Create Interesting Visualizations in Web Intelligence Report using HTML Components

Applies to:

BusinessObjects Web Intelligence XI Release 2, XI R3 For more information, visit the Business Objects homepage.

Summary

This document provides a guide for how to embed HTML components in a Web Intelligence Report. It also provides example of creating some interesting visualizations in a Web Intelligence Report using HTML. Limitations of the process are briefly mentioned.

Author: Arijit Das

Company: Tata Consultancy Services Ltd.

Created on: 2 April 2010

Author Bio

The author has been associated with Tata Consultancy Services Ltd. for 2.5 years and has worked in Telecom and Utilities domain as a BusinessObjects Consultant. He has mainly worked in development of Universe and Web Intelligence Reports in BusinessObjects Enterprise XI Release 2 and BusinessObjects Enterprise XI R3.

Table of Contents

Introduction

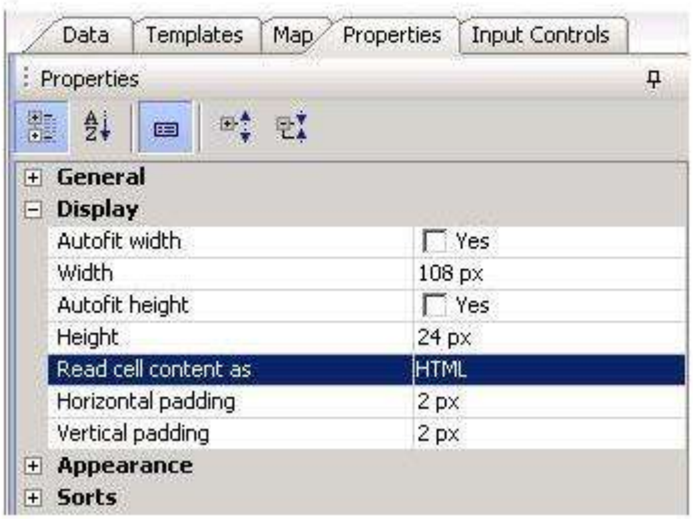
The purpose of a web browser is to read HTML documents and display them as web pages. When we use Web Intelligence as a web based application, embedding HTML contents in the report provides a good way to enhance the reporting capability of BusinessObjects Web Intelligence (WebI).

The purpose of this document is to show how to add HTML components in a WebI report using some simple examples of HTML components to enhance reporting in Web Intelligence.

Adding HTML Components in WebI Report

To use HTML components within a WebI report, follow the steps below:

1. In the Edit Report mode, add the HTML code in the formula editor for a cell.
2. In the Display property of the cell set the **Read cell content as** option to **HTML** as shown below:



Autofit width	☐ Yes
Width	108 px
Autofit height	☐ Yes
Height	24 px
Read cell content as	HTML
Horizontal padding	2 px
Vertical padding	2 px

3. You can add HTML code in a table cell(s) as well as in a free-standing cell.

Note: You can build HTML tagged objects in the universe as well. But while using that in WebI, in the display property of the cell using the object you must select Read Cell Content as HTML.

Note: In the Edit mode, only the HTML code will be visible inside the cell. In the view mode, the code will be translated to web content.

If browser setting allows, JavaScript and CSS can also be used to enhance HTML functionalities.

Examples of HTML components in WebI

This section shows some simple examples of using HTML components to enhance reporting in WebI.

Add a marquee in WebI Report

Requirement

Add a marquee in a WebI report. There will be a scrolling message inside the report. The message can be text or image.

Solution

To add marquee in a WebI Report with a simple text, follow the steps mentioned below:

1. Drag a blank cell in the report.
2. Enter the code below in the **Formula Editor** for that cell.

```
="<html><marquee direction=right
onmouseover=this.stop();onmouseout=this.start();>This is a marquee</marquee></html>"
```

3. Select the cell and go to **Properties** tab.
4. In **Display** section, select **HTML** in **Read cell content as** field.
5. Adjust the cell height and width to define the scroll area.
6. Save the report.

This will show a scrolling text “This is a marquee” in the defined scroll area within the WebI report when viewed in HTML report panel.

Customization

Create following variables in WebI:

Variable	Purpose	Allowed Values	Variable Type	Example
Content	Define the content to be scrolled.	Free form text	String	=“This is a Marquee”
Direction	Define the direction of scroll.	LEFT | RIGHT | UP | DOWN	String	=“RIGHT”
Behavior	Define the scroll behavior.	SCROLL | SLIDE | ALTERNATE	String	=“ALTERNATE”
BkgColor	Define the background color of scroll area.	Color value	String	=“#FAFAFA”
ScrollSpeed	Define the scroll speed.	Enter an integer. Higher value indicates higher speed. For better visibility, use a value between 5 and 20.	String	=“8”

In the cell defining the scroll area, give the formula as below:

```
="<html><MARQUEE direction="+[Direction]+" BEHAVIOR="+[Behavior]+"
BGCOLOR="+[BkgColor]+" SCROLLAMOUNT="+[ScrollSpeed]+" ONMOUSEOVER=this.stop();
ONMOUSEOUT=this.start();>"+[Content]+"</MARQUEE></html>"
```

If you want a scrolling image, define the variable Content with <IMG> tag, e.g. =”<img src=URL />”.

Add Bar Chart inside table cells

Requirement

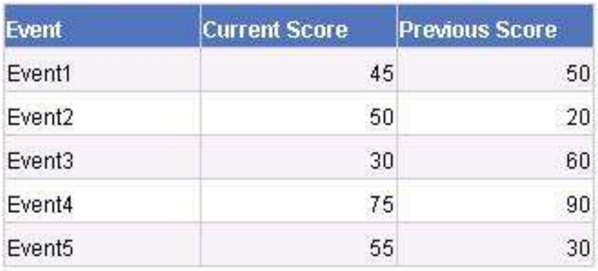
For simplicity, assume a requirement as below:

- A view having columns for Event, Current Score, and Score of Previous attempt.
- Show a table in the report with columns Event and Comparison. In the Comparison column, we have to show a horizontal bar chart comparing current and previous score.

Solution

The idea is to use a HTML table with background color where the width of the area filled by the color is dynamically varied by the score values.

Suppose, the data in the table is like below:



Event	Current Score	Previous Score
Event1	45	50
Event2	50	20
Event3	30	60
Event4	75	90
Event5	55	30

To achieve the requirement, follow the steps below:

1. Drag the following objects in the query:

   Event (dimension), Current Score (measure), Previous Score (measure)

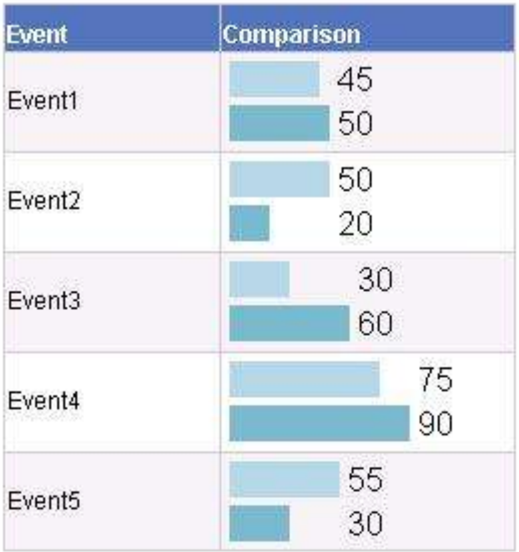
2. In the report, define a Vertical Table with two columns: one for Event and another for the comparison.
3. In the Event column, define the formula as =[Event] and in the Comparison column, define the formula as below:

```
="<html><table><tr><td><div style='background-color:#b5d7e7; width:"+[Current
Score]+";'/></td><td>"+[Current Score]+"</td></tr><tr><td><div style='background-
color:#7bbace; width:"+[Previous Score]+";'/><td>"+[Previous
Score]+"</td></td></tr></table>"
```

4. In **Display** property for the cells of Comparison column, select **HTML** in **Read cell content as** field.
5. Adjust the cell height and width to fit the chart.
6. Save the report.

Define two variables as percentage values of ([Current Score]+[Previous Score]) and use them in place of [Current Score] and [Previous Score] in the comparison formula so that the width of the column can be fixed.

In the HTML viewer, the report will look like below:



Event	Comparison
Event1	45 50
Event2	50 20
Event3	30 60
Event4	75 90
Event5	55 30

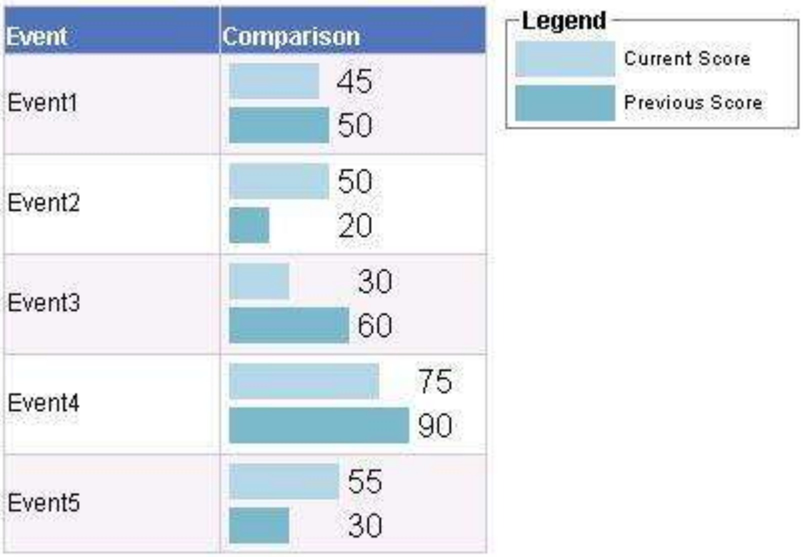
Then let us add one legend for the colors used.

1. Add one free-standing cell in the report.
2. In the **Formula Editor** of the cell, give the formula as below:

```
="<html><fieldset><legend>Legend</legend><table><tr><td><div style='background-
color:#b5d7e7;width:50'/></td><td>Current Score</td></tr><tr><td><div
style='background-color:#7bbace;width:50'/><td>Previous
Score</td></td></tr></table></fieldset></html>”
```

3. In **Display** property for the cell, select **HTML** in **Read cell content as field**.

In the HTML viewer, the report will look like below now:



Event	Comparison
Event1	45 50
Event2	50 20
Event3	30 60
Event4	75 90
Event5	55 30

Legend

🟦	Current Score
🟦	Previous Score

Add a tooltip text

Requirement

To add a tooltip text over a link or image within WebI report.

Solution

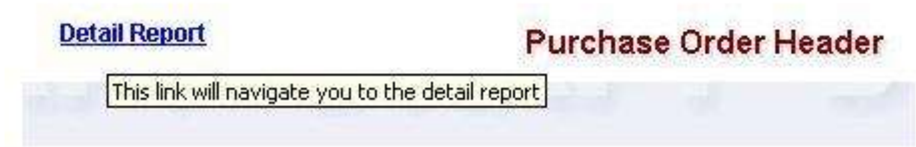
To add a tooltip text over a link within a WebI report, follow the steps below:

1. Define the link in HTML syntax. For example:

```
="<a title=”+Char(34)+”This link will navigate you to the detail report”+Char(34)+”
href=http://anvm21:8080/OpenDocument/opendoc/openDocument.jsp?iDocID=11004&sType=wid>
Detail Report</a>"
```

2. In **Display** property for the cell, select **HTML** in **Read cell content as field**.

When mouse is hovered over the link, the tooltip will be visible as shown below:



Detail Report

Purchase Order Header

This link will navigate you to the detail report

To add a tooltip text over an image within a WebI report, use the alt attribute in <IMG> tag. For example,

```
=”<img src=URL alt=Tooltip/>”
```

Limitations

The limitations of using HTML components within WebI reports are:

- The HTML components are visible only in HTML report panel.
- When saved as an Excel or PDF, the report will show the HTML code instead of the content.
- Use of HTML functionalities depends on the browser support.

Related Content

To know about Web Intelligence visit here.

To know about HTML visit here.

To know about HTML charts visit here.

For more information, visit the Business Objects 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.