

How to Add Dynamic Checkboxes in a Web Dynpro Java Page



Applies to:

SAP NetWeaver 2004s (EP7)

For more information, visit the [Web Dynpro Java homepage](#).

Summary

This article is about dynamically manipulating a Web Dynpro Java page with information available at run time.

Author: Tathagata Pal

Company: Tata Consultancy Services

Created on: 14 July 2009

Author Bio



Tathagata Pal is a Sun certified development consultant with Tata Consultancy Services Ltd. He has around 5 ½ years of experience industry experience including SAP Enterprise Portal and Webdynpro

Table of Contents

Problem Description	3
Solution Details	3
Step 1:	3
Step 2:	4
Step 3:	5
Step 4:	6
Step 5:	7
Step 6:	8
Related Content	9
Disclaimer and Liability Notice	10

Problem Description

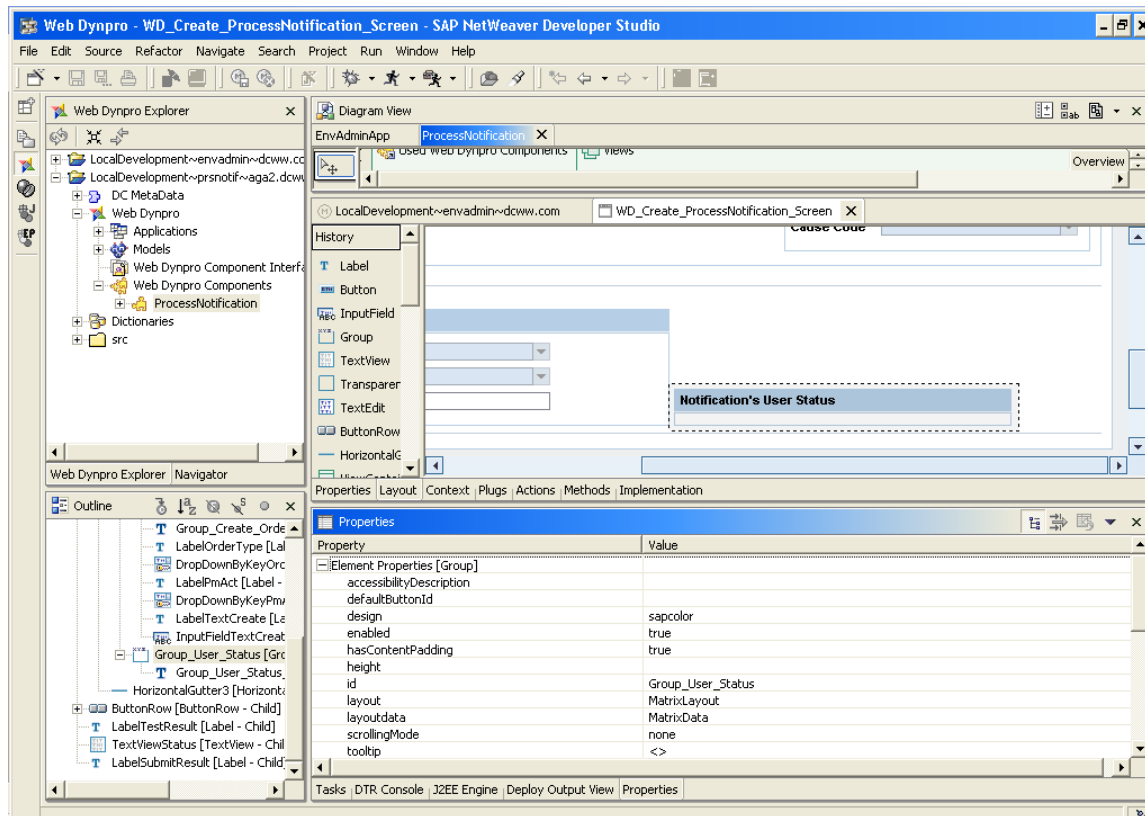
You need to have a group of checkboxes in a page but do not know the number of checkboxes to be included in the group until run time. Once the check boxes are rendered the check events also has to be captured and appropriate event handler has to be invoked.

Solution Details

The solution involves multiple steps to be followed in design time and code to be executed in run time.

Step 1:

Create a UI Element container in the page to hold the check box group. (Optional but good practice)



Step 2:

Create a **Value Node** with a **Value Attribute** in the context with parameters as shown below

The screenshot shows the SAP Web Dynpro IDE interface. The top part is the 'Context' editor, which displays a tree view of the context structure. The 'Context' node is expanded, showing several child nodes: 'Process_Notif_Detail_Input', 'Process_Notif_Submit_Input', 'UserGroupNode', 'userStatusGroup', 'causeCode', 'causeGroup', 'createOrderText', 'exceptionmessage', 'finalCheckedStatus', and 'localEffect'. The 'UserGroupNode' is selected and highlighted in blue. Below the tree view is a tabbed interface with 'Properties' selected. The 'Properties' tab shows a table with two columns: 'Property' and 'Value'. The table is organized into sections: 'Mapping' and 'Misc'.

Property	Value
Mapping	
mappingRef	ProcessNotification.userGroupNode
mappingType	collection_and_selection
Misc	
cardinality	0..n
collectionType	list
initializeLeadSelection	true
name	userGroupNode
selection	0..n
singleton	true
structure	
technicalDocumentation	
typedAccessRequired	true

Step 3:

This context element would be used to store the texts to be displayed along with the unknown number of check boxes which would be known only at run time. The following code illustrates this:

```
//1. Create context elements for node "UserGroupNode"
List checkBoxList = new ArrayList();
for (int i = 0; i < checkboxes.length; ++i) {

    IPrivateWD_Create_ProcessNotification_Screen
        .IUserGroupNodeElement groupNodeElement =
        wdContext.createUserGroupNodeElement();

    groupNodeElement.setUserStatusGroup(checkboxes[i]);
    checkBoxList.add(groupNodeElement);

}
```

Note: the variable checkboxes is an array of Strings containing the text to be displayed along side the check boxes.

```
//2. Bind node to element list
wdContext.nodeUserGroupNode().bind(checkBoxList);
```

Step 4:

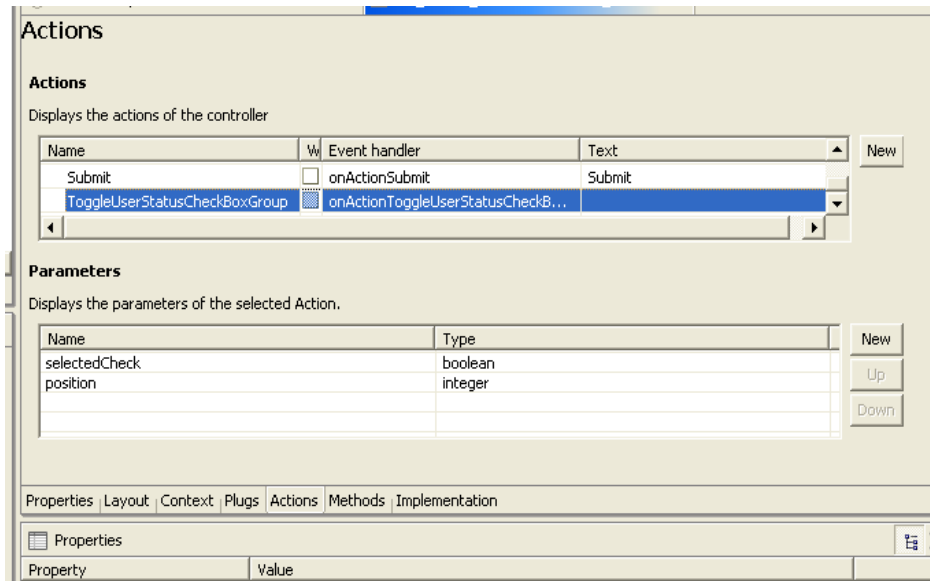
Now it's time to actually create the UI Elements and bind it to the context elements that we have created in the previous step. Since, UI Elements can be accessed only from the **wdDoModifyView** hook method, we write the following code in the method:

```
if (firstTime) {
    IWDGroup container =
(IWDGroup)view.getElement("Group_User_Status");
    IWDCheckBoxGroup checkBoxGroup =
        (IWDCheckBoxGroup) view.createElement(
            IWDCheckBoxGroup.class,
            "UserStatusCheckBoxGroup");
    checkBoxGroup.bindTexts("userGroupNode.userStatusGroup");
    container.addChild(checkBoxGroup);
}
```

This would have been sufficient if you wanted just dumb check boxes in your page and there was no need to capture events from them. Since that is hardly ever the case, you need to register events with the checkbox group just created and write event handlers to handle them

Step 5:

Define the event (Action) and the Event Handler



Note: That we have introduced two parameters “**selectedCheck**” and “**position**” in the event handler method. The purpose will be explained shortly.

Step 6:

Add the following immediately below the previous code:

```
// Event handling
IWDAction checkBoxAction = wdThis.wdCreateAction(
    IPrivateWD_Create_ProcessNotification_Screen
    .WDActionEventHandler
    .TOGGLE_USER_STATUS_CHECK_BOX_GROUP,
    " ");

checkBoxGroup.setOnToggle(checkBoxAction);
checkBoxGroup.mappingOfOnToggle().addSourceMapping(
    IWDCheckBoxGroup.IWDonToggle.CHECKED,
    "selectedCheck");
checkBoxGroup.mappingOfOnToggle().addSourceMapping(
    IWDCheckBoxGroup.IWDonToggle.INDEX,
    "position");
```

It is evident that the following event handler method will be “called back” by the container whenever a change is made to the visible check box group in the page. That is, you have modified the checked/unchecked state of the check box.

```
public void onActionToggleUserStatusCheckBoxGroup(
    com.sap.tc.webdynpro.progmodel.api.IWDCustomEvent wdEvent,
    boolean selectedCheck,
    int position) {
    //@@begin onActionToggleUserStatusCheckBoxGroup(ServerEvent)

    // Code your heart out here. You have selectedCheck and position of the // individual
    // check box that has fired this event

    //@@end
}
```


Related Content

help.sap.com

sdn.sap.com

[Web Dynpro Java API Javadoc](#)

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.