Oracle ADF, WebCenter Tech Tips: I Came, I Learned, I Share....

This blog shares some interesting use cases I encountered along with some standards on ADF, Webcenter (FMW).

Thursday, June 14, 2012

ADF Coding Standards & Check points

For any **ADF** Java code, **Model** and **UI layer**, here are few points to make code better from writing from performance and standard point of view, that comes from our coding practice and experience.

There is a set of checkpoints we should check before source controlling any changes.This ensures making code better, robust, standards compatible and more performant.

JSFF/JSPX code - Check points

- Make sure, you have used Resource Bundle where ever its required.
- For Bounded taskflows, make sure to use activation as **conditional/deferred** (Not Immediate) and use active condition EL expression.
- Id of the component, length should be <= 7.
- Make sure, you have removed **unwanted PageDef** Bindings.
- If af:popup is used, make sure to set the contentDelivery to "**LazyUncached**".
- If using af:contextInfo, make sure the af:dialog has modal="**false**".
- If using af:contextInfo, make sure the af:showPopupBehavior has triggerType="**contextInfo**".
- If using af:popup from af:commandLink, make the **partialSubmit to true** for af:commandLink.
- If using af:popup. make childCreation property to "**deferred**" so that only when the Popup is launched the WebBean hierarchy would get created in server memory.
- Run JAudit for the file you make changes, from **Menu->Run->JAudit** file.
- Right click->**Reformat** before checkin.
- If using JavaScript, can you think to avoid that.
- jsff/jspx files must reside under **/page** package.
- Taskflows must reside under **/flow** package.
- Menu model xml files must reside under **/menu** package.
- In Case of deriving the Url parameters, preferably use the following syntax to derive declaratively *#{facesContext.externalContext.requestParameterMap['Empno']}* (Empno is url parameter)

Any Java Code - Check points

- Managed Bean class name should end with "**Bean**".
- Don't use **SOP** statements.
- New methods introduced, make sure they are modular so that they can

Search This Blog

Search

About Me

Umesh Agarwal

An Oracle Application Development Framework 11g Certified Implementation Specialist and Oracle Certified Professional Java SE7 Programmer.

View my complete profile

Popular Posts

Select/Deselect All check box in a table

Select All/Deselect All check box in a table is a common requirement. This can be implemented in multiple ways. In this blog, I will expla...

How to set classpath in weblogic server

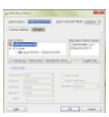
Developers like me might have struggled to figure out a way to set some jars in the CLASSPATH of the weblogic server to access and load t...

Implementation of SelectManyCheckBox component

The selectManyCheckbox component creates a component which allows the user to select many values from a series of checkboxes. ...

Use case of SelectManyListBox component

Oracle ADF is a powerful application framework for building next generation enterprise applications. This blog discusses one interesting ...



How to implement cascading LOV's in search criteria

Cascading LOV is a common requirement. It is dependencies of an LOV on the selected value of another LOV. In this post, I will explain ho...

be JUnitized.

- New Method should always return a value so that they can be JUnitized.
- Add Logger statements to the new methods.
- Make sure you have thrown sufficiently Exceptions.
- Run JAudit for the file you make changes, from **Menu->Run->JAudit** file.
- Make sure to define the Variables with a name starting with small letter.
- Do not use underscore (_) in Variable and Method names.
- Make sure to check for Null condition for all operation that could throw NullPointerException. Ex: CollectionModel, Row, ViewObject, ApplicationModule, String objects etc.
- Right click->**Reformat** before check in.

Application Module(AM) - Check Points

- Make sure to Use **JNDI** datasource instead of **JDBC URL** in the AM configuration(i.e **bc4j.xcfg**)
- Number of occurrences of **createRootApplicationModule**() or ***AMImpl.getInstance**() should be same as **releaseRootApplicationModule**().

Entity Object - Check Points

- Can you convert the code you have just written into Groovy?
- Run JAudit for the file you make changes, from **Menu->Run->JAudit** file.

View Object - Check Points

- Get **SQL explain plan** every time you touches a VO.
- For new Attributes added, make sure you use proper UI hints.
- If creating a New VO, make sure its is based on an EO (Exceptions to be considered).
- If createViewObject calls are present, then make sure, you remove those dynamic ViewObjects.
- If setting **rangeSize() to -1**, restore the rangeSize after you work on that.
- Don't set **ListRangeSize == '-1'** unless the ViewAttribute using the LOV has **CONTROLTYPE** 'choice' or 'radio' or 'default'
- **FetchSize** should be **<= 25** or same as number of rows to be displayed in UI.
- Avoid use of **vo.getRowCount()**
- View Criteria shouldn't have "**null checking**" checkbox checked.
- LOVs shouldn't have "**query automatically**" checked.
- SQL based VOs,should have query optimizer hint set to "**FIRST_ROWS(10)**".
- Always test the application with **ampooling= false** when our bc4j transaction involves any transient VO's.

Most of the above points were shared with me by one of my superior(Amulya Mishra) when I was a novice ADF developer. So, thought this check list will be helpful for other novice developers, like it helped me :)

Posted by Umesh Agarwal at Thursday, June 14, 2012

Labels: Application Module, Entity Object, View Object

1 comment:

subra manian Friday, June 29, 2012

nice points good. i personally appreciate you both the peoples

Implementation of SelectManyShuttle component

The selectManyShuttle provides a mechanism for selecting multiple values from a list of values by allowing the user to move items between...

ADF Coding Standards & Check points

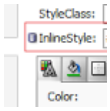
For any ADF Java code, Model and UI layer , here are few points to make code better from writing from performance and standard point of ...

SelectManyCheckBox component in a table

This post is very similar to my earlier post on Implementation of SelectManyCheckBox component. The use case/functionality remains same ....

Format: Date and Number fields in table

To format a date field on UI, use <af:convertDateTime> . For example, if a date has to be displayed in 12-Jan-2012 format, use conv...



Use case of Inlinestyle & ContentStyle

In this post, I will explain a use case of InlineStyle and ContentStyle on ADF UI components. In one of my project I had a requirem...

Translate

Select Language

Powered by Google Translate

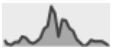
Blog Archive

Reply

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)

Pageviews



Visitors

There was an error in this gadget

Subscribe To

Posts

Comments

Followers

Join this site
with Google Friend Connect

Members (2)

Already a member? Sign in

Follow by Email

Email address... Submit

Google+ Followers

Simple template. Template images by sandoclr. Powered by Blogger.