

Retrieving Application Parameters Dynamically in Web Dynpro for ABAP Application



Applies to:

Web Dynpro for ABAP, Web Dynpro for Java and CE 7.1 Portal. For more information, visit the [Web Dynpro ABAP homepage](#)

Summary

This document is all about retrieving application parameters in Web Dynpro ABAP application sent by a Web Dynpro for Java application. The value of the parameter is dynamically decided by the source application and passed via URL parameters. The methodology is mentioned in this document is one of the ways of passing parameters to Web Dynpro for ABAP applications.

Author: Ramganesan Karuppaiah

Company: Infosys Technologies Limited, Bangalore, India

Created on: 12 September 2011

Author Bio



Ramganesan Karuppaiah is a NetWeaver Portal Consultant currently working with Infosys Technologies Limited, Bangalore. He has 5+ years of extensive experience in SAP Enterprise Portal and ABAP. He is an SAP Certified ABAP with NetWeaver 7.0 development Associate. Also he is a Sun Certified Java Programmer 1.5. His nature of work involves Portal Administration, Portal Customization, Personalization and Content Development & Customization using Web Dynpro for Java, JSP Dynpage, ABAP and Web Dynpro for ABAP.

Table of Contents

Introduction	3
Result Preview:	3
Creation of Web Dynpro for ABAP Project:	3
Steps:	4
Step 1:	4
Step 2:	4
Step 3:	4
Step 4:	4
Step 5:	5
Step 6:	5
Creation of WebDynpro for ABAP iView:.....	5
Creation of Web Dynpro for Java Project:	6
Implementation of Action Handler Method:	7
Creation of WD Java Page in Portal:	7
Result:.....	7
Related Content.....	9
Disclaimer and Liability Notice.....	10

Introduction

The purpose of this article is to receive the parameter in web Dynpro for ABAP application sent by Web Dynpro for Java Application as per the result below. The value of the parameter shall be dynamically decided.

Result Preview:

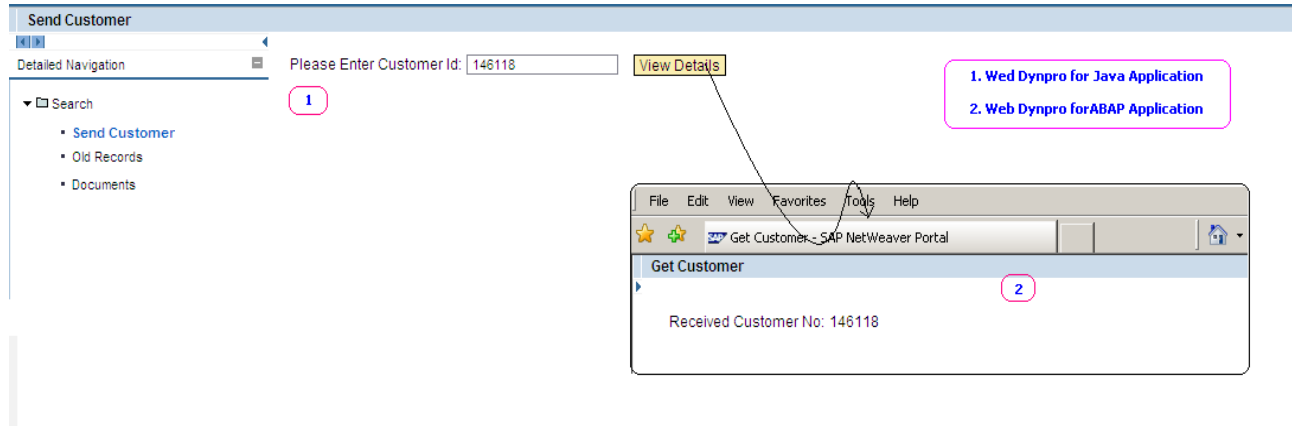


Fig: Result Preview

Creation of Web Dynpro for ABAP Project:

The below diagram demonstrate the application structure of the Web Dynpro for ABAP application.

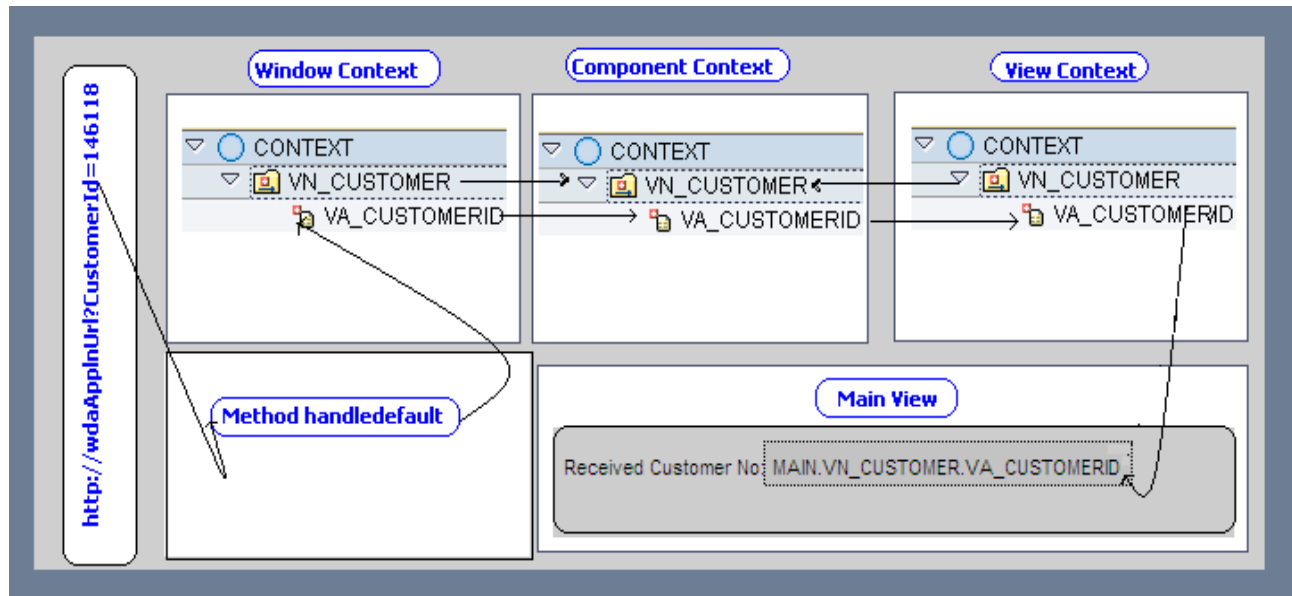


Fig: Application Structure of Web Dynpro for ABAP Application

Steps:**Step 1:**

Create a new Web Dynpro for ABAP project in ABAP Workbench (SE 80) as per the below details.

Project Name:	ZSample_wda
Window Name:	ZSample_wdaWin
View Name:	Main View
Application Details:	ZReceiveCustomerId_App

Step 2:

Create Value node and attribute as per the given details below in context of Component Controller, Interface Controller and View Controller. Establish the context mapping between them as per the below diagram.

Value Node & Attribute:	VN_CUSTOMER, VA_CUSTOMERID(String)
------------------------------------	---------------------------------------

Step 3:

Implement custom coding in the default method **handledefault** of Window. Add a new parameter "CUSTOMERID" with type string in the method. This is required to receive the parameter passed by the application URL. Implement the custom coding to set the customer id into context node's attribute Customer Id. The steps are as in the screenshot below.

The screenshot displays the SAP ABAP Workbench interface for configuring a Web Dynpro window. The left pane shows the project structure with the window **ZSAMPLE_WDAWIN** selected under **Views**. The right pane shows the **Methods** tab for the **HANDLEDEFAULT** event handler. A table lists the parameters for the method:

Parameter	RefTo	Opt	Associated Type	Description
WDEVENT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	CL_WD_CUSTOM_EVENT	
CUSTOMERID	<input type="checkbox"/>	<input checked="" type="checkbox"/>	STRING	

Below the table, the custom coding for the **method handledefault** is shown:

```

1 method handledefault .
2
3   data:lo_node type ref to if_wd_context_node,
4         lo_elmt type ref to if_wd_context_element.
5
6   "Get the Customer context node
7   lo_node = wd_context->get_child_node( wd_this->wdctx_Vn_Customer ).
8   "Get node element reference
9   lo_elmt = lo_node->get_element( ).
10  "Set the CustomerId to the context
11  lo_elmt->set_attribute( name = 'VA_CUSTOMERID' value = CUSTOMERID ).
12
13
14 endmethod.

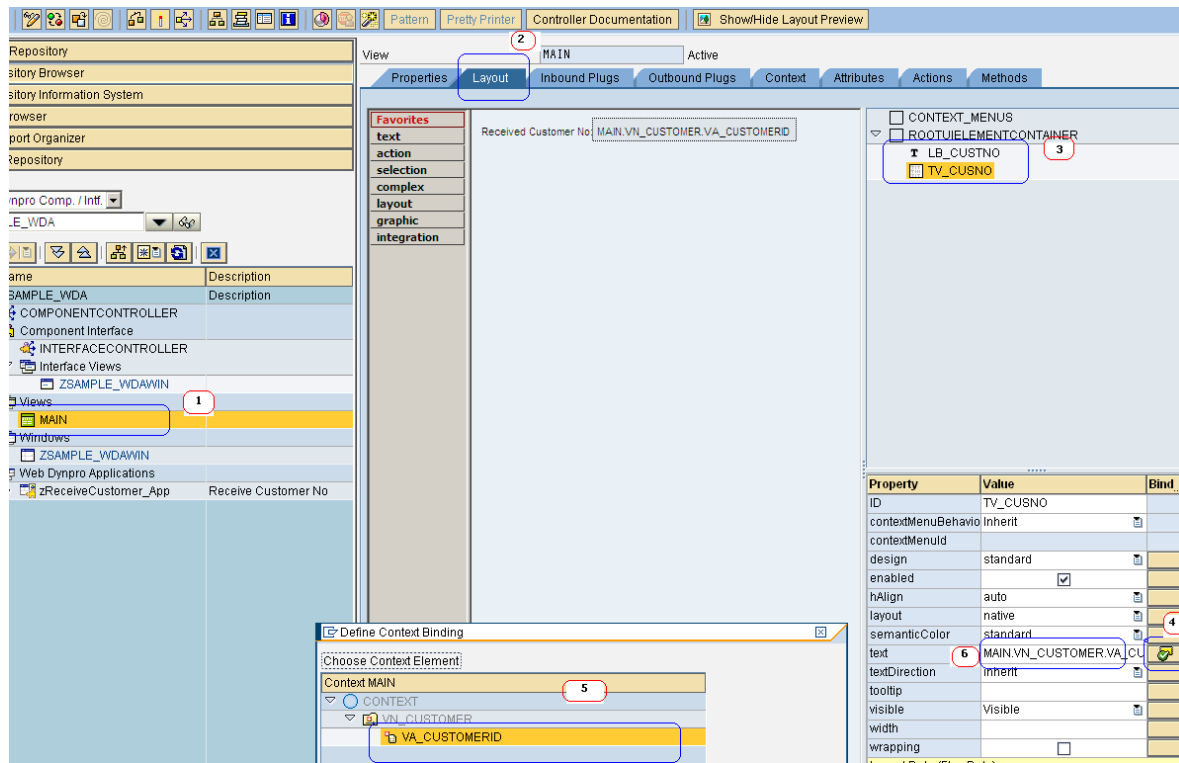
```

Step 4:

Create a Label and Text View UI elements in Main View.

Step 5:

Bind the value attribute of the node Vn_CUSTOMER in View Context with the Text view UI of the Main View as per the steps mentioned in the below screenshot.

**Step 6:**

Activate the WD Project.

Creation of WebDynpro for ABAP iView:

This is an important part of the article in which we map application parameter of source application to destination application i.e. Web Dynpro for ABAP iView.

The iView creation steps are as below:-

Content Administration → Portal Content → Navigate to your content folder → Right Click → Create New → iView → select WD Dynpro for ABAP → provide valid name, id, prefix details → Select a valid System Object → Set Namespace = [sap], Application Name = [zReceievCustomer_App], Application Parameters = [CUSTOMERID=<Request.CUSTOMERID>]

Note: The template of application parameters should be as below:

- param1=<Request.paramName1> (or)
- param1=<Request.paramName1>¶m2=<Request.paramName2>...¶mN=<Request.paramNameN>

Example: CUSTOMERID=<Request.CustomerId>

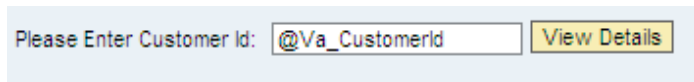
In the above example, the parameter name CUSTOMERID should be same as in the parameter name of handle default Method. Value of the parameter <Request.CustomerId> shall be retrieved from request object and filled at run.

Creation of Web Dynpro for Java Project:

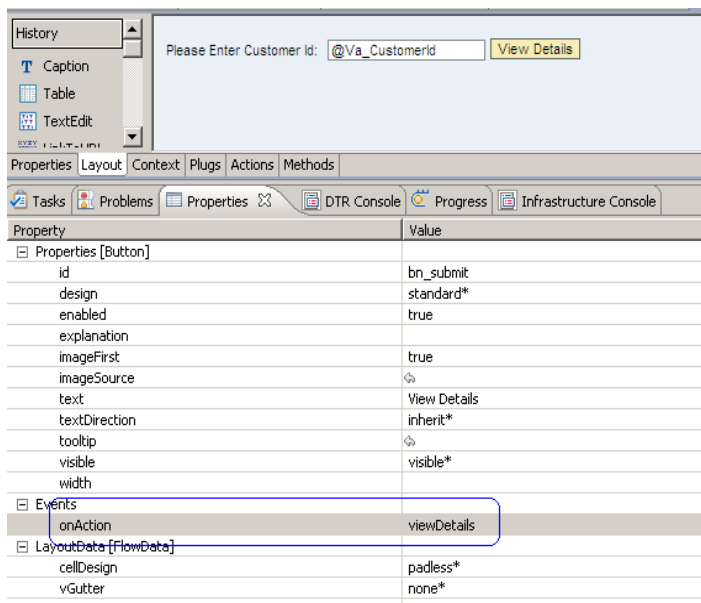
1. Create a new Web Dynpro for Java project in NWDS as per the below details.

Project Name:	sample_wd
Component Name:	Customers_Comp
Window Name:	Customers_CompWin
View Name:	Customers_CompView
Application Name:	SendCustomer_App

2. Create a Value attribute Va_CustomerId in component context and View context the establish context mapping between them.
3. Create UI elements Label, Input field and Button in the Customers View. Create an action with the name and bind the action to Button UI element.
4. Bind the context attribute Va_CustomerId to the Input field.



5. Implement action handler method for the action in order to call the Web Dynpro for ABAP Page using with the Dynamic parameter on click of the button. The navigation logic can be implemented using UME Navigation API.



Implementation of Action Handler Method:

```

public void onActionViewDetails(com.sap.tc.webdynpro.progmodel.api.IWDCustomEvent wdEvent )
{
    //@@begin onActionViewDetails(ServerEvent)
    //Retrieve Customer Id from context
    String customerId = wdContext.currentContextElement().getVa_CustomerId();
    if(customerId != null && !customerId.equals("")){ //Validate Input
        WDPortalNavigation.navigateAbsolute( //Call WD ABAP page using Navigation API
            "ROLES:PCDlocationOfWebDynproForAbapPage" + "?CUSTOMERID=" + customerId,
            WDPortalNavigationMode.SHOW_HEADERLESS_PORTAL, (String) null,
            (String) null,
            WDPortalNavigationHistoryMode.NO_DUPLICATIONS,
            (String) null, (String) null, " ");
    }
    //@@end
}

```

Note: PCD location of the Web Dynpro ABAP application can be retrieved from Location property of the page to which the WD ABAP iView is assigned in content admin.

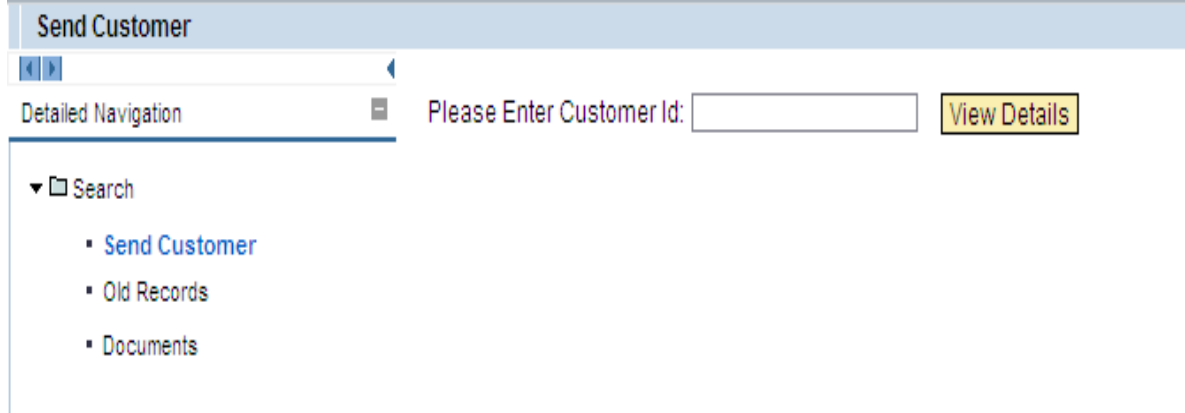
6. Deploy and Run Application.

Creation of WD Java Page in Portal:

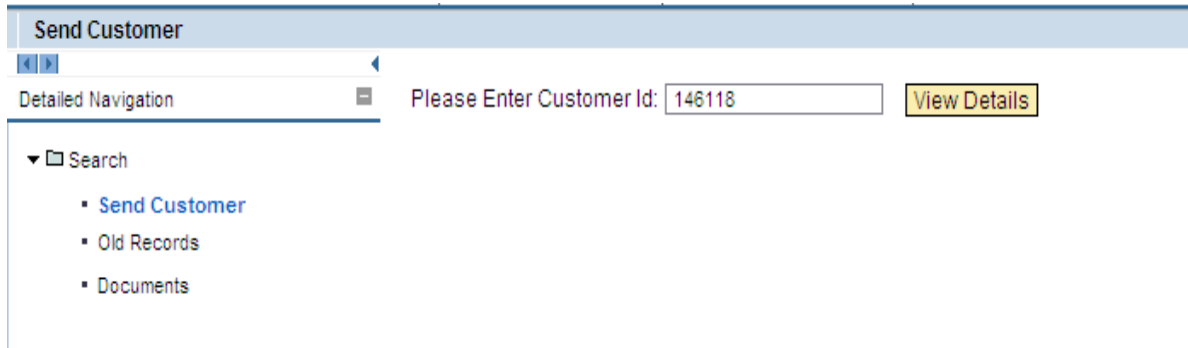
Content Administration → Portal Content → Expand Web Dynpro Applications → Navigate to the Deployed Web Dynpro Project → Expand and Select the WD application → Right Click and Copy → Navigate to your custom folder → Right Click → Select the option paste as PCD object → Enter required details and press Finish → Assign the newly created page to the Role.

Result:

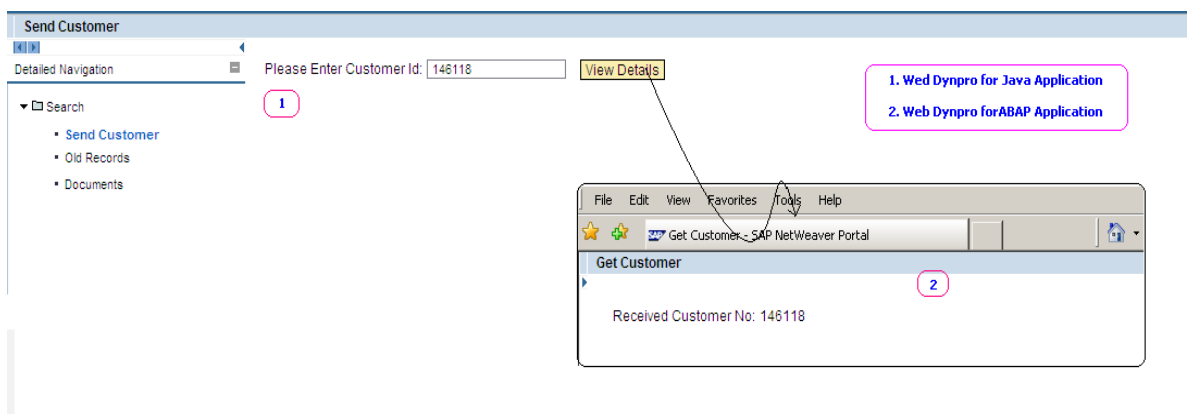
1. Login to Portal Navigate to the page which was assigned with Web Dynpro for Java iView.



2. Enter the Parameter to be passed (CustomerId) → Press the button “View Details”.



3. The button “View Details” opens Web Dynpro ABAP page in a new window with the parameter sent by Web Dynpro Java Application.



Related Content

[Creating a Simple Web Dynpro Java Application](#)

[Creating your first Web Dynpro Java Application](#)

[Other Web Dynpro references](#)

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

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.