

ADF Code Corner

047. How-to build a select one choice displaying hierarchical selection data

ORACLE
CODE CORNER



twitter.com/adfcodecorner

Abstract:

Apparently, in the JavaServer Faces RI you are able to build hierarchical select choice lists that display the list entries categorized by their parent records.

For ADF Faces RC, hierarchical select choices are an enhancement request filed and wont be available in the initial production release of JDeveloper 11. However, with a bit of programming - where the emphasize clearly is on the "bit" word - you can achieve exactly the same behavior. This how-to document explains how you build hierarchical tree lists based on the Oracle Fusion stack, which is ADF Faces RC, ADF for the binding and ADF BC for the persistence layer.

Author:

Frank Nimphius, Oracle Corporation
twitter.com/fnimphiu
06-JUN-2008

Oracle ADF Code Corner is a loose blog-style series of how-to documents that provide solutions to real world coding problems.

Disclaimer: All samples are provided as is with no guarantee for future upgrades or error correction. No support can be given through Oracle customer support.

Please post questions or report problems related to the samples in this series on the OTN forum for Oracle JDeveloper: <http://forums.oracle.com/forums/forum.jspa?forumID=83>

Introduction

The image below shows what this how-to document is shooting for:



As you see, the list of employees is categorized by departments. The department category is highlighted with a different background color and font-weight and is not selectable as a value.

Implementation

The implementation of hierarchical lists is based on a hierarchical tree binding in the ADF binding layer. You can create the binding manually in the pageDef file, or simpler, drag and drop a ViewObject as a tree onto the page, configure the tree nodes and eventually remove the tree component from the page. Note that if you go for the latter approach, remove the tree component from the JSF page source as otherwise JDeveloper will delete the binding as well.

To build a selectOneChoice with a hierarchical display, you need to iterate over the tree nodes and access the child nodes

```

<af:selectOneChoice id="selectBox" label="Choose Employee"
    valuePassThru="true"
    styleClass="employeeSelectBox"
    unselectedLabel="Choose Employee">
  <af:forEach items="#{bindings.DepartmentsView1.children}"
    var="departments">
    <af:selectItem label="#{departments.DepartmentName}"
      disabled="true"/>
    <af:forEach items="#{departments.children}"
      var="employees">
      <af:selectItem label="#{employees.LastName}"
        value="employees.EmployeeId"/>
    </af:forEach>
  </af:forEach>

```

The outer `forEach` loop iterates over the departments records, retrieving the employees within the department as "child".

```

<af:forEach items="#{bindings.DepartmentsView1.children}"
  var="departments">

```

The inner `af:forEach` loop accesses the child elements through the variable name that is specified on the outer loop.

```

af:forEach items="#{departments.children}" var="employees">

```

Note that the `af:selectItem` component for the parent node is set to `disabled=true` because we don't need this item to be selectable in the choice list

```

<af:selectItem label="#{departments.DepartmentName}" disabled="true"/>

```

To apply list indenting and the color coding, CSS is used, which is why the `af:selectOneChoice` has its `styleClass` property set to **"employeeSelectBox"**.

In my example, the CSS is added to the page where the listbox shows, in your development environment you may consider using external CSS files or a skinning file, if your application has its own look and feel configured.

```

<af:document>
  <f:facet name="metaContainer">
    <af:group>
      <![CDATA[
        <style>
          .employeeSelectBox option{text-indent:15px}
          .employeeSelectBox option[disabled]{color:black;
            background-color:#dddddd;
            font-weight: bold; text-indent:0px}
        </style>
      ]]>
    </af:group>
  </f:facet>
  ...
</af:document>

```

The stylesheet is added to the **metaContainer** facet of the **af:document** object, which is considered best practices in ADF Faces RC. The first style sheet applies indenting to all option elements that are children of the listbox with the "employeeSelectBox" style class configured. The second line of CSS then defines the rules for the parent entries, which should not be indented, should have a gray text background and black text color. The definition of black text color is important because otherwise the textcolor comes up

gray, as usual for disabled entries. Also note that the parent node is identified by its **disabled attribute** (option[disabled]).

And this is it already. You can download the sample workspace **from ADF Code Corner**

<http://www.oracle.com/technetwork/developer-tools/adf/learnmore/index-101235.html>

To run the sample, all you need to do is to configure the HR database connect to point to your database's HR schema.

RELATED DOCUMENTATION

<input type="checkbox"/>	Select one choice tag - http://download.oracle.com/docs/cd/E15523_01/apirefs.1111/e12419/tagdoc/af_selectOneChoice.html
<input type="checkbox"/>	
<input type="checkbox"/>	