

# Calling Custom Portal Application from XML Forms



## Applies to:

SAP Enterprise Portal. For more information, visit the [Portal and Collaboration homepage](#).

## Summary

This article would explain the method to call a custom portal application from xml form and transfer data to the application from the form.

**Author:** Mahesh Krishnapillai

**Company:** Larsen & Toubro Infotech Ltd.

**Created on:** 26 September 2008

## Author Bio



Mahesh Krishnapillai is a SAP certified Professional with more than 3 yrs. of experience as SAP NetWeaver-Portal and WebDynpro consultant.

## Table of Contents

Introduction .....	3
Creating a Custom Portal Component.....	3
XML Form Development.....	4
Related Content.....	8
Disclaimer and Liability Notice.....	9

## Introduction

With the launch of web page composer many corporate IT departments has started the process of converting there existing intranet web pages into SAP Enterprise Portal. This has led to extensive use of SAP XML Forms and portal component development to achieve a web look n feel for the portal. SAP xml forms would provide an easy content management option with minimal IT dependency for the business user. Custom developed portal components would serve to achieve some particular business functionality in the page.

In such a scenario wherein xml forms are used for content management, passing or reading data stored in xml forms through custom portal component becomes important. Either we can parse the xml file to read the data in portal component and achieve the business need or we can pass the parameter from xml form to the component.

## Creating a Custom Portal Component

Develop a custom portal component with the code to achieve the particular business condition. Let's think about a JSP Dynpage component to achieve the business condition. This component would get a parameter from xml form and execute the business logic.

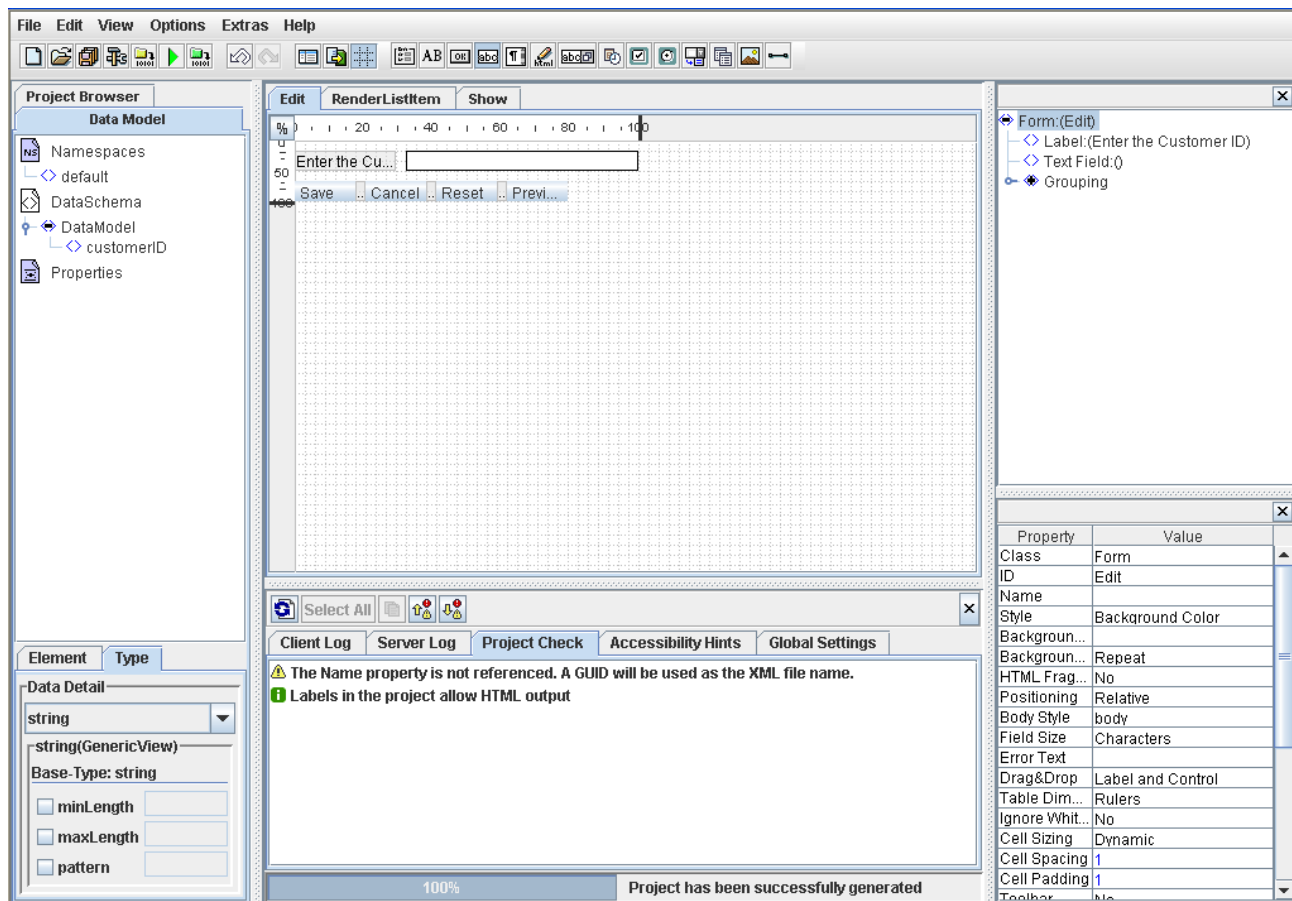
```
<%@ page import = "com.sapportals.portal.prt.component.*" %>
<%@ page import = "com.sapportals.htmlb.*" %>
<%
//Implement this code to read the parameter
com
    .sapportals
    .portal
    .prt
    .component
    .IPortalComponentResponse componentResponse =
    (IPortalComponentResponse) pageContext.getAttribute(
        javax.servlet.jsp.PageContext.RESPONSE);
    String customerID = request.getParameter("id");
%>
<%
//Code to display the paramter received from xml form
%>
<span>
<%=customerID%>
</span>
```

Here "**id**" is the parameter that is been passed to the portal component from the xml form. The above code reads the parameter "**id**" from the portal component request.

Create an iView form the above PAR and preview to get the url of the iView. Note the url part contain pcd location details "**pcd!3aportal\_content!2fcom.!2fsdn\_demo**".

## XML Form Development

Create a simple xml form with edit form having an input field for the business user to enter the data. The data here would be “customerID”. The logic is to pass this “customerID” to the custom portal application.



**Note:** Data Schema Element – customerID  
Type- String

Once the edit form is ready, create a show form with the text "View Customer ID" hyperlinked to Display.

Property	Value
Class	Label
Caption	View Customer ID
Hyperlink	Display
Style	Standard Text
Link Descri...	
Alignment	
Vertical Alig...	Top
Word Wrap	Allowed
Text Color	AUTO
Font	
Font Size	
Condition	
Font Decor...	
Display HT...	No
Empty	Empty Column
Schema R...	

Build the project and close the xml form builder. Once this is done navigate to the KM folder "/etc/xmlforms/<your form id folder>". Under this folder download the file "<your form id>Show.xml" to the local machine. Open the downloaded file using WordPad and find the section "MAIN ITEM TEMPLATE".

The xml structure under the section would be as shown below

```

- <!--
      MAIN ITEM TEMPLATE
-->
- <xsl:template name="item">
- <table width="100%" class="urGrpBdyBoxBgColor" cellpadding="1" cellspacing="1">
- <tr>
- <td height="7%" valign="top" width="24%" rowspan="1" colspan="1">
- <p>
- <a tabindex="0" class="urTxtStd">
- <xsl:if test="true">
- <xsl:attribute name="href">
  javascript:open_url_with_params('
  <xsl:value-of select="$previewervlet" />
  ?XMLFormID='+encodeXmlFormsUri('SDNDEMO')+'&show/test='+encodeXmlFormsUri('
  <xsl:value-of select="$dataurl" />
  ')+','width=465,height=273,toolbar=no,status=no,menubar=no,resizable=no,scrollbars=yes')
  </xsl:attribute>
  </xsl:if>
- <xsl:call-template name="outputvalue">
  <xsl:with-param select="$resources/TEXT[@AKEY='lnk_1222412022902']" name="value" />
  </xsl:call-template>
  </a>
  </p>
  </td>
  <td height="7%" valign="top" width="76%" />
</tr>
- <tr>
  <xsl:text disable-output-escaping="yes"><td height="93%" valign="top" width="24%" /><td height="93%"
  valign="top" width="76%" /></xsl:text>
</tr>
</table>
</xsl:template>

```

Change the xml data in between the tag as marked above with the below lines.

```

  javascript:open_url_with_params('pcd!3aportal_content!2fcom.!2fsdn_demo?
  id='+encodeXmlFormsUri('
  <xsl:value-of select="customerID" />
  ')+','width=465,height=273,toolbar=no,status=no,menubar=no,resizable=no,scrollbars=yes')
</xsl:attribute>

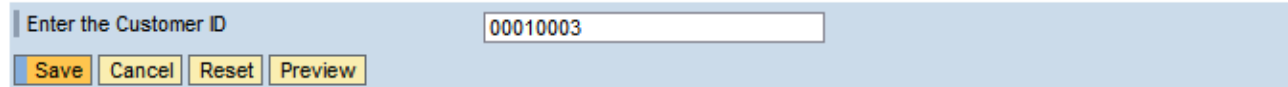
```

Now the open\_url\_with\_params function has the pcd location part earlier noted from the url of the iView attached to which is "id" the parameter which portal applications reads from its url.

Data schema element "customerID" which is passed from the xml form

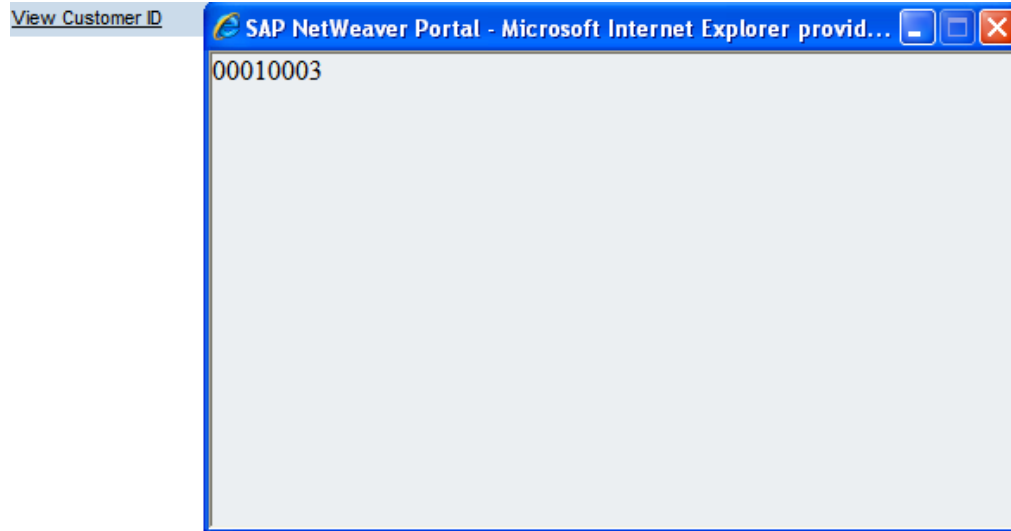
Upload the file from the local machine and replace the existing file. Once the changed XSL file is uploaded the form shouldn't be build again using the xml form builder. The above change will be overwritten once build the form from xml form builder. In that case we need to change the XSL file as mention earlier.

Now create an xml form using the above created xml project in some folder and enter the data.



A screenshot of a web form with a light blue background. At the top, it says "Enter the Customer ID". Below this is a text input field containing the value "00010003". At the bottom of the form, there are four buttons: "Save", "Cancel", "Reset", and "Preview".

View the show form from the folder in which the form was saved and click on the hyperlink text "View Customer ID". A pop up window would open showing the portal component displaying the customer id entered in the xml form as shown below.



As explained above, we could pass multiple parameters to a portal application and call multiple applications using different links in the xml form.

## Related Content

[A Beginner's Guide For: XML Forms Part 1](#)

[How to Create XML Forms \(NW2004\)](#)

[Easy Creation of Web-based Content Web-based Forms and the XML Forms Builder Exercise](#)

For more information, visit the [Portal and Collaboration 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.