

Hiding Columns in BPS Web Interfaces



Applies to:

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

Summary

Users normally use web interfaces for the planning applications. Sometimes there are certain business scenarios or backend technical requirement which leads to an unnecessary column in planning interfaces. This creates a need to hide those columns in the planning web interface when finally displayed to users. This document highlights how a certain column can be hidden using BSP code.

Author: Satyashilpa Bowalekar & Samruddhi Gadwe

Company: L&T Infotech Ltd

Created on: 3rd July 2009

Author Bio

Satyashilpa Bowalekar is SAP Certified in 'Solution Consultant SAP NetWeaver 2004 – Business Intelligence' with almost six years of experience SAP BI. During this tenure, she has worked on BPS/BI development and Support projects.

Samruddhi Gadwe has an experience of almost three years in SAP BI. She has been a part on production Support and development. She has also worked extensively on BPS projects

Table of Contents

Background.....	3
Closing Period Significance	3
Scenario.....	3
Step Wise Solution	5
Identifying BSP Application:.....	5
Locating the code where layout is created.....	5
Code Changes and Logic	5
Points to be taken care of:.....	8
Related Content.....	9
Copyright.....	9

Background

There are business scenarios where we need to hide few columns in the layout displayed in the web interface. In such cases, we can modify the BSP code and achieve the hiding of columns.

We will explain the same with a scenario we encountered pertaining to the closing period defined by the client.

In our application we had a layout which would always display columns for 12 months divided into actual and plan (based on close period). The actual data would be displayed in columns till the closing period. The plan data would be displayed after closing period

Eg. If closing period is July 2009, actual will be seen from Jan to July; plan will be seen from Aug to Dec

Closing Period Significance

On a particular day (say 8th cal day) in the current month, the closing period would be changed to previous month. So for all days before 8th cal day it would be two months prior to current month.

E.g.: If the current month is July 2009,

From 1st to 8th July 2009, the closing period would be May 2009.

From 8th July onwards, the closing period would be June 2009.

(I.e transactions for month of May would take place only till 8th July)

Scenario

Due to the business logic explained above, our layout would look like this:

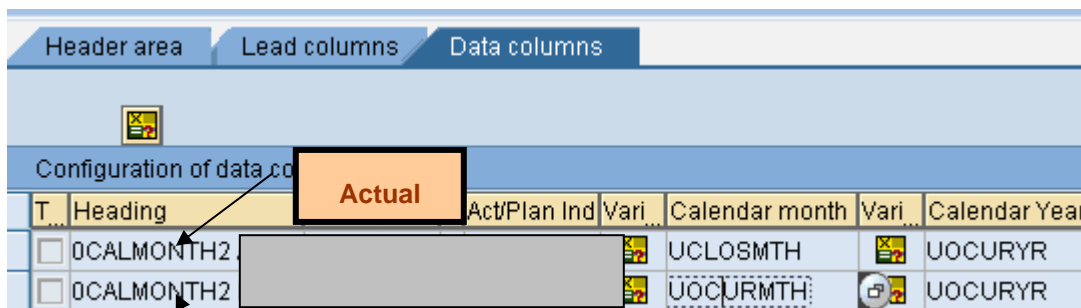
Case 1: Closing Period is Jul 2009, Current month: Aug 2009

Year 2009														
Customer	Country	Jan Act	Feb Act	Mar Act	Apr Act	May Act	Jun Act	Jul Act	Aug Pln	Sep Pln	Oct Pln	Nov Pln	Dec Pln	
C1	Customer 1	India	56	23	54	45	67	82	34	67	90	85	80	78

Case 2: Closing Period is Dec 2008, Current month: Jan 2009

Year 2009														
Customer	Country	Jan Pln	Feb Pln	Mar Pln	Apr Pln	May Pln	Jun Pln	Jul Pln	Aug Pln	Sep Pln	Oct Pln	Nov Pln	Dec Pln	
C1	Customer 1	India	56	23	54	45	67	82	34	67	90	85	80	78

In BPS0, the layouts were defined in following way:



We had two columns for Actuals and other for the Plans. The variables used along with the logic of population are me

UCLOSMTH :

Determine the range of months for Actual data based on closing period. This would normally be Jan to the closing period.

Eg. If closing period is 07, Range for actual would be 01 to 07.

In case the closing period is 12, we did not want range to be 01 to 12 as ideally we do not need actual. We just need the plan data, so the logic was such that the range would be 00 to 00.

UOCURMTH :

Determine the range of months for Plan data based on closing period. This would normally be closing period to Dec.

Eg. If closing period is 07, Range for actual would be 08 to 12.

UOCURYR :

Determine the year of the month next to closing period

The scenario in Case 2 would only arise when closing period = 12 and current month = 01.

Due to case 2, the logic to get the actual column would generate month 0 (explained above in logic for population of variables). The layout would be displayed as follows:

Year 2009														
Customer	Country	Not Assigned Act	Jan Pln	Feb Pln	Mar Pln	Apr Pln	May Pln	Jun Pln	Jul Pln	Aug Pln	Sep Pln	Oct Pln	Nov Pln	Dec Pln
C1	Customer 1	India	56	23	54	45	67	82	34	67	90	85	80	78

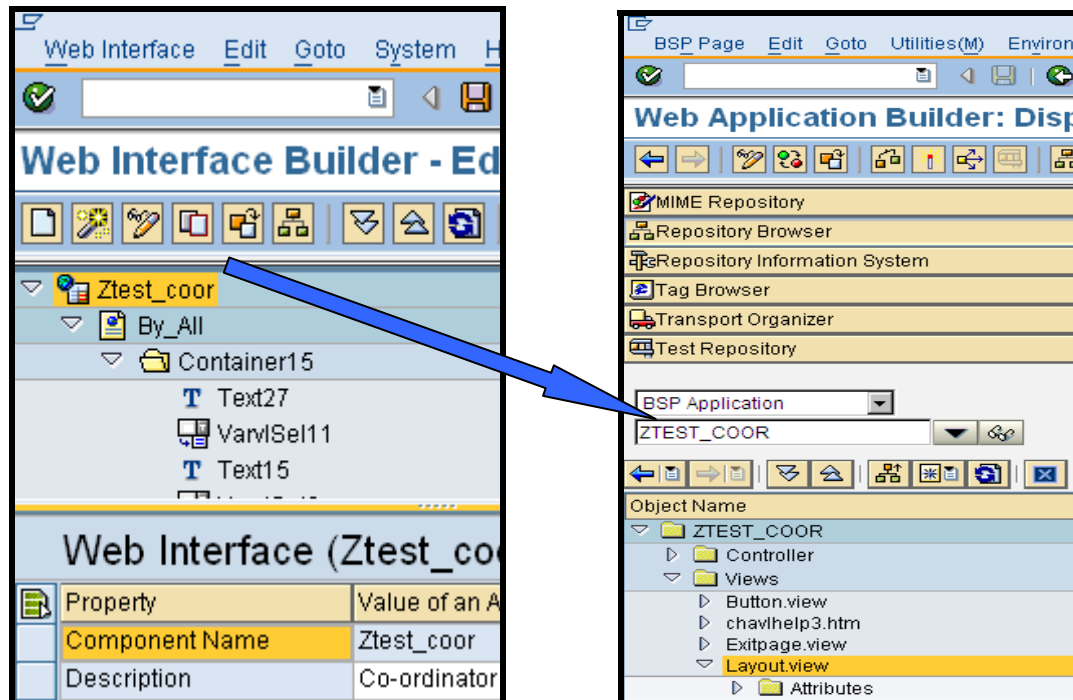
Here was the need to eliminate this newly generated column (Not Assigned Act) on this special business scenario.

Step Wise Solution

The following steps should be carried out to achieve the same:

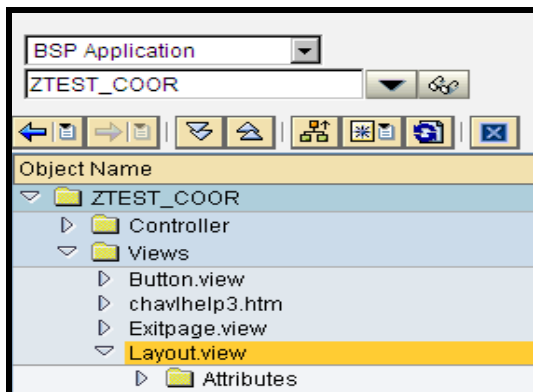
Identifying BSP Application:

The BSP application can be identified by the technical name of the BPS web interface. Here the technical name is 'ZTEST_COOR'. The same is the name of BSP application (Transaction → SE80 → BSP Application)



Locating the code where layout is created

Layout.view in the BSP application is the place which deals with layout component of the web interface.



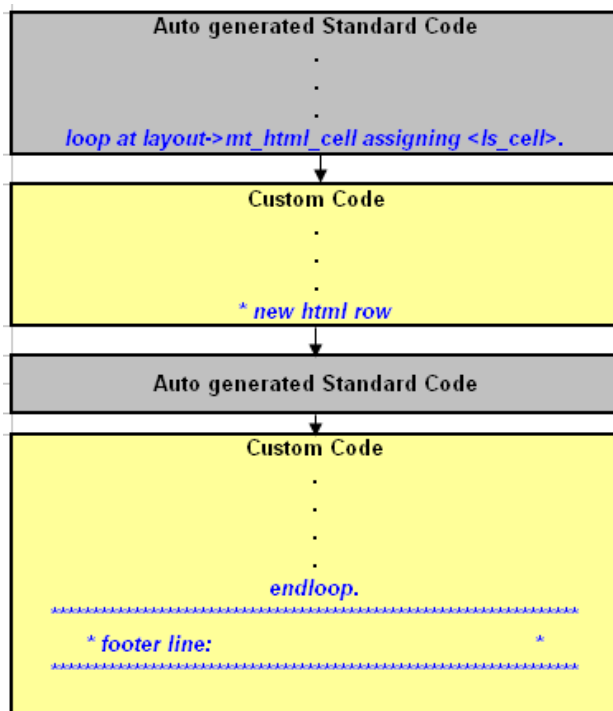
Code Changes and Logic

The change of code needs to be done in the Layout Rendering section. We need to do our code changes immediately after below statement:

loop at layout->mt_html_cell assigning <ls_cell>.

In case you have any IF statement under this loop and need to complete the ENDF statement, this should be done just above the Footer section.

So the place where code will be inserted is the custom code part in the below flow chart.



We have explained our coding in certain sections as mentioned below:

a. Find columns in Row 1 and Row 2

We need to figure out the count of columns is Row1 and Row2. The number of columns was different in these rows as we had Row 2 displaying the text along with the key for customer.

In our example, Row 1 had 15 columns; Row 2 has 16 columns

Year 2009														
Customer	Country	Not Assigned Act	Jan Pln	Feb Pln	Mar Pln	Apr Pln	May Pln	Jun Pln	Jul Pln	Aug Pln	Sep Pln	Oct Pln	Nov Pln	Dec Pln
C1	Customer 1	India	56	23	54	45	67	82	34	67	90	85	80	78

Pseudo Logic:

Loop at layout→mt_html_cell,
to get the number of columns in Row 1 and Row 2.
End the loop.

Sample Code:

```

loop at layout->mt_html_cell assigning <ls_cell>.
if <ls_cell>-row = 1 .
row1_col = row1_col + 1.
else.
if <ls_cell>-row = 2 .
row2_col = row2_col + 1.
else.
exit.
endif.
endif.
Endloop.
  
```

Result:

row1_col returns 15
row2_col returns 16

Note: This step will be required only when the number of columns in Row 1 and Row 2 are different.

b. Initialize column to be hidden

Initialize a variable to know exactly which column needs to be eliminated. In below case it is fourth column. Col_Elim = 4. This counting needs to be done with respect to row 2.

Year 2009														
Customer	Country	Not Assigned Act	Jan Pln	Feb Pln	Mar Pln	Apr Pln	May Pln	Jun Pln	Jul Pln	Aug Pln	Sep Pln	Oct Pln	Nov Pln	Dec Pln
C1	Customer 1	India	56	23	54	45	67	82	34	67	90	85	80	78

c. Add business logic to eliminate required column

The business logic to hide the column is closing period = 12 and current month = 01.

We need to ignore processing of the layout rendering for column number 4 (Col_Elim = 4) in case of this special business scenario.

Pseudo Code:

```
Loop at layout->mt_html_cell,
For that specific layout
  IF closing period = 12 AND current month = 01.
    IF Col_Elim = 4.
      CONTINUE.
    ENDIF.
  ENDIF.
```

Sample Code:

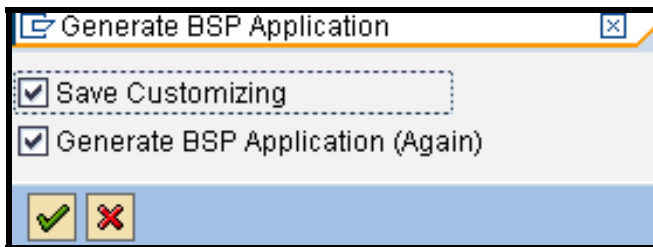
```
loop at layout->mt_html_cell assigning <ls_cell>.
  IF name = 'Layout3'.
    IF CLOSE_MTH = 12 AND CURR_MTH = 01.
      IF <ls_cell>-col = l_col3 .
        continue.
      ENDIF.
    ENDIF.
  ENDIF.
```

'Layout3' is the layout component used in the BPS web interface 'ZTEST_COOR'

Points to be taken care of:

We need to remember that the BSP code changes should be the last changed when working on any web interface. The steps would be:

1. Make changes in the web interface (like adding/removing any component)
2. Generate the BSP application



3. This results in auto generation of BSP code in the background.
4. The steps to hide any column should be done now.
5. In case there are still changes to be done in web interface, repeat steps 1 to 4.

This is necessary else, the auto generated code will overwrite your changes each time you generate the BSP application.

Related Content

http://help.sap.com/saphelp_nw70ehp1/helpdata/en/62/e5a562753511d5b3d70050dadfb23f/content.htm

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.