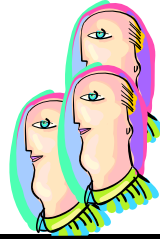


... and UIX Begat ADF Faces: How Rich is ADF Faces Rich Client?

Peter Koletzke
Technical Director &
Principal Instructor



quovera



From The Golden Age of Hollywood

We didn't need dialogue.
We had faces.

—Norma Desmond,
Sunset Boulevard (1950)

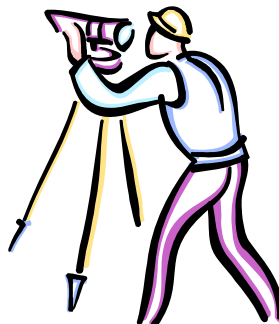


quovera

2

Survey

- Job responsibilities?
 - DBA, developer
- Frameworks?
 - MyFaces, Trinidad
 - ADF Faces, ADF Faces RC
 - ADF BC
 - Other
- Tools?
 - Eclipse
 - JDeveloper
 - Other



quovera

3

Agenda

- Who begat what?
- How rich?
- Atomic components
- Layout components
- Data visualization components
- Resources

See also:

2:30 (this room): *Visualize Your Data*, by Shay Shmeltzer

That's Rich! Putting a smile on ADF Faces, by Lucas Jellema, ODTUG 2009

Slides and white paper will be on the conference website.

The white paper contains some short hands-on practices.



quovera

4

In the Beginning Was EBS

- E-Business Suite (Oracle Applications) needed a lightweight client interface
 - Oracle created UIX for this
- User Interface XML (Extensible Markup Language)
 - First supported in JDeveloper 9i
 - Became “ADF UIX” in JDeveloper 10g (10.1.2)
 - UIX support dropped out of JDev 10.1.3
 - Set of *components*
 - Tag language (e.g., af:inputText)
 - Used to build UIs (e.g., for web applications)
 - Concurrent with JSP in the industry
 - Still used in EBS through R12
 - 9i version of UIX, not the 10g version



And UIX Begat ADF Faces

- Application Development Framework (ADF) came upon the land
 - Formalized first in JDeveloper 10g
 - Introduced specific ADF frameworks
 - ADF Business Components (formerly BC4J)
- JSF had become a Java standard
 - Not in the EE editions, but an industry trend
- Oracle converted UIX components to the JSF standards
 - And called it “ADF Faces”
 - And it was good.



And ADF Faces Begat ADF Faces RC

- Oracle released ADF Faces (non-RC) to open source Apache Project
 - Called “Trinidad” (part of MyFaces)
 - myfaces.apache.org/trinidad
- Oracle created ADF Faces Rich Client
 - Introduced in JDeveloper 11g
 - Concurrent with introduction of ADF Controller (task flows)
- Oracle used it to build Fusion Applications
 - A **very** heavy investment
- And it was **very** good.



Agenda

- Who begat what?
- How rich?
- Atomic components
- Layout components
- Data visualization components
- Resources

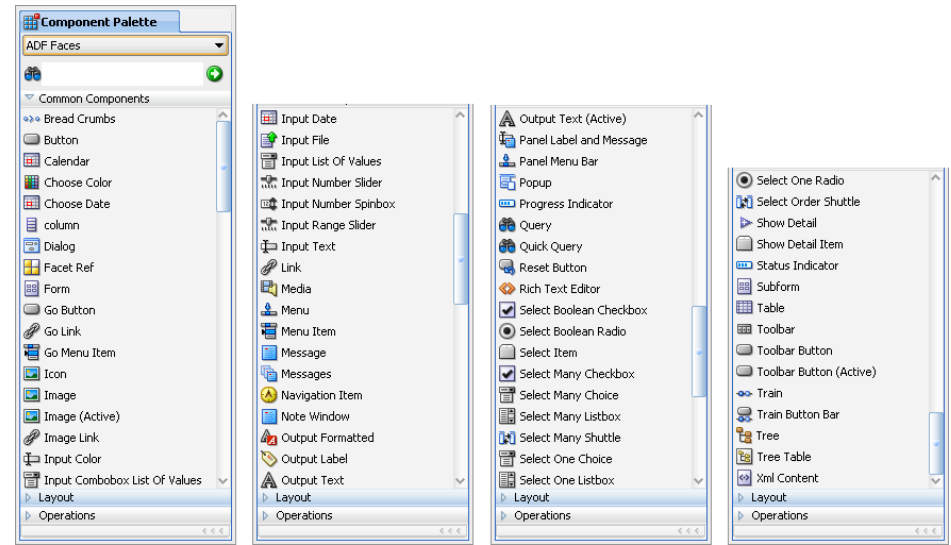


Agenda

- Who begat what?
- How rich?
- Atomic components
- Layout components
- Data visualization components
- Resources



Atomic Components



- Most are on the Common Components page of the Component Palette

Some Atomic Components

Labels in the screenshot:

- `af:inputText` (text input field)
- `af:commandButton` (Last button)
- `af:commandImageLink` (Logon and Help icons)
- `af:inputListOfValues` (Department dropdown)
- `af:menuItem` (Help menu)
- `af:inputDate` (Hire Date calendar)
- `af:selectBooleanCheckbox` (I Love JDev 11g checkbox)
- `af:selectOneChoice` (Job dropdown)

af:selectManyShuttle

- Single component for multiple-selection list
 - Checkmark selection or
 - Double click or
 - Select and click arrow buttons

`af` prefix identifies the ADF Faces RC library

Labels in the screenshot:

- `af:selectManyShuttle` (Available Favorite Things)
- Selected Favorite Things

Some Rich Features

- Atomic components are the main user interface items

- Features:

- Converters; e.g., number format

Convert+Number
sub-component
Pattern =
"###,###.##"

Label 1 Example format: ###,###.##
Label 2

- Validators; some are built in, e.g., required

- Messages appear next to items

*af:inputText
af:outputText
af:inputDate
af:selectOneChoice
Error: A value is required.
You must enter a value for *af:inputText.

- AJAX

- Drag and drop

Required = true
RequiredMessageDetail = "You must enter a value for {0}."

Using 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!



AJAX Interactions - Total Pay

Recalculate

PPR Demo - Mozilla Firefox

PPR Demo

Salary Total Pay
Raise

Done

Id	raise
Value	#{bindings.raise.inputValue}
AutoSubmit	true

Id	totalPay
Value	#{bindings.salary.inputValue + bindings.raise.inputValue}
AutoSubmit	false
partialTriggers	salary raise

Id	salary
Value	#{bindings.Salary.inputValue}
AutoSubmit	true

Drag and Drop

Drag Source **Attribute sub-component; Name = "dragSourceValue"**

Drop Target **Attribute Drop Target sub-component; Attribute = "value"**

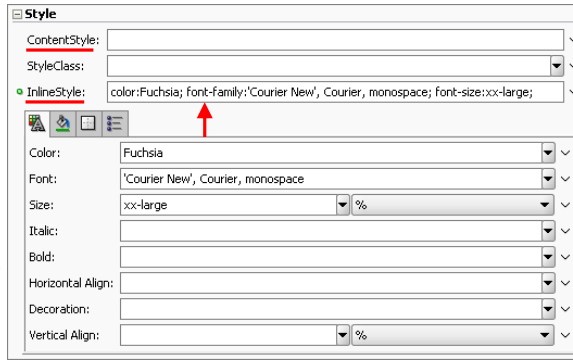
- The value from Drag Source will be copied into Drag Target



Modifying the Visual Aspects

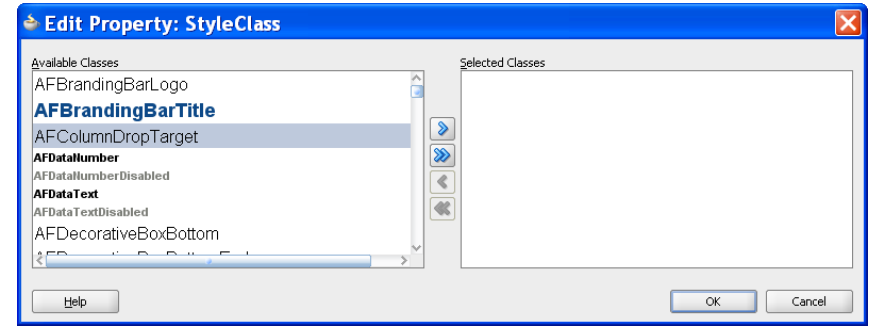
- Skins
 - First and foremost – get this right
- Properties - secondary
 - *ContentStyle*
 - For data inside the component (foreground)
 - *InlineStyle*
 - Set from tab area below it
 - Or just type it in

Demo



Another Visual Aspects Property

- *StyleClass*
 - Equivalent to the HTML *class* property
 - Apply existing style sheet selectors
 - Can apply more than one to a component



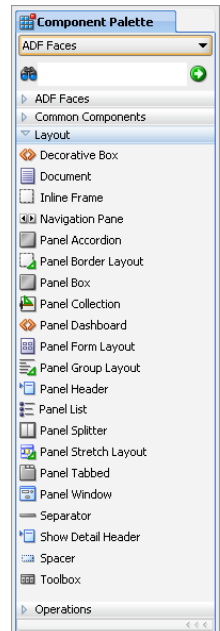
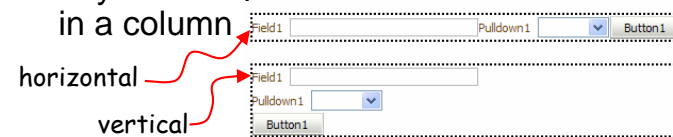
Agenda

- Who begat what?
- How rich?
- Atomic components
- **Layout components**
- Data visualization components
- Resources



Container Components

- Most are on the Layout page of the Component Palette
 - Use them to hold other components
- af:panelHeaderLayout
 - Region title
 - Stack components under it
- af:panelGroupLayout
 - Lay out components in a row or in a column



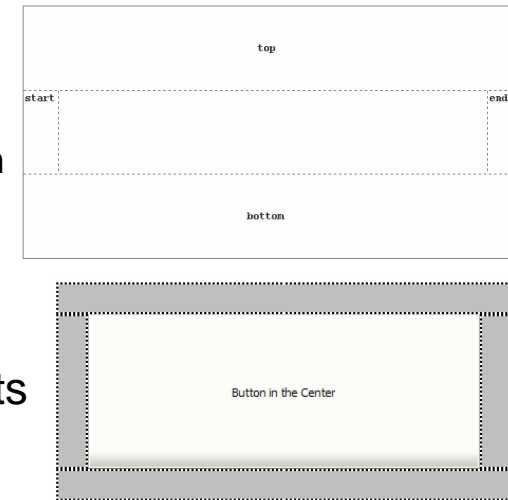
Some Rich Features

- Allow for sophisticated layouts
 - Nest layout components within layout components
 - Virtually limitless possibilities
- *Facets* (sub-components) provide special functionality
 - For example, relative placement of components – regardless of resolution or window size
 - `<f:facet>` component
- Use `af:spacer` to fine tune placement
- Hide or display the contents using the layout component properties



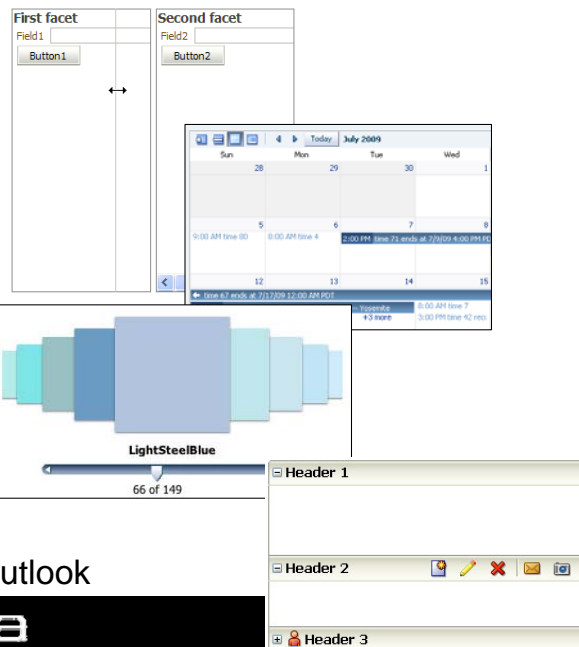
More Layout Components

- `af:panelBorderLayout`
 - Predefined layout areas
 - Uses *facets* to hold the contents of each area
 - Start, end, top, bottom (and more)
- Center area stretches its contents to fill the area



Super-Rich Layout Components

- `af:panelSplitter`
 - Split pane control
 - Optional: user can move the drag bar
 - Horizontal or
 - Vertical
- `af:calendar`
 - MS Outlook style
- `af:carousel`
 - Good for visual browsing
- `af:panelAccordion`
 - Also found in MS Outlook



Layout Tip

- Design This Container option in the pulldown menu for the layout component

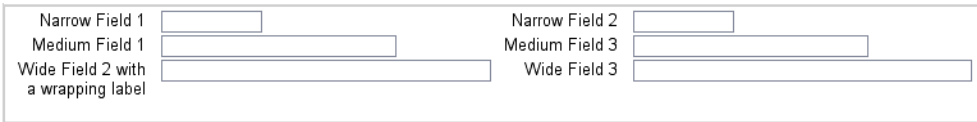
Demo



- Allows you to resize container elements visually
 - This action changes the property values
- Another tip: Assign color to the borders so you can see the containers



PanelFormLayout



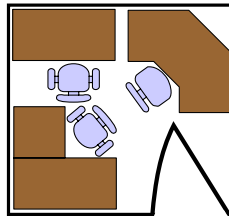
- Lay out fields in rows and columns
 - Perfect for most input forms
- Right justifies prompts
- Left justifies fields
- Tab order is down the first column, then across to the second column
 - Not necessarily intuitive; workaround:
 - Multiple PFLs in a single vertical PGL
 - Set *fieldWidth* and *labelWidth* of the PFLs



Details in the white paper

Achieving the Perfect Layout

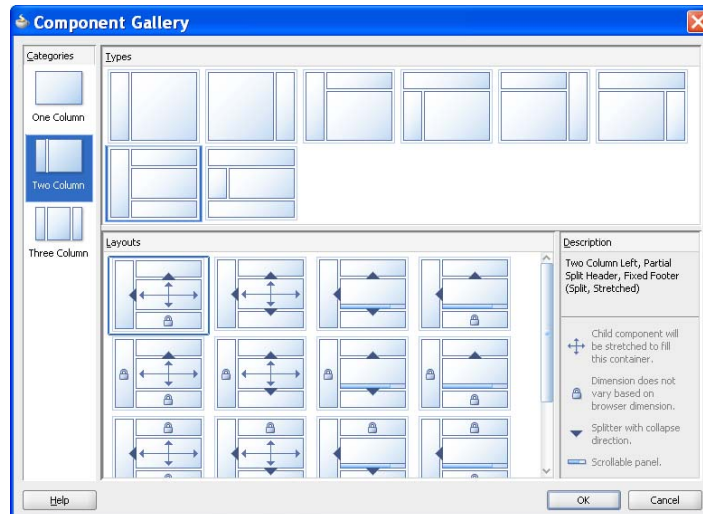
- Know your container components
 - Facets are powerful things
- More
 - af:popup
 - Drop in an af:dialog or af>window
 - Drop af:showPopupBehavior into an action item (button or menu choice)
 - af:panelStretchLayout
 - Expands contained components to fill width
 - Use if another container cuts a component
 - af:menuBar
 - Panel Menu Bar
 - Creates menu area, drop in af:menu then af:menuItem
 - af:panelLabelAndMessage
 - Provides a prompt for a group of objects
 - FirstName and LastName fields with a prompt of "Name"
 - Use inside af:panelFormLayout



More Perfect Layout Tips

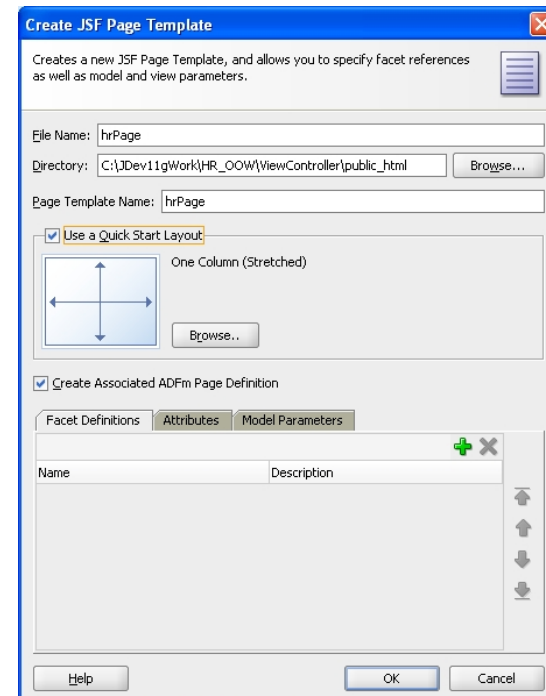
- Design page fragments, not separate pages
- Use Quick Start Layouts when creating the page or template

Find by category.
Provides prebuilt functionality indicated by symbols and description.



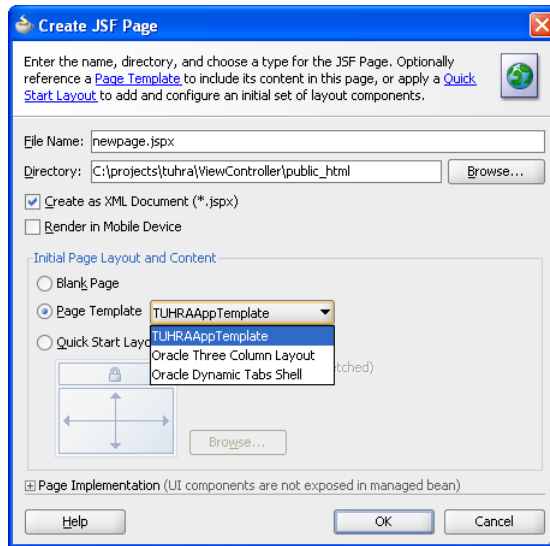
Create Templates

- New gallery item for JSF page template
- Define facets
 - Your own layout areas
- Add attributes
 - Can transfer data from page to template
- Add container components



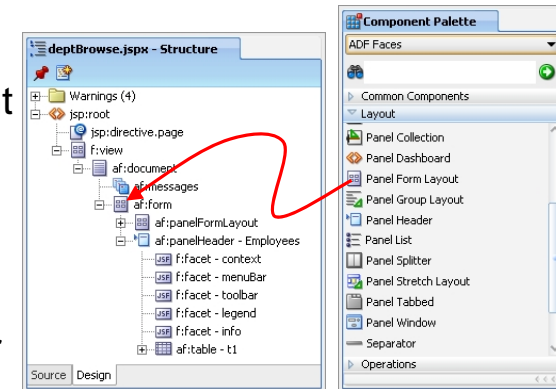
Using the Template

- Application's templates appear in the JSF Page dialog
 - Use Page Template pulldown
- The layout elements are referenced from the template



Tip: Drop Into the Structure Window

- Drop on top of container into which you want the component to appear
- Much more accurate
- Other options
 - Click the component after selecting the Structure window node
 - Use the bread crumbs in the editor



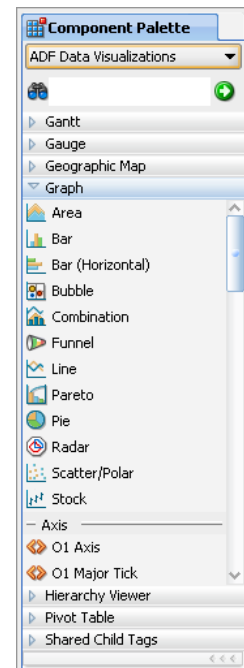
Agenda

- Who begat what?
- How rich?
- Atomic components
- Layout components
- Data visualization components
- Resources

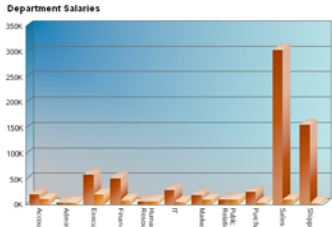


DVT Components

- Data Visualization Tools are on the ADF Data Visualizations page
- Each type has its own panel with components
 - Gantt
 - Gauge
 - Map
 - Graph
 - Hierarchy Viewer
 - Pivot Table



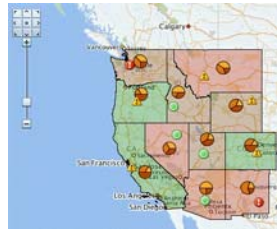
Some DVT Components



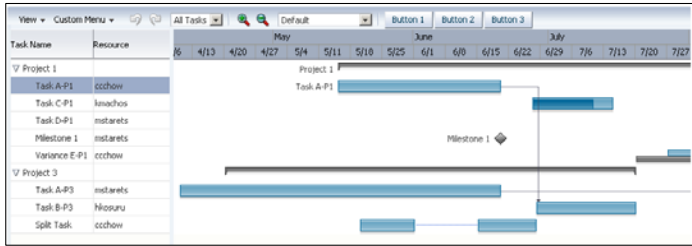
dvt:graph



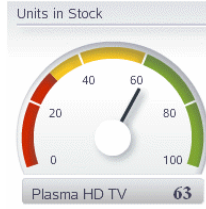
More graphs



dvt:map



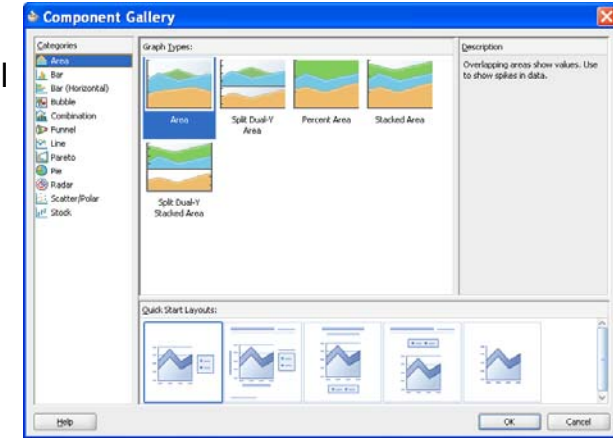
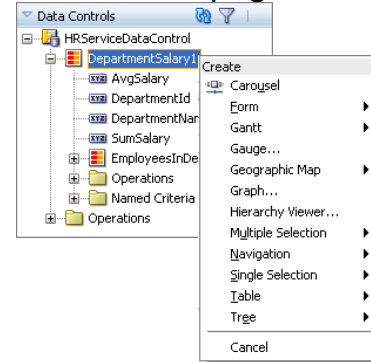
dvt:projectGantt



dvt:gauge

Some Rich Features

- Rich by nature
- Connect to data
 - Drop from the Data Control panel to the page
- Select from a multitude of styles
 - Drop from the Component Gallery

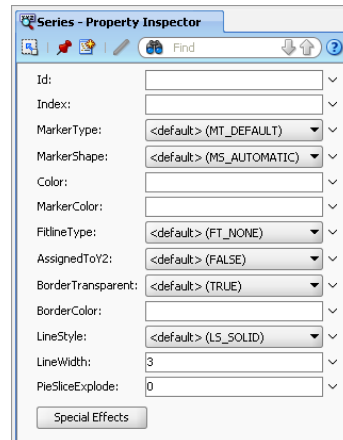


More Rich Features

- Specify data usage

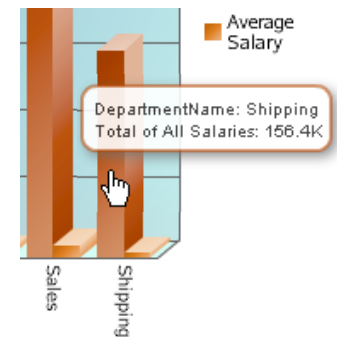


- Control visual aspects using properties



Other Rich Features

- Drill down capability can be mostly declarative
- Some elements of the DVT components are mouse-aware
 - For example, mouse over in the bar graph
 - Displays details about the data point



Details in the white paper

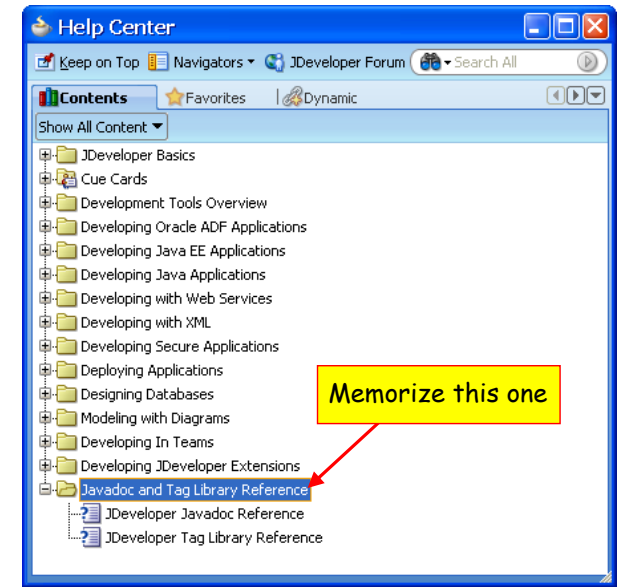
Agenda

- Who begat what?
- How rich?
- Atomic components
- Layout components
- Data visualization components
- Resources

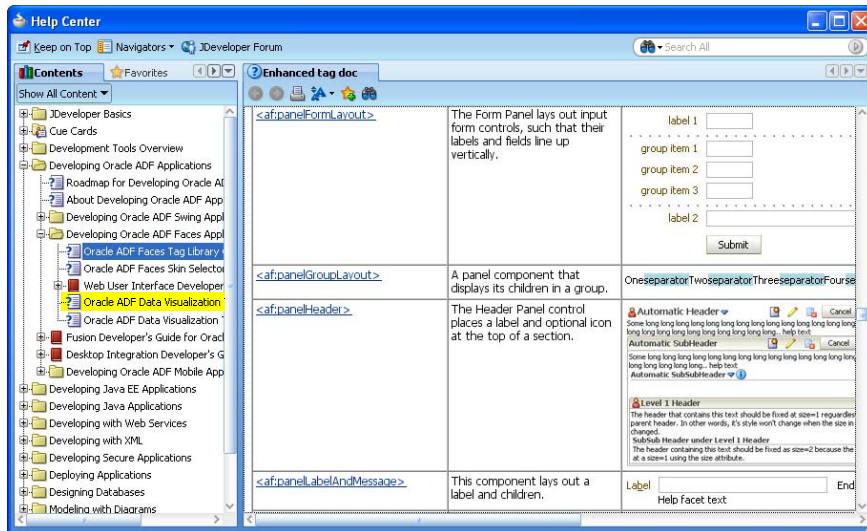


JDeveloper Help Center

- Help | Table of Contents
- Search engine
- Link to JDev forum
- Favorites tab
- Dynamic tab
 - Context-sensitive list based on the task at hand
- Opening a topic opens another tab



Visual Component Guide

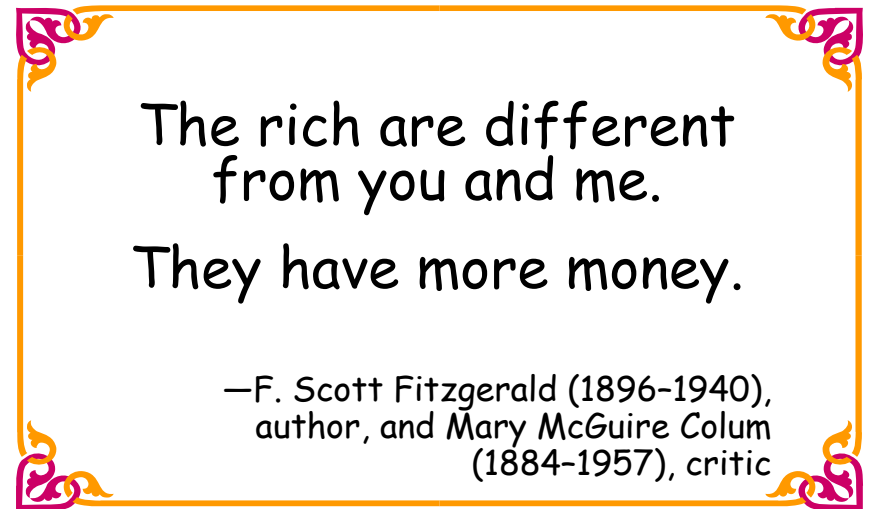
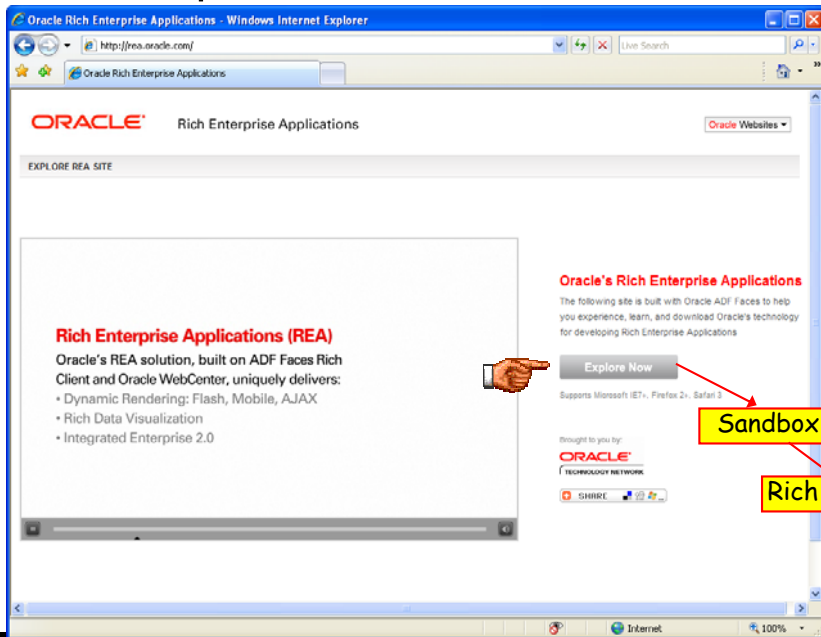


- JDev Help Center (help system)
 - Search for “enhanced tag doc”; also one for **DVT**

Other Resources

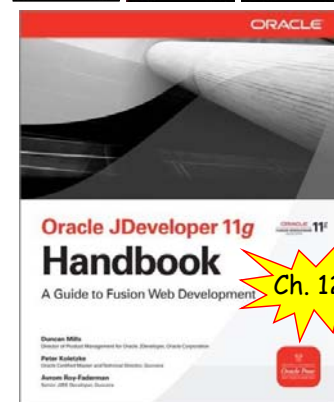
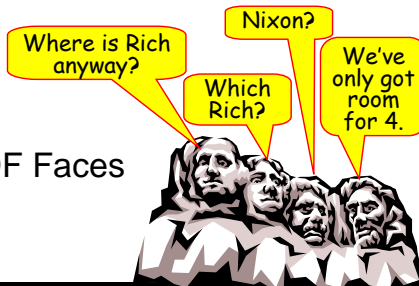
- www.oracle.com/technology/products/adf/adffaces
 - *The Web UI Developer's Guide for Oracle ADF*
 - Cheat Sheets
- ADF EMG – enterprise methodology group
- AMIS blog – ADF Faces white papers
- rea.oracle.com
 - Explore Now | Sandbox | Rich Client
- *Quick Start Guide*
 - Grant Ronald
- *Oracle Fusion Developer Guide*
 - Nimphius and Munsinger
- *Oracle JDeveloper 11g Handbook*
 - Mills and Yours Truly





Summary

- **UIX begat ADF Faces...**
 - ADF Faces RC has a long heritage
- **ADF Faces RC is pretty rich**
 - High-interactivity, AJAX, skins, popups, dialogs, DVT, drag-and-drop
- **Atomic components are mostly for interaction with users**
- **Layout Components provide sophisticated arrangement of components**
- **DVTs supply the functionality users often ask for**
- **There are many resources for help with learning and using ADF Faces**
- **And it is good.**



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



<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**