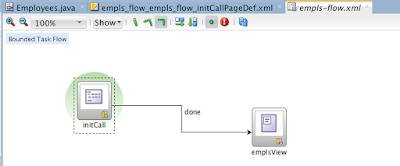
### [Avoid Blind SQL Call from ADF Task Flow Method](http://andrejusb.blogspot.in/2018/01/avoid-blind-sql-call-from-adf-task-flow.html)

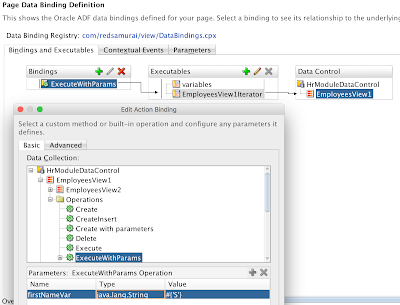
Keep an eye open on ADF Task Flow Method Call activities where methods from ADF Bindings are called. JDEV 12c sets deferred refresh for ADF binding iterators related to TF Method Call activities and this causing blind SQL to be executed. Blind SQL - query without bind variables.

Let me explain the use case, so that it will be more clear what I'm talking about.

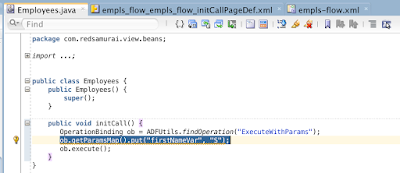
Common example - TF initialization method call where data is prepared. Typically this involves VO execution with bind variables:



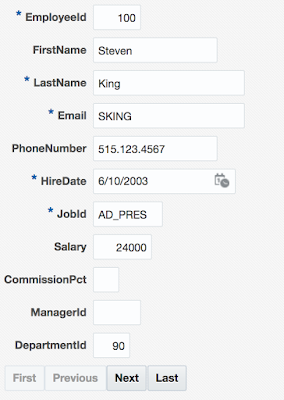
Such method call could invoke binding operation either directly (pay attention - bind variable value is set):



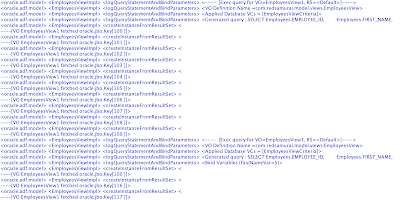
Or through Java bean method using API:



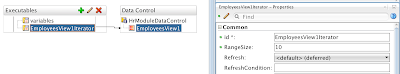
My example renders basic UI form in the fragment, after TF method call was invoked:



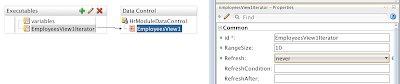
If you log SQL queries executed during form rendering, you will see two queries instead of expected one. First query is executed without bind variables, while second gets correct bind variable assigned:



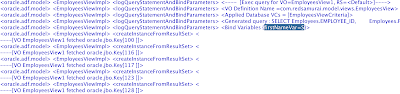
What is the cause for first query without bind variables? It turns out - iterator (with setting **Refresh = deferred**) from page definition mapped with TF method call is causing this. Somehow iterator is initialized not at the right time, when bind variable is not assigned yet and this causing blind SQL call:



Workaround is to set **Refresh = never**:



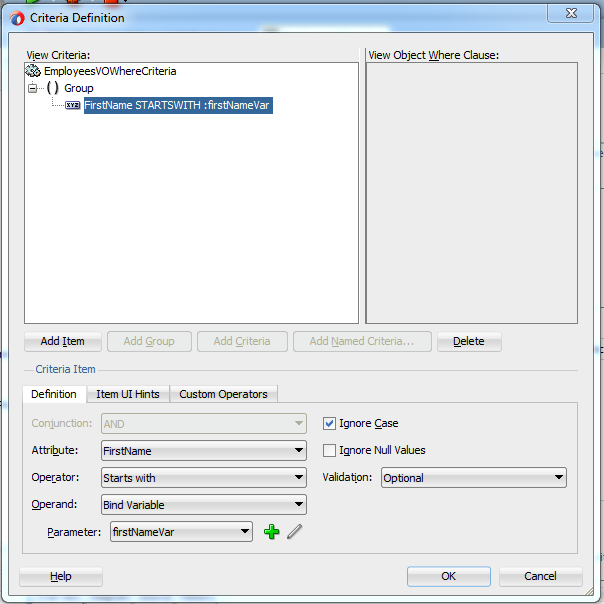
With **Refresh = never**, only one query is executed as expected, with bind variable assigned:

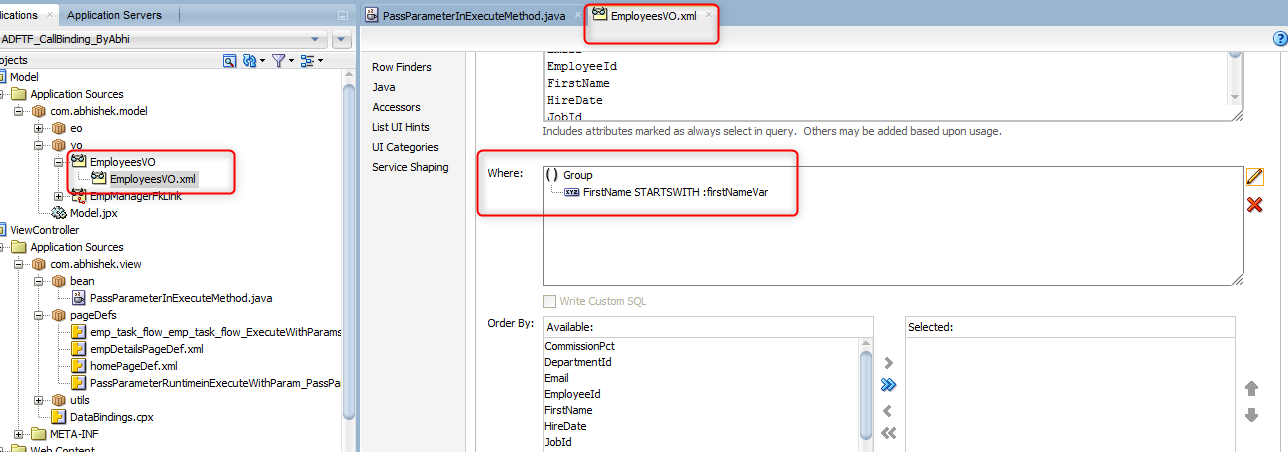


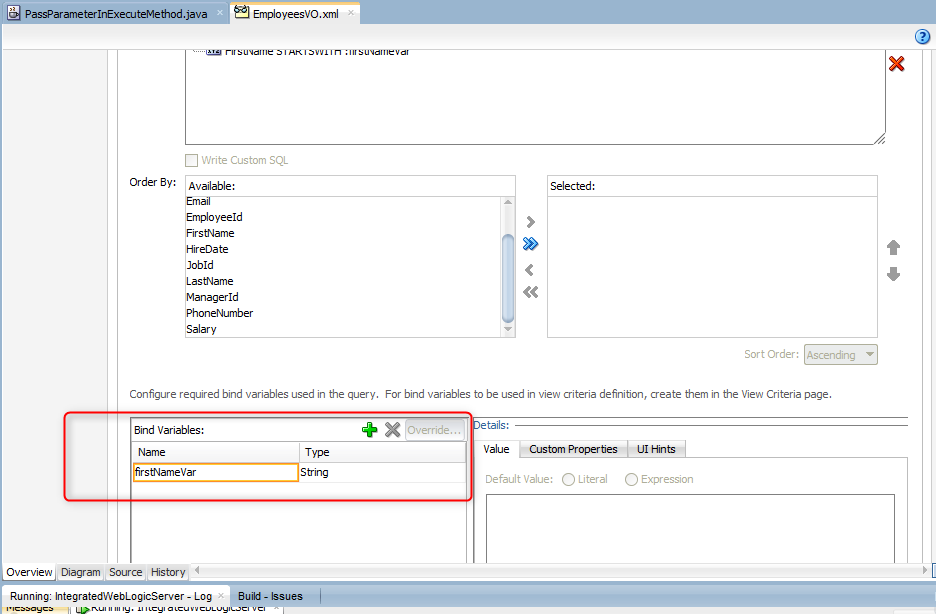
This may look minor, but trust me - with complex queries such fix could be a great help for performance tuning. Avoid executing SQL queries without bind variables.

Download sample application - [ADFTFCallBindingApp.zip](https://docs.google.com/uc?export=download&id=1hNDNOaPqc3TvWVXmPlFVctxyE2Radqtg).

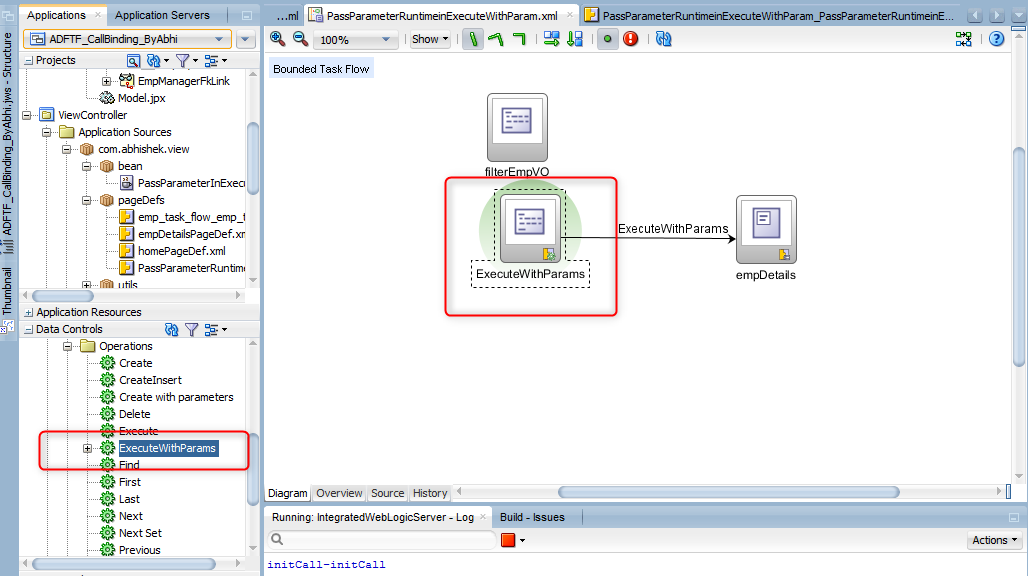
Tips and Tricks->

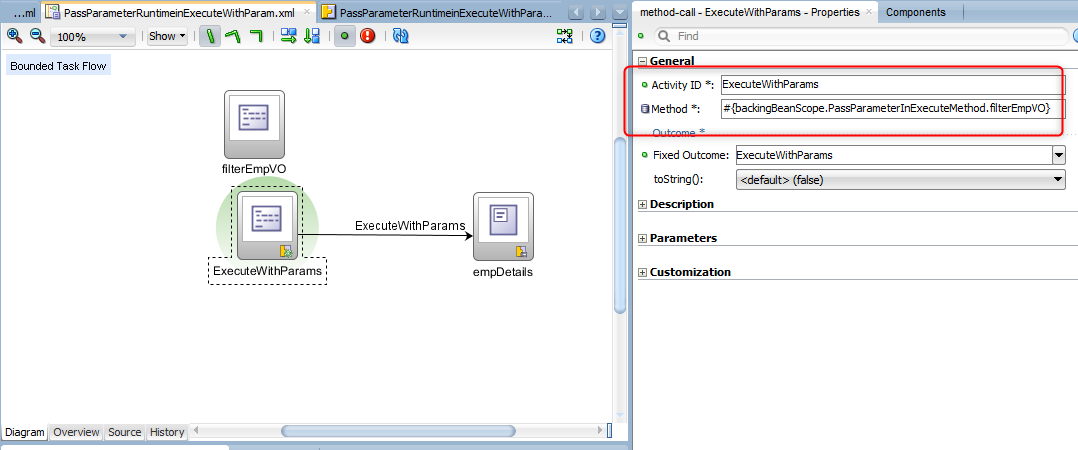


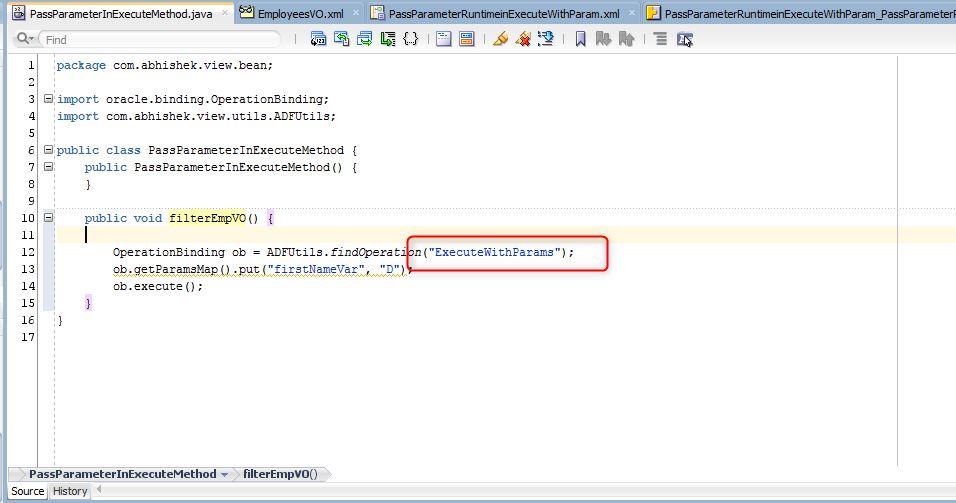


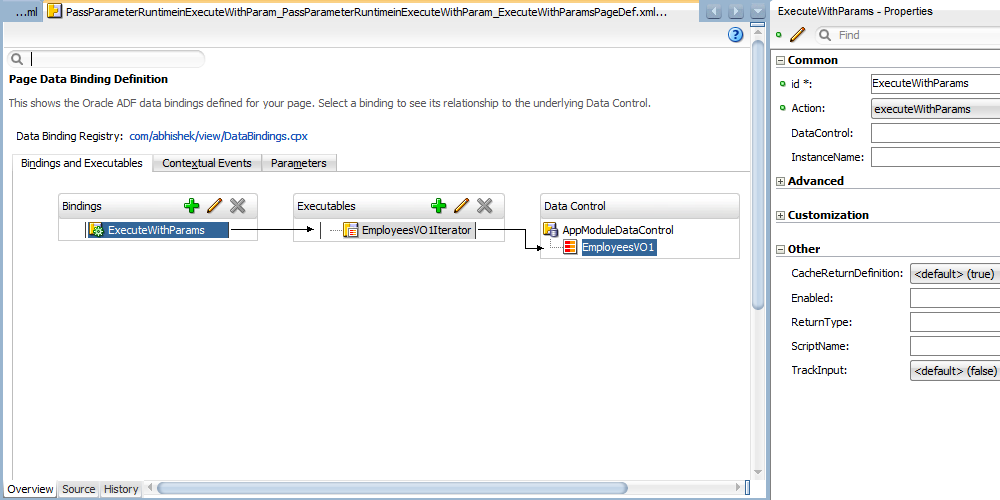


Drag and drop ExecuteWithParams method in a Task flow.



**Replace this execute method #{bindings.ExecuteWithParams.execute} with this custom method #{backingBeanScope.PassParameterInExecuteMethod.filterEmpVO}...** 

**Inside a custom method of task flow , we are calling ExecuteWithParam method with passing value**.

**ExecuteWithParam method must have there in Pagedefinition Page of Task flow. Otherwise it will give Null Pointer Exception as we are using this ExecuteWithParam method in Custom method for passing value at Runtime .**