The Tie That Binds: An Introduction to ADF Bindings

Peter Koletzke
Technical Director &
Principal Instructor

quovera

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

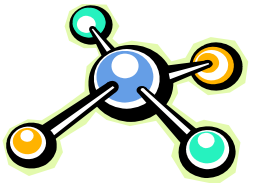
- "Traditional" Oracle development
  (Forms, Reports, Designer)
  - 1-2 years?
  - More than 2 years?
- Java development
  - 1-3 years?
  - 4-11 years?
  - More than 11 years?
- JDeveloper
  - 1-2 years?
  - More than 2 years?

Agenda

- ADF Model Layer
- ADF Data Bindings
- Expression Language
- PageDef File
- Binding Examples

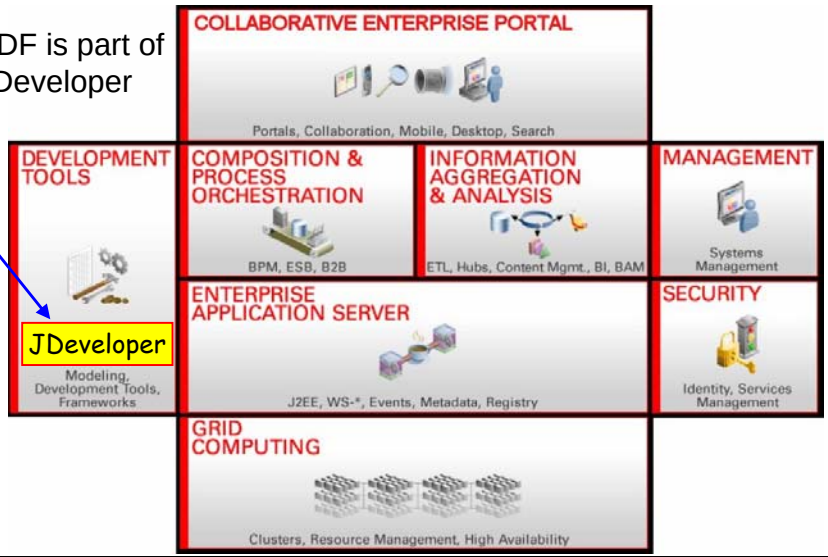
Slides will be available on the Quovera and NYOUG websites

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



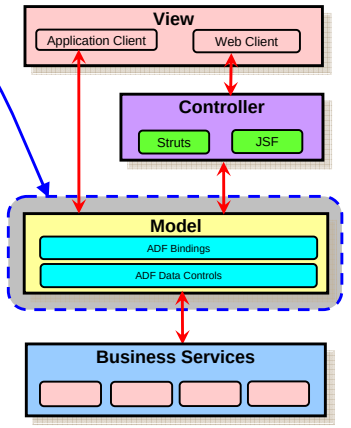
JDeveloper, ADF, and Oracle Fusion Middleware

- ADF is part of JDeveloper



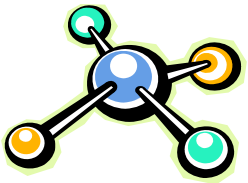
ADF Model (ADFm)

- 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



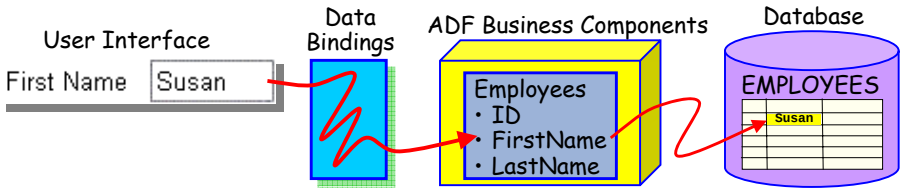
Agenda

- ADF Model Layer
- ADF Data Bindings
- Expression Language
- PageDef File
- Binding Examples



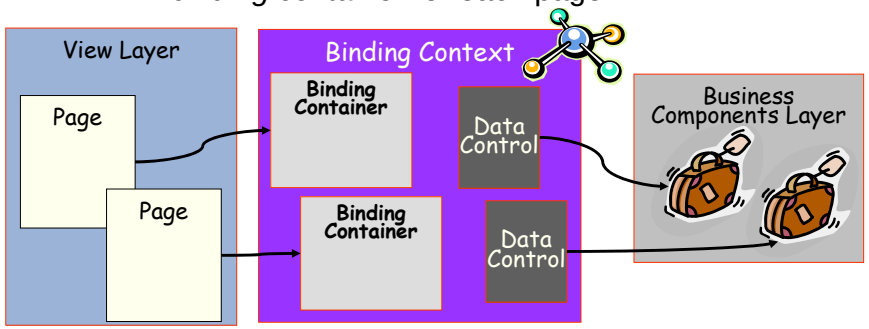
Bindings

- Association of a business service data element or action with a UI control
  - Relatively automatic in Oracle Forms
  - Definitely not automatic in native J2EE
- Binding normally takes a lot of coding
  - One-off solution is not the answer
  - Need a framework to assist



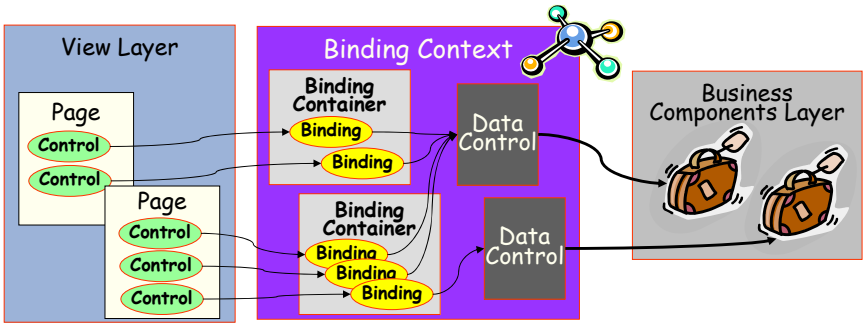
Bindings – Behind the Scenes

- Each user has a *binding context*
  - This contains two main layers
    - A *data control* for each ADF BC application module instance
    - *A binding container* for each page



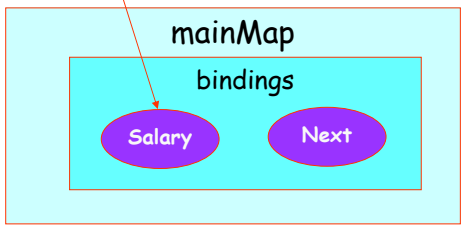
ADF Model Components

- Each control on the page can use its own *data binding*
  - Exposes a bit of data or an action
    - Commit, Rollback, First, Last, Execute (query), etc.



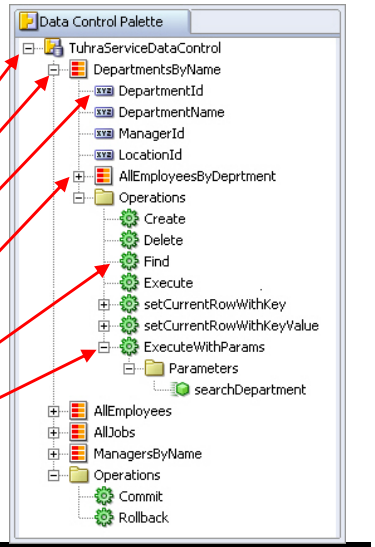
Accessing Bindings

- Progammatically, you use **`java.util.Map`**
  - An interface for organizing data
  - Stores *elements* – data of any Object type
- The **`bindings`** map contains all the bindings in the current page's binding container
- You can access these bindings using Expression Language `#{bindings.Salary}`



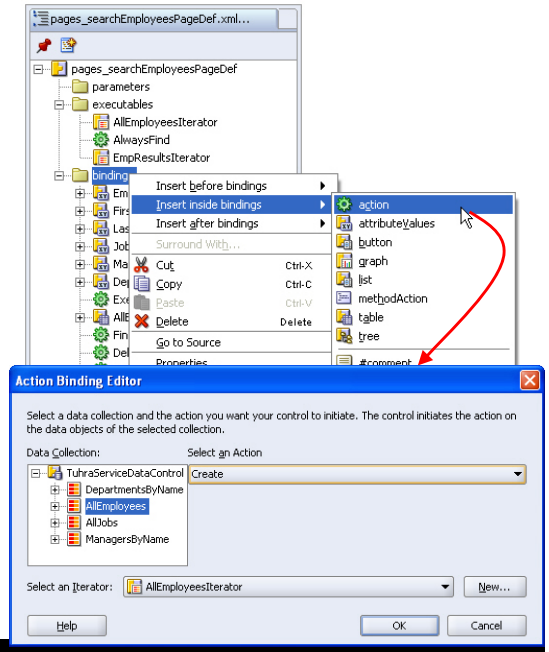
Creating Bindings in JDeveloper

- Use the Data Control Palette
  - Automatically appears when editing a JSF JSP
  - OR Ctrl-Shift-D
  - This creates and binds UI items
- Nodes for
  - Data control
  - Data collection
  - Attribute
  - Nested data collection
  - Operation
  - Method



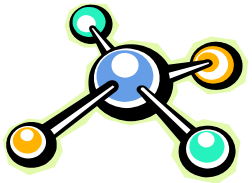
Creating Bindings

- Alternatively use the Structure window
- Right-click menu options
- Both methods create bindings entries in the PageDef file



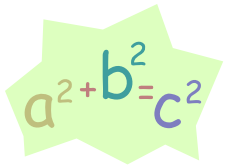
Agenda

- ADF Model Layer
- ADF Data Bindings
- Expression Language
- PageDef File
- Binding Examples



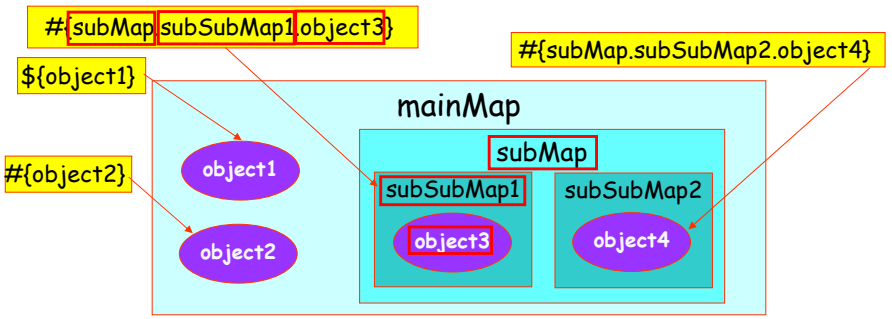
Expression Language

- 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

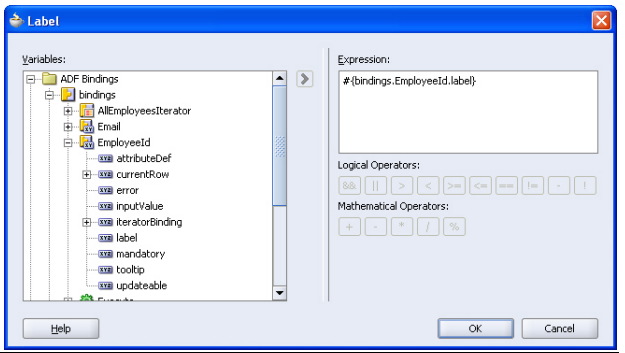
```
<af:inputText
  value="#{bindings.EmployeeId.inputValue}"
  label="#{bindings.EmployeeId.label}"/>
```

- 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);
```

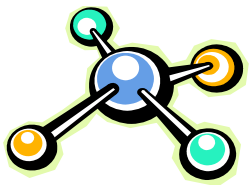

More About EL

- 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
- All bindings are stored in the PageDef file



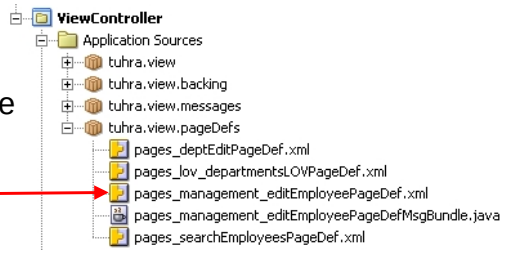
Agenda

- ADF Model Layer
- ADF Data Bindings
- Expression Language
- PageDef File
- Binding Examples



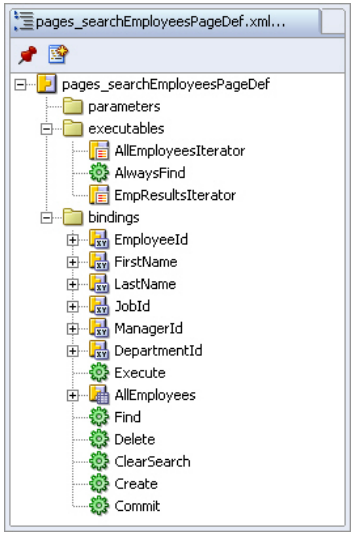
PageDef Binding File

- 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



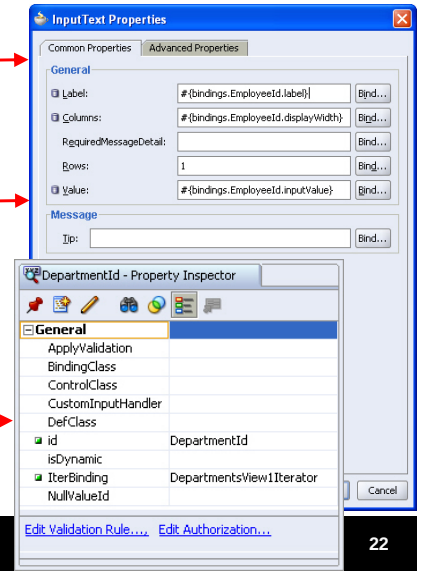
Contents of the PageDef File

- 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 Record Manager definitions in Forms



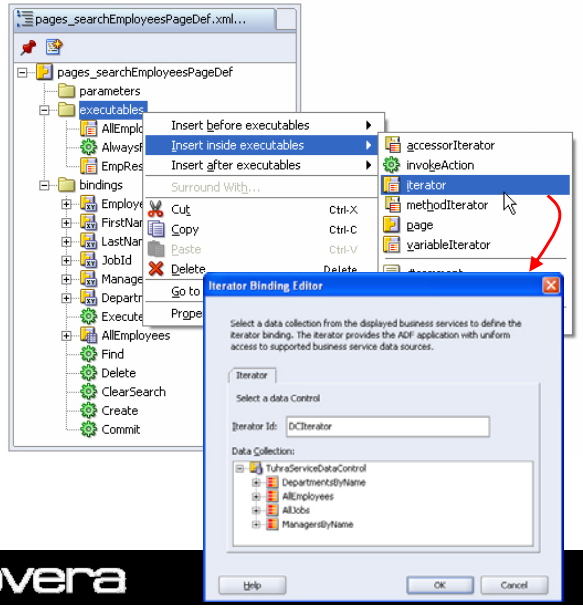
Editing Methods

- 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



Adding Elements to the PageDef

- Easiest in the Structure window
- Select from the right-click menu
- This opens the relevant dialog



PageDef Snippet: Bindings

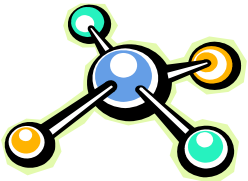
```
<pageDefinition xmlns=http://xmlns.oracle.com/adfm/uimodel ...
  <bindings>
    <attributeValues id="EmployeeId" IterBinding="AllEmployeesIterator">
      <AttrNames>
        <Item Value="EmployeeId"/>
      </AttrNames>
    </attributeValues>
    <action id="Commit" InstanceName="TuhraServiceDataControl"
            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

Action bindings

Agenda

- ADF Model Layer
- ADF Data Bindings
- Expression Language
- PageDef File
- Binding Examples

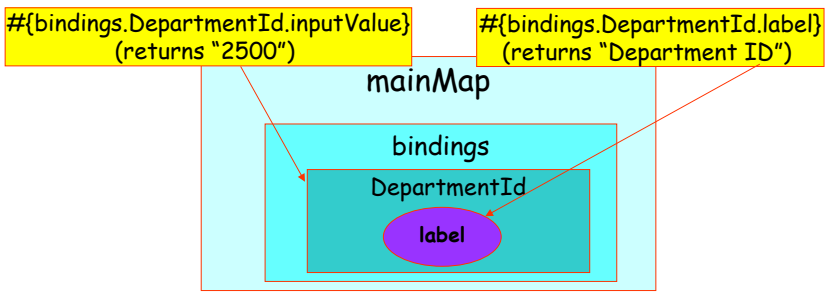


Binding Types

- 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



Attribute Bindings and JSF

- You can access attribute bindings from any component attribute

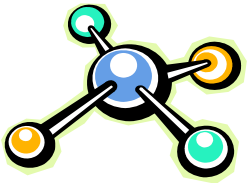
```
<af:outputText value="#{bindings.DepartmentId.inputValue}" />
```

- 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}"/>
```

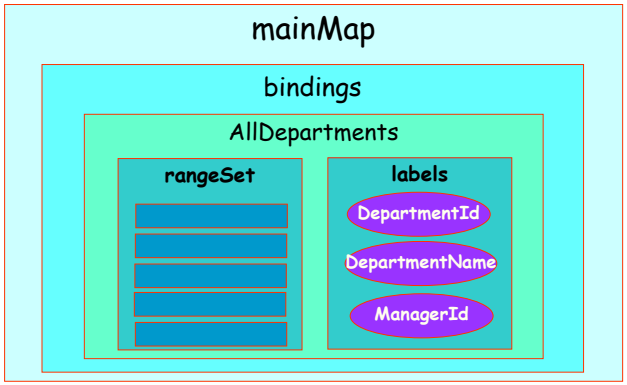
Table Bindings

- 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"



Range Bindings and Maps

- Range bindings are maps, too



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

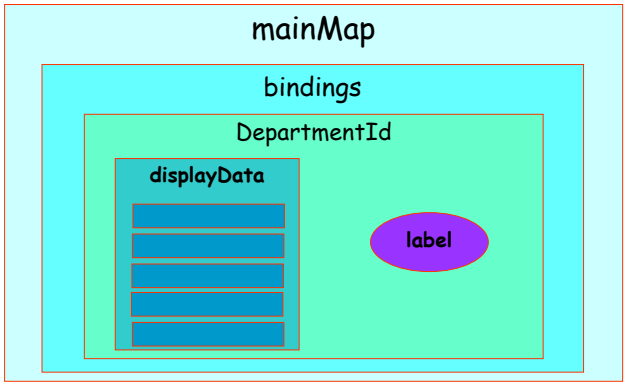
- Populate a single attribute in the current row, or navigate between rows
  - Dynamic or static
- Generally used as the **`value`** attribute for a pulldown list or similar element

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

See PageDef Snippet: Bindings slide for associated binding definition.

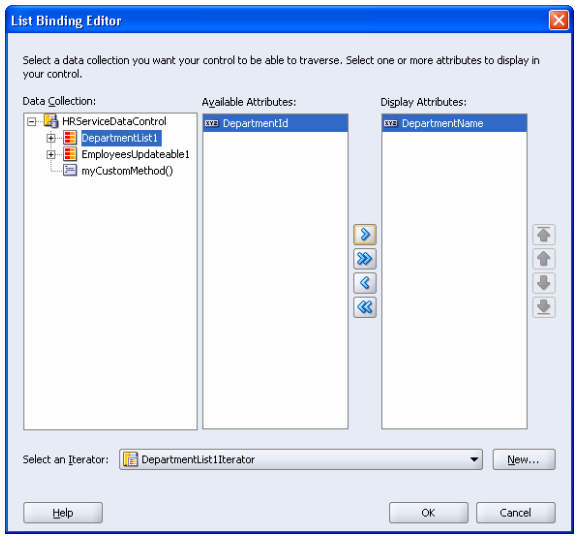
List Bindings and Maps

- List bindings are maps too



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



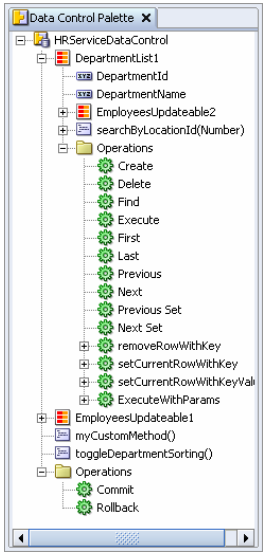
Action Bindings

- 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

Save

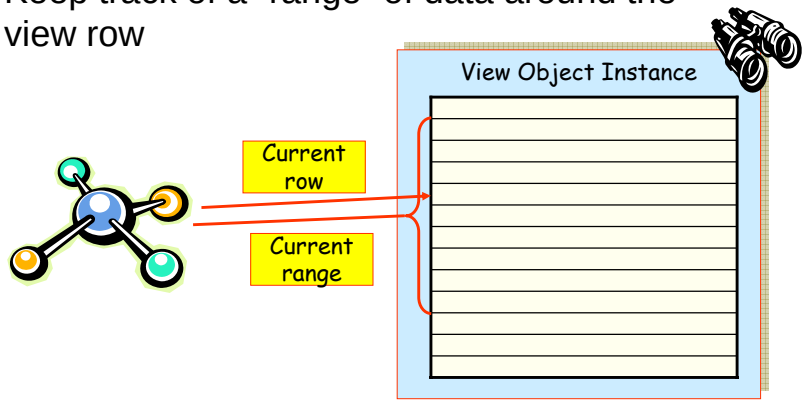
```
<af:commandButton
  text="Save"
  actionListener="#{bindings.Commit.execute}"
  disabled="#{!bindings.Commit.enabled}"/>
```

See PageDef Snippet: Bindings slide for associated binding definition.



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

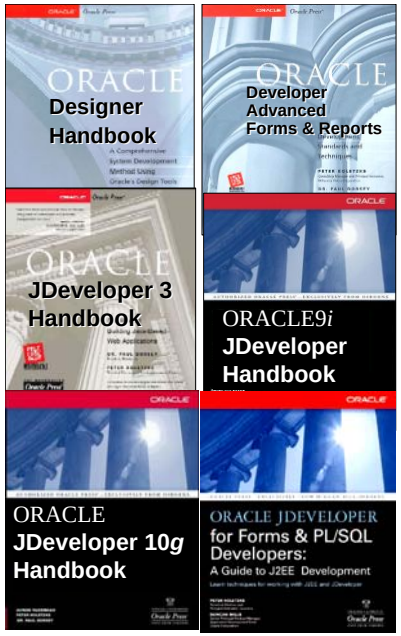


Iterator Details

- 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)

Summary

- ADF Model layer connects the ADF Controller or View and Business Services layers
- ADF Data 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



- **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`

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**