ADF Code Corner

023. How-to build a Generic Selection Listener for ADF bound Tables

twitter.com/adfcodecorner

Abstract:

Creating an ADF bound table with drag and drop from the ADF Data Control Palette configures the selectionListener property of the ADF Faces table component with EL.

The expression references the makeCurrent function that exists on the ADF Faces binding class at runtime. Unless you want to execute pre-processing or post-processing instructions in the context of the selection listener, there is no need to change this setting.

However, if you have a need to change it, then you can't just use the *makeCurrent* in your Java code because `FacesCtrlHierBinding`, the ADF Faces binding object for tables and trees, is located in an internal package and is for internal framework use only. In this article I explain how your build your own generic Java implementation of an ADF table selection listener to perform what the *makeCurrent* method does, just with a public API.

Author: Frank Nimphius, Oracle Corporation
twitter.com/fnimphiu
17-MAR-2010

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

You create an ADF bound ADF Faces table by dragging a collection from the ADF Data Controls panel to a page or page fragment. In the opened context menu, you choose an editable or read-only table that you then configure to support row selection, sorting and filtering.

If you configure the table for row selection, the following expression is added as the value of the table "selectionListener" attribute: #{bindings.DepartmentsView1.collectionModel.makeCurrent}. The selection listener is important and ensures that the selected row in an ADF Faces table component is synchronized with the current row setting in the ADF binding iterator.

The expression addresses the ADF binding layer through the *bindings* object and accesses the tree binding in the current page definition - the pagedef.xml file. In the example above, the tree binding is defined with the name "DepartmentsView1".

At runtime, DepartmentsView1 is represented by the FacesCtrlHierBinding object located in an internal framework package that extends the generic ADF `JUCtrlHierBinding` API. The *makeCurrent* method is defined in the `FacesCtrlHierBinding` class and is not available in the generic ADF API. Oracle discourages ADF customers from using this method in their application Java code to avoid upgrade issues due to possible future changes to `FacesCtrlHierBinding`.

One option to execute the *makeCurrent* method from Java, not using `FacesCtrlHierBinding` directly is to use a method expression in Java. This solution is documented many times and works pretty well and, as far as I can tell, is future safe.

However, as a Java developer you possibly prefer a pure Java API that does not have any strings attached. Therefore, in this article, I explain how to create a generic method that you can use in a managed bean to perform exactly what the makeCurrent method is doing.

Generic Selection Listener

When a user selects a table row, then the ADF Faces table component notifies a registered event handle about this selection, passing a SelectionEvent object. The event handler signature in a managed bean method looks as follows

```
public void handleTableSelection(SelectionEvent selectionEvent){ ...}
```

The selectionEvent object contains all the information required to re-build the *makeCurrent* method functionality of the `FacesCtrlHierBinding` class. To build a generic solution, I defined the event handler as a static method in a separate class. The source code is documented by added comments

```
package adf.sample.view;
import oracle.adf.model.binding.DCIteratorBinding;
import oracle.adf.view.rich.component.rich.data.RichTable;
import oracle.jbo.Key;
import oracle.jbo.uicli.binding.JUCtrlHierBinding;
import oracle.jbo.uicli.binding.JUCtrlHierNodeBinding;
import org.apache.myfaces.trinidad.event.SelectionEvent;
import org.apache.myfaces.trinidad.model.CollectionModel;

public class GenericTableSelectionHandler {
    public GenericTableSelectionHandler() {
        super();
    }

    /**
     * Synchronizes the table UI component row selection with the
     * selection in the ADF binding layer
     * @param selectionEvent event object created by the table
     * component upon row selection
     */
    public static void makeCurrent(
                              SelectionEvent selectionEvent){

     RichTable _table = (RichTable) selectionEvent.getSource();
     //the Collection Model is the object that provides the
     //structured data
     //for the table to render
     CollectionModel   _tableModel =
                       (CollectionModel) _table.getValue();
     //the ADF object that implements the CollectionModel is
     //JUCtrlHierBinding. It is wrapped by the CollectionModel API
     JUCtrlHierBinding _adfTableBinding =
                   (JUCtrlHierBinding) _tableModel.getWrappedData();
      //Acess the ADF iterator binding that is used with
      //ADF table binding
        DCIteratorBinding _tableIteratorBinding =
                               _adfTableBinding.getDCIteratorBinding();

      //the role of this method is to synchronize the table component
```

```
      //selection with the selection in the ADF model
      Object _selectedRowData = _table.getSelectedRowData();
      //cast to JUCtrlHierNodeBinding, which is the ADF object
      //that represents a row
      JUCtrlHierNodeBinding _nodeBinding =
                          (JUCtrlHierNodeBinding) _selectedRowData;
      //get the row key from the node binding and set it
      //as the current row in the iterator
      Key _rwKey = _nodeBinding.getRowKey();
        _tableIteratorBinding.setCurrentRowWithKey(
                                    _rwKey.toStringFormat(true));
    }
}
```

This method synchronizes the ADF binding layer with the table selection and is called from a selection event handler in a managed bean

```
package adf.sample.view.bean;
import adf.sample.view.GenericTableSelectionHandler;
import oracle.adf.model.BindingContext;
import oracle.adf.model.binding.DCIteratorBinding;
import oracle.binding.BindingContainer;
import oracle.jbo.Row;
import org.apache.myfaces.trinidad.event.SelectionEvent;

public class DepartmentsBacking {
    public DepartmentsBacking() {
    }

    public void onTableSelect(SelectionEvent selectionEvent) {

    // your pre-trigger code goes here
    // ...

      GenericTableSelectionHandler.makeCurrent(selectionEvent);

    //your post-trigger code goes here
    // ...
    }
}
```

Last, but not least, the custom selection listener method is referenced from the ADF Faces table component instead of the existing EL

```
<af:table value="#{bindings.DepartmentsView1.collectionModel}"
          var="row"
          rows="#{bindings.DepartmentsView1.rangeSize}"
          emptyText="…"
          fetchSize="#{bindings.DepartmentsView1.rangeSize}"
          rowBandingInterval="0"
          selectedRowKeys=
             "#{bindings.DepartmentsView1.collectionModel.selectedRow}"
          rowSelection="single" id="t1"
          selectionListener="#{DepartmentsBacking.onTableSelect}">
```

Note that if you don't need to customize the selection listener behavior, for example to process pre- and post instructions, or call multiple event handlers in a sequence, then there is no need to change the default setting.

Download Sample

This is a little JDeveloper 11g R1 PS1 tester application that prints the current selected row data when pressing a button. Configure the model to connect to a HR schema on your local database before running it. Download the sample from **ADF Code Corner**:

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