

Using Radio Buttons in Web Template



Applies to:

SAP BW 3.5. For more information, visit the [Business Intelligence homepage](#).

Summary

One of the ideal requirements in the BW Web Reporting is the user wants certain set of reports to be clustered together and should be able to navigate to a particular report from the set of reports. Most of the users prefer Radio Buttons to achieve this purpose. In BW 3.5 this can be easily achieved by using simple JavaScript code. We can also pass some default values to the variables while selecting the report.

Author: Rakesh Pattani

Company: Patni Computer Systems Ltd

Created on: 8th May 2009

Author Bio



Rakesh Pattani is currently working in Patni Computers Ltd. He is working on SAP BW 3.5 and SAP BI 7.0 and is mainly involved in Development and Enhancement work.

Table of Contents

Pre Requisites	3
Creating Main Web Template	4
Executing the Report	5
How the Code works?.....	7
Conclusion	8
Related Content.....	9
Disclaimer and Liability Notice.....	10

Pre Requisites

You need two different Web Templates which runs different reports respectively. In the following example I have used two templates which execute **Marketing** and **Sales** Report respectively.

Technical name for the Web Template are as follows.

Marketing Report: Web_Temp_Markt

When you run the Marketing web template independently you get the variable screen as shown below.

Variables for Ad Hoc Report

Company (*)

Company Hierarchy

Calendar Day To

Category =

Sales Report: Web_Temp_Sales

When you run the Sales web template independently you get the variable screen as shown below.

Variables for Ad Hoc Report

Company (*)

Company Hierarchy

Employee Name

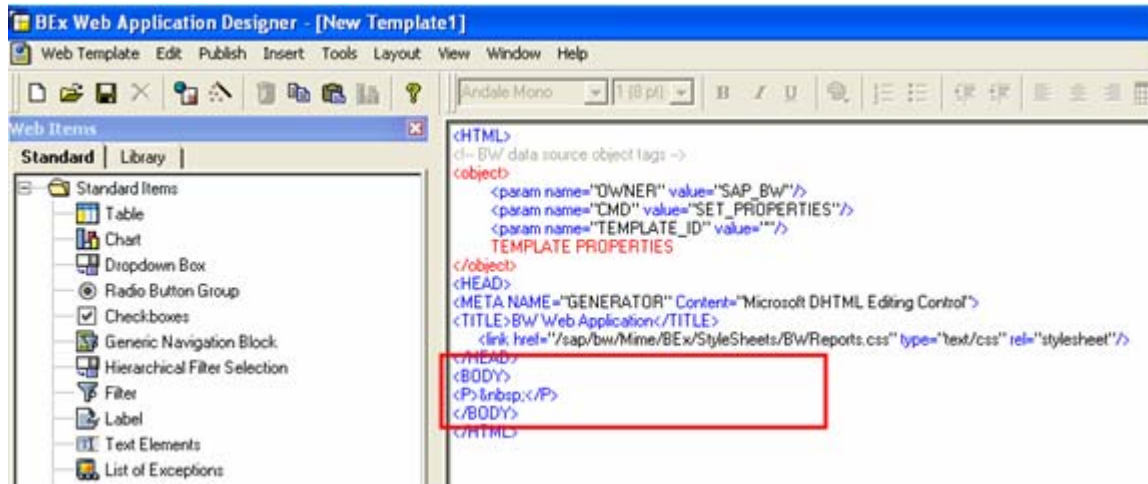
Employee Manager =

Calendar Day To

Category =

Creating Main Web Template

Open a new Web Template for creating the main report (**Category Report**) and goto the HTML tab of the web template as shown below.



Now between the BODY tags of the HTML code insert the following JavaScript code

```

<script language="JavaScript">
function fun(url)
{
  SAPBWOpenURL(SAP_BW_URL_Get() + url);
}

</script>
<P align=left><FONT face=Arial size=5><STRONG>Category Report</STRONG></FONT></P>
<form name="form1" >

<INPUT TYPE=RADIO NAME="Category"
onclick=fun("&CMD=LDOC&TEMPLATE_ID=Web_Temp_Sales&VAR_NAME_1=YUS_ATYP&VAR_VALUE
_LOW_EXT_1=SALES")>Sales Report<BR>

<INPUT TYPE=RADIO NAME="Category"
onclick=fun("&CMD=LDOC&TEMPLATE_ID=Web_Temp_Markt&VAR_NAME_1=YUS_ATYP&VAR_VALUE
_LOW_EXT_1=MARKETING")>Marketing Report<BR>

</form>

```

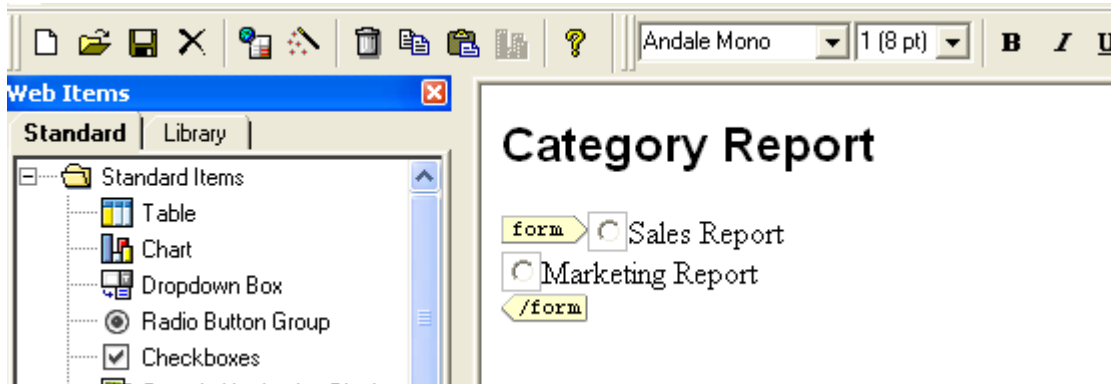
So you can see the code embedded between the BODY tags as shown below.

```

</HEAD>
<BODY>
<script language="JavaScript">
function fun(url)
{
  SAPBWOpenURL(SAP_BW_URL_Get() + url);
}
</script>
<P align=left><FONT face=Arial size=5><STRONG>Category Report</STRONG></FONT></P>
<form name="form1">
<INPUT TYPE=RADIO NAME="Category" onclick=fun"&CMD=LDOC&TEMPLATE_ID=Web_Temp_Sales&VAR_NAME_1=YUS_ATYP&VAR_VALUE_LOW_EXT_1=SALES">Sales Report<BR>
<INPUT TYPE=RADIO NAME="Category" onclick=fun"&CMD=LDOC&TEMPLATE_ID=Web_Temp_Mark&VAR_NAME_1=YUS_ATYP&VAR_VALUE_LOW_EXT_1=MARKETING">Marketing Report<BR>
</form>
</BODY>
</HTML>

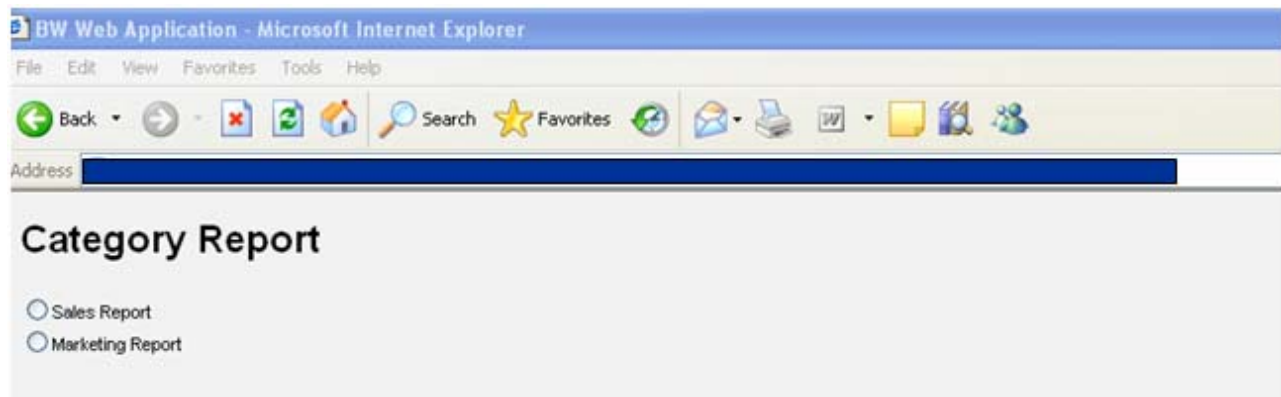
```

When you go to the Layout tab of the web template, you can see the Report name and the radio buttons as shown below.




Executing the Report

Now execute the Web Template and you can see the following screen appears.

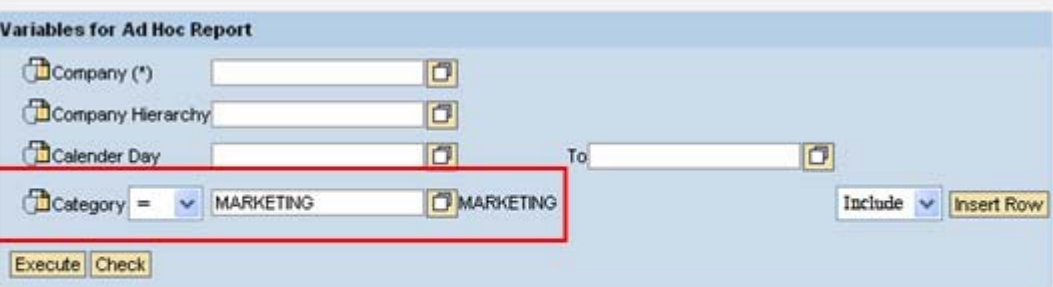


Now when you click on the **Sales Report Radio Button**, it will navigate you to the **Sales Web template**. So you can see the variable screen of the sales report and it will also fill the Category field with the value that was passed through the URL as shown below. **You can transfer any number of variable values through the URL.**



The screenshot shows a web form titled "Variables for Ad Hoc Report". It contains several input fields and buttons. The "Category" field is highlighted with a red box and contains the value "SALES". The "Include" dropdown is set to "Include" and the "Insert Row" button is visible. The "Execute" and "Check" buttons are at the bottom left.

If you click on the **Marketing Report Radio Button**, it will navigate you to the **Marketing Web template** and will display the screen as shown below. Here you can see the Category field is filled with the Marketing value as expected.



The screenshot shows the same "Variables for Ad Hoc Report" form, but the "Category" field is now highlighted with a red box and contains the value "MARKETING". The "Include" dropdown is set to "Include" and the "Insert Row" button is visible. The "Execute" and "Check" buttons are at the bottom left.

How the Code works?

```

<script language="JavaScript">
function fun(url)
{
  SAPBWOpenURL(SAP_BW_URL_Get() + url);
}
</script>
<P align=left><FONT face=Arial size=5><STRONG>Category Report</STRONG></FONT></P>
<form name="form1" >

<INPUT TYPE=RADIO NAME="Category"
onclick=fun("&CMD=LDOC&TEMPLATE_ID=Web_Temp_Sales&VAR_NAME_1=YUS_ATYP&VAR_VALUE
_LOW_EXT_1=SALES")>Sales Report<BR>

<INPUT TYPE=RADIO NAME="Category"
onclick=fun("&CMD=LDOC&TEMPLATE_ID=Web_Temp_Markt&VAR_NAME_1=YUS_ATYP&VAR_VALUE
_LOW_EXT_1=MARKETING")>Marketing Report<BR>
</form>

```

Function to open a
Web template

Displays Category
Report in the main report

Value for the
Category variable

Marketing Web
Template technical
name

Technical name of
the Category variable

Conclusion

Similarly you can include many reports in the main template and make them accessible via Radio buttons.

Following are the advantages of doing this.

- User has to run only one web template (the main template) and all others can be directly accessed from that template.
- User has to remember the name of only the main template
- Variable values can be directly passed to the sub templates making this more user friendly.

Related Content

[Using JavaScript Commands](#)

[Radio button Group](#)

For more information, visit the [Business Intelligence 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.