ADF Code Corner

053. How-to refresh a bounded task flow that is displayed as an ADF Region in a popup dialog

Abstract:

Every technology seems to have its superstar; a feature that developers love and use more than any other feature. Though in ADF Faces RC, there seem to be more than one super star feature, the clear winner of the Oracle ADF Faces RC pop idol feature contest is a bound taskflow that is added to a page in a region that is displayed in a popup dialog on demand. The most frequent question asked on the JDeveloper help forum on OTN is how to refresh such a bound taskflow - and this indeed doesn't seem to be obvious until you have a look at the developer guide documentation.

twitter.com/adfcodecorner

Author: Frank Nimphius, Oracle Corporation
twitter.com/fnimphiu
29-MAY-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.

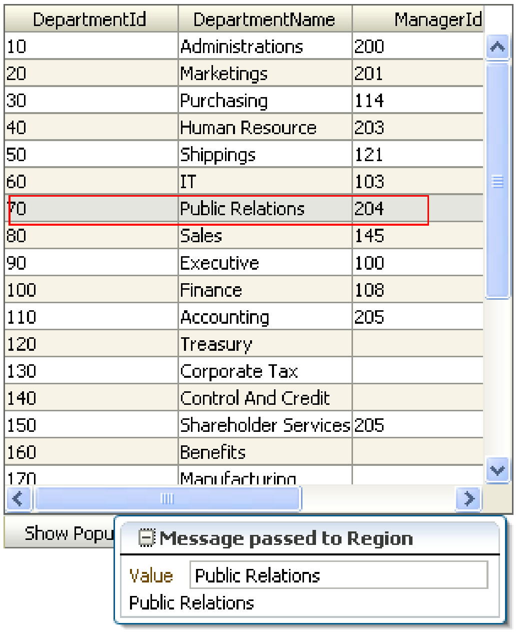
Please post questions or report problems related to the samples in this series on the OTN forum for Oracle JDeveloper: http://forums.oracle.com/forums/forum.jspa?forumID=83

Introduction

To picture this, a bound taskflow is a story board of a business process programmed in web technology, namely the Oracle ADF Faces component set and the Oracle ADF binding layer. Bound taskflows is an Oracle extension to the JavaServer Faces navigation model that enables developers to reuse page and method flows (better referenced as activities) within an application or across applications. Taskflows can contain page fragments - *.jsff files - that allow a taskflow to run embedded within a parent page. The surrounding container for the taskflow in a page is the af:region component.

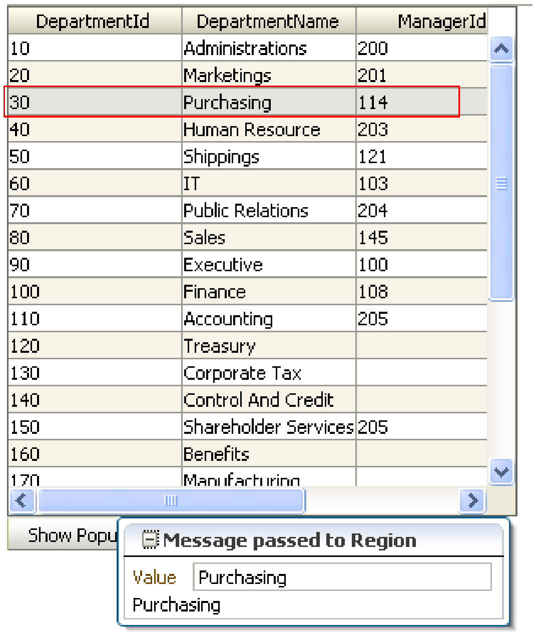
The example scenario chosen for this how-to article is a page with a table in it for the user to select a row. Once a row is selected, the user presses a button to launch a popup dialog with a one page (-fragment) taskflow in it. The selected row's department name is passed as an input parameter to the taskflow so it can be displayed in a textfield. If this usecase sounds trivial, then this is exactly what it is supposed to be.

Two third of the key to the solution is documented in chapter **17.1.7 What You May Need to Know About Refreshing an ADF Region** of the **Fusion Developer’s Guide for Oracle Application Development Framework.** The guide is available on OTN, the chapter I am referencing here is chapter 17.1.5 in that version.



DepartmentId	DepartmentName	ManagerId
10	Administrations	200
20	Marketings	201
30	Purchasing	114
40	Human Resource	203
50	Shippings	121
60	IT	103
70	Public Relations	204
80	Sales	145
90	Executive	100
100	Finance	108
110	Accounting	205
120	Treasury	
130	Corporate Tax	
140	Control And Credit	
150	Shareholder Services	205
160	Benefits	
170	Manufacturing	

Value Public Relations
Public Relations



DepartmentId	DepartmentName	ManagerId
10	Administrations	200
20	Marketings	201
30	Purchasing	114
40	Human Resource	203
50	Shippings	121
60	IT	103
70	Public Relations	204
80	Sales	145
90	Executive	100
100	Finance	108
110	Accounting	205
120	Treasury	
130	Corporate Tax	
140	Control And Credit	
150	Shareholder Services	205
160	Benefits	
170	Manufacturing	

Value Purchasing
Purchasing

What you should know about refreshing a Region

For simplicity, I copied the content from the developer guide for you as an overview

"An ADF region initially refreshes when the parent JSF page on which the region is located first displays. To execute the ADF task flow at the same time that the region renders, you must trigger a task flow binding refresh as the same time that the region renders.

During the initial refresh, any ADF region task flow binding parameter values are passed in from the parent page. The parameter values are used to display the initial page fragment within the ADF Region. If the ADF bounded task flow relies on input parameter values that are not available when the page is first created, ensure your task flow will behave correctly if the input parameter values are null.

An ADF region task flow binding can be refreshed again based on one the following task flow binding attributes:

***Neither Refresh or RefreshCondition attributes specified (default)**- If neither the Refresh or RefreshCondition task flow binding attributes are*
specified, the ADF Region is only refreshed once at the time the parent page is first displayed.

***RefreshCondition="#{EL.expression}"** The ADF region is refreshed a single time when its RefreshCondition evaluates true. The RefreshCondition must evaluate to a boolean value. At the time the RefreshCondition is evaluated, if the variable bindings is used within the EL Expression, the context refers to the binding container of the parent page, not the page fragment displayed within the ADF Region. RefreshCondition is independent of the change of value of binding parameters.If the task flow binding parameters do not change, nothing within the ADF region will change.*

***Refresh="ifNeeded"** Any change to a task flow binding parameter values causes the ADF region to refresh. There are no values other than ifNeeded for the Refresh attribute.*

If the ADF region task flow binding doesn't have parameters, Refresh="ifNeeded" is equivalent to not specifying the Refresh attribute. If you set Refresh to ifNeeded, the RefreshCondition attribute should not be specified.

Example 17–2 Refresh Option Specified in ADF region binding

```
<taskFlow id="Department1" taskFlowId="/WEB-INF/Department#Department"
   Refresh="ifNeeded"
   xmlns="http://xmlns.oracle.com/adf/controller/binding">
   <parameters>
   <parameter id="DepartmentId"
              value="#{bindings.DepartmentId.inputValue}"
   xmlns="http://xmlns.oracle.com/adfm/uimodel"/>
   </parameters>
</taskFlow>
```

If the variable, bindings, is used within the EL expression, the context refers to the binding container of the parent page, not the page fragment displayed within the ADF region. You do not need to refresh an ADF region to refresh the data controls inside the ADF region. During the ADF lifecycle, the refresh events telling the iterators to update will be propagated to the binding container of the current page of the ADF region".

For the usecase handled in this how-to document, the important information to carry away is that a region is refreshed after setting the refresh property on the region binding in the pageDef file of the parent page to **ifNeeded**. The region then refreshes as soon as the input parameter of the taskflow changes its value. In my how-to example that you can download at the end of this document, the parameter is linked to an attribute binding:

```
<executables>
  <iterator Binds="DepartmentsView1" RangeSize="25"
   DataControl="AppModuleDataControl" id="DepartmentsView1Iterator"/>
  <taskFlow id="tabletaskflowdefinition1"
   taskFlowId="/WEB-INF/table-task-flow-definition.xml
               #table-task-flow-definition"
   xmlns="http://xmlns.oracle.com/adf/controller/binding"
   Refresh="ifNeeded">

        <parameters>
        <parameter id="inputMessage"
                   xmlns="http://xmlns.oracle.com/adfm/uimodel"
        value="#{bindings.DepartmentName.inputValue}"/>
   </parameters>
   </taskFlow>
 </executables>
```

You may have noticed that my usecase is about reading the DepartmentName of the selected table row to then become the input argument for the taskflow property. In the code snippet above however you see me using an attribute binding. Well spotted! I use a little trick to simplify the work and to improve readability. I created an attribute binding for the row attribute I am interested in. As soon as the user selects a table row as current, the iterator too has this row set as current thanks to ADF. Thus the attribute binding always points to the current row in the iterator, which means that it exposes an easy to use interface to the current row's DepartmentName attribute.

The remaining one third of the key to success

Two third of the key is what is documented in the developer guide. The remaining hint is that the popup dialog that contains the region needs to have its content delivery property set to immediate:

```
<af:popup id="popup1" contentDelivery="immediate">
   <af:region value="#{bindings.tabletaskflowdefinition1.regionModel}"
   id="tabletaskflowdefinition1"/>
</af:popup>
```

If this property is left to deferred loading of data then the refresh of the region is never displayed. I assume this piece of information to be the main reason why developers didn't get this usecase to work.

What I didn't cover

In this how-to document I didn't cover how taskflows are created and regions are build. This however is well explained in the developer guide referenced below.

Download

The JDeveloper 11 TP4 workskpace requires you to setup the hr schema connection for your database. For this, select the hr connection entry in the database navigator and choose "properties" from the context menu. Change and test the connection and you are ready to run the application.

Download from ADF Code Corner:

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

Download

RELATED DOCOMENTATION

☒	Oracle® Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework11g Release 1 - http://download.oracle.com/docs/cd/E15523_01/web.1111/b31974/toc.htm
☒	
☒	