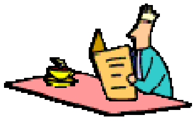
The Tie That Binds: An Introduction to ADF Bindings

Paper #425

Peter Koletzke

Technical Director &
Principal Instructor



IOUG FORUM
COLLABORATE08
Technology and Applications Forum for the Oracle Community

I'd Hammer in the Morning

> *All parts should go together*
> *without forcing...*
> *By all means,*
> *do not use a hammer.*
>
> —*1925 IBM Maintenance Manual*

Survey

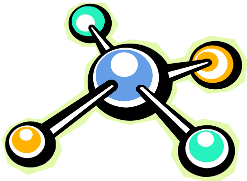
- "Traditional" Oracle development (Forms, Reports, Designer)
  - 1-2 years?
  - More than 2 years?
- Java development
  - 1-3 years?
  - 4-11 years?
  - More than 11 years?
- JDeveloper
  - 1-2 years?
  - More than 2 years?
  - ADF?

Agenda

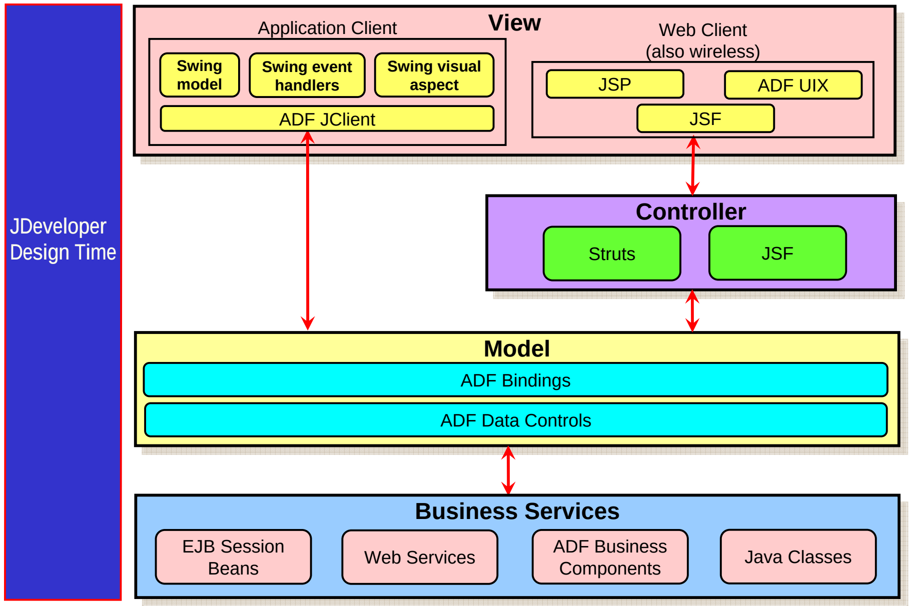
- ADF Model and Data Controls
- ADF Bindings
- Expression Language
- PageDef File
- Binding Examples

Slides and white paper will be available on the Quovera and IOUG websites

Rumor: There is a good book out about JDeveloper 10*g* for Forms and PL/SQL developers.



Application Development Framework



JDeveloper Design Time

View

Application Client

Swing model

Swing event handlers

Swing visual aspect

ADF JClient

Web Client (also wireless)

JSP

ADF UIX

JSF

Controller

Struts

JSF

Model

ADF Bindings

ADF Data Controls

Business Services

EJB Session Beans

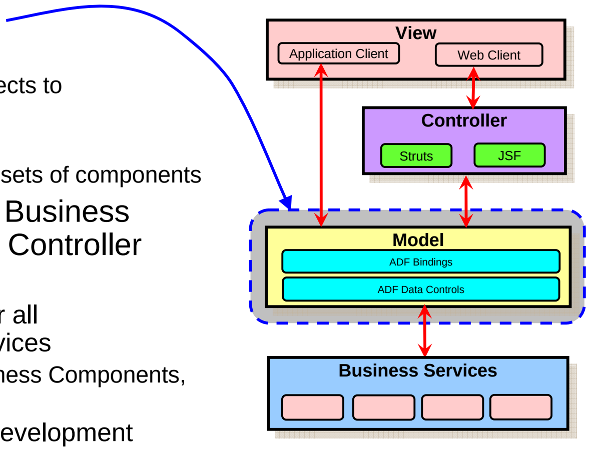
Web Services

ADF Business Components

Java Classes

ADF Model

- The most innovative part of ADF
- ADF Model functions
  - ADF Bindings
    - Bindings provide objects to link to components
  - ADF Data Controls
    - Automatically bound sets of components
- Communication from Business Services to View and Controller layers
  - One common layer for all types of business services
    - E.g., EJB, ADF Business Components, web services
  - The same code and development method for access to any business service
  - We will focus on ADF Business Components



View

Application Client

Web Client

Controller

Struts

JSF

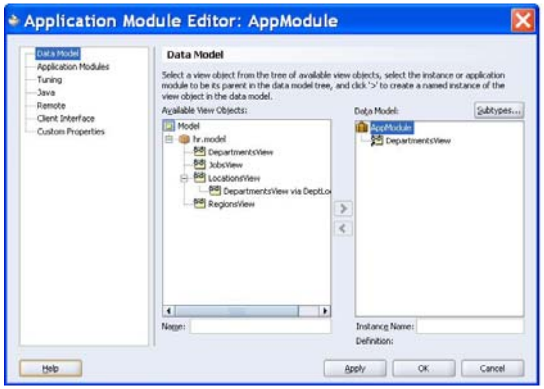
Model

ADF Bindings

ADF Data Controls

Business Services

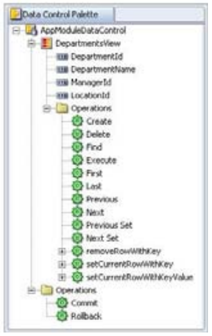
Model – ADF Data Controls



Application Module Editor: AppModule

Data Model

Select a view object from the tree of available view objects, select the instance or application module to be its parent in the data model tree, and click '>' to create a named instance of the view object in the data model.

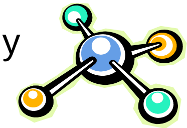


- Data controls represent the data model and provide data to the bindings
- Data Control Palette displays data model objects from the ADF BC application module

---

Data Controls

- Defined in DataControls.dcx
  - Automatically created in the view package of the ViewController project
  - Can represent more than one application module
  - Refers to the pages and PageDef files
    - A PageDef file contains binding definitions
- Allow you to drop UI controls (components) onto the page
  - The components will be automatically bound to Business Services objects

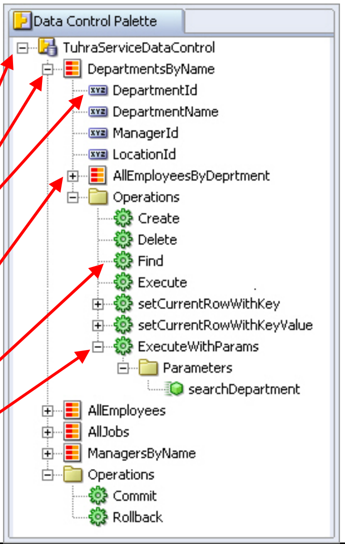


Simple DataControls.dcx

```
<?xml version="1.0" encoding="UTF-8" ?>
<Application xmlns="http://xmlns.oracle.com/adfm/application"
             version="10.1.3.40.66" id="DataBindings"
             SeparateXMLFiles="false"
             Package="hr.view" ClientType="Generic">
  <pageMap>
    <page path="/dept.jspx" usageId="deptPageDef"/>
  </pageMap>
  <pageDefinitionUsages>
    <page id="deptPageDef" path="hr.view.pageDefs.deptPageDef"/>
  </pageDefinitionUsages>
  <dataControlUsages>
    <BC4JDataControl
       id="AppModuleDataControl" Package="hr.model"
         FactoryClass="oracle.adf.model.bc4j.DataControlFactoryImpl"
         SupportsTransactions="true" SupportsFindMode="true"
         SupportsRangesize="true" SupportsResetState="true"
         SupportsSortCollection="true"
         Configuration="AppModuleLocal" syncMode="Immediate"
         xmlns="http://xmlns.oracle.com/adfm/datacontrol"/>
  </dataControlUsages>
</Application>
```

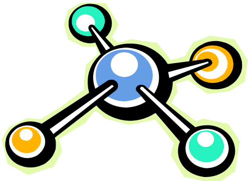
Data Control Palette Nodes

- DCP automatically appears when editing a JSF JSP
  - OR press Ctrl-Shift-D
  - Creates & binds UI items
- Nodes for
  - Data control
  - Data collection
  - Attribute
  - Nested data collection
  - Operation
  - Method



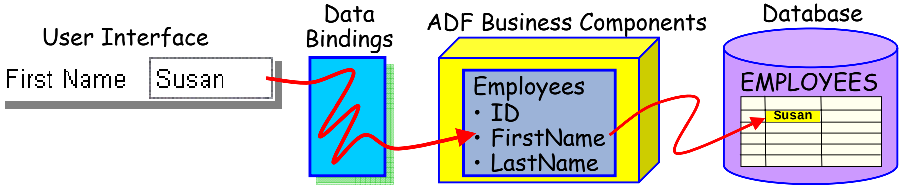
Agenda

- ADF Model and Data Controls
- ADF Bindings
- Expression Language
- PageDef File
- Binding Examples



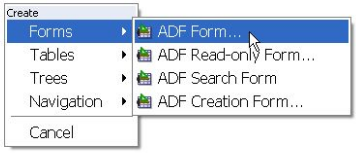
Bindings

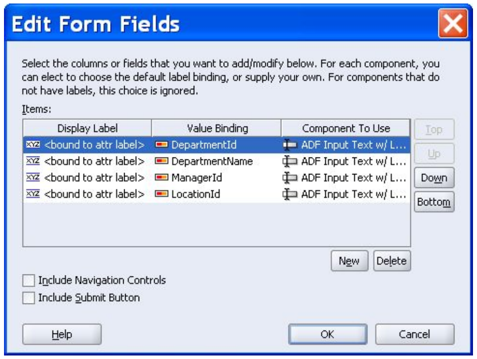
- Association of a business service data element or action with a UI control
  - Relatively automatic in Oracle Forms
  - Definitely not automatic in native Java EE
- Binding normally takes a lot of coding
  - One-off solution is not the answer
  - Need a framework to assist



Using the Data Control Palette

1. Drag a node from the DCP and select a control
2. Define the fields and some properties
3. Click OK





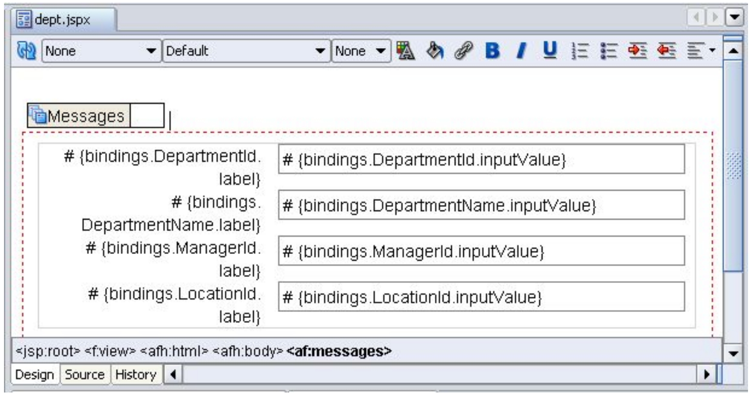
Select the columns or fields that you want to add/modify below. For each component, you can elect to choose the default label binding, or supply your own. For components that do not have labels, this choice is ignored.

Items:

Display Label	Value Binding	Component To Use
<bound to attr label>	DepartmentId	ADF Input Text w/ L...
<bound to attr label>	DepartmentName	ADF Input Text w/ L...
<bound to attr label>	ManagerId	ADF Input Text w/ L...
<bound to attr label>	LocationId	ADF Input Text w/ L...

☐ Include Navigation Controls
☐ Include Submit Button

Result



# {bindings.DepartmentId.label}	# {bindings.DepartmentId.inputValue}
# {bindings.DepartmentName.label}	# {bindings.DepartmentName.inputValue}
# {bindings.ManagerId.label}	# {bindings.ManagerId.inputValue}
# {bindings.LocationId.label}	# {bindings.LocationId.inputValue}

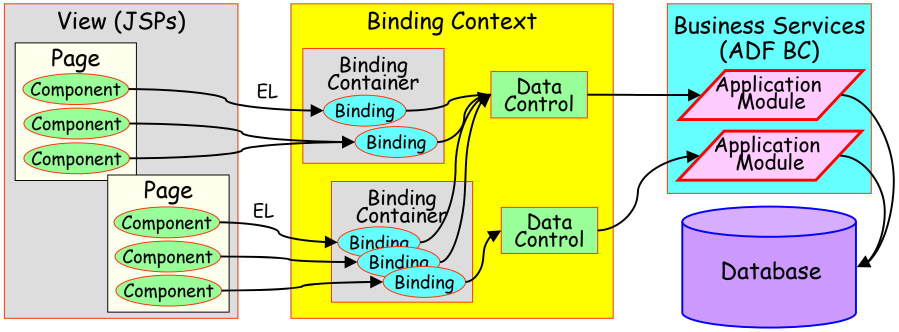
<jsp:root> <f:view> <afh:html> <afh:body> **<af:messages>**

- Components in the JSP displayed in the Visual Editor
- PageDef file loaded with binding definitions

Simple PageDef File

```
<?xml version="1.0" encoding="UTF-8" ?>
<pageDefinition xmlns="http://xmlns.oracle.com/adfm/uimodel"
                version="10.1.3.40.66" id="untitled1PageDef"
                Package="hr.view.pageDefs">
  <parameters/>
  <executables>
    <iterator id="DepartmentsView1Iterator" RangeSize="10"
              Binds="DepartmentsView1"
              DataControl="AppModuleDataControl"/>
  </executables>
  <bindings>
    <attributeValues id="DepartmentId"
                     IterBinding="DepartmentsView1Iterator">
      <AttrNames>
        <Item Value="DepartmentId"/>
      </AttrNames>
    </attributeValues>
    <!-- more attribute tags like the above -->
  </bindings>
</pageDefinition>
```

Behind the Scenes



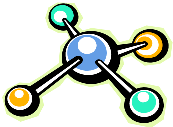
Components on the page are linked to a binding in the binding container (PageDef file) through EL

The binding is linked to the application module through the data control

The application module provides the interface to the database

You Can Bind More Than Data

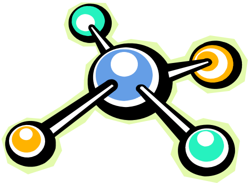
- Bindings can also point to operations (actions) in the application module
  - Predefined actions like Commit, Next Record, Previous Record, Create, Delete, etc.
  - Custom methods you create in the application module
- These bindings are called *action bindings*
- The components are action oriented
  - Command buttons and command links



ADF Bindings = data bindings + action bindings

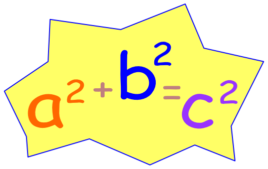
Agenda

- ADF Model and Data Controls
- ADF Bindings
- Expression Language
- PageDef File
- Binding Examples



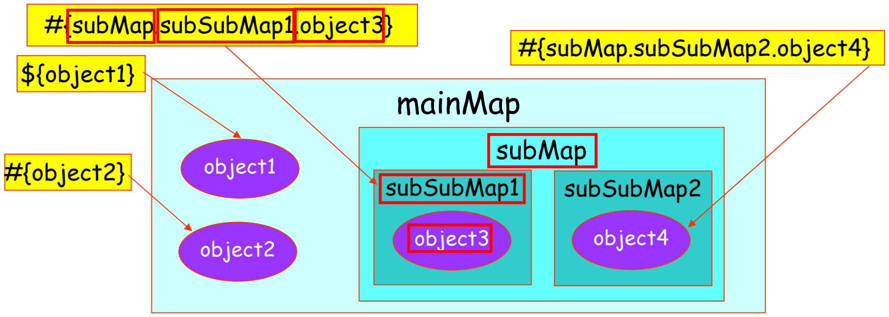
Expression Language

- A.k.a.:
  - “JSP Expression Language”
  - “EL”
- Part of JavaServer Pages Standard Tag Language (JSTL)
  - Procedural language within tags
    - `forEach; if; choose; set; when`
- Many other technologies can use it
  - JSF, UIX, Struts, Swing
- Can be used to refer to elements stored in maps
  - Collections of objects



EL Syntax

- All EL expressions have the form
  `${…}` or **`#{…}`**
  - JSF uses the # variation for component properties
- Refer to map elements by specifying the path to the element within the map, separated by “.”
- The main map is determined by context



EL for ADF

- ADF's submap is "bindings"
- You can use EL expressions that refer to this map in attribute values, e.g.

ID 203

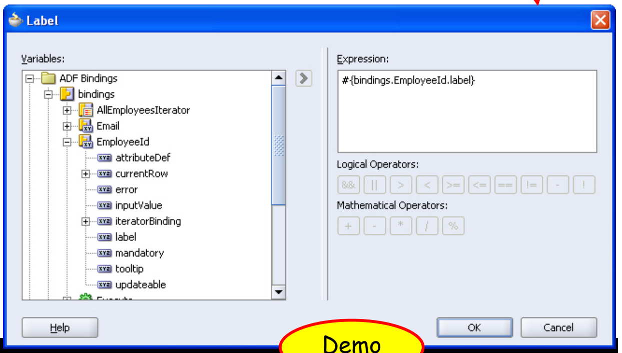
```
<af:inputText
  value="#{bindings.EmployeeId.inputValue}"
  label="#{bindings.EmployeeId.label}"/>
```

- The af:inputText label and value are derived from the bindings context
  - EmployeeId – a submap referring to the ADF BC view object instance
  - In Forms, those EL expressions are like this:

```
GET_ITEM_PROPERTY('EMP.EMPLOYEE_ID',DATABASE_VALUE);
GET_ITEM_PROPERTY('EMP.EMPLOYEE_ID',PROMPT_TEXT);
```

More About EL

- JSF expressions must use the "#" prefix
  - Distinguishes them from JSTL expressions
- Use the "Bind to data" dialog to build expressions
  - Click the "Bind to data" button in the Property Inspector
- Warning:
  - Click "…" button for existing bindings

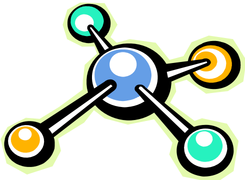


Expression: #{bindings.EmployeeId.label}

Demo

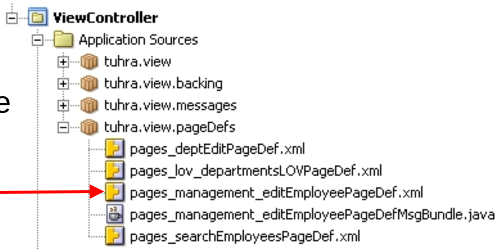
Agenda

- ADF Model and Data Controls
- ADF Bindings
- Expression Language
- **PageDef File**
- Binding Examples



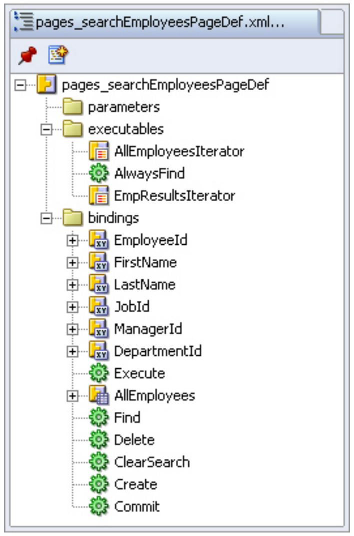
PageDef File

- The *PageDef* (or *page definition)* XML file stores binding definitions for the page
  - One PageDef file for each JSP
  - Called *package_filename*PageDef
  - For example, pages_editEmployeePageDef.xml
- Access it with "Go to Page Definition"
  - Right click menu in code editor or visual editor
- Also available in the navigator


ViewController
Application Sources
tuhra.view
tuhra.view.backing
tuhra.view.messages
tuhra.view.pageDefs
pages_deptEditPageDef.xml
pages_lov_departmentsLOVPageDef.xml
pages_management_editEmployeePageDef.xml
pages_management_editEmployeePageDefMsgBundle.java
pages_searchEmployeesPageDef.xml

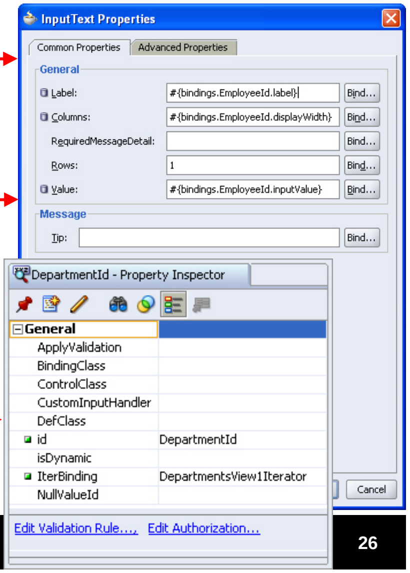
Contents of the PageDef File

- Executables
  - Definitions of actions that will be run when the PageDef file is loaded
    - Like trigger code in Forms
- Bindings
  - Values and operations required on the page
  - Like database-oriented item properties in Forms



Editing Methods

- Code Editor
- OR select **Edit Binding** from a UI component
- OR select **Properties** from the right-click menu on a binding in the Structure Window
- OR select the binding in the Structure Window and use the Property Inspector

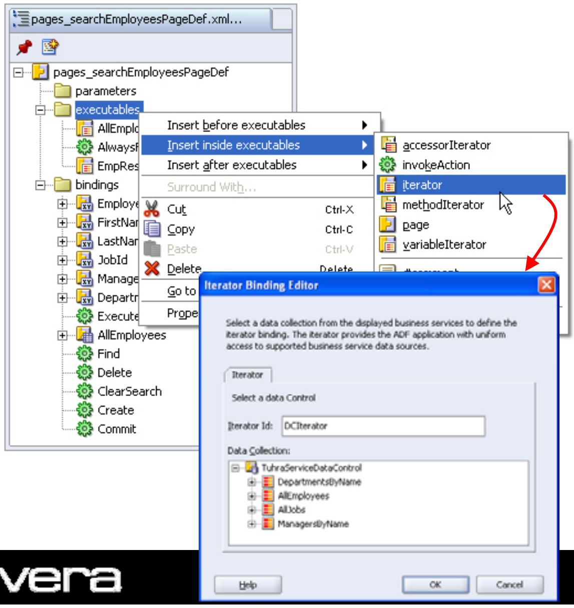


Label: #{bindings.EmployeeId.label}
Columns: #{bindings.EmployeeId.displayWidth}
RequiredMessageDetail:
Rows: 1
Value: #{bindings.EmployeeId.inputValue}
Tip:

General
ApplyValidation
BindingClass
ControlClass
CustomInputHandler
DefClass
id DepartmentId
isDynamic
IterBinding DepartmentsView1Iterator
NullValueId

Adding Elements to the PageDef

- After the DCP, the easiest way uses the Structure Window
- Select from the right-click menu
- This opens the relevant dialog



Select a data collection from the displayed business services to define the iterator binding. The iterator provides the ADF application with uniform access to supported business service data sources.

Select a data Control

Iterator Id: DCIterator

Data Collection:
TuhraServiceDataControl
DepartmentsByName
AllEmployees
AllJobs
ManagersByName

PageDef Snippet: Bindings

```
<pageDefinition xmlns=http://xmlns.oracle.com/adfm/uimodel ...
  <bindings>
    <attributeValues id="EmployeeId" IterBinding="AllEmployeesIterator">
      <AttrNames>
        <Item Value="EmployeeId"/>
      </AttrNames>
    </attributeValues>
    <action id="Commit" InstanceName="TuhraServiceDataControl"
            DataControl="TuhraServiceDataControl"
            RequiresUpdateModel="true" Action="100"/>
    <list id="AllEmployeesJobId" IterBinding="AllEmployeesIterator"
          StaticList="false" ListOperMode="0" ListIter="AllJobsIterator"
          NullValueFlag="1" NullValueId="AllEmployeesJobId_null">
      <AttrNames>
        <Item Value="JobId"/>
      </AttrNames>
      <ListAttrNames>
        <Item Value="JobId"/>
      </ListAttrNames>
      <ListDisplayAttrNames>
        <Item Value="JobTitle"/>
      </ListDisplayAttrNames>
    </list>
  </bindings>
```

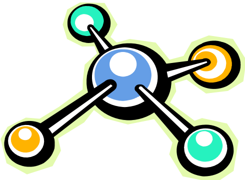
List bindings

Demo

Action bindings

Agenda

- ADF Model and Data Controls
- ADF Bindings
- Expression Language
- PageDef File
- Binding Examples

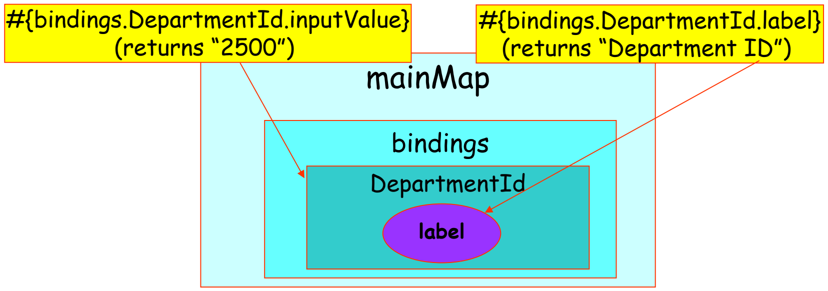


Binding Types

- Attribute
  - For single attribute in a collection
- Table (or range)
  - For table components bound to collections
- List
  - For data-bound list elements
- Action
  - For standard operations like Commit
- Navigation List
  - To manipulate the current row in a set
- Method
  - For custom methods
- Boolean
  - For checkboxes
- Tree
  - Set of master-detail data for hierarchical controls

Attribute Bindings

- A single view attribute value on the iterator binding's current view row
- Attribute bindings are maps



#{bindings.DepartmentId.inputValue}
(returns "2500")

#{bindings.DepartmentId.label}
(returns "Department ID")

Attribute Bindings and JSF

- You can access attribute bindings from any component's attribute

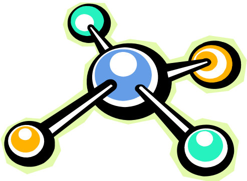
```
<af:outputText value="#{bindings.DepartmentId.inputValue}" />
```

- Other component attributes access properties on the Model level object
  - Control hints or attribute properties (for example, label and width):

```
<af:inputText
  value="#{bindings.PhoneNumber.inputValue}"
  label="#{bindings.PhoneNumber.label}"
  columns="#{bindings.PhoneNumber.displayWidth}"/>
```

Table Bindings

- A.k.a., *range bindings*
- Expose data from all rows in the iterator binding's current range
- Expose some or all columns
- Iterator's *collectionModel* property is used as the table's value
- A variable in the table UI component is assigned to the table binding's data
  - Usually "row"



PageDef Snippets

```
<bindings>
 <table
   id="AllEmployees"
   IterBinding=
     "EmpResultsIterator">
  <AttrNames>
    <Item Value="EmployeeId"/>
    <Item Value="FirstName"/>
    <Item Value="LastName"/>
    <Item Value="Email"/>
    <Item Value="PhoneNumber"/>
    <Item Value="HireDate"/>
    <Item Value="JobId"/>
    <Item Value="Salary"/>
    <Item Value="ManagerId"/>
  </AttrNames>
</table>
```

Table Binding

```
<executables>
 <iterator
   id="EmpResultsIterator"
   RangeSize="10"
   Binds="AllEmployees“
   DataControl=
   "TuhraServiceDataControl"/>
 <iterator
   id="AllJobsIterator"
   RangeSize="-1"
   Binds="AllJobs“
   DataControl=
   "TuhraServiceDataControl"/>
</executables>
```

Iterator

JSF Snippet: af:table

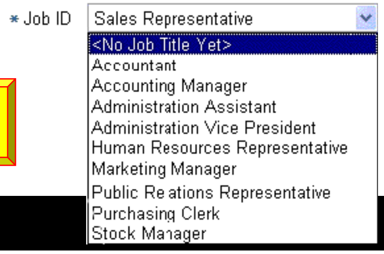
```
<af:table
  value="#{bindings.AllEmployees.collectionModel}" var="row"
  rows="#{bindings.AllEmployees.rangeSize}"
  first="#{bindings.AllEmployees.rangeStart}"
  selectionState=
      "#{bindings.AllEmployees.collectionModel.selectedRow}"
  selectionListener=
      "#{bindings.AllEmployees.collectionModel.makeCurrent}"
  width="100%">
  <af:column
    headerText="#{bindings.AllEmployees.labels.EmployeeId}"
    sortProperty="EmployeeId" sortable="true">
    <af:outputText value="#{row.EmployeeId}"/>
  </af:column>
  <af:column
    headerText="#{bindings.AllEmployees.labels.FirstName}"
    sortProperty="FirstName" sortable="true">
    <af:outputText value="#{row.FirstName}"/>
  </af:column>
```

List Bindings

- Populate a single attribute in the current row, or navigate between rows
  - Dynamic or static
- Generally used as the **`value`** attribute for a pulldown list or similar element

```
<af:selectOneChoice
  value="#{bindings.AllEmployeesJobId.inputValue}"
/>
```

Associated binding definition on the PageDef Snippet: Bindings slide

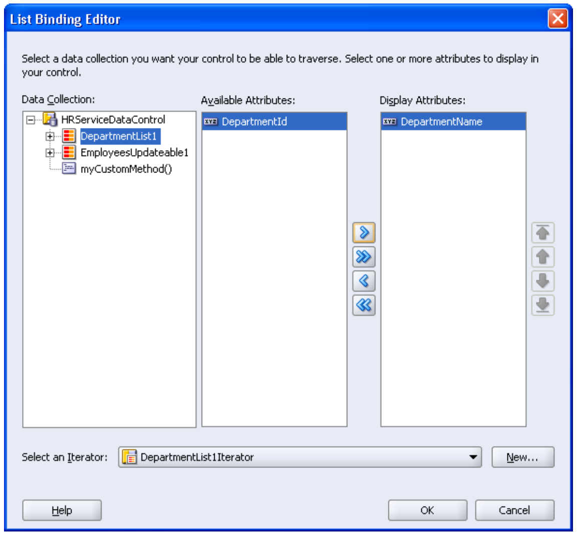


* Job ID: Sales Representative

- <No Job Title Yet>
- Accountant
- Accounting Manager
- Administration Assistant
- Administration Vice President
- Human Resources Representative
- Marketing Manager
- Public Relations Representative
- Purchasing Clerk
- Stock Manager

Navigation List Bindings

- Changes the current row in the iterator
- Selecting a value from a navigation list sets that row as current
- This List Binding Editor appears when you drop a collection as a navigation list



List Binding Editor

Select a data collection you want your control to be able to traverse. Select one or more attributes to display in your control.

Data Collection:
- HRServiceDataControl
  - DepartmentList1
  - EmployeesUpdateable1
  - myCustomMethod()

Available Attributes: DepartmentId

Display Attributes: DepartmentName

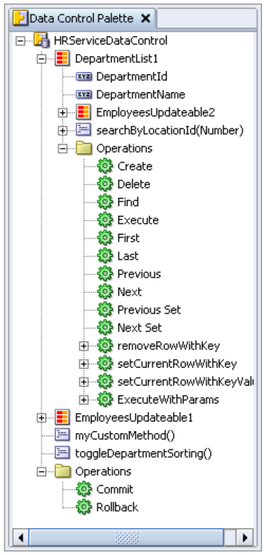
Select an Iterator: DepartmentList1Iterator

Action Bindings

- Added when buttons or links are dropped into the UI
- Run *operations*
  - Like Forms' built-ins that operate at the block level
  - Collection operations
    - Create, Delete, Find, First, Last, etc.
  - Data control operations
    - Commit, Rollback

```
<af:commandButton
  text="Save"
  actionListener="#{bindings.Commit.execute}"
  disabled="#{!bindings.Commit.enabled}"/>
```

Associated binding definition on the PageDef Snippet: Bindings slide

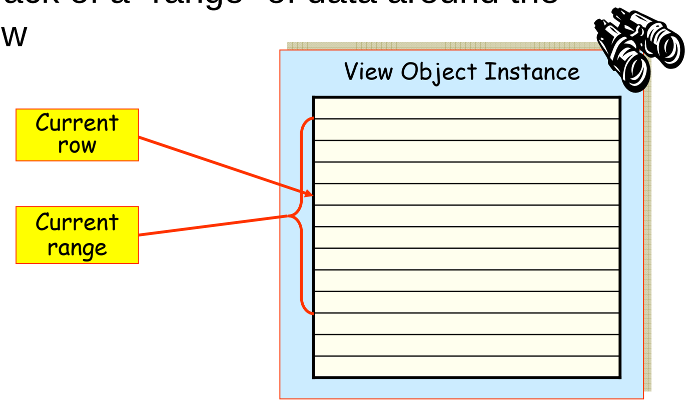


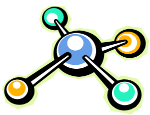
Data Control Palette

- HRServiceDataControl
  - DepartmentList1
    - DepartmentId
    - DepartmentName
    - EmployeesUpdateable2
    - searchByLocationId(Number)
    - Operations
      - Create
      - Delete
      - Find
      - Execute
      - First
      - Last
      - Previous
      - Next
      - Previous Set
      - Next Set
      - removeRowWithKey
      - setCurrentRowWithKey
      - setCurrentRowWithKeyVal
      - ExecuteWithParams
  - EmployeesUpdateable1
  - myCustomMethod()
  - toggleDepartmentSorting()
  - Operations
    - Commit
    - Rollback

Iterator – An Executable

- These keep track of the current view row in a view object instance's query result
- Keep track of a "range" of data around the view row





Iterator Details

- Consider this binding:
  `#{bindings.EmployeeId.inputValue}`
  - The employee ID of whatever record is currently pointed to by the iterator
- Iterators are defined in the PageDef file executables section
  - The query associated with the iterator's view object is run when the page loads
    - If more than one iterator per collection, the query executes once
- Important property: *rangeSize*
  - Number of rows displayed (-1 is all)

Don't Write Iterators Like This

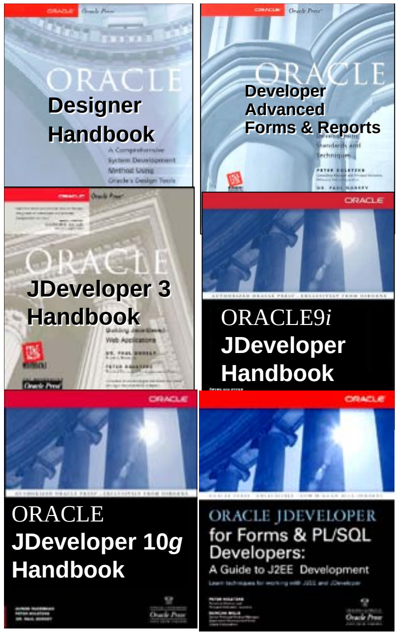
endless loop
n., see loop, endless.

loop, endless
n., see endless loop.

—Data Processing Dictionary,
Random House Publishing

Summary

- ADF Model layer connects the ADF Controller or View and Business Services layers
- ADF Bindings provide association of UI components and business services
- Simple coding in EL (and XML) tap into the ADF Model framework code
- XML code in the PageDef file defines the bindings
- EL in the UI properties access the binding by name


ORACLE Designer Handbook
ORACLE Developer Advanced Forms & Reports
ORACLE JDeveloper 3 Handbook
ORACLE9i JDeveloper Handbook
ORACLE JDeveloper 10g Handbook
ORACLE JDEVELOPER for Forms & PL/SQL Developers: A Guide to J2EE Development

- **Please fill out the evals – paper 425**
- **Books co-authored with Dr. Paul Dorsey, Avrom Roy-Faderman, & Duncan Mills**
- **Personal web site:**

`http://ourworld.compuserve.com/homepages/Peter_Koletzke`

quovera

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**

---

quovera

www.quovera.com