How-To Freeze Lead Columns and Header Rows in a BPS Web Application Layout

Author: Cynara Kidwell Email: cynarasap@yahoo.com

1) Business Scenario

This document contains information on how to freeze lead columns and header rows for an SEM-BPS ALV layout in a SEM-BPS web interface. This is accomplished by making changes to the generated web interface BSP (Business Server Page). This means that this solution will need to be re-implemented every time the web interface is re-generated. This solution is generic for all layouts, which means it will freeze the lead columns and headers for all layouts on a given page.

To implement this solution, it will be useful to have basic knowledge of html, CSS and ABAP.

This solution does not cover freezing of lead columns and header rows using an excel web layout. This functionality is delivered standard with BW-BPS 3.5.

2) System Requirements

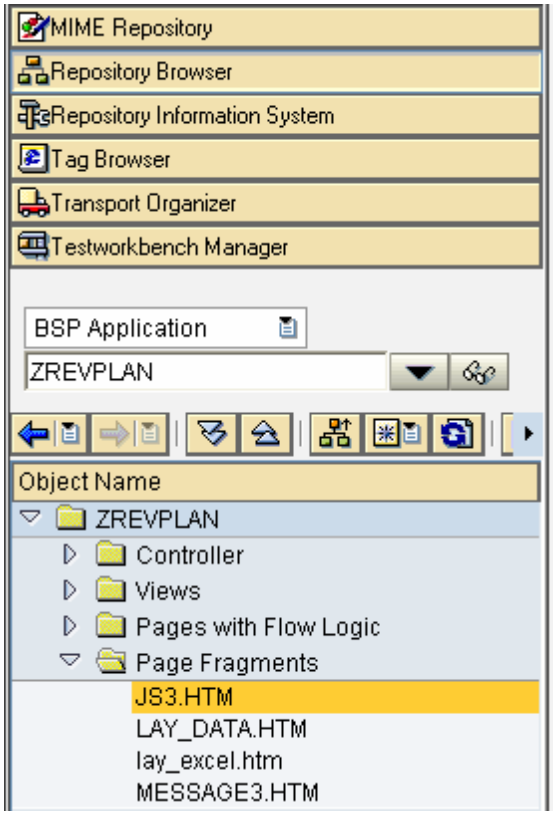
This solution has been tested on BW-BPS 3.5, though it theoretically should work on old versions of BW-BPS as well. The end users must be using Internet Explorer 5.0 or above.

3) Step – By – Step Solution

a) Add the custom java script functions to recalculate lead column and header row positions.

i) Open the java script page for the web interface.

In the repository browser (SE80), open the web interface you would like to modify, find the html fragment page for java script functions, under the folder “Page Fragments”. In BW-BPS version 3.5 and above, this file will be called “JS3.htm”. In past versions of BW-BPS, this file was called “JS.htm”.



ii) Insert the column and row repositioning functions in the java script file.

At the end of the javascript file, **before** the end script tag (</SCRIPT>), add the following code snippet:

```
// ***************************** custom functions
//APPLICATION SPECIFIC VARIABLES FOR TABLE LEAD COLUMN FREEZING
//the num. of header rows to freeze (change to meet your needs)
var lrows = 2;
//the num. lead column to freeze (change to meet your needs)
var lcol = 2;

//Table Height screen percentage
var LMaxHeightPerc = .80;

//Table Width screen percentage
var LMaxWidthPerc = .94;

//*******Other Global Variables
var divcont; //the div object containing the table w. column
freeze
var first_run = true; //flag to indicate if the func. has been run
var laytable; //the table for column and header freezing
```

```
//********Function SetStyles*****************/
//Set the css style for the table to
// freeze header rows and lead columns
  function SetStyles()
  {
   //find the table to be manipulated
     divcont = document.getElementById("tblcont");
   //find the table in the document
     if(divcont != null){
     laytable = divcont.children(0);
     if( laytable != null){
     var docbody = window.top.document.body;

  //calculate the desired table heigh
     LMaxHeight = docbody.clientHeight * LMaxHeightPerc;
   if(laytable.clientHeight > LMaxHeight && LMaxHeight > 1){
      divcont.style.height = (LMaxHeight + "px");
      divcont.style.overflowY  = "scroll";
   }

  //calculate the desired table width
     LMaxWidth = docbody.clientWidth * LMaxWidthPerc;
     if(laytable.clientWidth > LMaxWidth && LMaxWidth > 1){
      divcont.style.width  = (LMaxWidth + "px");
      divcont.style.overflowX  = "scroll";
  }

//specify table parameters
         laytable.style.zIndex = 5;
         laytable.style.position = "relative";

//specify locked columns
if( lrows < laytable.rows.length){
    for (i=lrows; i < laytable.rows.length; i++){
        for (j=0; j < lcol; j++) {
       if( j >= laytable.rows(i).cells.length) break;
         laytable.rows(i).cells(j).style.position = "relative";
         laytable.rows(i).cells(j).style.zIndex = 10;
        }
     }
}

//specify locked rows
if( lcol < laytable.rows(0).cells.length){
    for (i=0; i < lrows; i++) {
        if( i >= laytable.rows.length) break;
      for (j=lcol; j < laytable.rows(i).cells.length; j++) {
         laytable.rows(i).cells(j).style.position = "relative";
         laytable.rows(i).cells(j).style.zIndex = 20;
       }
    }
}

//Give special priority to cells in the upper right-hand corner
    for (i=0; i < lrows; i++) {
        if( i >= laytable.rows.length) break;
      for (j=0; j < lcol; j++) {
         if( j >= laytable.rows(i).cells.length) break;
           laytable.rows(i).cells(j).style.position = "relative";
         laytable.rows(i).cells(j).style.zIndex = 30;
       }
```

```
        }
      }
      }
    } //end  function SetStyles

//***********SetLengths function *************************/
//Set the positions of the lead columns
//and header rows after a scroll event
  function SetLengths()
  {
//adjust lead column
    for (i=0; i < laytable.rows.length; i++) {
        for (j=0; j < lcol; j++) {
          if( j >= laytable.rows(i).cells.length) break;
          laytable.rows(i).cells(j).style.left =
divcont.scrollLeft;
        }
       }

//adjust header row
    for (i=0; i < lrows; i++) {
          if( i >= laytable.rows.length) break;
      for (j=0; j < laytable.rows(i).cells.length; j++) {
         laytable.rows(i).cells(j).style.top = divcont.scrollTop;
       }
    }
  } // end function SetLengths
-----------------------------------------------------------------------
```

Note: Adjust the values of variables lcol and lrows, to match the number of lead column or header rows you wish to freeze. This will correspond to the number of lead columns and header in your layout. Adjust the value of the javascript variable LMaxHeightPerc to specify the desired maximum height percentage of window(in decimals). Adjust the value of the javascript variable LMaxWidthPerc to the desired maximum width as percentage of the window (in decimals).

Deleted: *LMaxHeight*

Deleted: *pixels).*

Deleted: *LMaxWidth*

Deleted: *pixels*

iii) Change the handleLoad function to include call to custom javascript function

In the same page fragment, find the javascript function called “handleLoad”, at the end of this function, before the closing bracket (“}”) insert a call to the function SetStyles.

Example:

```
-------------------------------------------------------------
  function handleLoad(statefull, title)
  {
    try {
      parent.handleChildLoaded(statefull == 'X', title);
    } catch(e) {}
    SetStyles();
  }
-------------------------------------------------------
```

iv) Activate the Changes

Click on the activation button  to activate the changes to the page.

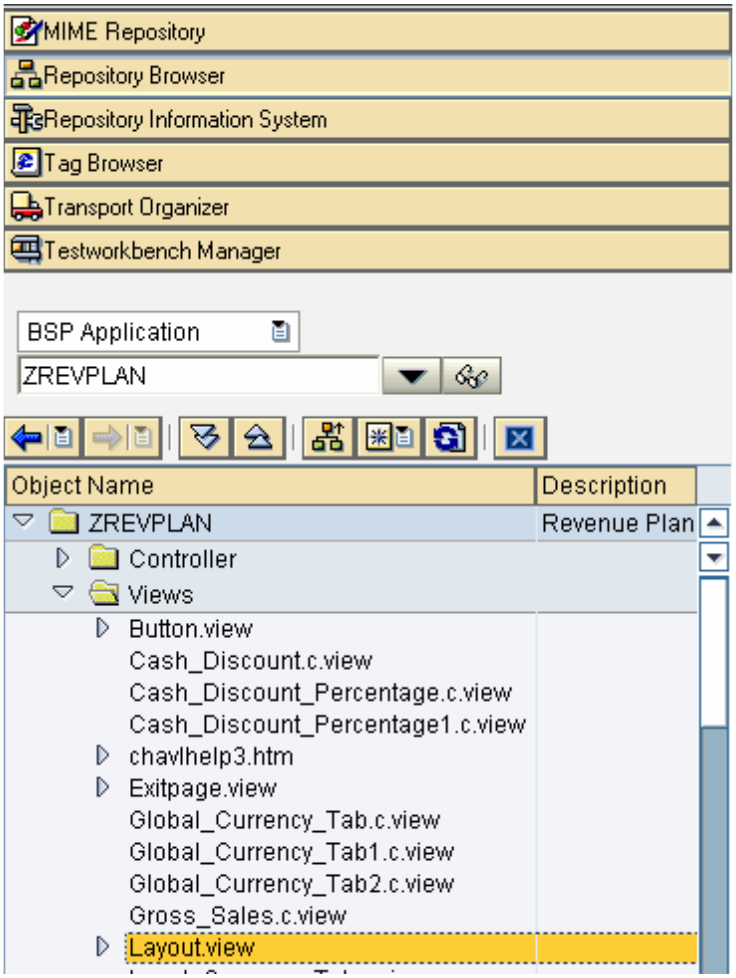
HINT: The javascript page will not change during customizations. This means that after implementing the solution, you may download a copy of the altered javascript file to your desktop, and re-upload it to the server every time the BSP page is re-generated to avoid typographical errors.

b) Add Division HTML Element to Layout view.

Layout.view is the generated BSP where the html for the table layouts are generated. This layout needs to be modified to include a division html element ("DIV") with scroll bars for scrolling when the columns are freezed.

i) Open the Layout.view for the web interface.

In the repository browser (SE80), open the web interface you would like to modify, find the Layout.view file, under the folder "Views".

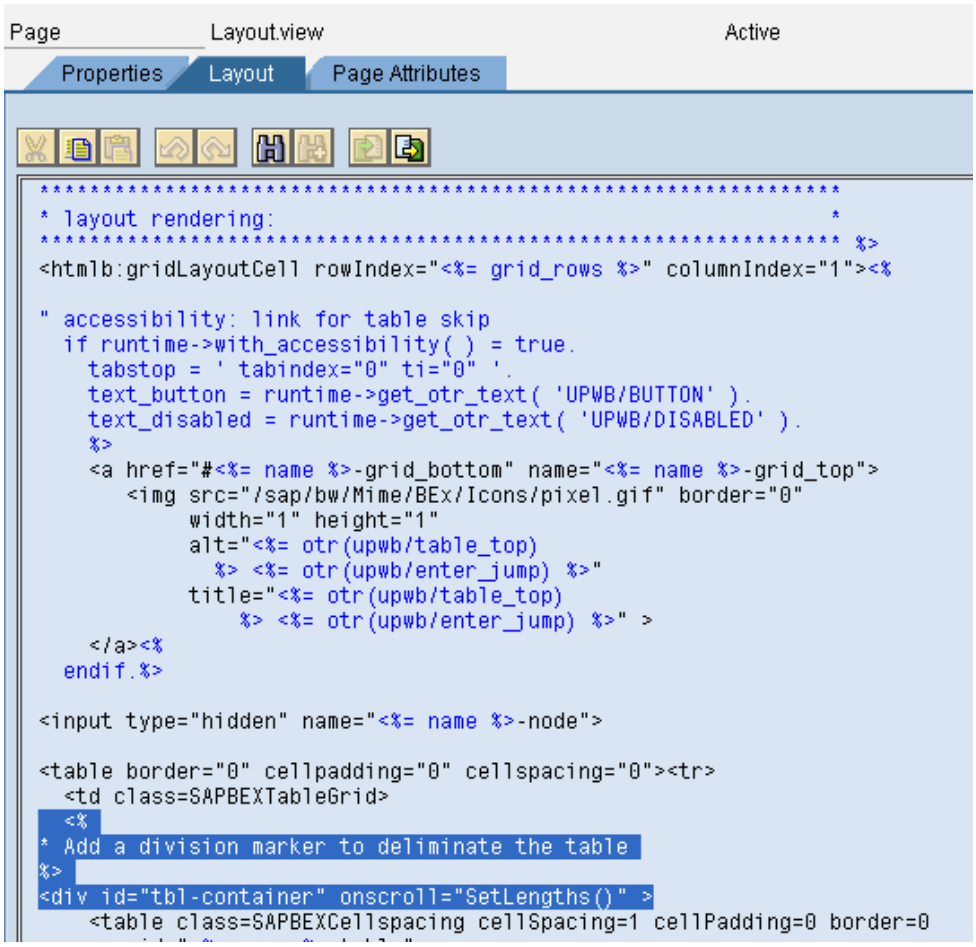


ii) Insert the opening division (<DIV…>) tag.

Under the heading “Layout Rendering” there are two open table tags (“<TABLE…..>). Before the second open table tag (the one with the class “SAPBEXCellspacing”), please insert the following code:

```
------------------------------------------------------------------
  <%
* Add a division marker to deliminate the table
%>
<div id="tbl-container" onscroll="SetLengths()" >
------------------------------------------------------------
```

Example:

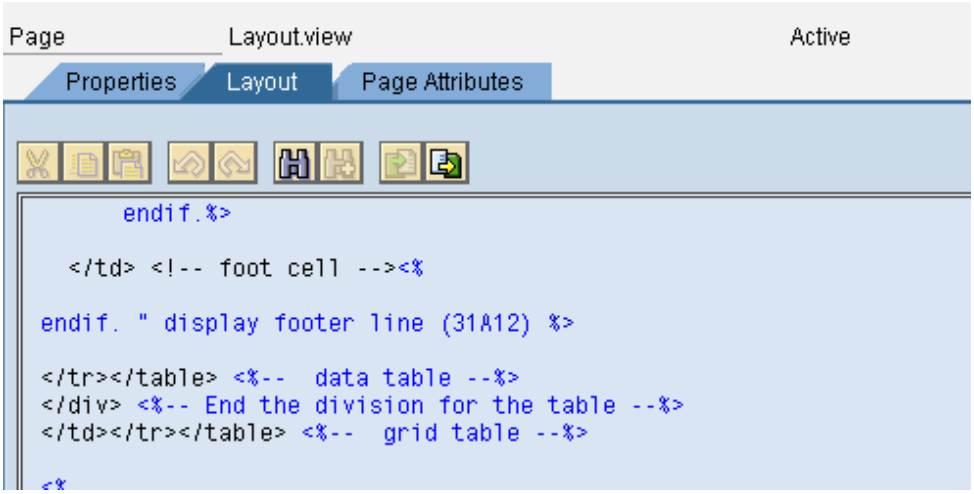


iii) Insert the closing division (<DIV…>) tag.

Under the heading “footer line” there are two closing table tags (“</TABLE>). Before the second close table tag, (commented as “grid table”), please insert the following code, this will specify that the division html tag is around only the data table.

```
</div> <%-- End the division for the table --%>
```

Example:



iv) Activate the Changes

Click on the activation button  to activate the changes to the page.

HINT: The JS3.htm and the Layout view will not change during customization. This means that after implementing the solution, you may download a copy of the altered Layout.view and JS3.htm files to your desktop, and re-upload them to the server every time the BSP page is re-generated to avoid typographical errors.