**ADF - Refresh page programmatically**

Some times when you develop your ADF application, you need to refresh programmatically a page. You can achieve this using the internal page ID in backing bean class usgin the following code:

ViewHandler vh = fctx.getApplication().getViewHandler();  
String pageId = fctx.getViewRoot().getViewId();  
UIViewRoot vr = vh.createView(fctx, pageId);  
vr.setViewId(pageId);fctx.setViewRoot(vr)

**ADF - Refresh child Iterator via View Link Accessor**

I'm back... after few months I found five minutes to write a short post on my blog, from Sydney, my new home.

There are a lot of use cases where you need to refresh child iterators of a master ViewObject via View Link Accessor, for example in tree component. In this post I'm writing a simple implementation to achieve that.

DCIteratorBinding masterIterato = .....  
ViewObject vo = masterIterato.getViewObject();  
Row[] deptRow = vo.findByKey(new Key(new Object[] { objectId }), 1);  
RowSet childRows = (RowSet)deptRow[0].getAttribute("ViewLinkAccessorName");  
childRows.executeQuery();

In the code you can show the way to obtain the RowSet representing the child rows of a specific master Row, then re-execute the query of RowSet. Master Row is found via findByKey method on ViewObject.

# [Oracle ADF Step by Step Guide - Oracle ADF Tutorial](http://www.learnoracleadf.com/)

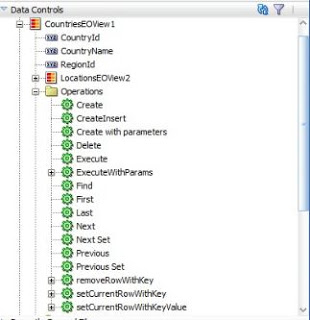
Information about the Oracle ADF,Oracle Application Development Framework,ADF, Oracle ADF, J2EE Framework, BC4J.

* [Home](http://www.learnoracleadf.com/)
* [All Posts](http://www.learnoracleadf.com/search/label/ADF)
* [Privacy Page](http://www.learnoracleadf.com/p/privacy-page.html)

## Monday, May 13, 2013

### ADF DataControl Built-in View Operations - Part I

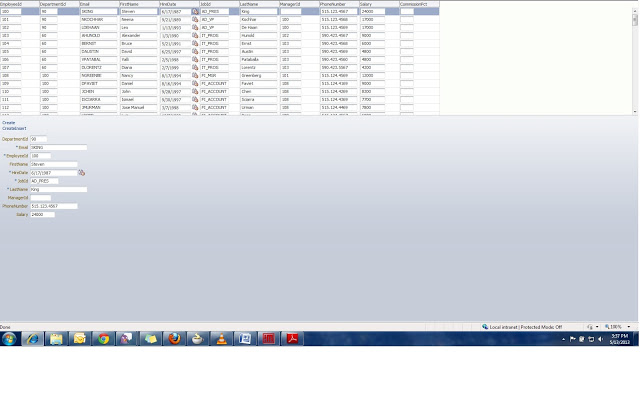
Every view object instance added to ADF AM data model appears in the Data Controls panel. The attributes in the view object will also be displayed under the corresponding data control view object . Along with it, there are many Built-in view object operations available .Lets see each one of them with an example.  
  
Data Control View object operations

[](http://4.bp.blogspot.com/-5jlVuHbS5Gc/UZCAky09mCI/AAAAAAAAANQ/OXUwhO5-htI/s1600/ADF+Datacontrol.jpg)

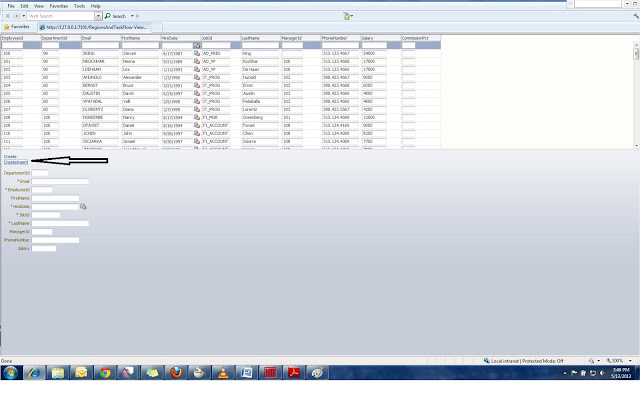
you can drag and drop operations on to the UI page as a ADF Link /ADF Button and with the available options .   
  
Drag and Drop the Employee Data control on to the UI page as ADF Table .

1) Create and CreateInsert :

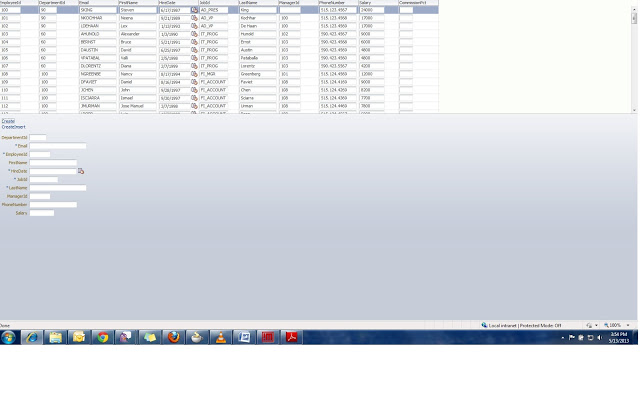
I have quite a big story to explain and how did i understood this . I have dragged and dropped EmpVO1(Employee VO data control ) on to the page as ADF Table and as ADF Form . I have also dragged and dropped Create and CreateInsert operations as ADF Link . See the below image

[](http://2.bp.blogspot.com/-O_5JG1yZvXA/UZC7ied7-VI/AAAAAAAAANg/6pCfM-I8aJM/s1600/ADF+Datacontrol1.jpg)

Create will creates a new row but does not insert it . But what does it mean , it creates but does not insert ?  
Yes.... Create will just open a slot . Row will be initialized. But it does not insert a row .  
  
see this code :  
  
vo.insertRow(row); // This is how insertion happens . In the case of Create , this will not be invoked   
  
Create will just initialize a row whereas CreateInsert initialize the row as well as insert row.  Look at the image , when we click CreateInsert a new blank row is being created and displayed in the ADF table as well as in ADF Form. CreateInsert will work in ADF Form component and ADF Table

[](http://3.bp.blogspot.com/-e-FKCGVt-Mk/UZC97n6c9QI/AAAAAAAAANs/ZhYYbiEanyM/s1600/ADF+Datacontrol+CreateInsert.jpg)

Create will work only on ADF Form component.  
When we click Create , new row added being displayed ADF Form component and not to ADF Table

[](http://2.bp.blogspot.com/-qIMY5oWM1S4/UZC_karaWNI/AAAAAAAAAOA/0URJAx5wf74/s1600/ADF+Datacontrol+Create.jpg)

OK.. we can directly use CreateInsert instead of Create. why confuse? There are situations we need to bulk insert or selective insert based on the user choice .  
Remember operation carried by 'Create' is not part of transaction so need not to be roll backed when don't want commit the new rows . when a row is marked as initialized it will be removed from Transaction pending changes.The new row which is initialized with any default values that the underlying EO has defined. If the user never enters any data , then it means no row is being created logically. .no worries during commit time  
  
Just to check this, write a custom method and bind it to the both the links. Look at the snippet .  
  
<af:commandLink actionListener="#{backingBeanScope.DCOperations.onCreate}" text="Create"  
                    disabled="#{!bindings.Create.enabled}" id="cl1"/>  
 <af:commandLink actionListener="#{backingBeanScope.DCOperations.onCreateIns}"  
                    text="CreateInsert"  
                    disabled="#{!bindings.CreateInsert.enabled}" id="cl2"/>  
  
I have made action listener of both the links pointing to a custom method .  
  
  
  
After clicking the Create link in the UI when u check the output  row count before and after will be the same. But in CreateInsert , row count will not be same and it will be +1 always compare to the before row count because it  has already inserted  a row in the iterator.  
  
When u want to remove a row once CreateInsert is made , u need to rollback and careful what you are removing because CreateInsert operation is part of transaction.  
  
Finally the difference between Create and CreateInsert is : User can just navigate away from the page without worrying about the created row in Create operation. User has to justify the created row  because it is already inserted in the iterator and a part of transaction too..  Another important case is , difference between two really matters if it is ADF Table.  
  
2) Create with Parameters :   
  
            It creates a new row  accepting parameter values.  
  
3) First,Last,Next :   
  
           Sets the current row to the First,Last,Next row in the rowset.  
  
4) Next and Previous Set :  
  
       Sets the set of rows forward to one set and backwards (i.e) previous.