Check the website for details

www.rmouhg.org
See you in Denver Colorado!

COLLABORATE 13

TECHNOLOGY AND APPLICATIONS FORUM
FOR THE ORACLE COMMUNITY

April 2013 – Get Ready to Go!
ODTUG
Kscope13
NEW ORLEANS, LA • JUNE 23-27

Topics
Application Express
ADF and Fusion Development
Developer’s Toolkit
The Database
Building Better Software
Business Intelligence
Essbase
Planning
Financial Close
EPM Reporting
EPM Foundations & Data Management
EPM Business Content

ODTUG - Leading Developers into the Future with Oracle Tools

www.kscope13.com
Oracle ADF Task Flow
Beyond the 10-Minute Demo

To contact the author:

John King
King Training Resources
P. O. 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