



LMI

GOVERNMENT CONSULTING

THE OPPORTUNITY TO MAKE A DIFFERENCE HAS NEVER BEEN GREATER



Developing Web Applications With Oracle ADF and ADF Faces

Rick Wang

Agenda

- About Developing Web Applications
- About ADF
- About ADF Model
- About ADF Faces
- Demonstration
- Q&A

Agenda

- **About Developing Web Applications**
- About ADF
- About ADF Model
- About ADF Faces
- Demonstration
- Q&A

About Developing Web Applications

- Web Standards
 - HTML, XHTML, Javascript, CSS
- Microsoft
 - ASP
 - ASP.NET
- Java
 - Servlet
 - JSP
 - Custom Tag Libraries
 - JSTL
 - JSF



Agenda

- About Developing Web Applications
- **About ADF**
- About ADF Model
- About ADF Faces
- Demonstration
- Q&A

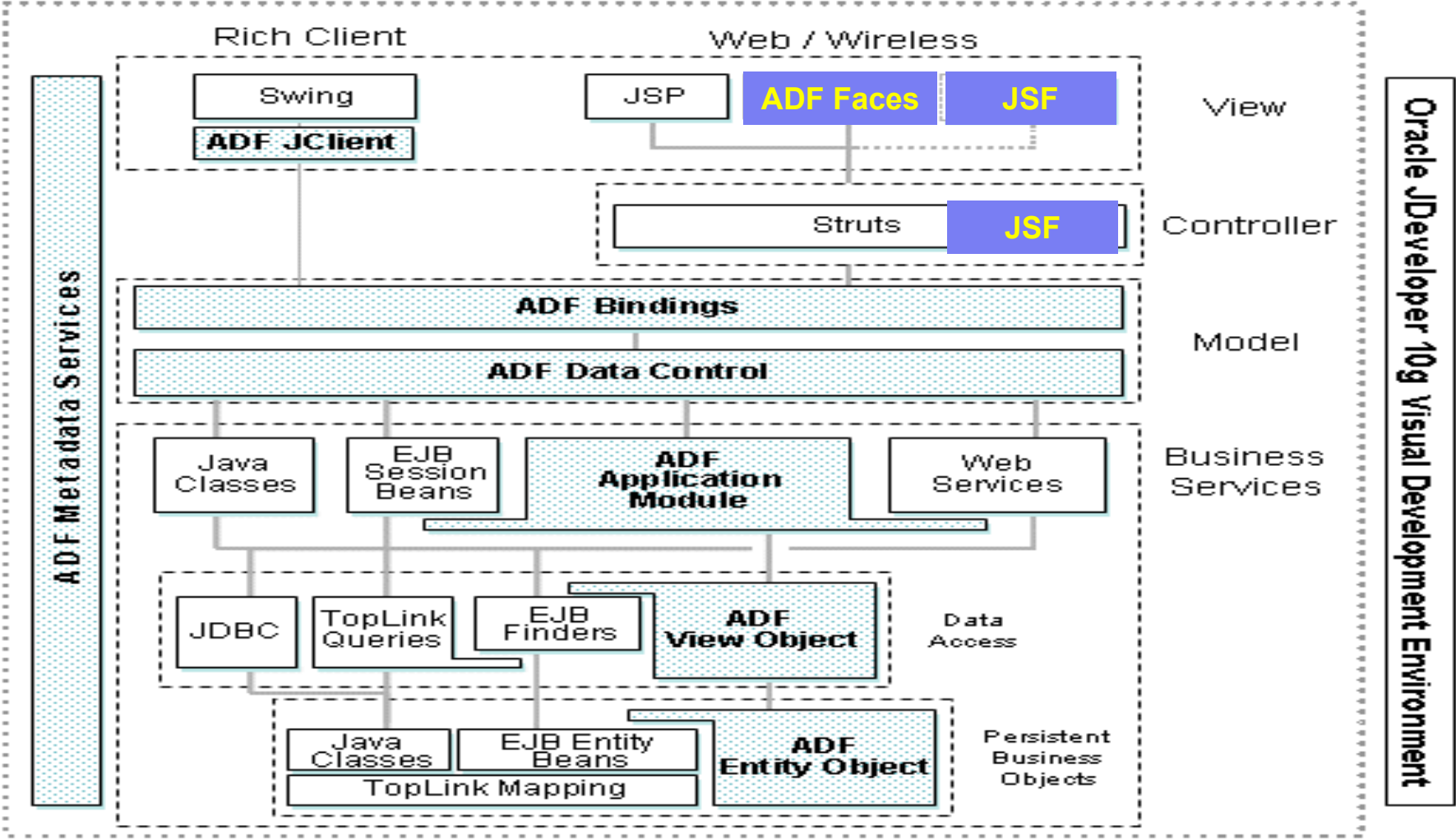
About Oracle ADF

Application Development Framework (ADF)

- Simplify developing J2EE applications
- Increase development productivity
- Provide a flexible, extensible environment
- Encourage J2EE best practices



Oracle ADF Architecture



ADF Business Components (ADF BC)

- Entity Objects
 - Encapsulate business rules, logic, defaults for a table
- View Objects
 - Encapsulate SQL queries, Join, Filter, Order data for external client interaction
- Application Module
 - Defines data model of view object usages for an application



Agenda

- About Developing Web Applications
- About ADF
- **About ADF Model**
- About ADF Faces
- Demonstration
- Q&A

ADF Model Layer

- Provides a client-side wrapper and abstraction of your business service's data model and operations.
- The view and controller accesses business services through the model layer
- Manages and presents data from different Business service types in a common way



Benefits of the ADF Model Layer

- A layer of abstraction
- A consistent design-time experience
- Databinding for any type of business service
- Support for standards (JSR227)
- Fully extensible framework



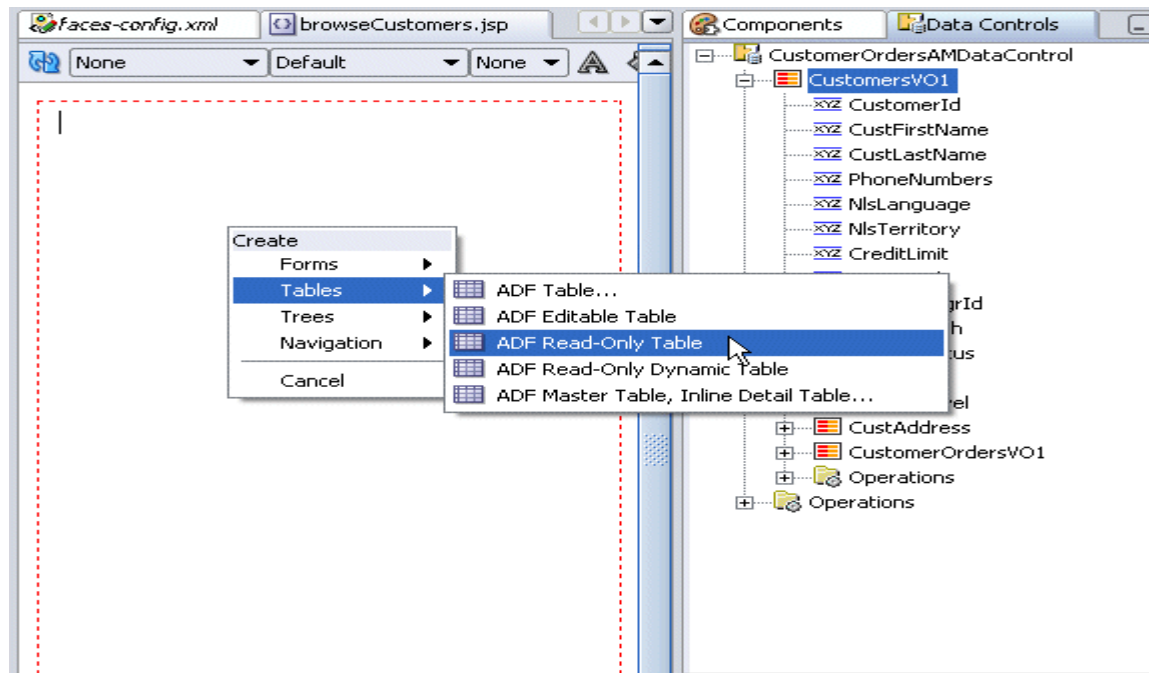
Components of the ADF Model Layer

- ADF Data controls
 - Metadata describing the data model returned by the Business Service.
 - Middle-tier-facing components
- ADF Bindings
 - Metadata describing how the UI components on a page use the values and actions provided by the Business Service.
 - Client-facing components



ADF Model Design Time

- Data Controls Palette
 - Exposes the data model.
 - Shows the available UI components.
- Drag and drop collections, objects and operations.



ADF Model Metadata and Objects

- ADF Model Metadata
 - Are generated at design time
 - Are specific to your application
- ADF Model Objects
 - Are instantiated at run time
 - Pull application-specific information from metadata



ADF Model Objects

- **Binding context**
 - The handle through which the client accesses the data binding layer
- **Binding container**
 - A container for all binding objects per page
 - Control binding
 - Defines the interaction between a UI component and the business service
 - Two types: value binding and action binding
- **Data control**
 - A client-side interface to the business services



Agenda

- About Developing Web Applications
- About ADF
- About ADF Model
- **About ADF Faces**
- Demonstration
- Q&A

ADF Faces

- JSF components
- Support Variety of Clients
 - Web Browsers, Mobile Devices
- Rich Components, Layouts, and Render Kits
 - Over 100 Components
- Advanced and interesting components
 - Table and Tree
- Powerful Runtime Features
 - Partial Page Rendering (PPR)
 - Look and Feel "Skinning"

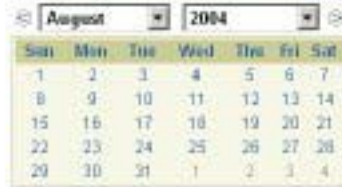
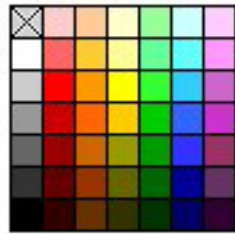


ADF Faces

- Deployable on any compliant implementation of JSF (e.g. JSF RI, MyFaces)
- Provides much of what is missing in JSF 1.1
 - The component set is small
 - No client-side behavior
 - No support for customization
 - Little direct support for working with databases
- Accessibility and internationalization support across the board



ADF Faces Components Examples



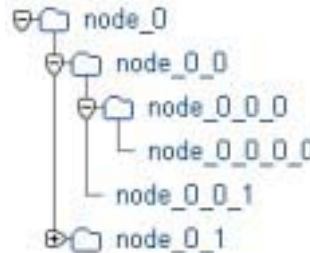
Label



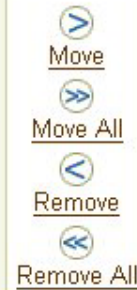
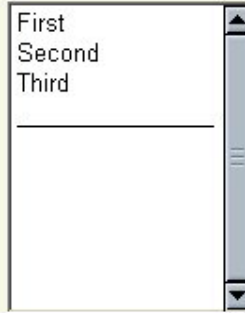
Previous 1-5 of 18 Next 5

Show All Details | Hide All Details

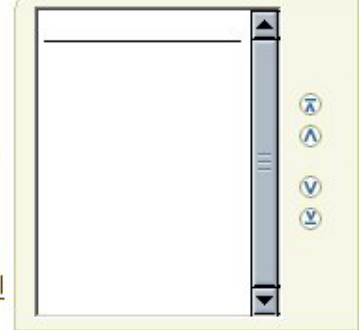
Details	Int	String	Operation
Show 0	String 0		<input type="button" value="Action"/>
Show 1	String 1		<input type="button" value="Action"/>
Show 2	String 2		<input type="button" value="Action"/>
Show 3	String 3		<input type="button" value="Action"/>
Show 4	String 4		<input type="button" value="Action"/>



Available values:



Selected values:



Expand All | Collapse All

0 >

Focus Name	Parent	Index	Depth
AC	A	1	1
ACA	AC	0	2
ACC	AC	1	2
ACCA	ACC	0	3
ACCC	ACC	1	3
ACCG	ACC	2	3
ACCT	ACC	3	3
ACG	AC	2	2
ACT	AC	3	2



Label and message support

Using JSF RI

```
<h:panelGrid columns="2">  
  <h:outputLabel for="nameInput" value="Name: ">  
  <h:panelGroup>  
    <h:inputText id="nameInput" value="Enter a name"/>  
    <f:verbatim><br></f:verbatim>  
    <h:message for="nameInput"/>  
  </h:panelGroup>  
</h:panelGrid>
```

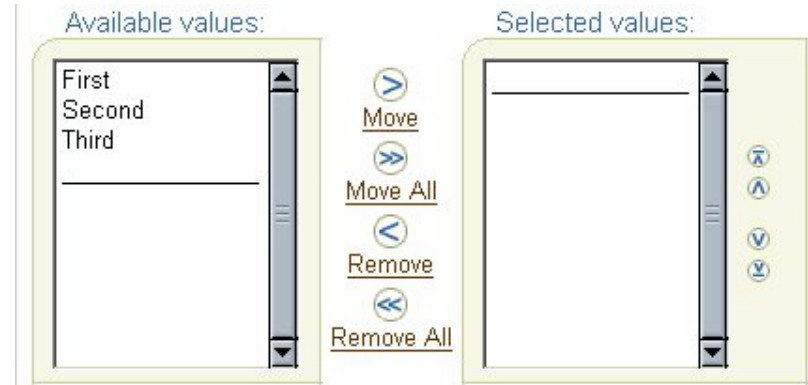
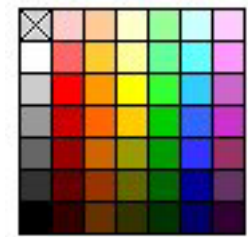
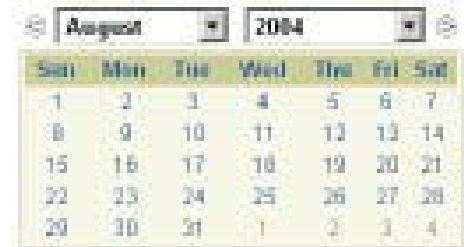
Using ADF Faces

```
<af:inputText label="Name:" value="Enter a name"/>
```



More input components...

- Date field: `<af:selectInputDate>`
- Color field: `<af:selectInputColor>`
- “Shuttle”: `<af:selectManyShuttle>` and `<af:selectOrderShuttle>`



A Better Table

- Built-in paging
- Column sorting
- Partial-page rendering support
- Built-in single/multiple row selection
- Row/Column banding



The screenshot shows a table interface with the following elements:

- Paging controls: "Previous", "1-5 of 18", "Next 5", and a right arrow.
- Toggle controls: "Show All Details" and "Hide All Details".
- Table columns: "Details", "Int", "String", and "Operation".
- Table rows: Five rows, each with a "Show" link, an integer value (0-4), a "String" value (String 0-4), and an "Action" button.

Details	Int	String	Operation
Show 0	0	String 0	Action
Show 1	1	String 1	Action
Show 2	2	String 2	Action
Show 3	3	String 3	Action
Show 4	4	String 4	Action



Communicating Between Pages: processScope

- Standard JSF scopes
 - applicationScope
 - sessionScope
 - requestScope
- Do we need another scope?
- ADF Faces provides processScope



Client-side validation

- A major missing feature of JSF
- ADF Faces provides it
 - Server- and client-side validation/conversion from the same code



Partial-page rendering

- Why do we need PPR?
- Main attributes
 - partialSubmit
 - autoSubmit
 - partialTriggers



Dialog framework

- Show a dialog from an ADF Faces command

```
<af:commandButton text="Get Value"  
  action="dialog:getValue"  
  returnListener="#{backing.handleReturn}"/>
```

- When the dialog finishes, the command component gets a `ReturnEvent` delivered to a `ReturnListener`.



Agenda

- About Developing Web Applications
- About ADF
- About ADF Model
- About ADF Faces
- **Demonstration**
- Q&A

Agenda

- About Developing Web Applications
- About ADF
- About ADF Model
- About ADF Faces
- Demonstration
- **Q&A**

Resource -- Websites

- Oracle JSF Resources
 - <http://otn.oracle.com/jsf>
- Dive into BC4J and ADF
 - <http://radio.weblogs.com/0118231/>
- JSF Specification
 - <http://java.sun.com/j2ee/javaserverfaces/>
- JSFCentral.com
 - JSF Community Website
- Apache Myfaces Website
 - <http://myfaces.apache.org>



Resource -- Books

