

Working with Events in Web Dynpro ABAP



Applies to:

SAP ECC 6.0 (Release 700, SP 12)

Summary

This article basically explains the usage of events in a Web Dynpro component and across multiple components.

Author: Abhimanyu Lagishetti

Company: Satyam Computer Services Ltd.

Created on: 12 March 2008

Author Bio



Abhimanyu L. works at Satyam Computer Services Ltd and is a B.Tech Computer Science Graduate. He is working on Technologies like Web Dynpro Java, Web Dynpro ABAP, Enterprise Portals and Business Workflows.

Table of Contents

Introduction	3
Get Started with an Application	3
Events across Web Dynpro Components.....	6
Summary.....	7
Related Content.....	7
Disclaimer and Liability Notice.....	8

Introduction

Event is “**Change of State of an Object**”. When user performs an action in one view of a Web Dynpro Component and we want the action to be handled in another view of the same component or another Web Dynpro Component, Events are helpful.

Let’s work with a simple example where click of a button in one view is handled in another view.

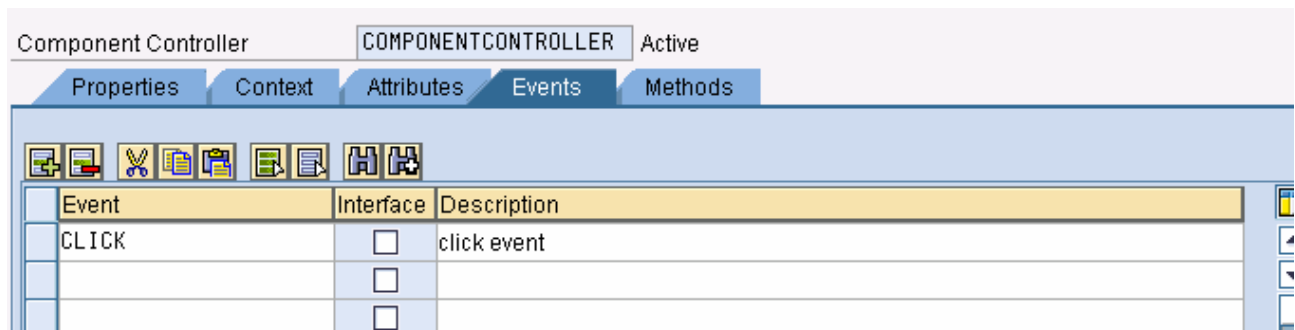
This is View1 This is View 2

On click of the button

This is View1 This is View 2 - Button clicked

Get Started with an Application

- Create a web Dynpro component, and create two views view1 and view2
- In View1 place a caption and Button
- In View2 create a context attribute called TEXT of type STRING, and place a caption control on the layout and bind it with the TEXT Attribute.
- Create a Main View and place two view container ui elements to display view1 and view2
- Add main view to the window and embed view1 and view2 in the corresponding view container UI elements
- Create an Event in the component Controller (Picture 1)

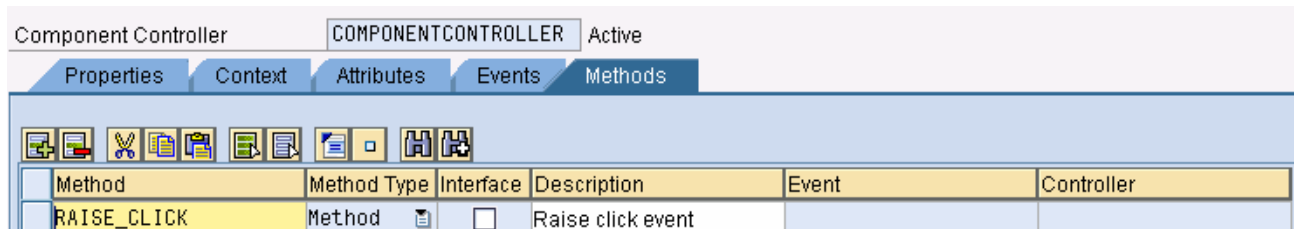


Picture 1

When you create an event, observe the a method generated in the IF_COMPONENTCONTROLLER (Picture 3)

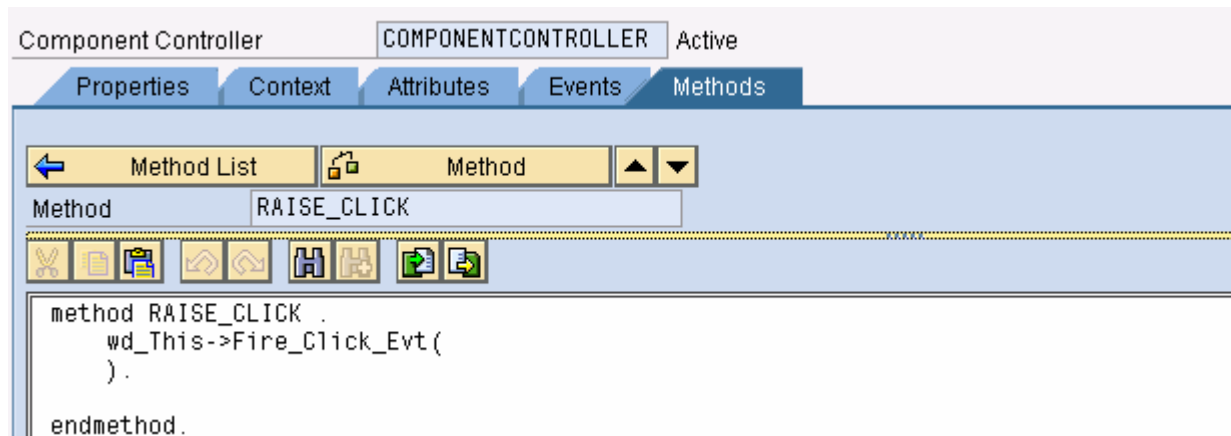
Events cannot be fired from Views, for this reason create a method in the component controller, fire the event inside the method and call the created method from any view

- Create a Method RAISE_CLICK which fires the event CLICK

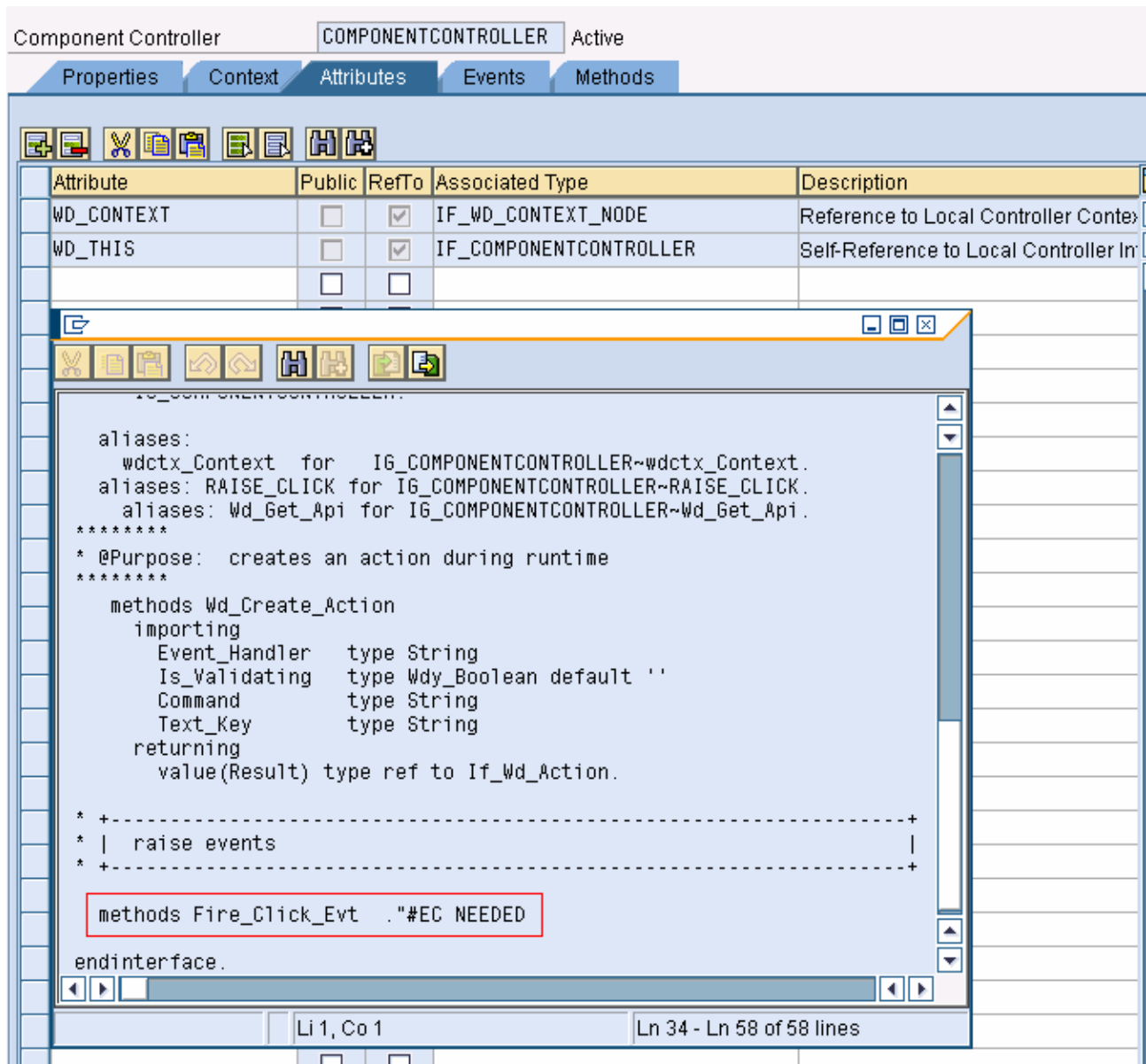


Picture 2

- Call the fire event method inside RAISE_CLICK method

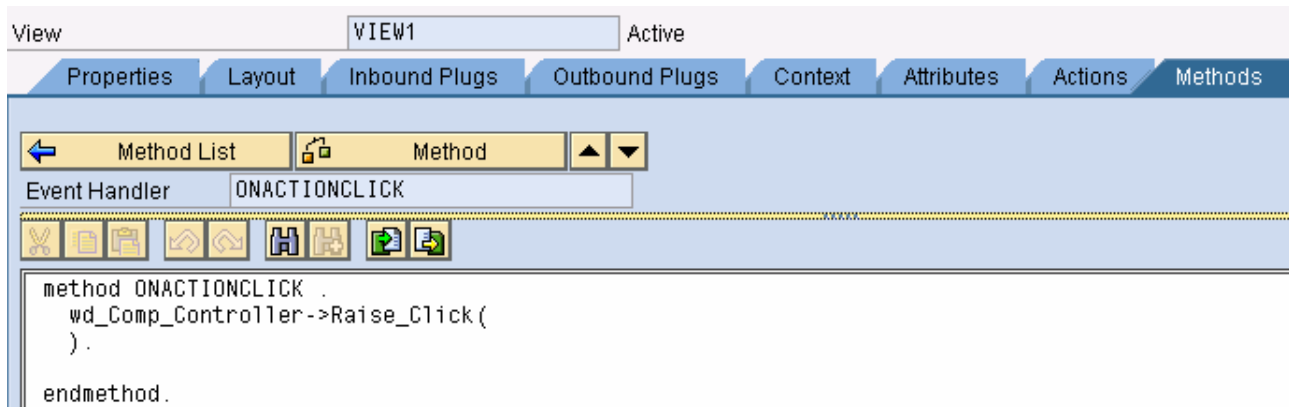


Picture 3



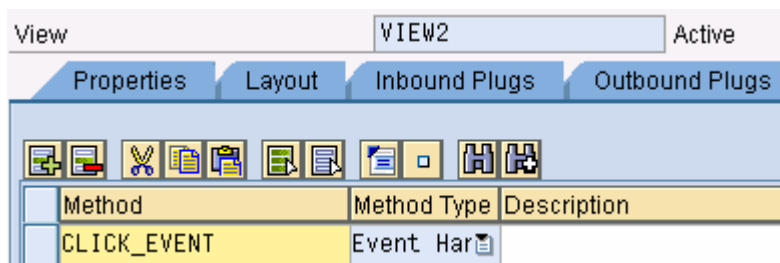
Picture 4

- Call the RAISE_CLICK method of the component controller in view1's button click handler



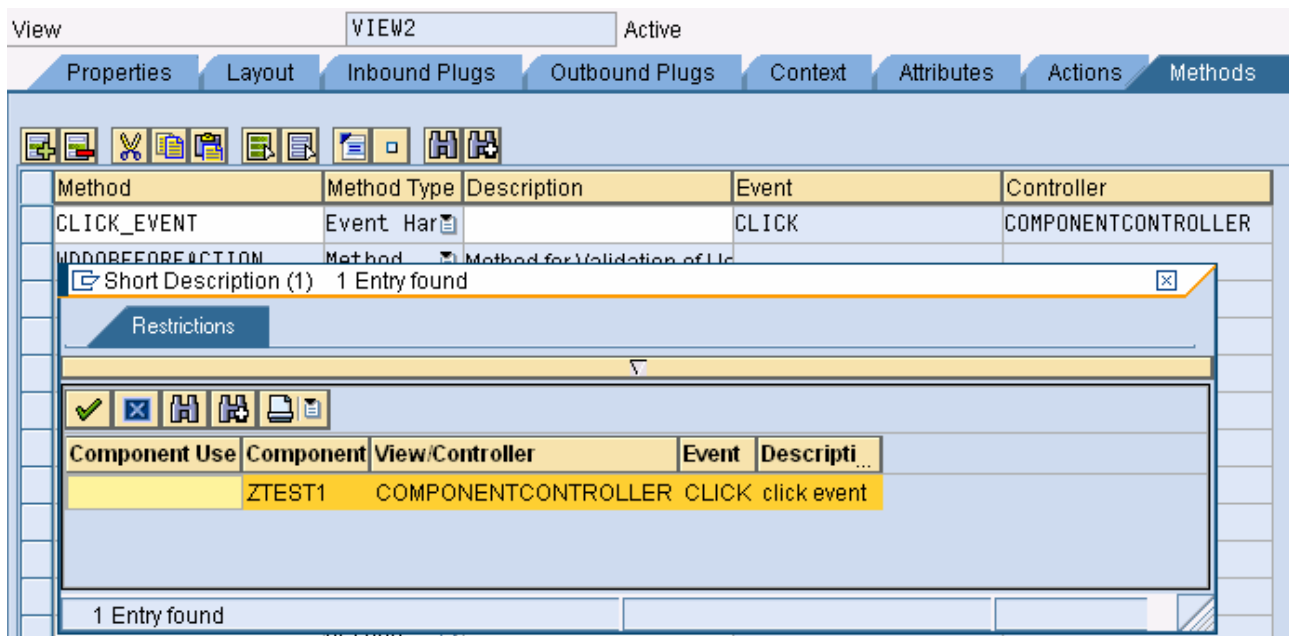
Picture 5

- In View2, go to the methods tab and add a method CLICK_EVENT and change the method type to event handler



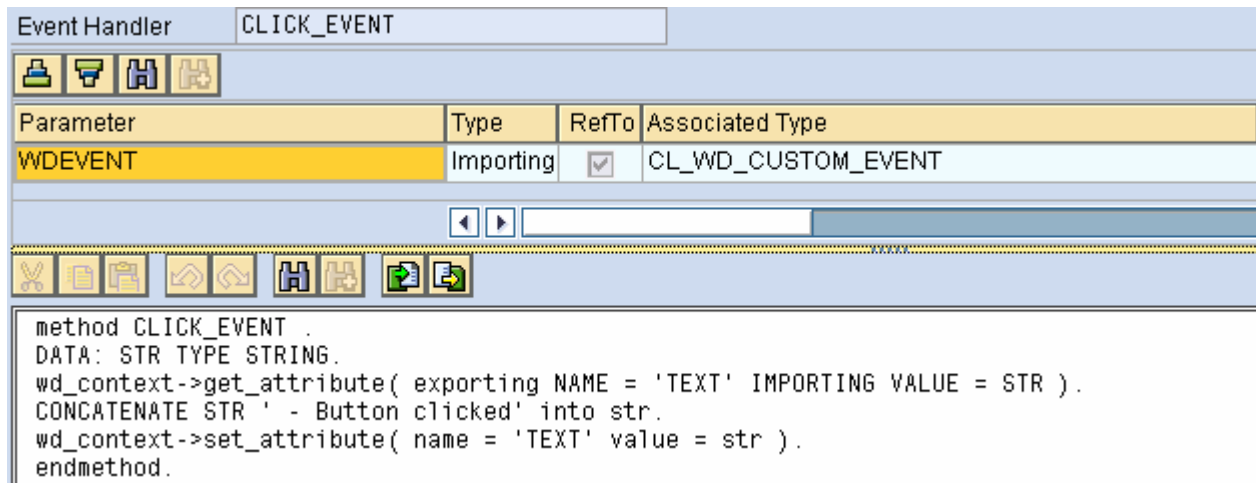
Picture 6

- Subscribe the Event CLICK of the component controller for the method CLICK_EVENT, by selecting the value help from the Event Column in the Methods Tab



Picture 7

- Code the following the CLICK_EVENT method



Parameter	Type	RefTo	Associated Type
WDEVENT	Importing	<input checked="" type="checkbox"/>	CL_WD_CUSTOM_EVENT

```

method CLICK_EVENT .
  DATA: STR TYPE STRING.
  wd_context->get_attribute( exporting NAME = 'TEXT' IMPORTING VALUE = STR ).
  CONCATENATE STR ' - Button clicked' into str.
  wd_context->set_attribute( name = 'TEXT' value = str ).
endmethod.

```

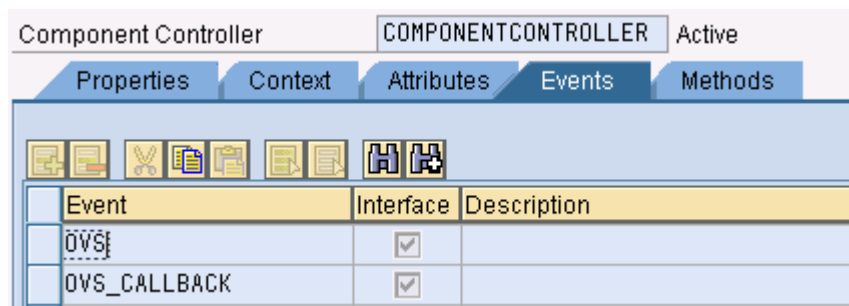
Picture 8

- Create an Application and run

Events across Web Dynpro Components

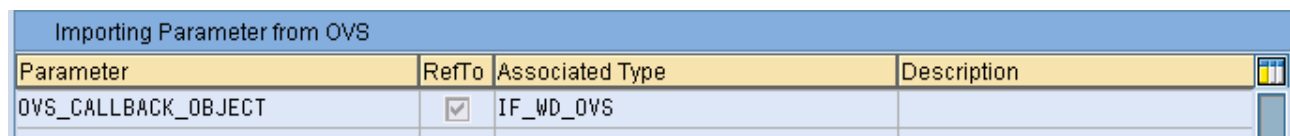
Note: to use the events across multiple components while creating the event check the check box for the column interface. For standard component WDR_OVS see the OVS event in the component controller

If we use WDR_OVS Component in our component now we can handle OVS event just as shown in Picture 7



Event	Interface	Description
OVS	<input checked="" type="checkbox"/>	
OVS_CALLBACK	<input checked="" type="checkbox"/>	

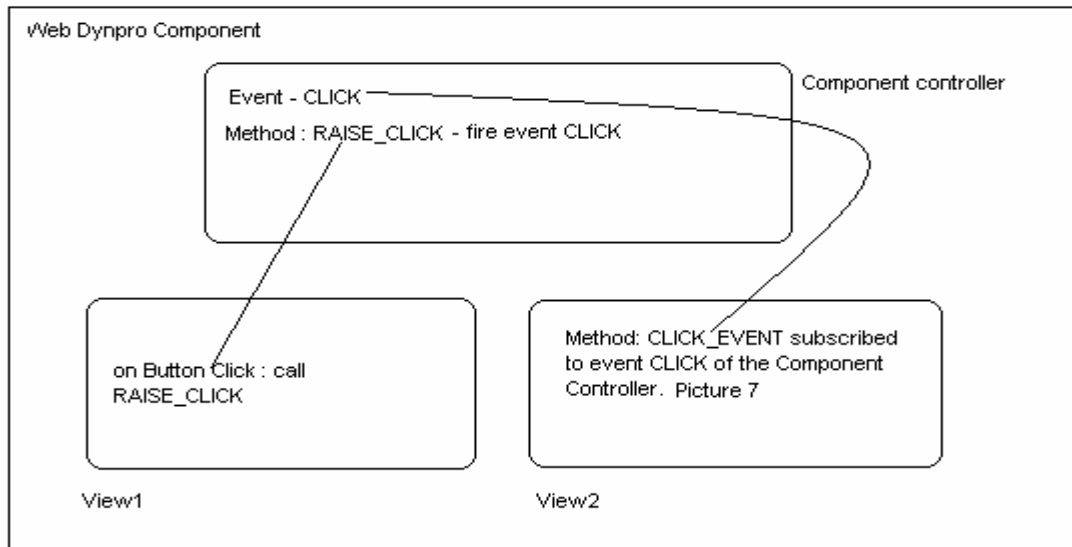
Parameters can be passed along with the event, to add parameters double click the event and add parameters at the down.



Parameter	RefTo	Associated Type	Description
OVS_CALLBACK_OBJECT	<input checked="" type="checkbox"/>	IF_WD_OVS	

Summary

A RAISE_CLICK method is created in the component controller to fire the event CLICK, because the event is not accessible from the view.



Related Content

[Web Dynpro Architecture](#)

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.