

ADF Code Corner

041. How-to conditionally prevent dialogs from closing

ORACLE
CODE CORNER



twitter.com/adfcodecorner

Abstract:

ADF Faces Rich Client provides in browser popup dialogs and panels. Compared to the still existing and still good to use ADF Faces dialog framework, the popups are lightweight dialogs that have their content defined within the parent page. At runtime these popups render as DHTML dialogs on the parent page. The available categories of popups include inline selectors, dialogs, note windows and menus. A dialog component, added as the first child of af:popup, may contain buttons like ok, cancel, yes and no. Each of the buttons allow the developer to respond to the user selection in a managed bean method before the dialog is closed. And this is where this how-to comes in. How can closing a dialog made conditional ?

Author:

Frank Nimphius, Oracle Corporation
twitter.com/fnimphiu
01-JUL-2008

Oracle ADF Code Corner is a loose blog-style series of how-to documents that provide solutions to real world coding problems.

Disclaimer: All samples are provided as is with no guarantee for future upgrades or error correction. No support can be given through Oracle customer support.

Please post questions or report problems related to the samples in this series on the OTN forum for Oracle JDeveloper: <http://forums.oracle.com/forums/forum.jspa?forumID=83>

Introduction

In general it is okay to close a dialog when the user presses the "Ok" or "Cancel" button,. But there also exist usecases where the dialog should only be closed if a specific condition is met. For example: If your dialog box is for a user to acknowledge and confirm a specific licence agreement, then the ok button should not close the dialog until the checkbox with the terms and conditions is checked. Cancel however should close the dialog as it is equivalent to the user saying "no" to your terms of licence.

Code Sample

ADF Faces RC is a JavaServer Faces framework and we recommend to find what you need in the ADF Faces components first. Only if you cannot find in ADF Faces what you are looking for, or if you have a good excuse to not use the ADF Faces component Java API you are good to use JavaScript. Its not that JavaScript doesn't work well, but its recommended best practices that should be obeyed. An example, where JavaScript is needed, is the usecase of conditional dialogs closing. There is no equivalent Java API available on the component framework to stop a dialog from closing, thus JavaScript is a good choice.

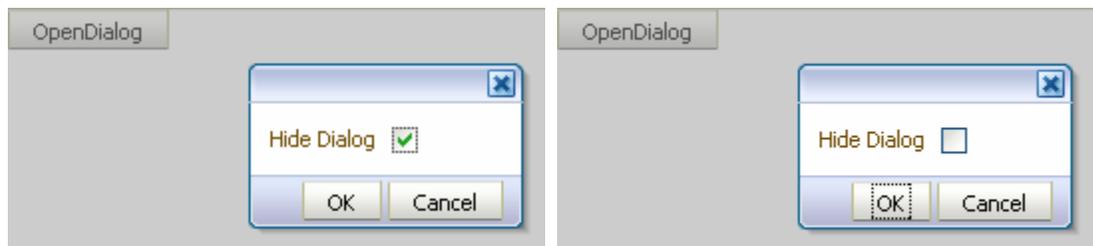


Figure 1: Allow dialog to close when pressing Ok

Figure 2: Don't allow dialog to close when pressing Ok

The page source code that produces the images above looks as follows

```
<?xml version='1.0' encoding='windows-1252'?>
<jsp:root xmlns:jsp="http://java.sun.com/JSP/Page" version="2.1"
  xmlns:h="http://java.sun.com/jsf/html"
  xmlns:f="http://java.sun.com/jsf/core"
  xmlns:af="http://xmlns.oracle.com/adf/faces/rich">
<jsp:directive.page contentType="text/html; charset=windows-1252"/>
<f:view>
  <af:document>
```

```

<f:facet name="metaContainer">
  <af:group>
    <![CDATA[
      <script>
        function checkCloseDialog(event) {
          var checkbox =
            event.getSource().findComponent("checkBox1");
          var eventOutcome = event.getOutcome();
          if(checkbox.getSubmittedValue()==false &&
            eventOutcome == AdfDialogEvent.OUTCOME_OK) {
            event.cancel();
          }
        }
      </script> ]]>
  </af:group>
</f:facet>
<af:form>
  <af:commandButton text="OpenDialog">
    <af:showPopupBehavior popupId="dialog1"
      triggerType="action"/>
  </af:commandButton>
  <af:popup id="dialog1">
    <af:dialog
      dialogListener="#{pageFlowScope.dialogOpenCloseBean
        .onDialogAction}">
      <af:selectBooleanCheckbox id="checkBox1"
        clientComponent="true"
        label="Hide Dialog"
        value="#{pageFlowScope.dialogOpenCloseBean
          .closeDialog}"/>
      <af:clientListener method="checkCloseDialog"
        type="dialog"/>
    </af:dialog>
  </af:popup>
</af:form>
</af:document>
</f:view>
</jsp:root>

```

The JavaScript is added to the metaContainer facet of the af:document, as it is recommended by Oracle. The JavaScript function **checkCloseDialog** is called by a clientListener component that is added to the af:dialog component (highlighted in red). The **event** argument contains a reference to the event source, which is the dialog. Since the component of interest, the checkbox, is contained in the dialog, we can start the component search from this reference. Once we have a handle to the checkbox component, we can use its **submittedValue** to determine whether or not the dialog is allowed to close. Until the dialog is allowed to close, the event.cancel() method is called to prevent the broadcast of the close event.

A word about the checkbox:

```

<af:selectBooleanCheckbox id="checkBox1"
  clientComponent="true"
  label="Hide Dialog"
  value="#{pageFlowScope.dialogOpenCloseBean
    .closeDialog}"/>

```

The **clientComponent** of the `selectBooleanCheckbox` must be set to **true** to make it accessible to JavaScript. The **value** property references a managed bean method in `pageFlowScope`. Using ADFc, a managed bean has additional scopes to use than the default request, session and application scope of Java EE. The `pageFlowScope` lasts longer than request and shorter than session. It spans over a set of pages that belong to the same taskflow. The managed bean method is used to memorize the checkbox state in this sample.

RELATED DOCUMENTATION

<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	