Web Application Designer for Beginners

Applies To:

Web Application Designer, SAP BW Version: 3.5, SAP GUI version 6.40/Patch Level 13

BW 3.5 front end patch 7 (bw350_7-10001615)

Summary

The document will run the readers through a step by step example of how to create a simple web template which contains multiple queries & navigation blocks. It has been written for users who have no prior knowledge or experience of the Web Application Designer & HTML. All the required HTML code for the example has also been provided.

Pointers to more detailed information have also been provided for readers who might want to explore advanced functionalities

By: Nikhil Chowdhary

Company: Infosys Technologies Limited

Date: 20 March 2006

Introduction:

The Web Application Designer (WAD) is the front end application provided by SAP to create and modify web templates in order to display your query on the web. The default template used for displaying queries on the web is 0ADHOC. You can set your own template to be used as the default template.

A web template is essentially HTML code which contains some special SAP BW objects & tags in order to display various web items provided by SAP. The BW specific objects are always in red color in the HTML code generated. The WAD provides drag & drop functionality for all standard web items and also allows the user to set the behaviour & properties of these web items through a graphical interface. There are three main windows in the WAD:

1) The Web Items window: This is at the top left hand corner of the WAD interface by default. It contains a list of the SAP provided web items ready fro use on a template.

A comprehensive description of all the web items & their uses & functionalities may be found at:

http://help.sap.com/saphelp_bw33/helpdata/en/83/54973910a9bc12e10000000a114084/content.htm

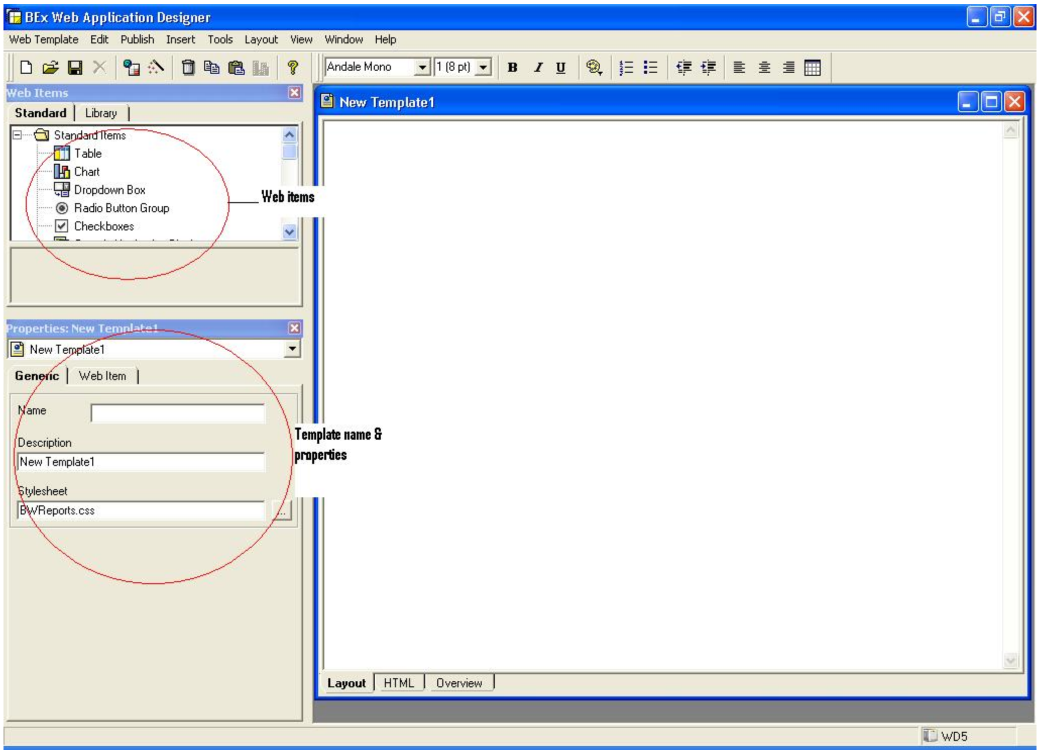
2) The Properties window: This is (by default) at the bottom left hand corner of the WAD interface (right below the web items window) and allows the user to set the properties of the selected web item. This has two tabs – “Generic” tab - to set the Data Provider (Query + Infoprovider) from which the web item will get its data and the “Web Item” tab to set the behaviour of the selected web item. This tab is itself divided into two main sections “Generic” properties – properties which are generic in nature and are available for all web items and “Specific” properties – properties which are specific to the selected web item type.

3) The template window: This is the workspace where the web template is created by dragging and dropping various web items from the web items window. This has three tabs shown at the lower left hand corner in the window: “Layout” – the layout designer, “HTML” – the page where the code is generated behind the scenes and can be modified for advanced functionality, and “Overview” – lists all the web items on the template in the order in which they appear.

Step by step solution (to create a template with two queries & two navigation blocks):

Let us run through the creation of a simple web application which contains just two queries displayed in two tables and a two navigational blocks to enable the end user to filter and drilldown:

Step 1: Open WAD and login to your account. This is what the WAD interface looks like:



Name

Description: New Template1

Stylesheet: BWReports.css

Step 2: Drag and drop the table web item onto the blank template. In the properties window change the query name to the desired name, also you may change the dataprovider name if desired – its advisable to use the dataprovider name “DP” since this is the name used by SAP provided sub templates which might be used later on. In the context of WAD, a dataprovider is a combination of a query and an InfoProvider.

Step 3: Drag & drop the Navigation block web item onto your template. This object would by default pick up the dataprovider that you had assigned in the previous step.

This essentially sets up the basic web application with one query & navigation block. You may stop here and save your template if this is all you need. You may click on the “Execute in the Browser” button (5th button from left in the toolbar) to see the result.

Step 4: You would want to change the default behaviour of the web items you have used, like you might want to hide the free characteristics section from the navigation block, or you might want the table item to show 30 records per page instead of the default 25. All these properties can be accessed by first clicking on the relevant web item in the template window & then changing the properties window tab to “web items” instead of “generic”. This would list down all the modifiable properties of the selected web item. The properties are divided into two sections – generic properties which are common for all the web items, and specific properties which are, obviously, specific to the selected web item. A description of the properties of the table & navigation block web items can be found at:

Table:

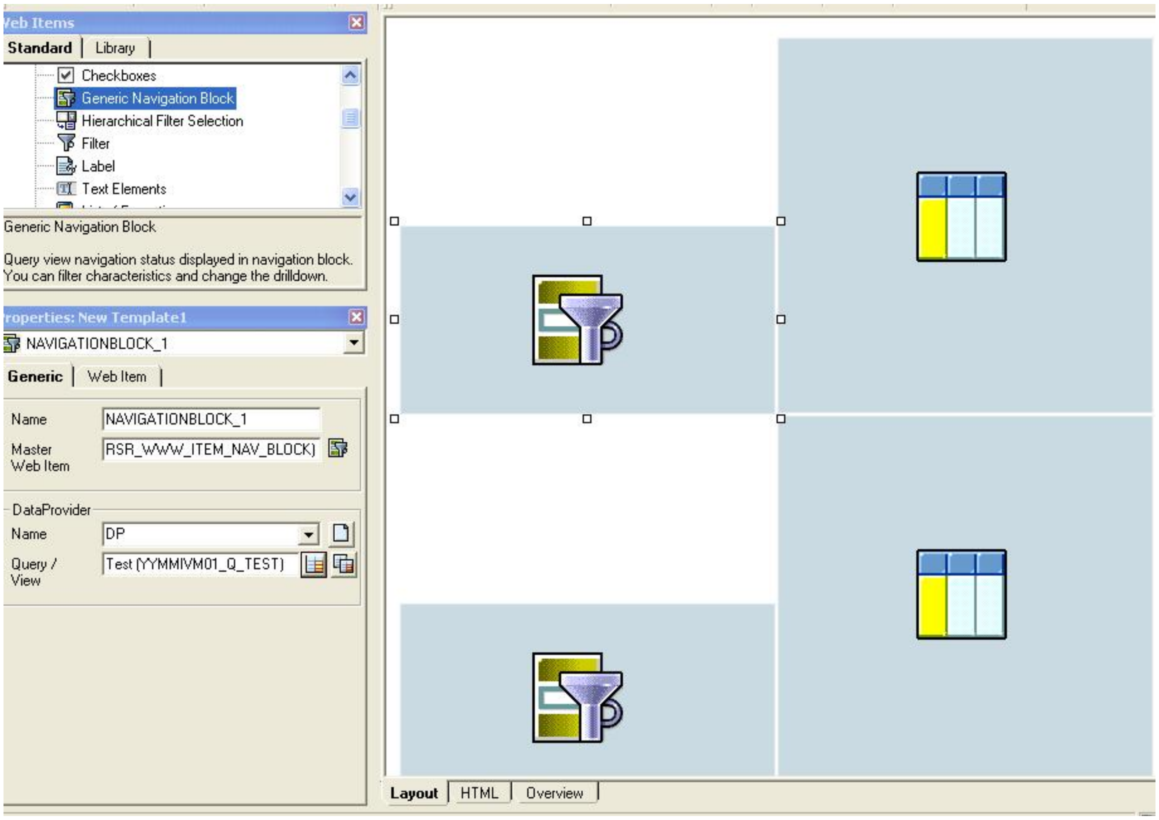
http://help.sap.com/saphelp_bw33/helpdata/en/76/489d39d342de00e10000000a11402f/content.htm

Navigation Block:

http://help.sap.com/saphelp_bw33/helpdata/en/1d/499d398b45160ae10000000a11402f/content.htm

Now suppose you need to add another table & navigation block to this template. This new table & navigation block use a different query than the one used earlier (you want to display two queries on the same web template).

Step 5: Drag & drop another table web item and another navigation block web item onto the template window. You template at this stage should look like this:



Generic Navigation Block

Query view navigation status displayed in navigation block. You can filter characteristics and change the drilldown.

NAVIGATIONBLOCK_1

Name: NAVIGATIONBLOCK_1

Master Web Item: RSR_WWW_ITEM_NAV_BLOCK)

DataProvider

Name: DP

Query / View: Test (YYMMIVM01_Q_TEST)

Step 6: Go to the HTML tab page on the web template window (lower left hand corner).

This would open up the HTML code which was so far generated behind the scenes by WAD. Towards the top of the code you will notice this code:

```
<!-- BW data source object tags -->

<object>

    <param name="OWNER" value="SAP_BW"/>

    <param name="CMD" value="SET_DATA_PROVIDER"/>

    <param name="NAME" value="DP"/>

    <param name="QUERY" value="YYMMIVM01_Q_TEST"/>

    <param name="INFOCUBE" value="ZMMIVM01"/>

    DATA_PROVIDER:          DP

</object>
```

You need to create another such segment which would allow you to declare the new data provider. Copy the entire code between this <object> & </object> and paste it below right this segment. Now change the web item name right before the </object> tag from DP to DP1. Also change the query name mentioned in the line <param name=”QUERY” value=”YYMMIVM01_Q_TEST”/> & the INFOCUBE name in the line <param name=”QUERY” value=”YYMMIVM01_Q_TEST”/>. So now your code would look like this:

```
<!-- BW data source object tags -->

<object>

      <param name="OWNER" value="SAP_BW"/>

      <param name="CMD" value="SET_DATA_PROVIDER"/>

      <param name="NAME" value="DP"/>

      <param name="QUERY" value="YYMMIVM01_Q_TEST"/>

      <param name="INFOCUBE" value="ZMMIVM01"/>

      DATA_PROVIDER:          DP

</object>

<object>

      <param name="OWNER" value="SAP_BW"/>

      <param name="CMD" value="SET_DATA_PROVIDER"/>

      <param name="NAME" value="DP"/>

      <param name="QUERY" value="ZZSDSRM01_Q_0025"/>

      <param name="INFOCUBE" value="ZSDSRM01"/>

      DATA_PROVIDER:          DP1

</object>
```

Step 7: Now scroll down to this block of code:

```
<object>

      <param name="OWNER" value="SAP_BW"/>

      <param name="CMD" value="GET_ITEM"/>

      <param name="NAME" value="NAVIGATIONBLOCK_2"/>

      <param name="ITEM_CLASS" value="CL_RSR_WWW_ITEM_NAV_BLOCK"/>
```

```
    <param name="DATA_PROVIDER" value="DP"/>
    ITEM:         NAVIGATIONBLOCK_2
</object><object>
    <param name="OWNER" value="SAP_BW"/>
    <param name="CMD" value="GET_ITEM"/>
    <param name="NAME" value="TABLE_1"/>
    <param name="ITEM_CLASS" value="CL_RSR_WWW_ITEM_GRID"/>
    <param name="DATA_PROVIDER" value="DP"/>
    ITEM:         TABLE_2
</object>
```

These two objects are your new table and your new navigation block. You would need to change the tag <param name="DATA_PROVIDER" value="DP"/> in both these objects to DP1 (to assign these to your new dataprovider.

This completes your web template with two queries and two navigation blocks.

Now go back to the layout tab. The system would ask you whether the code should be verified, click yes. Save your template with the desired technical name & description.

Additional information:

There are essentially three ways to start working on web templates in the Web Application Designer (WAD):

Make a new template from scratch

Would be useful in case the required web application is simple - maybe using just a couple of Navigation blocks & Tables

Start from one of the standard templates

0Query_Template: This will reduce the amount of work required to make a web template from scratch if you are required to bring in functionality like icon based buttons for performing activities such as graph generation, buttons for export to EXCEL/CSV, button to show the variable screen etc.

0ADHOC. This will allow you to use a tabbed screen to separate logical segments like tables, graphs, information on the query & query broadcast while simultaneously providing you with all the standard functionality.

The first method is the one we used in the example.

I would suggest using the last method since it will essentially provide you with all the generic functionality which you might require for your template. You can just delete the code for whatever functionality is not required and save it with the desired name. But this would require some understanding of HTML & the SAP tags used for web templates.

In case a standard template is being used (either 0QUERY_TEMPLATE or 0ADHOC, take care to first save the template by another name (e.g. ZADHOC) before making any modifications so that the original template is available for reuse if something goes wrong, or if another format template is required to be built.

Before using the 0QUERY_TEMPLATE you would require the following modifications, since this template is not meant to be used as a standalone, but rather as a subtemplate called from the mail template:

When you open the 0QUERY_TEMPLATE it will generate an error saying:

“’ITEM_ID’ for library item or ‘ITEM_CLASS’ parameter for standard item is missing”

To get rid of this error change the mentioned object code (on the HTML tab page) to:

```
<object>

    <param name="OWNER" value="SAP_BW"/>

    <param name="CMD" value="GET_ITEM"/>

    <param name="NAME" value="GR2Chart"/>

    <param name="ITEM_ID" value="0ADHOC_COLUMN_CHART"/>

    <param name="DATA_PROVIDER" value="DP"/>

    <param name="HIDDEN" value="X"/>

    ITEM:       GR2Chart

</object>
```

Also, in order to use this query as a standalone template, you would need to delete the following lines which appear right at the top of the template:

```
<!-- Set the data provider --->
<!--specialbwcomment<OBJECT>
<PARAM NAME="OWNER" VALUE="SAP_BW"></PARAM>
<PARAM NAME="CMD" VALUE="SET_DATA_PROVIDER"></PARAM>
<PARAM NAME="NAME" VALUE="DP"></PARAM>
DATA_PROVIDER: DataProvider
</OBJECT>specialbwcomment-->
```

The above mentioned code segment is used to inherit the value of the dataprovider from the calling template. But, since we are using this template as a standalone template, we need to delete these lines.

Now save this as ZQUERY_TEMPLATE and use it for further modifications as required

The errors might get corrected in future versions.

Author Bio

Nikhil Chowdhary is a SAP Certified Solution Consultant for Netweaver 2004 BI and is currently working as an Associate Consultant for Infosys Technologies Limited, India.

Disclaimer & 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.