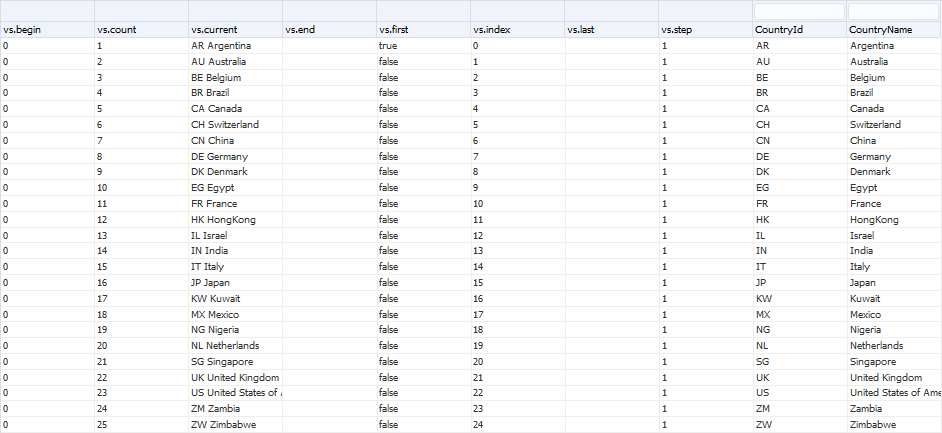
**resolve first and last row in the af:table using varstatus**

During my current project i've been asked about rendering a button for first and last row in a table. There is very simple way to achieve that without a single line of Java code. The keyword to the success is varStatus. I've prepared simple application where on the testPage i've drag&dropped sortable and filterable read only table. Also i've added few columns where varStatus properties are displayed.

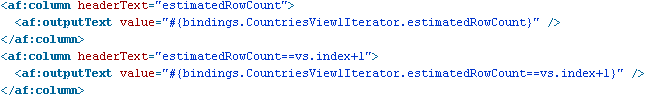
[](https://3.bp.blogspot.com/-WWRLFa1Pb2Q/VBv47EkDZZI/AAAAAAAAAY8/SDTawAuUkVs/s1600/1%2Bjsf%2Bcode.png)

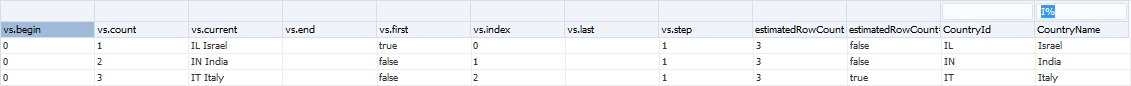
[](https://2.bp.blogspot.com/-G_KCGM_ipYg/VBv47Jj6FbI/AAAAAAAAAZA/R0Ey3Op9soo/s1600/2%2Bpage.png)

What can be noticed here:

* **begin** always returns 0, it's the index of first element in the collection
* **count** returns the number of objects in the collection for the current step
* **current** returns the reference to the current row (in the iterators loop)
* **end** is supposed to return index of the last element - unfortunately it returns null
* **last** is supposed to return true for the last element and false otherwise, it's null (see [bug: varStatus.last working?](https://community.oracle.com/thread/594043?start=0&tstart=0))
* **index** returns row index, could be used to display order number value for each row as #{vs.index+1} because it starts from 0
* **step** is always 1 - each row from the collection should be displayed

OK, so what needs to be done to properly point the last row? Just small and simple trick:

[](https://4.bp.blogspot.com/-kOg4anCa_0U/VBv5eCuTVNI/AAAAAAAAAZM/43lk73zLrfA/s1600/3%2Bjsf%2Btrick.png)

[](https://4.bp.blogspot.com/-MK0F0IAd8aQ/VBv5eiPUL7I/AAAAAAAAAZQ/HusL2RbR2ac/s1600/4%2Bpage%2Btrick.png)

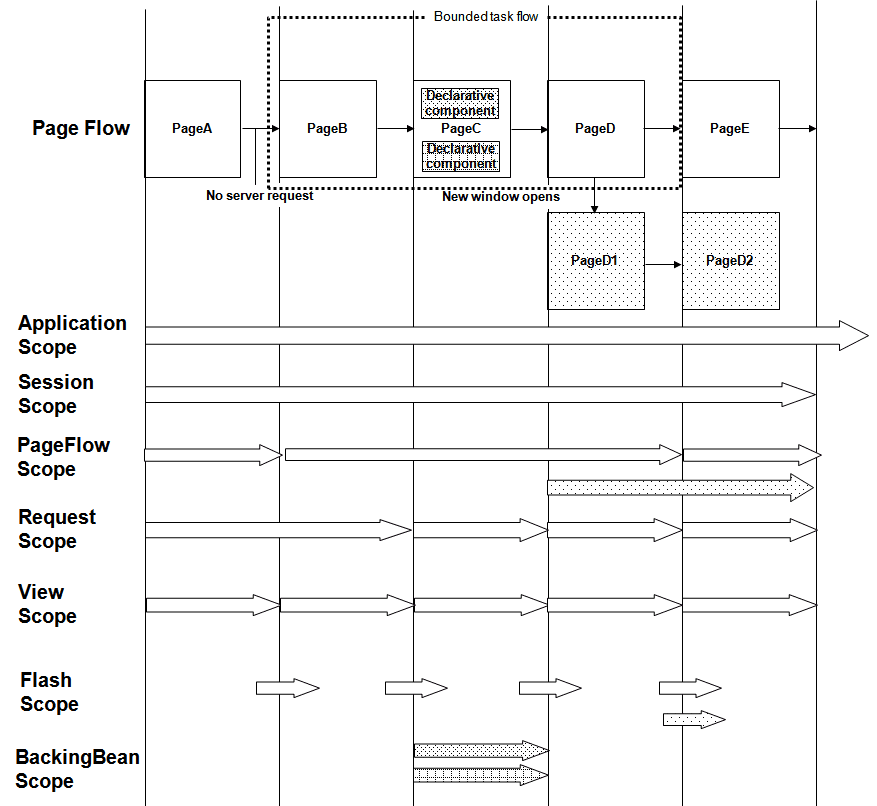
You need to compare current row index with row count in the collection #{bindings.TheNameOfTheIterator.estimatedRowCount==vs.index+1}. The screenshots above presents filtered collection. Filtering and sorting don't corrupt the solution.  
  
Please notice that varStatus is not only related with <af:table> but also with other components like <af:iterator> or <af:forEach> and its attribute set (and returned values) may differ there.

**meaning of a transient word in ADF**

In plain Java applications there is a transient keyword that you could use to notice the JVM that values are from some sensitive and they shouldn't be persisted. ADF is an J2EE framework, so while developing Java code in your application you have an ability to use transient keyword. Such values won't be serialized. Even more, there are situations where you must to use the transient keyword. If your application needs to work in a clustered environment, the object needs to implement Serializable interface. But object in the framework don't implement it. That's why almost each time when you bind an UI component to a bean, you must to mark the references as transient.

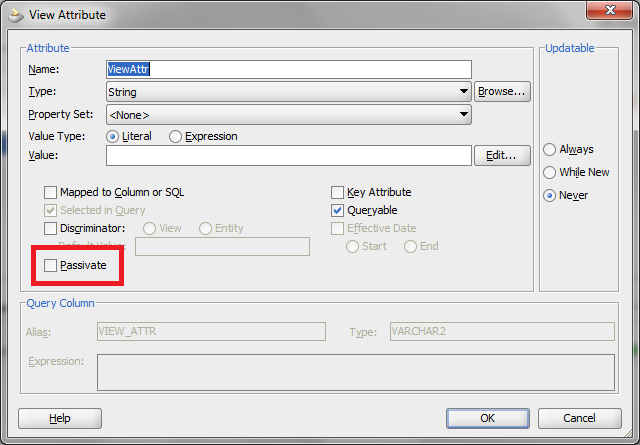
[https://2.bp.blogspot.com/-3nKe-PUJSK0/U9eWIpYGKiI/AAAAAAAAARI/7b8tmeiGfiM/s1600/1+bean.png](https://2.bp.blogspot.com/-3nKe-PUJSK0/U9eWIpYGKiI/AAAAAAAAARI/7b8tmeiGfiM/s1600/1+bean.png)

There is only one exception to that rule. And it depends on scope within your bean is defined. If the scope is request or less (shorter/more precise like backing bean) the rule doesn't apply. Because the user won't notice that the previous request server is now down.

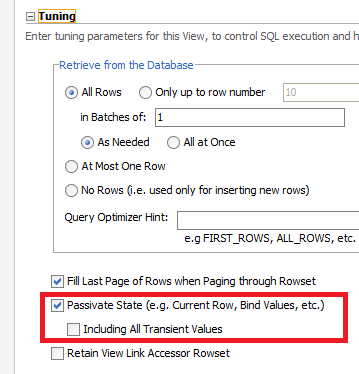
[](https://3.bp.blogspot.com/-g5qQkB76ZYE/U9eXjMpXFxI/AAAAAAAAARU/xviUSMAwYAI/s1600/adf_scopes3.png)

(image source: http://docs.oracle.com/cd/E16162\_01/web.1112/e16182/adf\_lifecycle.htm)

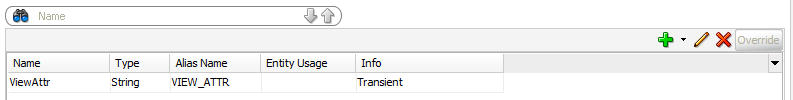
There are also another places in the ADF application where you can find transient values. In the BC4J model layer there are view objects. An attribute of a view object could be entity-based, calculated in the SQL query or ... Transient. It is possible to mark an java.lang.String attribute as transient, but does it mean that it won't be serialized? It will be. To visualize the concept of the keyword, transient means that the attribute value won't be persisted in the database. But that sentence needs more clarification. As ADF developer you make a decision if the attribute value is being recalculated all the time (i.e. using Groove) or if it stores some values during the session - in other words you're making a decision if the attribute will be passivated or not. You can achieve it on the attribute level:

[](https://1.bp.blogspot.com/-CxfJN0Epk5Y/U9eb2KydubI/AAAAAAAAARg/h8fJf_ZQgzA/s1600/2+attr+def.png)

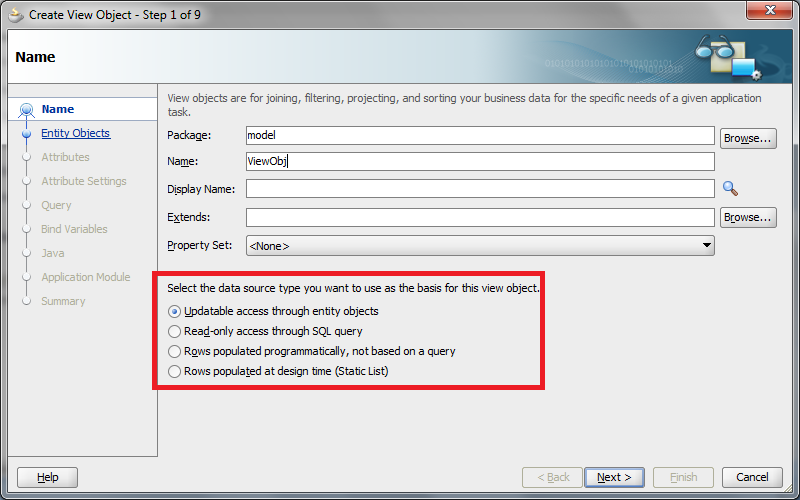
or globally for each transient value in the view object. You change settings in the Tuning section in General tab.

[](https://3.bp.blogspot.com/-pHNw9Rkdcjg/U9eb9lc12VI/AAAAAAAAARo/ubveDDAmpnM/s1600/3+tuning.png)

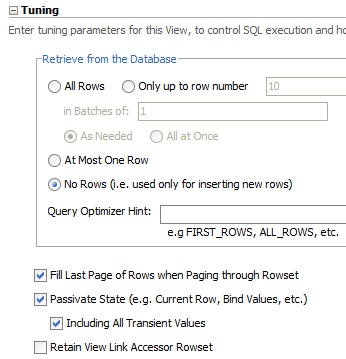
Saving the state requires that the class type of an attribute must be serializable, for example the java.lang.String is. To clarify the previous sentence - if your passivation store is set to the database (default), the value will be stored there.  
  
You can easily recognize if the view object contains transient attributes or not. Just check the value in the Info column next to the attribute name and class.

[](https://1.bp.blogspot.com/-t2fbU9AebiM/U9eiYZxZvnI/AAAAAAAAASk/3mqhWqz7op0/s1600/2b+attr+list.png)

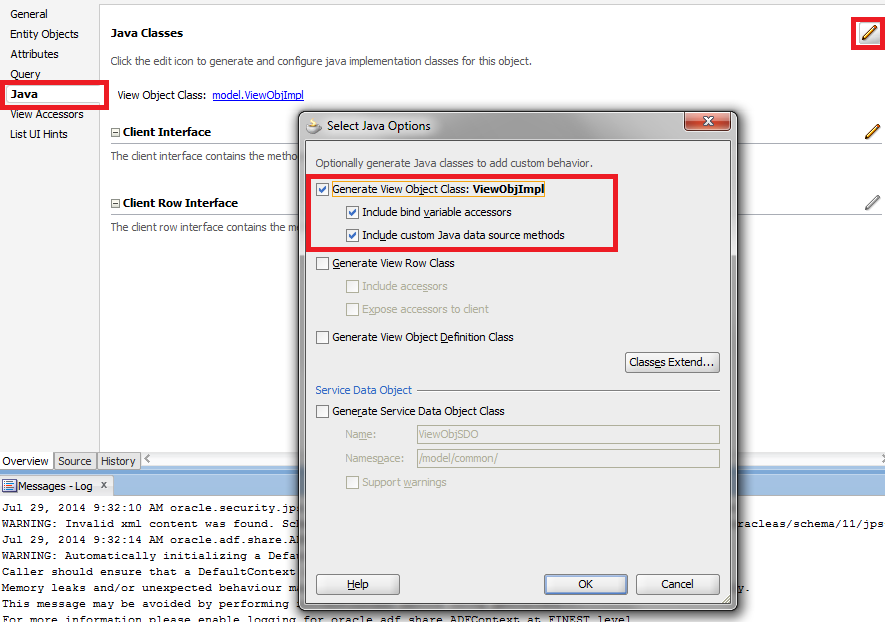
As we talk about view objects, a VO itself can be transient. Please review the first page of wizard when you want to create a new one.

[](https://1.bp.blogspot.com/-pQBwmOszBVY/U9edkgHNJdI/AAAAAAAAAR0/x0sPWD_D6Ys/s1600/4+vo+types.png)

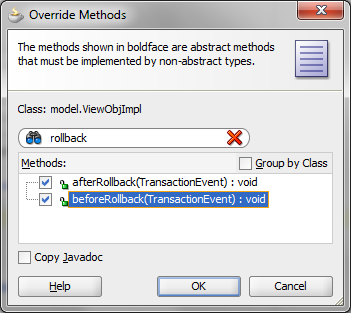
The third option is for transient view objects. Please go through the wizard. Remember to add at least one primary key attribute. You must to perform additional steps to make it works as any other type. Go to view object's General tab and expand the Tuning section. In Retrieve from the Database group select No rows option and check mentioned Including All Transient Values checkbox.

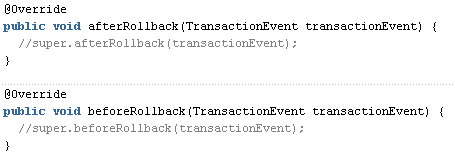
[](https://2.bp.blogspot.com/-P5tP_lTO6yE/U9efc6-kKAI/AAAAAAAAASA/v5BwJQggKpI/s1600/5+tvo.png)

If it is not done yet automatically, please create the ViewObjectImpl implementation class for that view object.

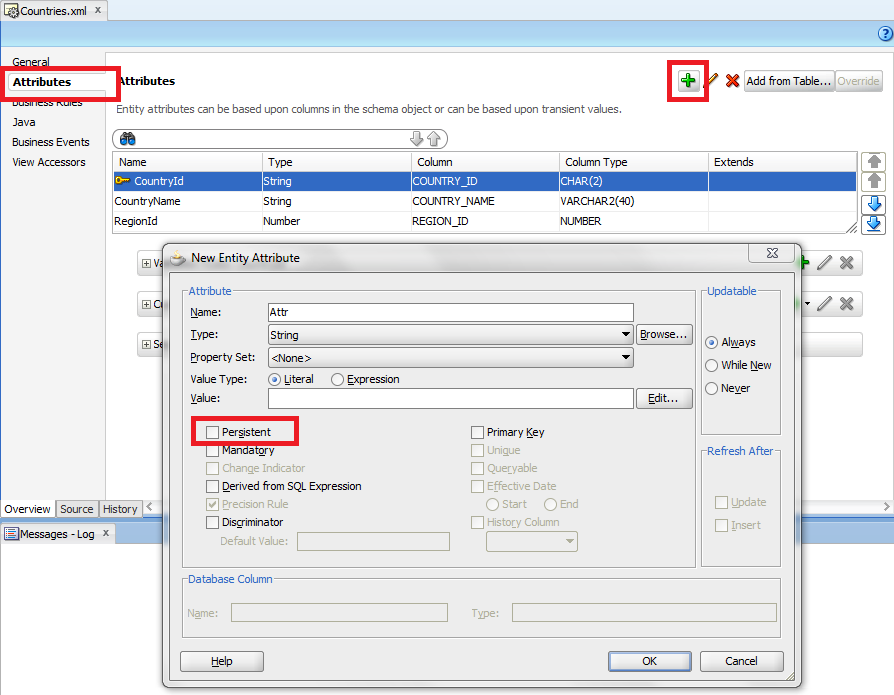
[](https://2.bp.blogspot.com/-EkK8h7eVzKY/U9egNxdzjcI/AAAAAAAAASI/jkwkwGkYKA0/s1600/6+impl.png)

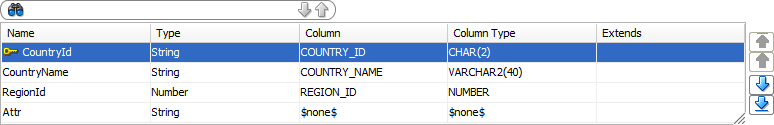
While editing Java class, click on Override Methods in the toolbar Source menu. Override beforeRollback and afterRollback methods. Comment the super calls.

[](https://2.bp.blogspot.com/-0_GCLSufv7g/U9ehO4fjk_I/AAAAAAAAASU/A6wq-jSrhJU/s1600/7+override+methods.png)

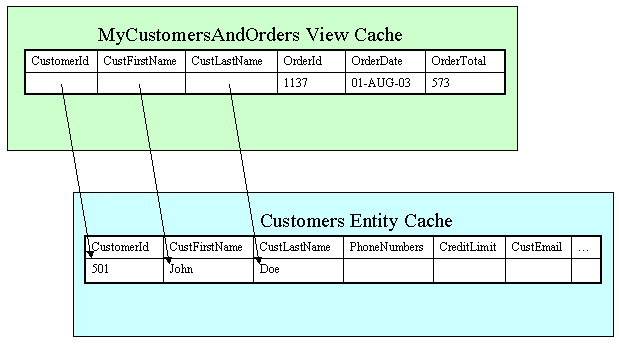
[](https://2.bp.blogspot.com/-_hj5naQk8NM/U9ehO7aqMBI/AAAAAAAAASY/nmHMWGgSf4w/s1600/8+methods+impl.png)

Transient attributes are one of the approach as value holders for web service invocation results.  
  
There is another place to define ADF-meaning transient values, at the entity attribute level. As you can see on the screen below, it is possible not to persist it.

[](https://3.bp.blogspot.com/-qyw9r8pxIUw/U9ingl4YMwI/AAAAAAAAAS0/Z_DxPWRINoQ/s1600/9+entity+transient+attr.png)

[](https://1.bp.blogspot.com/-1WAQhsA5Xoc/U9in6sCuK6I/AAAAAAAAAS8/XaVLLFw2r2Y/s1600/10+attr+added.png)

The attribute is not mapped to the database table. There is no such thing like passivation available on the entity level. So what is going to happen with it? Do I loose the value between requests? The answer is no. You are not going to loose the value. Furthermore you're able to share the value between view objects based on that entity. The persistence store here is the entity cache.

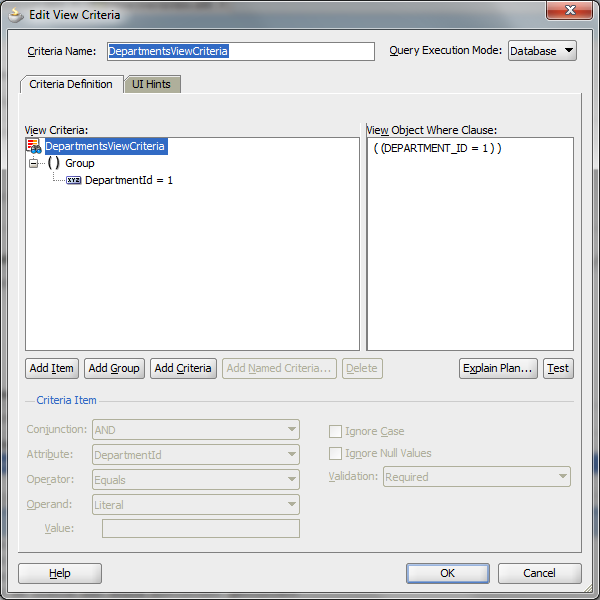
[](https://3.bp.blogspot.com/-yKLBcQjywfY/U9isv0lrocI/AAAAAAAAATM/cCjF49RxsMU/s1600/persistent_caching.gif)

(image source: http://download.oracle.com/otn\_hosted\_doc/jdeveloper/1012/bc4j/intro/bc\_acaches.html)

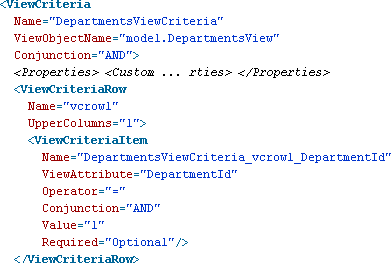
Just to summarize - ADF meaning of transient keyword is not to persist directly in the database. View objects can be whole transient or contain transient attributes (in addition to entity-based and calculated attributes). Entities can have their own transient attributes that are stored in entity cache and can be consumed and shared by view objects.

### tricks with view criterias

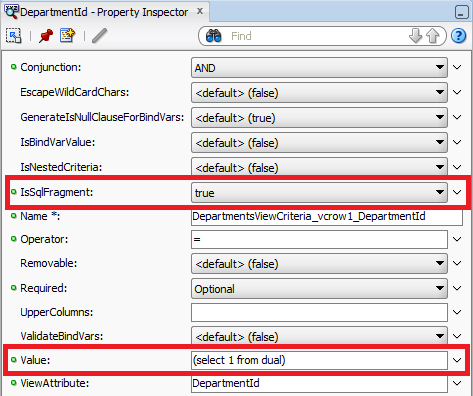
View Criterias provide useful mechanism to extend where clause in your SQL query. Most of the cases can be provided using wizards. But, as real life shows, wizard is only a wizard. It doesn't allow you to put an inline query inside. What if such query needs to be parametrized by a binded variable? Using wizard i've created the sample view criteria:

[](https://4.bp.blogspot.com/-606vBFUSuiI/U_Y4vmPSpWI/AAAAAAAAAU8/d1i5ZaPgess/s1600/1%2Bwizard.png)

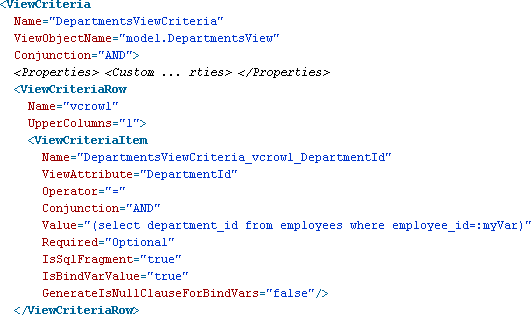
Let's go to the source (or view object structure if you like) to give the answer for that kind of the questions.

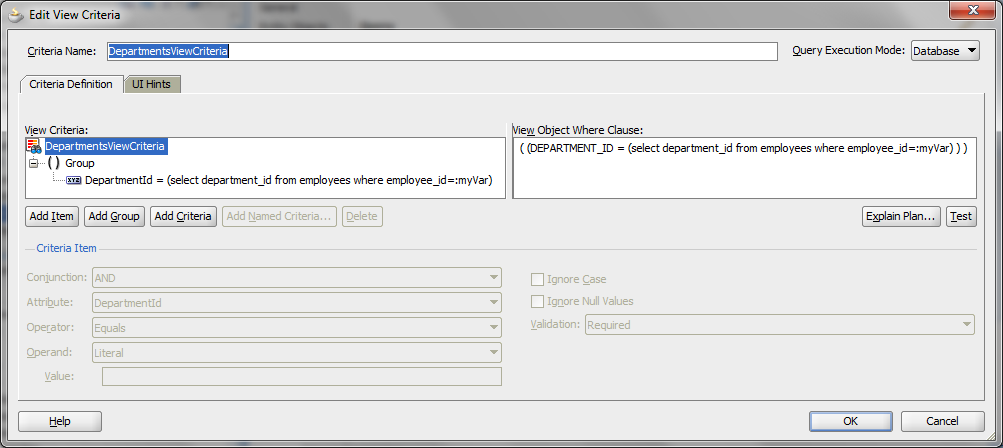
[](https://1.bp.blogspot.com/-8LHcGx7ko8U/U_Y41jCnOoI/AAAAAAAAAVE/znTBl6NAxaU/s1600/2%2Bsource.png)

Please select ViewCriteriaItem node and go to its Property Inspector.

[](https://1.bp.blogspot.com/-htencvHuPKc/U_Y49awukkI/AAAAAAAAAVM/HxoXsU_dTJs/s1600/3%2Bvci%2Bprop%2Binsp.png)

To put an SQL query to the view criteria you need to do two things. First you need to change IsSqlFragment flag to true. Second is to write an SQL query in Value attribute. Please remember about putting it into brackets.  
  
If you need to parametrize the query with bind variable, you need to set two additional flags. When you set IsBindVarValue to true, the BC4J framework will treat the value as a bind variable. I won't paste the screenshot from the wizard after leaving GenerateIsNullClauseForBindVars flag with its default. This flag needs to be set to false to avoid SQL comparison Values attribute value with null.

[](https://4.bp.blogspot.com/-Gc8i7-DhbVU/U_Y5QlnDz9I/AAAAAAAAAVc/cctVG2kmI4Q/s1600/4%2Bvci%2Bsql%2Bbind.png)

[](https://4.bp.blogspot.com/-PFc_a1BfEFs/U_Y5QohvmdI/AAAAAAAAAVg/s_vtec97m8M/s1600/5%2Bwizard%2Bscreen.png)

Of course the bind variable needs to be defined as in default, wizard way. Screenshot from the view object's source:

[https://2.bp.blogspot.com/--e6h3UNW_Ts/U_cK_UEowCI/AAAAAAAAAVs/Ed5CPX1Ofaw/s1600/6%2Bvar%2Bdef.png](https://2.bp.blogspot.com/--e6h3UNW_Ts/U_cK_UEowCI/AAAAAAAAAVs/Ed5CPX1Ofaw/s1600/6%2Bvar%2Bdef.png)

After changing values using Property Inspector, please be aware that the wizard may change flag values and whole query won't work as you expect.

Posted by [Kamil Krasowski](https://plus.google.com/101202787860923401081)at [11:25](https://kamilkrasowski.blogspot.com/2014/08/tricks-with-view-criterias.html)

[Email This](https://www.blogger.com/share-post.g?blogID=4320068064869669094&postID=7040246099868581354&target=email)[BlogThis!](https://www.blogger.com/share-post.g?blogID=4320068064869669094&postID=7040246099868581354&target=blog)[Share to Twitter](https://www.blogger.com/share-post.g?blogID=4320068064869669094&postID=7040246099868581354&target=twitter)[Share to Facebook](https://www.blogger.com/share-post.g?blogID=4320068064869669094&postID=7040246099868581354&target=facebook)[Share to Pinterest](https://www.blogger.com/share-post.g?blogID=4320068064869669094&postID=7040246099868581354&target=pinterest)

#### 1 comment:

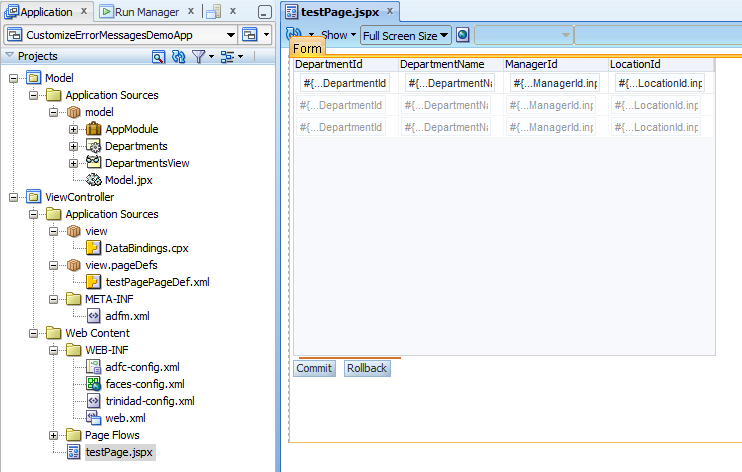
1. https://2.bp.blogspot.com/-7FlL3ofZac8/U03PJqXEvvI/AAAAAAAACUA/_HXlQz76FAk/s35/*

[**Renan Monteiro's Sharepoint**](https://www.blogger.com/profile/14170907780262017802)[25 September 2014 at 08:14](https://kamilkrasowski.blogspot.com/2014/08/tricks-with-view-criterias.html?showComment=1411658059507#c2419328771648636046)

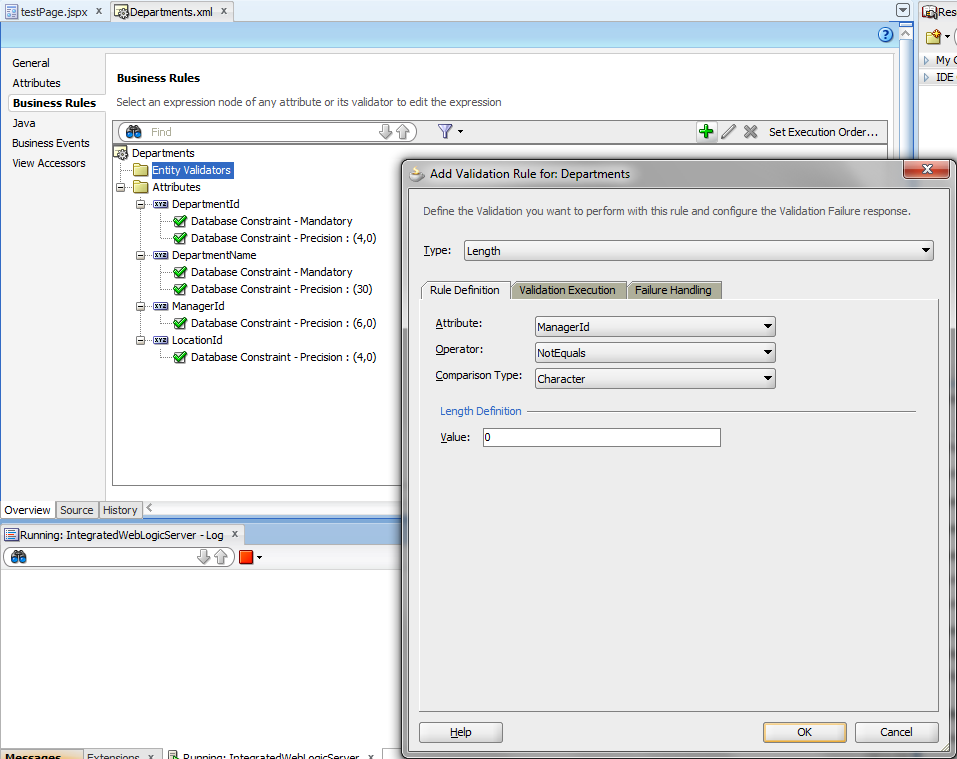
Very useful tutorial! Here is a example using Java API:  
  
ViewCriteria vc = this.createViewCriteria();  
ViewCriteriaRow vcRow = vc.createViewCriteriaRow();  
ViewCriteriaItem vcItem =  
vcRow.ensureCriteriaItem("DocumentId");  
vcItem.setOperator(JboCompOper.OPER\_IN);  
vcItem.setIsSqlFragment(true);  
vcItem.setValue("select 1 from dual");

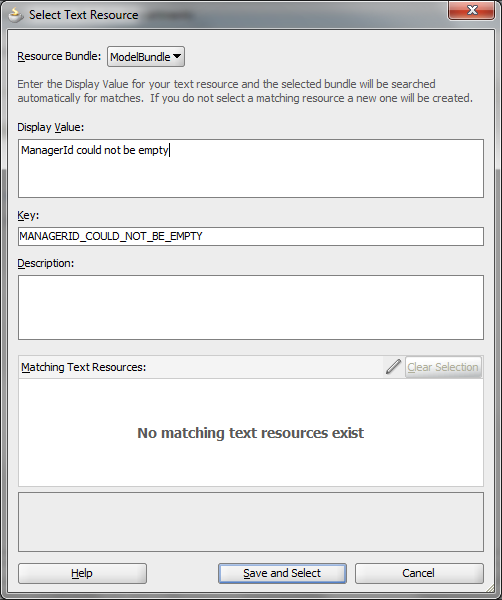
**how to customize error messages in the BC4J**

That exercise requires you to prepare sample fusion application that bases on departments table in HR schema. There will be only one page that displays updateable grid of data from the ViewObject with additional Commit and Rollback buttons (always enabled).

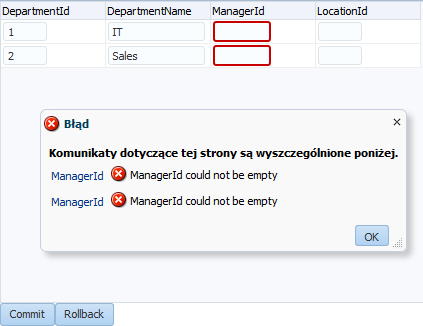
[](https://1.bp.blogspot.com/-s3JhRneB6ek/U9a2oVUmaHI/AAAAAAAAAPw/u58_WW6ppek/s1600/1+project+structure.png)

For test purposes assumption is that each department has assigned a manager to it (ManagerId is not empty). Please go to the Department entity, select Business Rules tab. Click on the Entity Validators node and green plus icon in the top right corner.

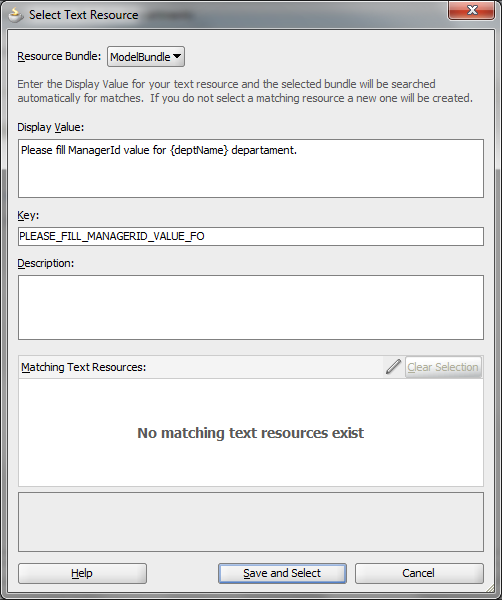
[](https://2.bp.blogspot.com/-3d70vcqDxh0/U9a4kuWa6EI/AAAAAAAAAP8/MZfaQKO27l8/s1600/2+entity+validator.png)

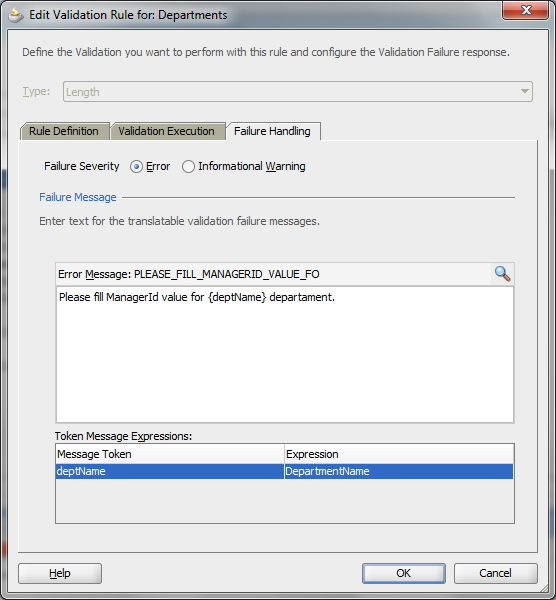
[](https://4.bp.blogspot.com/-kW_lErOMrb0/U9a5kPLS6zI/AAAAAAAAAQE/Z-6xRQS4uDk/s1600/3+error.png)

The rule disallows null values in the ManagerId attribute. Sample is ready to run. Remove values from two or more rows in the table and click Commit button.

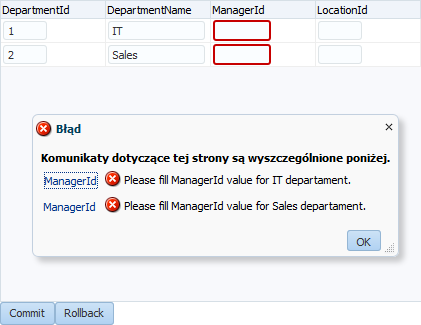
[](https://1.bp.blogspot.com/-bc1EBeDjhy4/U9a64b_HySI/AAAAAAAAAQQ/cjYTLff8hco/s1600/4+first+run.png)

Entity validators are fired properly. But here you can see all of the attributes, it is really easy to point the row that causes an error. The real life is not so simple. For the user it'll be a lot easier with the human readable message that contain hints how to fix the problem.  
  
Please go back to the error message definition and change them as on the screens below. Note that Expression field executes Groove scripts.

[](https://1.bp.blogspot.com/-l73XD2lHdI4/U9a7taaTK8I/AAAAAAAAAQc/M9iYnEXM-rI/s1600/5+msg+change.png)

[](https://1.bp.blogspot.com/-YqFug-NNCIw/U9a7tq7y4VI/AAAAAAAAAQk/lDq-qEhF8WU/s1600/6+msg+change2.png)

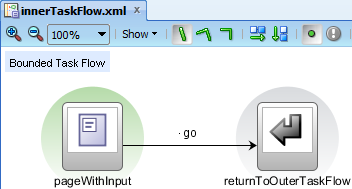
Re-run the test and the user know exactly what went wrong.

[](https://2.bp.blogspot.com/-iAxIA7bleG0/U9a80tr2njI/AAAAAAAAAQw/1xuWsPHNEOQ/s1600/7+second+run.png)

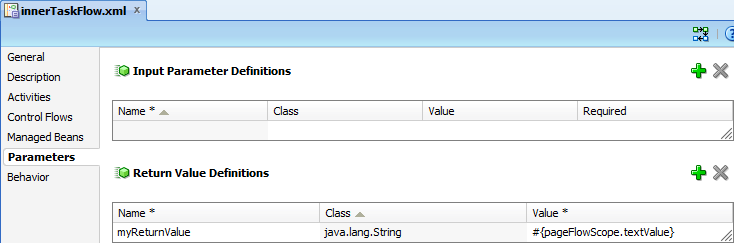
Few clicks to achieve very nice effect. While writing this post i've found hundreds of usages :)

**TaskFlow Return Value example (communication between two task flows)**

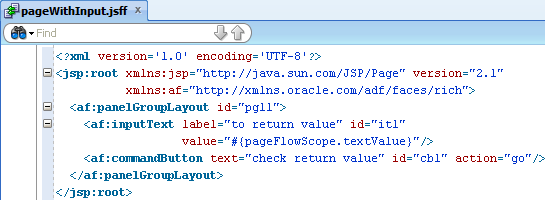
One of the interesting patterns regarding communication between task flows are return value parameters. It is possible to expose more than one return parameter. Also it is possible not to use them at all. OK, prerequisites are two task flows, one will be named innerTaskFlow and the second one outerTaskFlow. First one should contain two activities, one will be the view and the second will call an action in the parent (outer) task flow.

[](https://1.bp.blogspot.com/-DYxmH61lBdc/VAhRkNMmaPI/AAAAAAAAAWk/AL9T5ABbmHE/s1600/1%2Binner%2Btf%2Bdiagram.png)

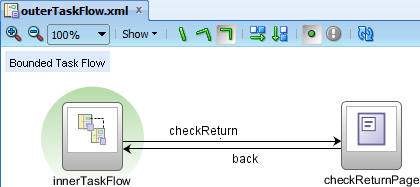
It should also define an output parameter. For example purposes I've defined it as a String. EL expression  in the Value column points to the attribute's value.

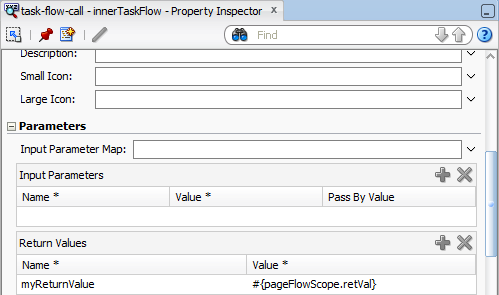
[](https://1.bp.blogspot.com/-iSfTjeTrXUQ/VAhRpGc_9UI/AAAAAAAAAWs/xS8JjaYJFF0/s1600/2%2Btf%2Bparam.png)

The page itself is as plain as it could be. I've decided to put the value directly (and implicitly) into pageFlowScope memory.

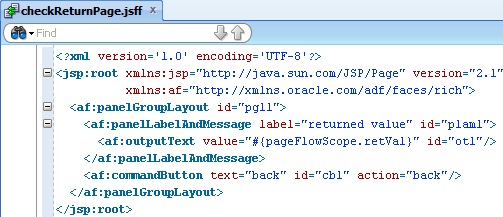
[](https://3.bp.blogspot.com/-Rgg9u7aXOSQ/VAhRtiYyHlI/AAAAAAAAAW0/q9nnxkLf2U0/s1600/3%2Bpage%2Bwith%2Bparam.png)

As for outerTaskFlow, it consumes innerTaskFlow as default activity. In its Property Inspector I've defined what needs to be done with innerTaskFlow output attribute - put it into current pageFlowScope.

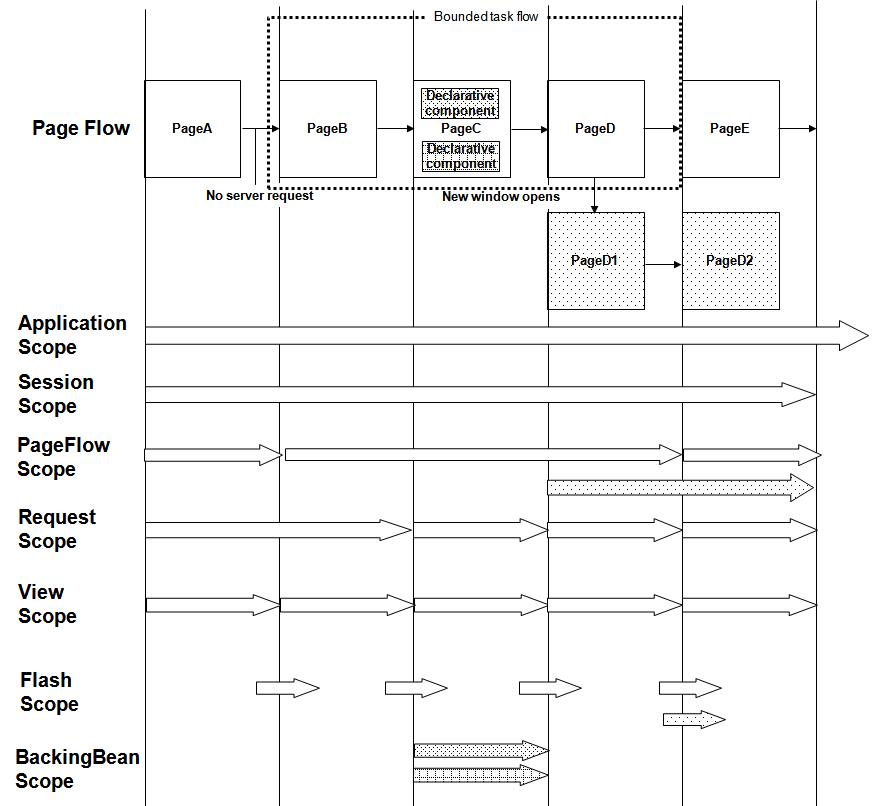
[](https://2.bp.blogspot.com/-TnVpivd3NuM/VAhR0O58fVI/AAAAAAAAAW8/Xo-Hn70F1OM/s1600/4%2Bouter%2Btf%2Bdiag.png)

[](https://1.bp.blogspot.com/-FtjrpDtPL4s/VAhR0LIP19I/AAAAAAAAAXA/95EvDEq6_3w/s1600/5%2Binner%2Binclude%2Bpi.png)

The page is just to proof that value is properly consumed there. I've defined also back flow to allow you check it few times :)

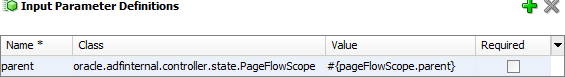
[](https://4.bp.blogspot.com/-gRUHILNi-cE/VAhR7hd7nAI/AAAAAAAAAXM/TVNbGKvzgy8/s1600/6%2Bcheck%2Breturn%2Bpage.png)

Please create also test JSPX page to run and verify the example (drag and drop outerTaskFlow on it).   
  
The question you may have is "why do I need to assign the pageFlowScope.varName1 to pageFlowScope.varName2 instead of using pageFlowScope.varName1 in both task flows?". To answer that please review (once again, I suppose :)) ADF memory scopes.

[](https://3.bp.blogspot.com/-FadoVjaSk9E/VAhSCDNL7TI/AAAAAAAAAXU/0voPobEh4uE/s1600/adf_scopes3.png)

(source: http://docs.oracle.com/cd/E24382\_01/web.1112/e16182/img/adf\_scopes3.png)

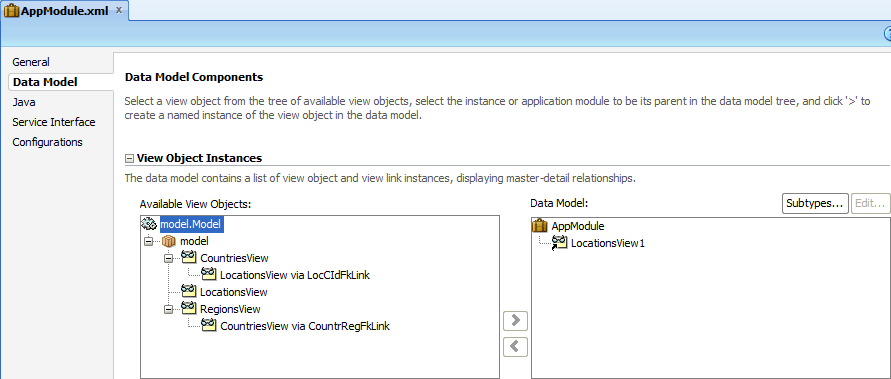
The pageFlowScope from innerTaskFlow is something different than pageFlowScope in outerTaskFlow. There is another way to interact between two task flows - bean injection. More or less you could pass outerTaskFlow pageFlowScope as an input parameter to innerTaskFlow:

[](https://3.bp.blogspot.com/-15l8DCGrrbI/VAhSJsNH0GI/AAAAAAAAAXg/But16CFMUWg/s1600/7%2Bpf%2Binjection.png)

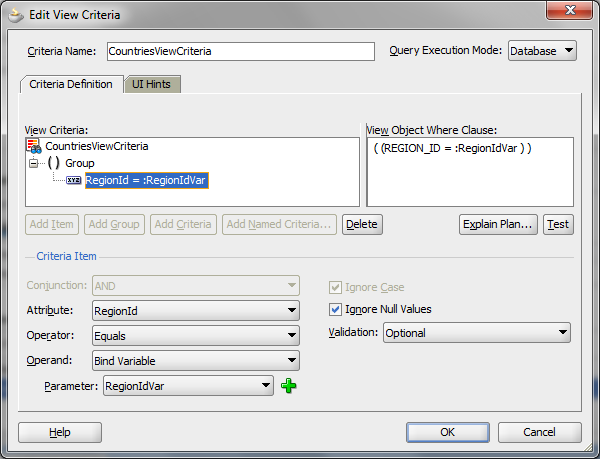
And after that in the innerTaskFlow view activity call #{pageFlowScope.parent.retVal}. But about bean injections I'll talk about in the future :)

**list of values hierarchy in af:query**

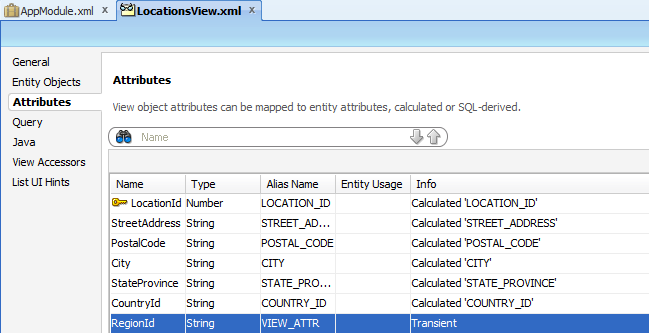
The requirement is to provide ability to search locations by counties from HR schema using af:query component. Query panel should contain list of values for countries attribute which is filtered by regions list of values field. In other words - af:query contain regions select one choice UI component which influences on countries component.   
  
As prerequisits please create three readonly view objects based od REGIONS, COUNTRIES and LOCATIONS tables. In the application module's data model put only the LocationsView.

[](https://4.bp.blogspot.com/-d9o3_MmGjY4/VBmJAtwZ6XI/AAAAAAAAAXs/kkpA0hITR74/s1600/1datamodel.png)

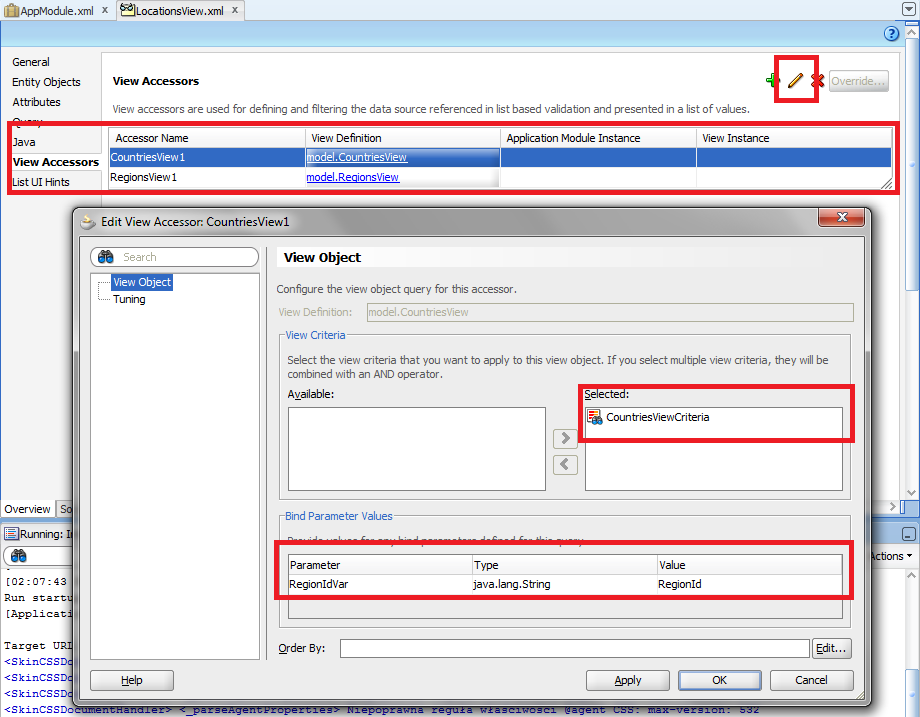
In CountriesView please create view criteria on RegionId attribute. Ignore null values not needed here. Bind variable is created using the wizard, so don't change it.

[](https://3.bp.blogspot.com/-AOvkzsqQXzU/VBmJL_TLCZI/AAAAAAAAAX0/MumL3q9S4W0/s1600/2%2Bview%2Bcriterias.png)

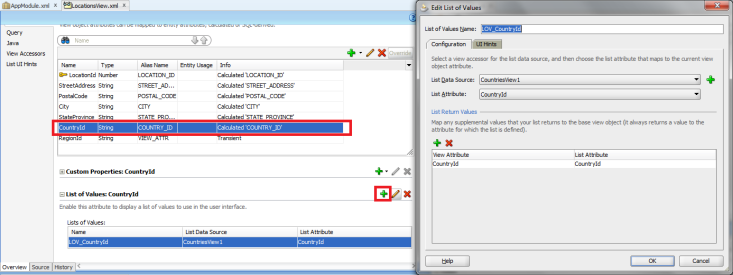
Please open LocationsView and add there new transient attribute called (in this example) RegionId.

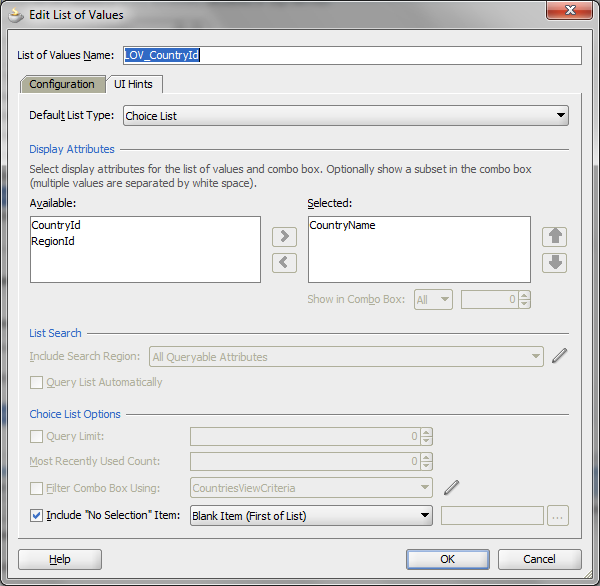
[](https://4.bp.blogspot.com/-_ovuVsVh6U0/VBmJSkist8I/AAAAAAAAAX8/pD8r6fJMsAs/s1600/3%2Btransient%2Battr.png)

Now go to View Accessors section. Please add there CountriesView and RegionsView as accessors. Additionaly please select the CountriesView1 (default name) and click on the pencil icon. Apply view criterias there and, it is important, fulfill RegionIdVar value with RegionId. It doesn't mean it'll put it as a string but it'll evaluate Groove there - it'll get RegionId transient attribute.

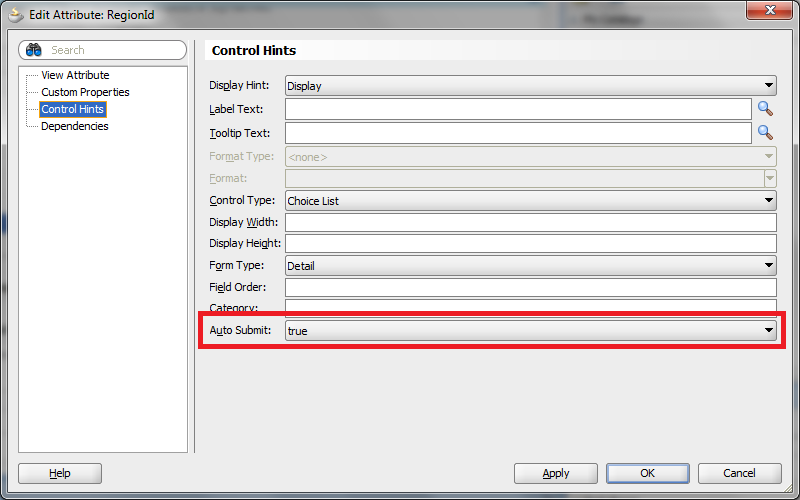
[](https://1.bp.blogspot.com/-tJhD82X0rj4/VBmJZ_zqV3I/AAAAAAAAAYE/e66oq6Mumhg/s1600/4%2Baccessors.png)

For CountryId and RegionId in Attributes tab please create list of values. Screenshots for begginers:

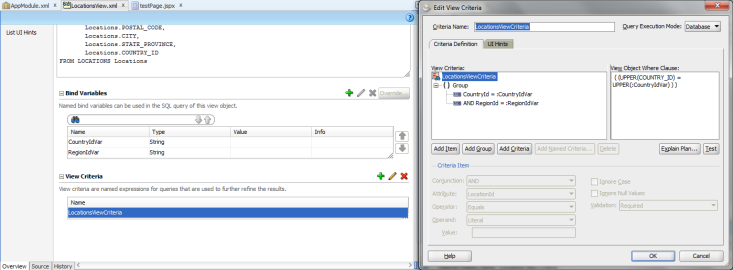
[](https://1.bp.blogspot.com/-rMZpfftyfS4/VBmJiO6mQyI/AAAAAAAAAYQ/j132_xympeo/s1600/5%2Blov%2Bdefinition.png)

[](https://3.bp.blogspot.com/-aEueUbvUOcA/VBmJiAT-c0I/AAAAAAAAAYM/KtryMeam6yE/s1600/6%2Blov%2Bdef%2B2.png)

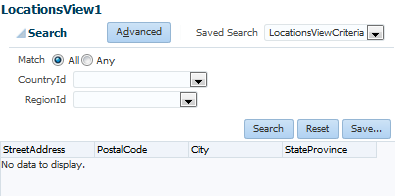
Do similar to RegionId attribute and then click pencil icon to edit it. Select Control Hints from the menu and set Auto Submit to true.

[](https://3.bp.blogspot.com/-000kYpkLwns/VBmJomMUtOI/AAAAAAAAAYc/Wz1_uZ7rPrI/s1600/7%2Bctrl%2Bhints.png)

You can use implicit view criteria to develope proof of concept but it'll add view criterias and then drag and drop them on the page as query panel with table.

[](https://3.bp.blogspot.com/-BVtEENu6SpE/VBmJv_wUjhI/AAAAAAAAAYk/5w94prBeTgQ/s1600/8%2Blocations%2Bvc.png)

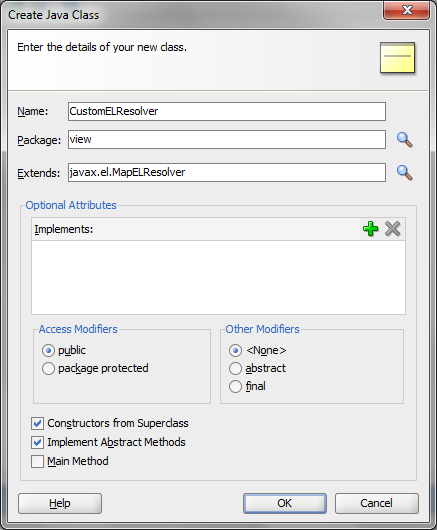
Running example:

[](https://1.bp.blogspot.com/-adXWj-Qd9aw/VBmJ1boK68I/AAAAAAAAAYs/JVH9nLnMPVc/s1600/9%2Bpoc.png)

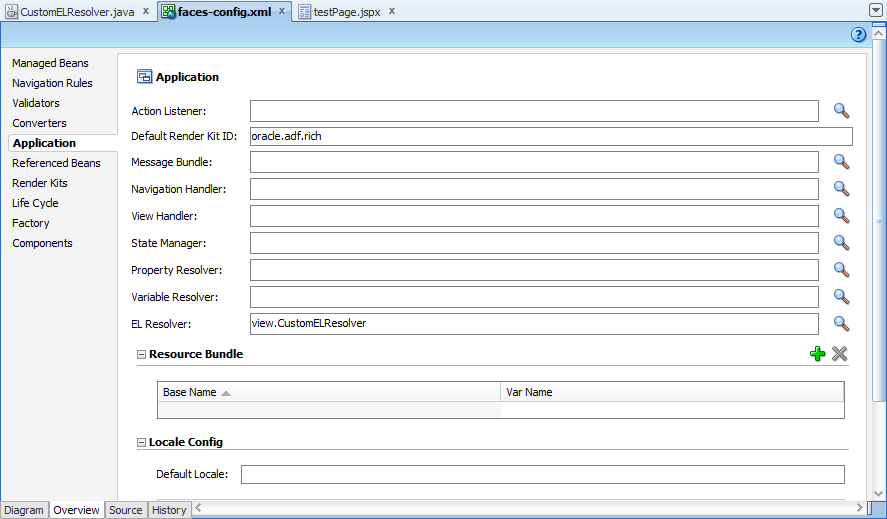
What else could be done - as you can see RegionId is not used in the query, it's transient. To make it useful, please read about [SQL tricks in view criterias](http://kamilkrasowski.blogspot.com/2014/08/tricks-with-view-criterias.html) post and try to adapt it here :)

**debugging EL expressions in your application**

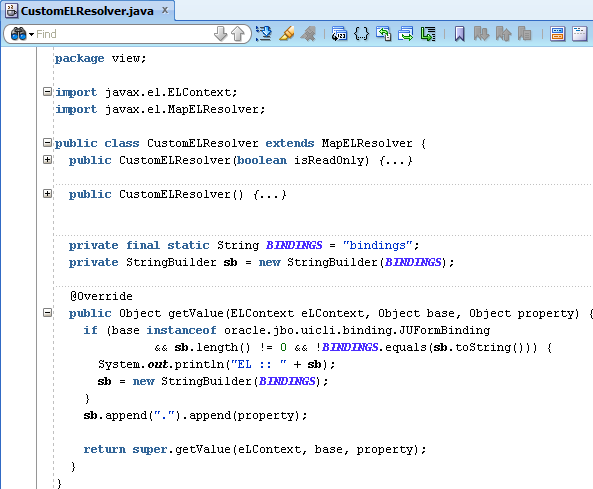
Debugging java code is relatively simple. But sometimes the stack trace which you can get is unreadable. You see a lot of lines from the core of the framework and... nothing more. No additional hints where to search for the cause. But stack trace contains a lot of UI-related classes. The reason may be in the EL expression.  
  
One approach is to comment all UI components on the page and check which expression generates an error.   
  
Second approach is a try to trace processed EL expressions on the page. Pages are usually very complex and usually contain many EL...  
  
It is possible to override the default EL resolver in your application. First of all, please create the class that extends abstract ELResolver class or one of its subclasses. In this example I've used MapELResolver (reasons why I've used it you'll find in the documentation).

[](https://4.bp.blogspot.com/-kLmahGXDfoE/VFC3HO5CcBI/AAAAAAAAAaE/Z1MLCt4qSh4/s1600/1%2BcustomElresolver.png)

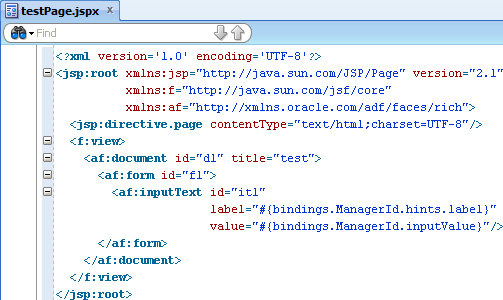
Then point it in faces-config.xml, as shown below:

[](https://3.bp.blogspot.com/-3CRfn-Yj2ds/VFC3MxqnSSI/AAAAAAAAAaM/Vo8YPKApS2Q/s1600/2%2Bpoint%2Belresolver.png)

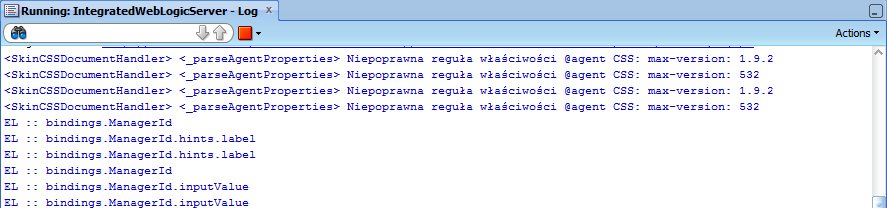
I've decided to override getValue method becasue usually an error will be thrown there.

[](https://3.bp.blogspot.com/-Cm1EoYuvL1E/VFC3Y1XtWzI/AAAAAAAAAaU/8OssZzMCurA/s1600/3%2Bel%2Bresolver%2Bimpl.png)

Please note, that this is only a simple implementation that I've used to solve my problem. To fulfill the example, the page that will be examinated.

[](https://1.bp.blogspot.com/-eEdtszA8y38/VFC3jzr-PAI/AAAAAAAAAac/XOMcaHsjZ3g/s1600/4%2Bpage%2Bsource.png)

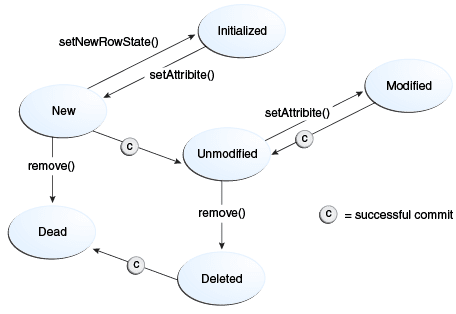
It bases on DEPARTMENTS table in HR schema. Finally, logs from the application after opening the page:

[](https://2.bp.blogspot.com/-rNOKhNMzXEc/VFC3ps9dwGI/AAAAAAAAAak/y5o0cd2MgIs/s1600/5%2Blogs.png)

Hope it'll simplify your debugging processes in the future :)

**differences between getPostState() and getEntityState()**

If you don't pay a lot attention to the subject of this post it'll sound the same. First of all what is the role of an entity. It is something like a mirror image of database table which takes a part during object-relational mapping process. It could contain transient attributes which aren't persisted. It can be based on other database objects than a table. But more or less we can understand an entity as database table definition in ADF.   
  
ORM role implies on it states. I think that everyone understand that the state machine implemented there will be quite complex and the transitions will depends on operations performed on the entity. It is illustrated on the graph below:

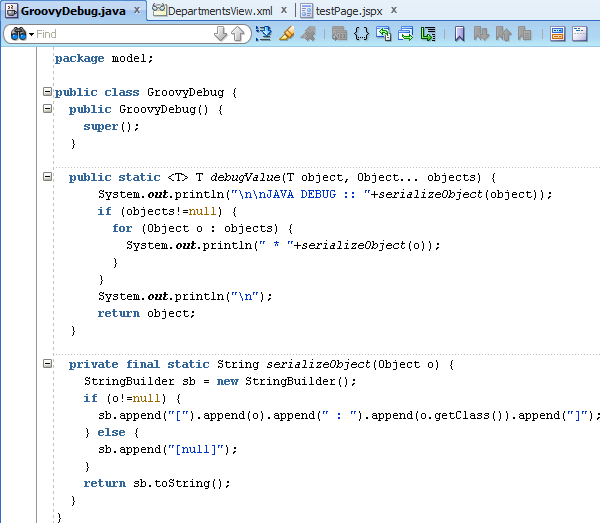
[](https://3.bp.blogspot.com/-lVTOB5hOibQ/VEDEh48Pi0I/AAAAAAAAAZ0/YkCC4nbJG7s/s1600/adffd_jd_008.png)

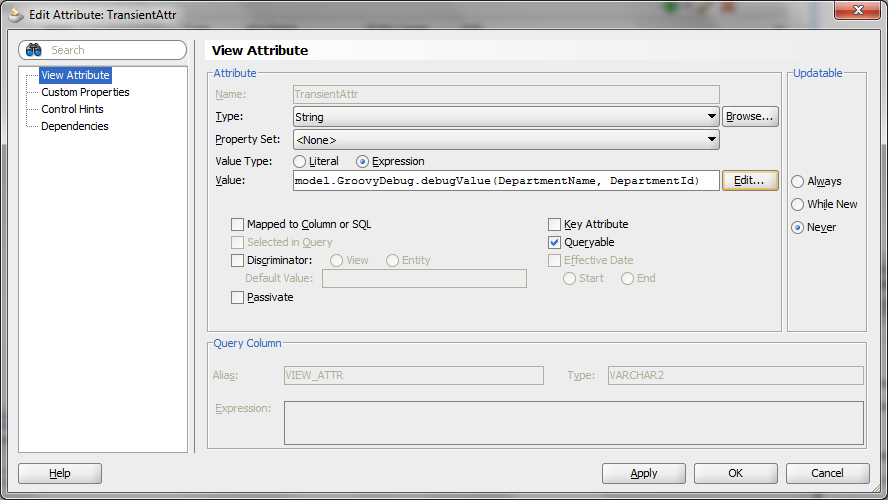
(source: http://docs.oracle.com/middleware/1212/adf/ADFFD/bcvalidation.htm)

For both of the cases the graph look similar but the meaning of the states is a bit different. Entity state represents corellation between object attributes value and the database. Post state is related to database transaction itself.  
  
Lets deeper look to entity state first. There is no default initial state for the entity. There are two states that can be assumed as defaults depends on origin. STATUS\_NEW, with byte value equal 0, while current entity is created in the middleware and will be commited within the database transaction. STATUS\_UNMODIFIED, byte value 1, if the current changes entity has been commited or it has been synchronized with database (read from the database).  
  
After at least one attribute is set, entity goes to STATUS\_MODIFIED (2). STATUS\_DELETED with 3 byte value is when a row will be deleted. The last option is STATUS\_DEAD (4). Entity has been created and after that removed within the same db transaction or it has been deleted and DELETE DML operation has been performed on the database.  
  
Post state holds information for the transaction manager what have already been done with the data within database transaction. Post state is new when a row is created in the middleware and waits to be commited. If you delete such row it'll go to STATUS\_DEAD. STATUS\_MODIFIED and STATUS\_DELETED are also markers for related DML operations. Interesting is the STATUS\_INITIALIZED with -1 value. It excludes the row from the database transaction, so it marks it as temporary one.  
  
These two states are tightly coupled but usually you won't need to worry about it. Framework itself will carry on the transitions. The problem that i've struggled with was while doDML entity method thrown an exception. Framework thought that the entity has been commited (getPostState()==STATUS\_UNMODIFIED) but getEntityState() was still STATUS\_NEW.

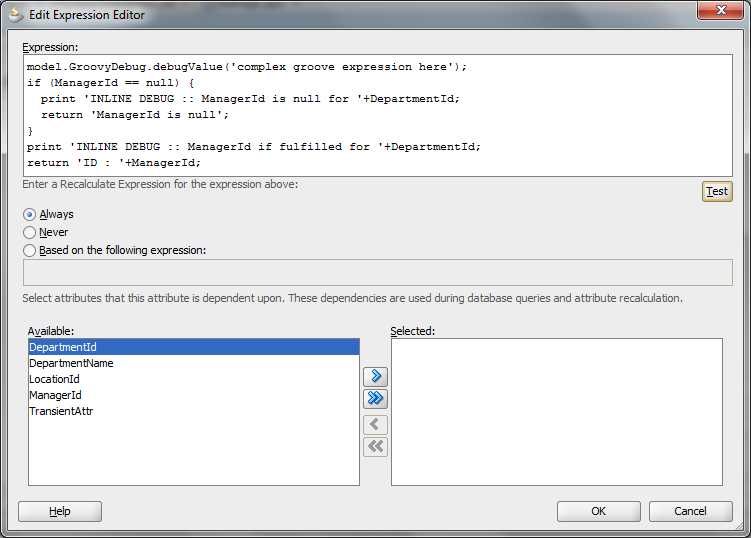
**Groovy expressions in BC4J - how to debug them?**

Today I'd like to share how to debug Groovy scripts in BC4J. At the beginning you must be aware that it is not possible to debug it step by step. But you can help yourself and print comments to the console in the runtime.  
  
Two approaches are possible. First bases on [Debugging groovy in ADF](http://kingsfleet.blogspot.co.uk/2009/11/debugging-groovy-in-adf.html) post. In short few words, in Model project you need to create a Java class and use it in Groovy:

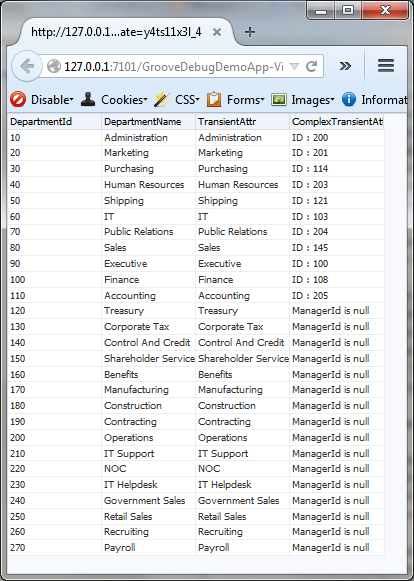
[](https://2.bp.blogspot.com/-vVJW6ol3Mqw/VFDHBe4OkyI/AAAAAAAAAa0/edpsYJ7nq3k/s1600/1%2Bclass.png)

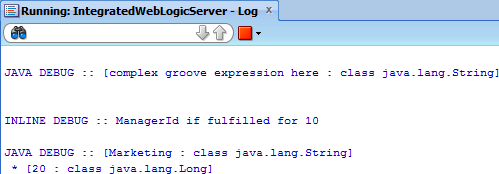
[](https://2.bp.blogspot.com/-2vM9wbOekHw/VFDHBU-sdVI/AAAAAAAAAa4/xIQ97cJdLdg/s1600/2%2Btransient%2Battr.png)

But what if the expression is much more complicated. Ok, you can call it inside the expression. There is also another way to print a line in the console in Groove - use **print** keyword.

[](https://3.bp.blogspot.com/-2SJ4cKby5Mw/VFDHJFu_7lI/AAAAAAAAAbE/dUniAgKJgzc/s1600/3%2Bcomplex%2Bgroove.png)

And, as usually, rendered page and logs output at the end of the post.

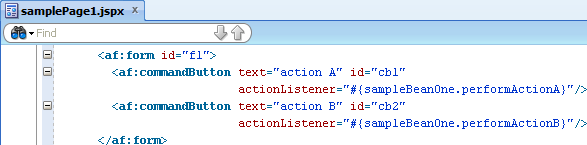
[](https://2.bp.blogspot.com/-Sw5aUXqXUIQ/VFDHOd2qYII/AAAAAAAAAbQ/VIg256o3-HU/s1600/4%2Bpage%2Brunned.png)

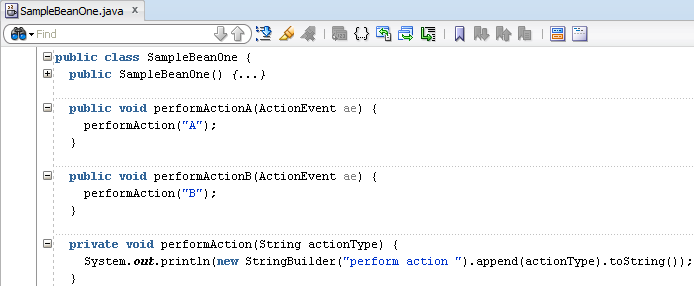
[](https://1.bp.blogspot.com/-gi-TdHiO68k/VFDHOUlAIfI/AAAAAAAAAbM/b6A4gbdS9bw/s1600/5%2Blogs.png)

According to [ADF 12.1.3 features](http://www.oracle.com/technetwork/developer-tools/jdev/documentation/1213nf-2222743.html), in 12c it'll be possible to debug it in JDeveloper :)

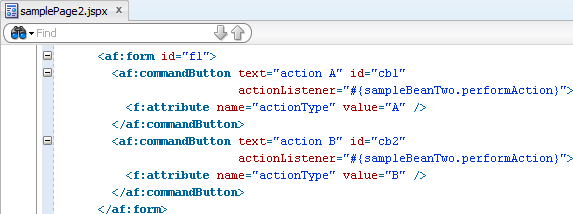
**simplify your code by using f:attribute tags**

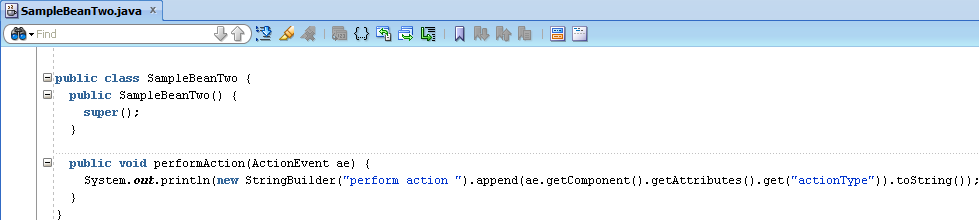
There are keywords which you're keep telling to your developers like write DRY code - Don't Repeat Yourself. It is very important to write easy, clear and self-describing code. The less code you develope the easier you'll maintain and enhance it. Please check the "base" solution:

[](https://2.bp.blogspot.com/-5RI4FQ2EitQ/VHRR0McWDhI/AAAAAAAAAbk/TqmUIH3DMD0/s1600/1.png)

[](https://2.bp.blogspot.com/-fEU0TVFMSN8/VHRR8m9qk1I/AAAAAAAAAbs/q6uMezMh9xU/s1600/2.png)

One page, one bean, two very similar actions to perform. I hope that body of the method is not copy-pasted :)  
  
And another more declarative approach that simplifies the solution - f:attribute tag inside af:commandButton. In your page source you need to add the attributes as shown below and in the action handle method consume it/them. The effect is the same.

[](https://4.bp.blogspot.com/-j8ncq1lFnFA/VHRSIZoDP7I/AAAAAAAAAb4/soDR0gNdSr0/s1600/3.png)

[](https://4.bp.blogspot.com/-7jQuVBrHmRA/VHRSH5xgrCI/AAAAAAAAAb0/hXrCBFG_hgQ/s1600/4.png)

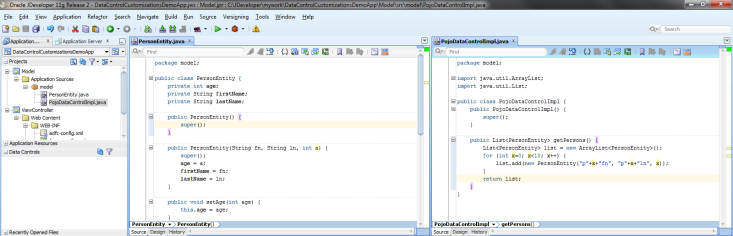
 Which is more important, you can use an EL expression there.

**create ViewObject in runtime**

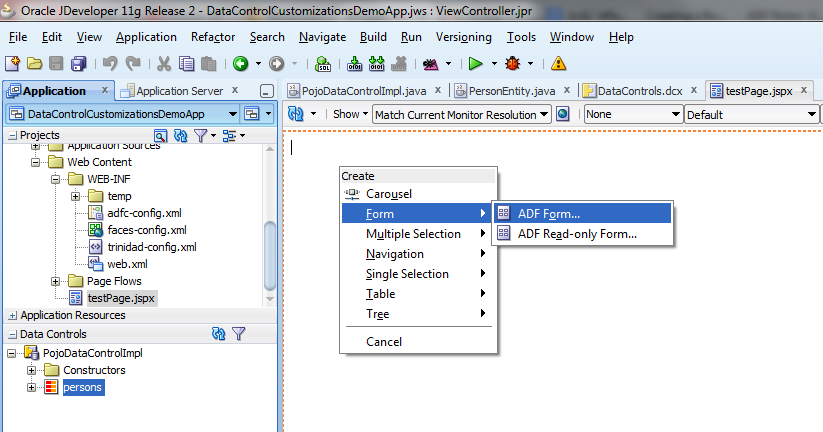
Usually, during ADF development, you create view objects at the design time. Using java call it is possible to create them also in runtime. The code will describe itself, so don't waste time and:  
  
ViewObject vo = getDBTransaction().getRootApplicationModule().createViewObjectFromQueryStmt("tempDynamicVo", "select 1 from dual");  
System.out.println("fitst selected attribute from the first row is: "+ vo.first().getAttribute(0));  
  
And that's it. The VO is there and can be accessed in different part of the code as those designed in common way.  
  
ViewObject vo = findViewObject("tempDynamicVo");  
System.out.println("fitst selected attribute from the first row is: "+ vo.first().getAttribute(0));  
  
Because it's dynamic, it doesn't obscure the Data Control and other developer won't use it in their part of the design.   
  
The view object name must be unique. Depends on your use case it may be consider as a best practice to clean it after data consumption. Just call:  
  
vo.remove();  
  
The use cases? There are, the simpliest one is an execution of stored function in the database:  
  
select my\_stored\_funct(:myBindVar) from dual  
  
Runtime VO approach sounds more BC4J than plain PreparedStatement calls. At least you've got an alternative way to implement it :)

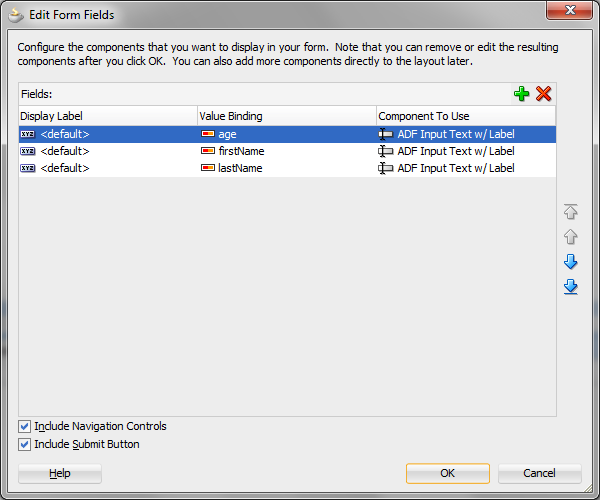
**useful tricks for ADF and services integration**

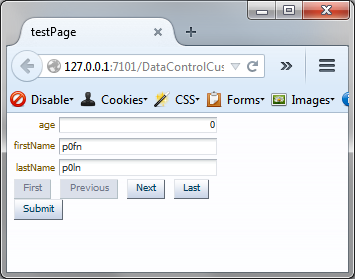
Nowadays it's though to imagine an application that doesn't consume web services. In my [previous post](http://kamilkrasowski.blogspot.com/2015/02/data-control-validation-level.html) I've described how to add declarative validation on Data Control level. WebService Data Control have been the best example in my opinion. Now I would like to make a step forward about how to combine ADF and web services.  
  
I believe that you have already watched ADF Architecture channel about [services integration](https://www.youtube.com/watch?v=lSdBbHQ0z7E) in ADF. Accodring to the architecture that Frank has described, you'll have web service proxy and kind of java facade. It helps to avoid UI changes caused by WSDL definition changes and gives a mean to cache the values. I assume that all of this is done. Let us explore POJO DC a little bit more.  
  
The start point of this post is creation of transfer object and data access object: Person entity and PojoDataControlImpl class.

[](https://3.bp.blogspot.com/-2Rwwoyoo5QA/VS-yNYLDgeI/AAAAAAAAAhk/ib2mmAW04zI/s1600/1%2Bmodel%2Bimpl.png)

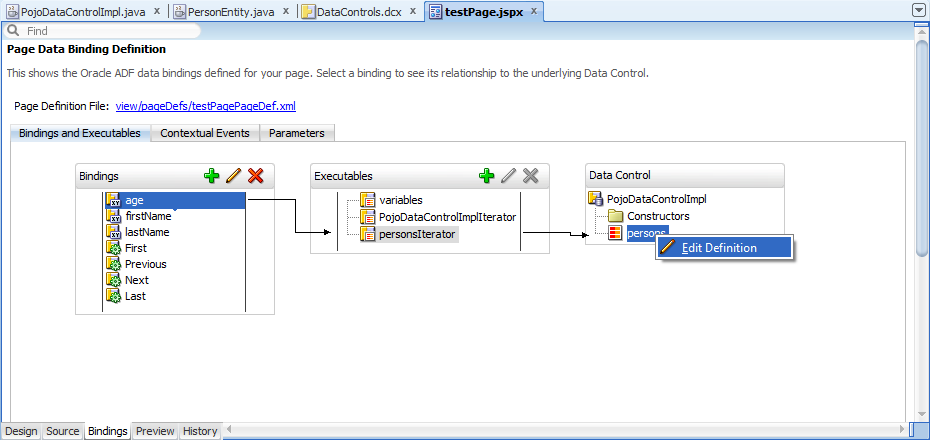
The second file is POJO DataControl implementation file. Having all this, please create the Data Control and a test page. Drag and drop persons collection as ADF Form and run it.

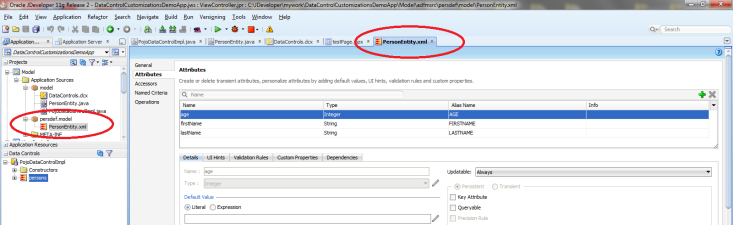
[](https://1.bp.blogspot.com/-MMep62HdJg8/VS-yWk5e8pI/AAAAAAAAAhw/ZOyL4V3V_J8/s1600/2%2Bdrag%2Band%2Bdrop%2Bform.png)

[](https://4.bp.blogspot.com/-zHjRs2VFU-0/VS-yWccQ1eI/AAAAAAAAAhs/A4L6zXYvtX4/s1600/3%2Bform%2Bcreation.png)

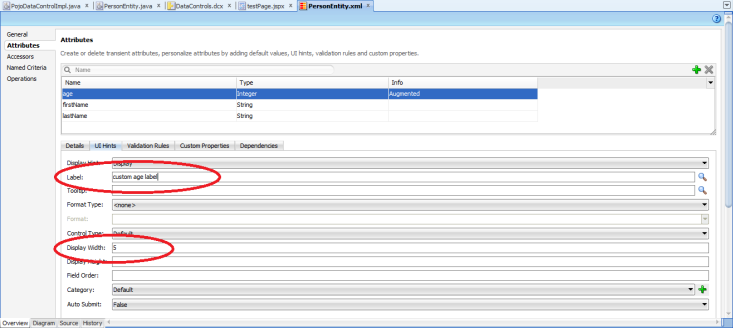
[](https://2.bp.blogspot.com/-4UpFbJUmxow/VS-yWaZX5zI/AAAAAAAAAh0/TTtsjgAhfEI/s1600/4%2Bfirst%2Brun.png)

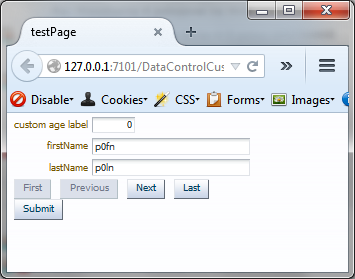
Let us focus on the first field which is age. Go to Bindings section and explore it. Right click on the collection in Data Control section. Click on Edit Definition.

[](https://3.bp.blogspot.com/-a7DOzIF1xCo/VS-ygixIgkI/AAAAAAAAAiE/4jTvW_lw5dE/s1600/5%2Bedit%2Bdef.png)

[](https://2.bp.blogspot.com/-564CC1D5Vv0/VS-ygnR64mI/AAAAAAAAAiI/V4iW3q3myfg/s1600/6%2Bediting.png)

What has happend? Framework creates XML file that corresponds with PersonEntity java class. Do you recognize the central part of JDeveloper? It is very similar to BC4J configuration wizard. And it allows to change it and use it in the same manner as BC4J entities. Please do some changes and run the page again.

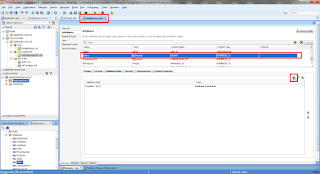
[](https://3.bp.blogspot.com/-tBvFUsOBPRM/VS-ynCwm1gI/AAAAAAAAAiY/ZRJ_tdg3jyY/s1600/7%2Bmake%2Bchange.png)

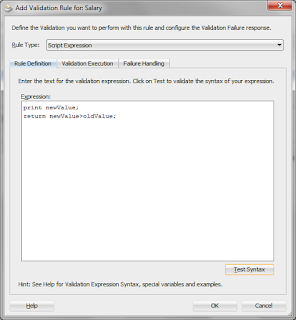
[](https://2.bp.blogspot.com/-JAxLsNJ0OsE/VS-ynN4yjaI/AAAAAAAAAiU/EXg31pRJyWI/s1600/8%2Bre-run.png)

The changes are applied to view layer by the framework. Nothing more to add. Need to iInternationalize your application? You've got model bundle file. Need to create new page? Just drag and drop attribute or collection and all of the validation, labels, read-only settings are applied.  
  
Hope it is helpful :)

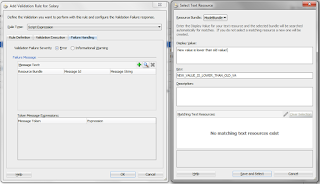
**state machines validation using Groove expressions**

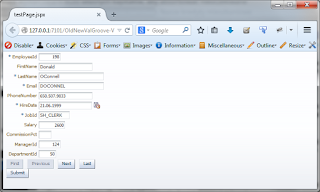
Very often there is a business requirement that defines some kind of a state machine. Each business entity has its own lifecycle, it can be created, published, modified, archived and so on. Transitions between states are usually one way. If a bank transfer is complete it cannot be rolled back to the prepared state. In other words, it means that validation depends on new and old value of the same attribute.   
  
Recently I've worked on such requirement and the validation have been implemented as valueChangeListener on af:selectOneChoice. There is simplier solution and development must be done in model layer. Reasonable because valueChangeListener is assigned to one instance of the attribute usage in the project. And such attribute may be used in various pages. And there is no guarantee that developers there know about such validation.   
  
Because new value depends on the previous, it fits to other patterns as well. Employees in a company can have their salary attribute to be updated. But who wants to have the new salary lower than current? :) Nobody I think.  
  
Please open entity Employees, select Salary attribute and specify new script validation rule:

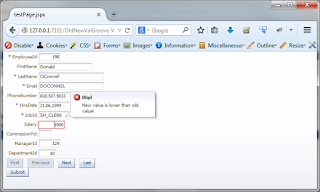
[](https://1.bp.blogspot.com/--3nyHdffZAA/VYFlCXpMZEI/AAAAAAAAAkA/hV7qcY4cYf0/s1600/01%2Bdefine%2Bnew%2Bvalidator.png)

[](https://1.bp.blogspot.com/-Q90WSDd8ZTQ/VYFlCcY_fWI/AAAAAAAAAj8/uznkOfQX3LQ/s1600/02%2Bthe%2Bdefiniton.png)

Print command is just to log the new value in the JDev console. Further, specify the error message and run the example page:

[](https://2.bp.blogspot.com/-5YhKm7iikV0/VYFlNQbVuPI/AAAAAAAAAkk/26Z61qTXRn8/s1600/03%2Berror%2Bmsg.png)

[](https://4.bp.blogspot.com/-tmVi2_KEDyc/VYFlNcARoWI/AAAAAAAAAkQ/BulHi_xi_Bw/s1600/04%2Bpreview.png)

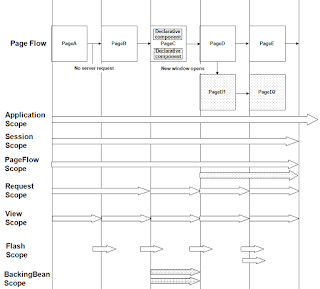
[](https://3.bp.blogspot.com/-IN4JJylJFS4/VYFlNXd2UMI/AAAAAAAAAkM/bd9Q-Wa1dSk/s1600/05%2Bvalidation%2Btest.png)

[https://4.bp.blogspot.com/-4i4SiAzwvXg/VYFlNnkO09I/AAAAAAAAAkU/NG9gbYqK0oY/s320/06%2Blog.png](https://4.bp.blogspot.com/-4i4SiAzwvXg/VYFlNnkO09I/AAAAAAAAAkU/NG9gbYqK0oY/s1600/06%2Blog.png)

Which is the most important here - Groove allows to refer to new and old values by using **newValue**and **oldValue** keywords.

**what is page definition scope? how many instances of the data controls will be created at runtime?**

Asking question about pageFlowScope, sessionScope and others leads to one of the most important (in my opinion) graphs:

[](https://4.bp.blogspot.com/-iL9OrdrxE9E/VadvDpI058I/AAAAAAAAAk0/8m8oXM4rRJo/s1600/01%2Bscopes.png)

source:http://docs.oracle.com/cd/E24382\_01/web.1112/e16181/img/lc\_scope.png

What is worth to mention, applicationScope lives with an application within the JVM container. If you have clustered environment, you'll have at most one instance per node (lazy init principle).  
  
Nevertheless, what is page definition scope or data control scope? How many instances of data controlls will be created in runtime? Quote from the [docs](http://otndnld.oracle.co.jp/document/products/as10g/101300/B25221_03/web.1013/b25386/web_DCP006.htm):

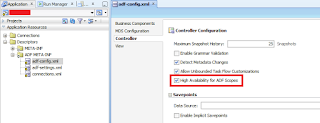
"*(...) the binding container, the binding objects it contains, and the values the binding objects reference are defined in session scope.*"

But it doesn't mean that there is equation one session == one binding container (data control instance). Please notice that ADF introduces taskflows and task flow transactions. If your application uses a lot of task flows as regions on one page and some of them are always creating new transactions (on the controller level), per each session you'll have more than one binding frame.  
  
As a summary - from the time point of view, each data control instance (each binding container) will live as long as user session exist. From the instances count point of view, per each session you'll have 0..n instances of the data controls (depends on total number of used task flows on your pages, task flow transaction setting and the page definition contents).

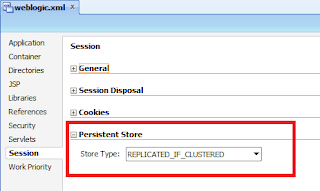
**High Available (HA) checklist for ADF applications**

Making ADF application HA enabled, there are two groups of actions that should be performed. One is the configuration and the second is the development part of work. Below you'll find my checklist:

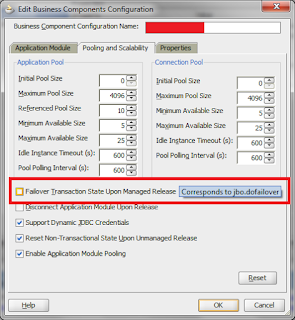
1. in **adf-config.xml** descriptor put code <adf-scope-ha-support>true</adf-scope-ha-support>

[](https://2.bp.blogspot.com/-wNREut9W2QI/VbjIvdgIigI/AAAAAAAAAlE/SPamJ9WrSEY/s1600/01%2Badf-config.png)

1. in **weblogic.xml** descriptor set session persistent store type as REPLICATED\_IF\_CLUSTERED

[](https://1.bp.blogspot.com/-GeiUqdkRPqQ/VbjI4Gk594I/AAAAAAAAAlM/TGEtYb1CqyA/s1600/03%2Bweblogic.png)

1. each application module must have jbo.dofailover attribute set to true

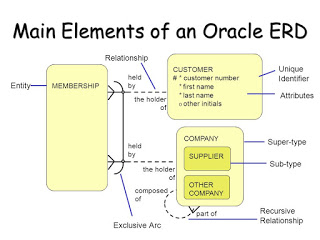
[](https://4.bp.blogspot.com/-2RErbTZgaZ4/VbjI96RbBuI/AAAAAAAAAlU/Fup9ZUXKZSw/s1600/04%2Bbc4j.png)

1. if you're using POJO DataControls, you need to implement three methods there: createSnapshot(), restoreSnapshot(Serializable handle) and removeSnapshot(Serializable handle); more about that here: [Creating and Configuring Bean Data Controls](https://docs.oracle.com/middleware/1212/adf/ADFDC/bean.htm#ADFDC751)
2. each bean defined in the memory scope higher than request level must implement java.io.Serializable interface
   * avoid using java transient keyword regarding to the bean attributes
   * each attribute that extends UIComponent aren't serializable! if UI component must be binded with the code, use org.apache.myfaces.trinidad.util.ComponentReference wrapper there
   * make the framework aware of changes that needs to be replicated against the cluster, mark the scope as dirty, ie. using ControllerContext.getInstance().markScopeDirty(AdfFacesContext.getCurrentInstance().getViewScope());
3. each page flow and session attributes must implement java.io.Serializable interface

 As short as possible. Hope it'll be helpful :)

**List of Values switcher example in ADF 12c**

I would like to start with a little bit of history. A long time ago, while people were still modelling ERD diagrams, there was an interesting relation, that I'm used to call an "exclusive or" but in the various sources it can be called also "exclusive arc" or "exclusive bar". I'll use an image found using Google to picture the situation:

[](https://4.bp.blogspot.com/-ZdkF7QJDlpc/VyDKkLFyeEI/AAAAAAAAAtU/QNuWaTQRcR0bZXHhfpDGiYW56N4NHeSFwCLcB/s1600/slide_6.jpg)

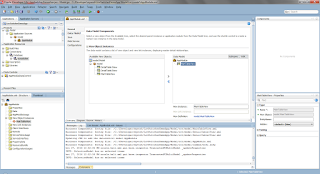
source: <http://images.slideplayer.com/23/6559166/slides/slide_6.jpg>

Such notation implies that in the logical model, the MEMBERSHIP table contains one foreign key column that can point to CUSTOMERS table or (exclusively) to COMPANY table.  
  
In the BC4J model layer there exist an ability to define list of values. Maybe you haven't noticed that one column can have assigned more than one list of values. Moreover, such an LOV model can be switched in the runtime.  
  
In this post I'll focus on the middleware implementation, the database is just to provide background for the demo. To complicate more, the TYPE column won't be threated directly as switching attribute. It'll be used by a Groove expression on the VO level.

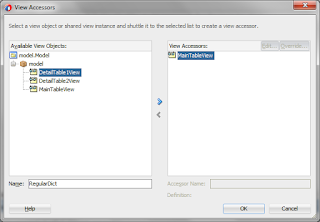
Definition for the database:

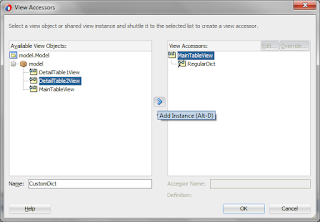
|  |  |
| --- | --- |
|  | -- schema creation CREATE USER LOVSWITCHER IDENTIFIED BY lovswitcher; GRANT  ALL PRIVILEGES TO LOVSWITCHER;  -- test structure and data CREATE TABLE LOVSWITCHER.MAIN\_TABLE ("ID" NUMBER NOT NULL ENABLE, "TYPE" VARCHAR2(100 CHAR), "ONE\_KEY\_COLUMN" NUMBER, CONSTRAINT "MAIN\_TABLE\_PK" PRIMARY KEY ("ID")); CREATE TABLE LOVSWITCHER.DETAIL\_TABLE1 ("ID" NUMBER NOT NULL ENABLE, "LABEL" VARCHAR2(100 CHAR), CONSTRAINT "DETAIL\_TABLE1\_PK" PRIMARY KEY ("ID")); CREATE TABLE LOVSWITCHER.DETAIL\_TABLE2 ("ID" NUMBER NOT NULL ENABLE, "LABEL" VARCHAR2(100 CHAR), CONSTRAINT "DETAIL\_TABLE2\_PK" PRIMARY KEY ("ID")); INSERT INTO LOVSWITCHER.DETAIL\_TABLE1 (ID, LABEL) VALUES (1, 'tab 1 label 1'); INSERT INTO LOVSWITCHER.DETAIL\_TABLE1 (ID, LABEL) VALUES (2, 'tab 1 label 2'); INSERT INTO LOVSWITCHER.DETAIL\_TABLE1 (ID, LABEL) VALUES (3, 'tab 1 label 3'); INSERT INTO LOVSWITCHER.DETAIL\_TABLE2 (ID, LABEL) VALUES (1, 'tab 2 label 1'); INSERT INTO LOVSWITCHER.DETAIL\_TABLE2 (ID, LABEL) VALUES (2, 'tab 2 label 2'); INSERT INTO LOVSWITCHER.DETAIL\_TABLE2 (ID, LABEL) VALUES (3, 'tab 2 label 3'); COMMIT; |

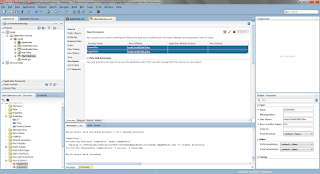
I'll create an ADF Fusion Middleware application with Model created with "ADF Business Components from Tables" wizard. There is one updateable view object based on MAIN\_TABLE and two dictionaries will be just for read only purposes.

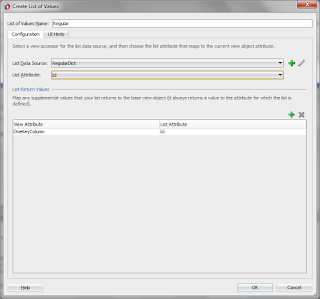
[](https://3.bp.blogspot.com/-iLKLxyhOsHk/VyDMKQXhPyI/AAAAAAAAAtg/_DfAA31OndQOB2BEB55IKhNIJGVNjYtJwCKgB/s1600/001%2Bdata%2Bmodel.png)

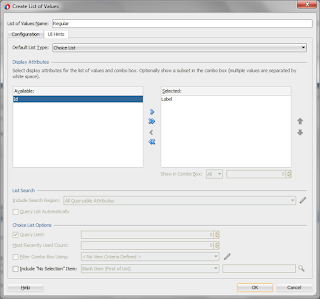
In the MainTableView view object please add read only DetailTable view accessors as following. After that please go to the Attributes tab and select OneKeyColumn. There, on the List of Values tab please assign two list of values using green plus icon.

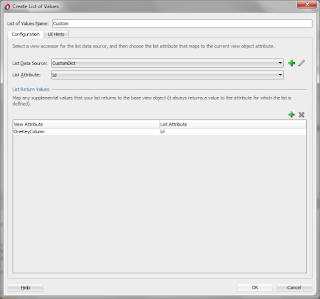
[](https://1.bp.blogspot.com/-WB6GobD-zmc/VyDNhOsMmJI/AAAAAAAAAts/uY3s4JfD6isNUjVoJzY204Sgt68bpLoMwCLcB/s1600/002%2Baccessors1.png)

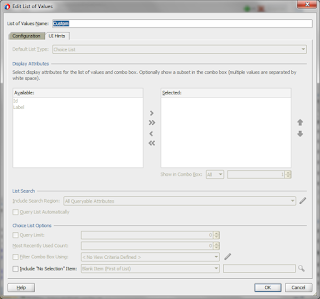
[](https://1.bp.blogspot.com/-TNzDypVkYwY/VyDNhUqMFiI/AAAAAAAAAtw/PQojAm7OwaEQDoVWndZCz8YcDbBHZvGzwCLcB/s1600/003%2Baccessors.png)

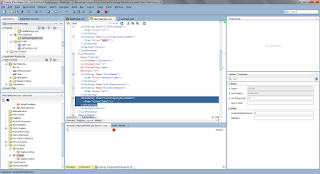
[](https://4.bp.blogspot.com/-lSDUcwAnPJA/VyDNhdMLDSI/AAAAAAAAAt0/rRpVe5WGMHghhDKfYfoVT8a9suyusbMEQCLcB/s1600/004%2Baccessors.png)

[](https://1.bp.blogspot.com/-bQnGk7-Owug/VyDNhbhWJVI/AAAAAAAAAuI/47xemc0xO84bPyhON3D-zf4tFwspTU6sACLcB/s1600/005%2Blov.png)

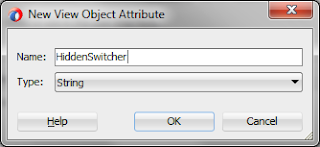
[](https://3.bp.blogspot.com/-u_lYWn88GI4/VyDNhsmAKUI/AAAAAAAAAt4/pB7iT5XnR54mBxMo8xptbhK8VEXIeuONgCLcB/s1600/006%2Blov.png)

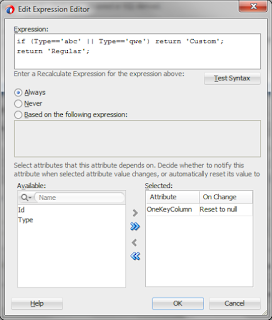
[](https://2.bp.blogspot.com/-qaSYKh2GYLc/VyDNhnSfGDI/AAAAAAAAAt8/nt_mJwlbR_AlYqZmqmmTiylxmsZhwRaYACLcB/s1600/007%2Blov.png)

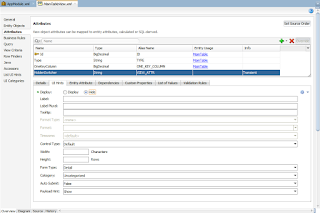
[](https://1.bp.blogspot.com/-2BomzWa5HZs/VyDNhxHZb1I/AAAAAAAAAuA/VWUz030_Mjsl3EvhBHcNdHNGypZQCoTlwCLcB/s1600/007b%2Blov.png)

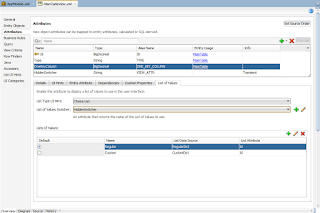
[](https://2.bp.blogspot.com/-ab6cEXth2jE/VyDNh9LZFBI/AAAAAAAAAuE/5ET6gg8aKlQ4MBgwgn5r46e4ooi9IIM6wCLcB/s1600/007c%2Blov.png)

At the moment please notice few things. First of all there, one of the wizards is disabled. That's why please ensure in the view object source that the label is selected as a display attribute for the second one.  
  
Each list of values has its own unique name. It is obvious that there must be just one default list of values.   
  
This is the moment for add technical field to control which model of the list should be used. I'll use transient attribute based on a Groove expression. It'll be pointed as a List of Values Switcher. Key idea is that it must return the name of LOV, that's why it is String based. It is calculating return value using Type attribute (which is persisted in the database). And, because it is technical field, it won't be exposed in the DataControl (Display attribute set to Hide).

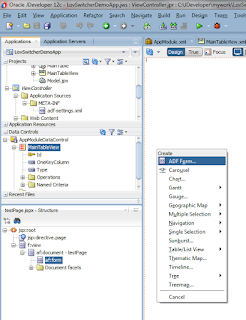
[](https://3.bp.blogspot.com/-W_R8iXU8Hh0/VyDPYcEK07I/AAAAAAAAAuU/KR7Xn5fA-7wl3jsZyky4ON_nQbq-4U5pACLcB/s1600/008%2Bswitcher.png)

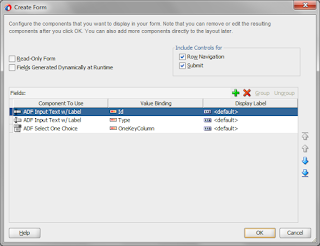
[](https://2.bp.blogspot.com/--oVgNAiKhlw/VyDPYdVGCsI/AAAAAAAAAuc/zQqLdrWA1e0gk5fJqWG2LR8iprWKN_VugCLcB/s1600/009%2Bswitcher.png)

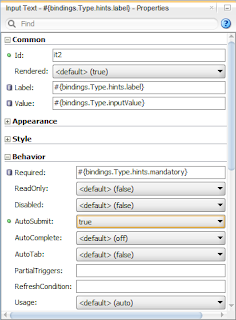
[](https://3.bp.blogspot.com/-OFKR3EY-Y1Y/VyDPYUNr0kI/AAAAAAAAAuY/-VmTv3jCKjM_wc4S3-UNinHNYSk1YdUWgCLcB/s1600/010%2Bswitcher.png)

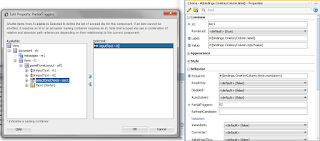
[](https://4.bp.blogspot.com/-MXEZ-i9eI5w/VyDPYk83nnI/AAAAAAAAAug/qEDHTr6Ttn04d-siiNIRFQd_e51hBndygCLcB/s1600/011%2Bswitcher.png)

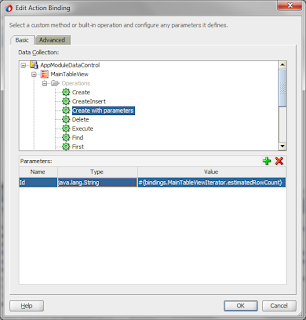
For the test page, please drag&drop MainTableView from the Data Control to the page. Please include navigation and commit buttons. The Type input field must submit its value after each change (autoSubmit set to true) and the OneKeyColumn must refresh itself after such change (partialTrigger on the Type input field). Because the SQL script doesn't contain test data, please also drag&drop CreateWithParams operation. Quick trick for setting the primary key with the row count value is used in this example.

[](https://4.bp.blogspot.com/-QSLYLGjYZmQ/VyDQUAUcIiI/AAAAAAAAAu0/vBh5N6ObQfsiyqlIEcW8WpC0JaXFd-aPQCLcB/s1600/012%2Bform%2Bdragdrop.png)

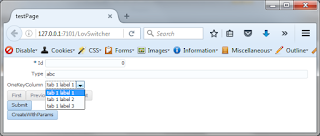
[](https://3.bp.blogspot.com/-pZBECHWCO2k/VyDQUOX2GiI/AAAAAAAAAuw/nLRpv3_C774gPU_oByn8F67EMoXilYNBgCLcB/s1600/013%2Bform.png)

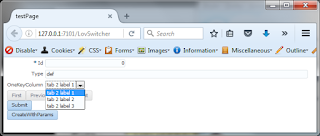
[](https://1.bp.blogspot.com/-n-vVbDuPucg/VyDQULf39OI/AAAAAAAAAus/JIdUD0JOSGI_toebsBLO4WF7oBGvCl_6wCLcB/s1600/014%2Bautosubmit.png)

[](https://1.bp.blogspot.com/-QEv_c9stcx8/VyDQUdck5eI/AAAAAAAAAu4/vnRDPnYg8x0EU-SAYQzx0NK4w_z2-mzSQCLcB/s1600/014%2Bpartialtrigger.png)

[](https://4.bp.blogspot.com/-AbRUTH6z1q8/VyDQUWSjWdI/AAAAAAAAAu8/puN7xrPGZmcxcJkjzdHIv08duDNuSMAyQCLcB/s1600/015%2Bcreaterow.png)

And please take a look on the list of values model after setting Type attribute with *abc* and not-*abc*string value.

[](https://1.bp.blogspot.com/-v4KB691skR4/VyDQnwhuSgI/AAAAAAAAAvE/TNybgkBCFao6-mPMCjEovGeLZuSK5TG4gCLcB/s1600/016%2Bdemo1.png)

[](https://4.bp.blogspot.com/-DzekrkThCbU/VyDQn7BeYvI/AAAAAAAAAvA/KG11OX5p5Cst0hxWJheeerFpzhZkVKwmgCLcB/s1600/017%2Bdemo2.png)

Posted by [Kamil Krasowski](https://plus.google.com/101202787860923401081)at [02:42](https://kamilkrasowski.blogspot.com/2016/06/list-of-values-switcher-example-in-adf.html)

**how to implement tokenized messages in ADF-Faces**

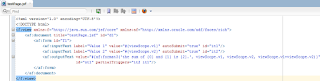
    Some time ago I've posted an article about [customizing error messages in BC4J](http://kamilkrasowski.blogspot.com/2014/08/how-to-customize-error-messages-in-bc4j.html). Now's the time to think more about the UI. From time to time there is a requirement to write a sentence or a string concatenation on the screen which won't be static. As an example: *There are XYZ rows from which ABC are shown on the screen*. Of course my application must be internationalized, I'm using the bundle file. I can split my sentence into three different entries in the bundle file but I must be sure that the syntax in each language is the same. And it mustn't be like this.

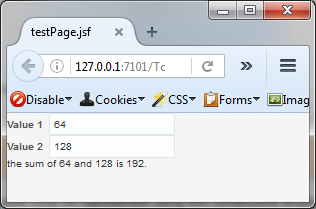
    The solution is simple and, as usually, no extra Java code. ADF-Faces allows to tokenize text message. My simple use case is to display two fields and write "a+b is aANDbSUM" at the bottom of the page. For that purpose I'll use [af:format3](http://jdevadf.oracle.com/adf-richclient-demo/docs/tagdoc/functions/af_formatNamed3.html) function. It is a part of EL expression and it can be mixed with other valid EL invocations inside.

    The code snippet (it is using static text and viewScope implicit variables just to simplify the example):

<af:outputText value="#{af:format3('the sum of {0} and {1} is {2}.', viewScope.v1, viewScope.v2, viewScope.v1+viewScope.v2)}" id="ot1" partialTriggers="it2 it1"/>

    And the screenshots of the code and of the running page:

[](https://2.bp.blogspot.com/-ktcxCWRR9Dc/V6DZnuQob5I/AAAAAAAAAzo/oniv2aHRHMoaR8AhoRy1GUavstVpjQwRwCLcB/s1600/source%2Bcode.png)

[](https://4.bp.blogspot.com/-8PvUi6F3vH0/V6DZqmTN7tI/AAAAAAAAAzs/nx-ZEzBgoH47EYfRVF8SSNXLYPNFe82wgCLcB/s1600/result.png)

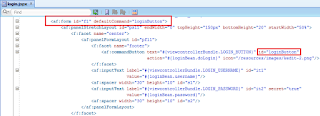
Posted by [Kamil Krasowski](https://plus.google.com/101202787860923401081)at [10:37](https://kamilkrasowski.blogspot.com/2016/08/how-to-implement-tokenized-messages-in.html)

**af:form default command use-case**

   Recently I've done some simple application. As usually, there was a logging page, which I have developed following rules from the documentation. The mechanisms were working fine, but there was one user who has pointed a change request - he wanted to facilitate the logging in process. Now he need to use a mouse to click the button, instead he wants the form to be submitted with Enter key. Useful change, to improve user experience with the app I should start with the login page.

   I've been checking for <af:commandButton> properties in order to find an attribute that would solve it. Unfortunately there was nothing suitable, and, it is not a place for specifying default action. In many cases developers may set such property to more than one components. During short and temporary insanity I've thought about JavaScript function to do the work :)

   Fortunately there is a solution. Simple and quick. Similar to rendered attribute, it is not the component that should be triggered in order to present (or hide) itself. It is the container. There is an attribute, defaultCommand, which must be set in <af:form> tag. And that's it :)

[](https://1.bp.blogspot.com/-MrajNAbQYGo/V_ZA0b5tgYI/AAAAAAAAA0E/25bA1trLSM8kHlTWS9IyjK-nvXz2eovcgCLcB/s1600/defaultCommand.png)

   And short quick note at the end - it supports EL expressions, so the action may be parameterized

**Effective use of getEntityState() and getPostState() methods in EOImpl class in Oracle ADF**

You can use the **getEntityState()**and **getPostState()**methods to access the current state of an entity row in your business logic code.  
  
The **getEntityState()** method returns the current state of an entity row with regard to the transaction  
  
**getPostState()**method returns the current state of an entity row with regard to the database after using the postChanges() method to post pending changes without committing the transaction.  
  
The sample code is as below  
  
public void postChanges(TransactionEvent transactionEvent) {  
    /\* If current entity is new or modified \*/  
    if (getPostState() == STATUS\_NEW || getPostState() == STATUS\_MODIFIED) {  
    }  
}

The possible outcomes of the getPostState() method are as follows

* STATUS\_UNMODIFIED - if this Entity Object has been queried from the database and is unchanged, or if it has been posted to the database.
* STATUS\_MODIFIED - if this Entity Object has been queried from the database and has changed.
* STATUS\_NEW - if this Entity Object is new and not yet posted to the database.
* STATUS\_INITIALIZED - if this Entity Object is new and the client code marks it to be temporary by calling Row.setNewRowState method.
* STATUS\_DELETED - if this Entity Object has been marked for deletion.
* STATUS\_DEAD - if this Entity Object is new, but has been deleted.

## How to get entity state in Managed bean ?

February 25, 2015 by [Rajesh](https://rajeshideas.wordpress.com/author/rajeshideas/)

EmployeeRowImpl row =

(EmployeeRowImpl )ADFUtils.findIterator(“Employee1Iterator”).getCurrentRow();

if (valueChangeEvent.getNewValue() != null) {

byte state = row.getEntity(0).getEntityState();

if (Entity.STATUS\_NEW == state) {

Int radomNum = 0; //do some business logic

row.setCodUsuario(“SWS” + randomNum);

row.setPassword(“SWS” + randomNum);

}

}

**How to get the entity status in VO impl ?**

String status = “UNMODIFIED”;

ViewRowImpl vil = (ViewRowImpl)r;

if (null != vil) {

byte state = vil.getEntity(0).getEntityState();

if (Entity.STATUS\_NEW == state) {

status = “NEW”;

}

if (Entity.STATUS\_MODIFIED == state) {

status = “MODIFIED”;

}

}

**Programmatically reading the entity state in Oracle ADF**

Entity row has two states one is associated to the transaction and the other is corresponding to the database(post state).

Use getEntityState() to read the Entity state corresponding to the transaction and getPostState() to read the Entity state corresponding to the DB.

**Points to be remembered**

When you create a new Entity row both the getEntityState() and getPostState() gives you the status as Entity.STATUS\_NEW.

When you post the EO to the DB(Without Committing the transaction) the getEntityState() gives you the Entity.STATUS\_NEW, whereas the getPostState() returns you the Entity.STATE\_UNMODIFIED.

When the transaction is committed the state both returns the Entity\_STATE\_UNMODIFIED.

You can observe the above points using the below code. Place the below code in the Application Module Implementation class.

Create a EO based on the Department table.

    public void findEntityState() {

        //Get defnition object

        EntityDefImpl departmentEODef = DepartmentEOImpl.getDefinitionObject();

        //Create the entiy instance in the current transaction

        DepartmentEOImpl newDept1 = (DepartmentEOImpl) departmentEODef.createInstance2(this.getDBTransaction(), null);

        newDept1.setDepartmentId(100);

        newDept1.setDepartmentName("Oracle Fusion");

        newDept1.getPostState() // NEW

        newDept1.getEntityState() // NEW

        //Post changes to DB

        getDBTransaction().postChanges();

        newDept1.getPostState() // UNMODIFIED

        newDept1.getEntityState() //NEW

        //Commit the Transaction

        getDBTransaction().commit();

        newDept1.getPostState() // UNMODIFIED

        newDept1.getEntityState() // UNMODIFIED

    }

**Restricting the display of future dates in af:inputDate component**

The input date component in Oracle ADF has some useful attributes for validating the dates like minimum acceptable date and maximum date allowed.

But to disable the future date in the input date comportment we can use the max value property by pointing it to a backing bean method as explained in the blog. <http://www.adftutorials.com/disable-future-date-on-adf-inputdate-component.html>. After implementing the code, the input date is throwing an error The date must be on or before 08-Jun-2016 when you select the current date.

Use the below solution to make it work.

1. Create a transient date attribute of type oracle.jbo.domain.Date.
2. Mark the default value as expression and paste new java.sql.Timestamp((adf.currentDate +1 ).time-1)
3. Create an attribute binding to the page that shows the input date.
4. Now use the transient attribute value in the max value of  the input date in the page like

                  <af:inputDate id="id4" label="Date"   autoSubmit="true"

**maxValue="#{bindings.LimitDate.inputValue}"**

                      disabled="true"/>

**Code for finding the Entity or Rows by using the Primary Key in ADF**

Use the below process/code to find the Entity instance based on the primary key.

1. Get the EntityDefinition object by using the EntityImpl class.
2. Use createPrimaryKey() method available in the EntityImpl to create the key.
3. Use the EntityDefImpl object findyByPrimaryKey method and pass the key you created in the previous step.
4. The findByPrimaryKey method gives the EntityImpl, which is an representation of row.

 In this example we are considering the EmployeeEO and empId as primary key.

Private EmployeeEOImpl findEmpById(String empId){

EntityDefImpl empDef= EmployeeEOImpl.getDefinitionObject();//Finding the Defintion

Key empKey= EmployeeEOImpl.createPrimaryKey(new Integer(empId));//Creating the Key

Return (EmployeeEOImpl) empDef.findByPrimaryKey(getDBTransaction(),empKey);

}

**Programmatically handling the Entity Objects in Oracle ADF**

**Entity Object is**n object oriented representation of data base objects like table. It represents a table or view. It encapsulates the properties of a Row like the validation rules, security and presenting values. It is used to perform all the DML operations.

Files related to Entity Object

1. Entity definition XML file- Defines the structure of  the entity object. It acts as a template for the creation of entity objects.
2. Entity Object class - It represents an entity instance. It wraps the business logic for a row like having default values at the time of row creation, lock() method to lock the row and doDML() method to perform DML operations.
3. Entity Collection Class- Caches the rows particular to a DB transaction.

Use the below code to create the entity rows. You can have it in the Application module implementation classes.

**Creating a new entity row**

public void createEntityObject(){

EntityDefImpl sampleEODef=sampleEOImpl.getDefinitionObject();//Here get the definition object using the EOImpl like EntityImpl

SampleEOImpl newEOInstance=sampleEODef.createInstance2(this.getDBTransaction(), null);

//The entity status is STATUS\_NEW now and qualifies to participate in a transaction.

//Use the newEOInstance object to set the attribute values.

}

**Finding an entity row and updating it**

public void updateEntity(String primaryKey){

//Key creation by passing the primary key

Key key= SampleEOImpl.createPrimaryKey(primaryKey);

//get the definition object

//The row status here is STATUS\_UNMODIFIED now

EntityEOImpl rowTobeUpdated =

(DepartmentEOImpl)SampleEOImpl.

getDefinitionObject().

findByPrimaryKey(getDBTransaction(), key);

rowTobeUpdated .setAttributes()

//The row status here is STATUS\_MODIFIED and qualifies for row the transaction.

}

**Removing an entity row**

The below method takes a key attribute value in a row as input and performs find and delete operations.

public void removeEntity(Integer key){

            //Key creation by passing the key

            Key key= SampleEOImpl.createPrimaryKey(key);

            EntityEOImpl rowTobeDelted =

             (SampleEOImpl)SampleEOImpl.

             getDefinitionObject().

            findByPrimaryKey(getDBTransaction(), key);

           //Row status is STATUS\_UNMODIFIED

           rowTobeDelted.remove();

          //Row status is STATUS\_DELETED

 }

**Committing a transaction**

     public void commitTransaction(){

       getDBTransaction().commit();

      }

**Changing the VO query at run time programmatically**

Use the below method to modify the query of a view object run time programmatically.

    public void modfiy(ViewObjectImpl voImpl, String newQueryString){

        voImpl.setFullSqlMode(voImpl.FULLSQL\_MODE\_AUGMENTATION);

        voImpl.closeRowSet();

        voImpl.setWhereClause(null);

        voImpl.setWhereClauseParams(null);

        voImpl.setQuery(newQueryString);

        }

**How to find modified rows of EO/VO**

    public void findModifiedRow(){  
        Iterator itr = EmpEOImpl.getDefinitionObject().getAllEntityInstancesIterator(getDBTransaction());  
        while (itr.hasNext()) {         
            EmpEOImpl eEOImpl = (EmpEOImpl) itr.next();  
             
            if (eEOImpl.getEntityState() == Entity.STATUS\_NEW){  
                System.out.println("Emp "+eEOImpl.getEmpId()+"is a new Row");  
            }  
            if(eEOImpl.getEntityState() == Entity.STATUS\_MODIFIED ){  
            System.out.println("Emp "+eEOImpl.getEmpId()+"is a modified Row");  
            }             
            if(eEOImpl.getEntityState() == Entity.STATUS\_DELETED){  
                System.out.println("Emp "+eEOImpl.getEmpId()+"is a deleted Row");  
            }                 
        }  
    }

**AM -executeCommand()**

public int executeCommand(java.lang.String command): Executes a SQL command using a JDBC Statement under the current transaction. Applications should use this method to execute application-specific JDBC statements. This method provides a way of bypassing the framework to query the database directly. Internally, the method passes the specified SQL command to a statement on the JDBC connection and executes it.  
  
The following code example uses executeCommand. The SQL string is designed to update the EMP table. This example passes the string to executeCommand, then prints a message to report how many rows were actually updated.  
  
 public static void demoUpdateColumn(ApplicationModule appMod) {  
     String sqlStr = "UPDATE EMP " +  
                     "SET MGR=7007 " +  
                     "WHERE MGR=7698 ";  
     int n = appMod.getTransaction().executeCommand(sqlStr);  
     System.out.println("Updated " + n + " rows.");  
   }  
  
Be careful when using executeCommand, because it will execute any valid SQL statement. For example, you could perform an operation like the following DDL command:  
  
 appMod.getTransaction().executeCommand("DROP TABLE MYTEMPTABLE");  
  
A pending database transaction could be committed inadvertently due to the implicit commit performed by DDL operations, as well as having any row locks released.

**How to know whether current row is new row or from database?**

A row or view has four states:

* New
* Modified
* Un-Modified
* Initialized

To get the state using expression: #{row.row.entities[0].entityState}   
  
You can use this expression to control the display or rows in the UI. Like to display new rows in diff colors.

Ex: #{row.row.entities[0].entityState==0 ? 'background-color:yellow' :  ''}

Now in RowImpl, you can access the state using: 

Ex: byte entityState = this.getEntity(0).getEntityState()

**Get modified rows from Entitiy Cache**

To get the modified rows from entity cache we have getEntityState() method at EntityImpl class.  
  
Refer to my previous blog [Accessing EO impl methods from VO impl](http://kadamshn.blogspot.in/2016/04/accessing-eoimpl-methods-from-voimpl.html) where i am overriding the getEntityState() in EOimpl and calling it in VOImpl.  
We can use methods written or overridden in VOImpl class to AMImpl class.  
  
There are different states associated with an entity object.  
  
STATUS\_UNMODIFIED  
STATUS\_MODIFIED  
STATUS\_NEW  
STATUS\_DELETED  
STATUS\_DEAD  
  
We have to check the state or row in our AmImpl class by using the VOImpl method and through this we can distinguish the rows present at vo.  
  
  
Add below code in AMImpl class along with my previous post.  
  
public void geCachedRowsCount(){  
        JobsVOImpl jobsVo = (JobsVOImpl)this.getJobsVO();  
        RowSetIterator iter = jobsVo.createRowSetIterator(null);  
            while(iter.hasNext()){  
            Row row = iter.next();  
            byte state = jobsVo.getEntityState(row);  
            System.out.println("Job\_id -> "+row.getAttribute("JobId") +" and state ->"+state);  
// use state variable to compare and write your logic  
        }  
        
    }  
  
  
As you can see in above example, you will get the state of a row in above loop and you have to compare it with either  
0 for new created row (but not committed to DB)  
1 for unmodified state  
2 modified state (but not committed to DB).

**How to identify whether a row has been modified or not**

There is an easy way to find whether a row has been modified/deleted/created in ADF framework, This works only for Entity based view objects which are based on entity object.

Iterator tbiter =

EmployeeEOImpl.getDefinitionObject().getAllEntityInstancesIterator(getDBTransaction());

while (tbiter.hasNext())

{

EmployeeEOImpl eoImpl = (EmployeeEOImpl) tbiter.next();

if (eoImpl.getEntityState() == Entity.STATUS\_NEW ||

eoImpl.getEntityState() == Entity.STATUS\_MODIFIED ||

eoImpl.getEntityState() == Entity.STATUS\_DELETED)

{

// Add your logic here

}