

How to maintain comments (or Text fields) in BPS planning layout. How to interface Transactional ODS with BPS Planning layout.

Applies to:

SAP SEM (BPS) 3.20.

SAP BW 3.50.

Summary

This document is about building an interface between a BPS planning layout (excel frontend) and transactional ODS to maintain comments (short text upto 60 char long) directly in the planning screen. During planning, many a times it is required to maintain a comment per data record. As an out-of-the-box functionality, SAP has provided to attach documents in the planning layout. In this article it is described (with sample ABAP codes) to maintain text fields like comments directly in the planning layout.

Author: Tarun Malik

Company: Infosys Technologies Ltd. India.

Created on: 05 November 2006

Author Bio



Tarun Malik is currently working as a consultant with Infosys Technologies Ltd. India. He has over 3 years of SAP-BW and SAP-BPS experience with more than 6 years of working experience in IT industry. He has 2 full lifecycle implementation experience of SAP-BW and 2 full life cycle implementation experience of SAP-BPS. He is certified SAP BW Consultant.

Table of Contents

| | |
|--|----|
| Applies to: | 1 |
| Summary..... | 1 |
| Author Bio | 1 |
| Business Scenario | 3 |
| Assumptions – | 3 |
| BW data model – | 3 |
| InfoObjects – | 3 |
| InfoCube – | 3 |
| Transactional ODS – | 4 |
| Integration of Planning layout and ABAP function Module – | 4 |
| ABAP Function Modules - | 5 |
| Include of function group– | 5 |
| Read Module : To read comments from layout and store in ODS | 5 |
| Write Module : To read comments from ODS and display in Planning Layout..... | 7 |
| How does it works – | 9 |
| Benefits – | 9 |
| Disadvantages – | 9 |
| Related Content..... | 10 |
| Disclaimer and Liability Notice..... | 11 |

Business Scenario

While planning Sales Values for different products per Calendar Year/Month, the sales planner wants to maintain a comment per record. This comment can be of maximum 60 characters long.

| | A | B | C | D |
|---|----------|---------------------|--------------|------------------------------|
| 1 | Char 01 | Calendar Year/Month | Sales Amount | Comments |
| 2 | | | | |
| 3 | PRODUCT1 | 11.2006 | 100.00 USD | Due to a very good reason. |
| 4 | PRODUCT1 | 12.2006 | 200.00 USD | Test comment. 100% Increase. |
| 5 | PRODUCT2 | 11.2006 | 2,000.00 USD | Schumacher is the Best. |
| 6 | PRODUCT2 | 12.2006 | 1,000.00 USD | |

Assumptions –

- Planning environment is Excel Based
- Comments should not be more then 60 characters long.

BW data model –

(Please note that technical name of the objects are provided to understand the ABAP codes easily. You can make your own objects with your own technical names as per your requirements.)

InfoObjects –

Following InfoObjects are used while defining the BW data model for the current scenario-

| InfoObject | Type | Purpose |
|------------|----------------|---|
| ZCHAR01 | Characteristic | To Store Products. |
| ZCOMMENT | Characteristic | To Store Comments. It can be max up to 60 characters long with lowercase letters allowed. |
| 0CALMONTH | Time-Char | To store month & year |
| 0AMOUNT | Key-figure | To store sales amount |

InfoCube –

Define a **Transactional** InfoCube as per your planning requirements. **Do not** include “Comments” infoobject in the InfoCube.

For Current scenario, my planning infocube has object –

| InfoObject | |
|------------|--------------------------|
| ZCHAR01 | As a characteristic |
| 0CALMONTH | As a time characteristic |
| 0AMOUNT | As a keyfigure |

Transactional ODS –

To store comments you need to define a **Transactional** ODS. You need to include all the infoobject as KEY fields on which you are storing comments. For example in current scenario I've to store comments on the combination of Product and CalYear-Month. So my key fields in the ODS will be ZCHAR01 and 0CALMONTH. **Include “Comments” infoObject as a Data field** in the ODS.

Technical Name of ODS - RA_BPS_1

Type – Transactional OSD

| InfoObject | Key/Data Field |
|------------|----------------|
| ZCHAR01 | Key field |
| ZCOMMENT | Data field |
| 0CALMONTH | Key field |

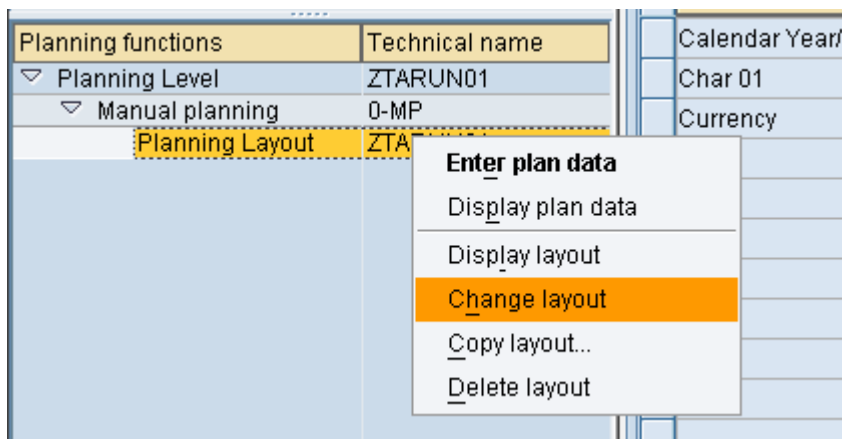
Integration of Planning layout and ABAP function Module –

You can call 2 function modules on a planning layout. One is “Write Module” and second is “Read Module”.

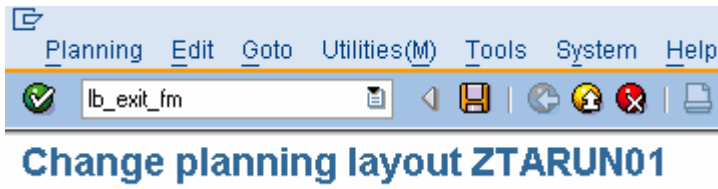
- Write Module :- This exit function module is used for writing data in the Excel Frontend. This FM executes while putting data into excel frontend.
- Read Module :- You can use this function module to read your own data from Excel frontend during manual planning.

Given below is the step by step method to link these exit function module with Excel planning layout.

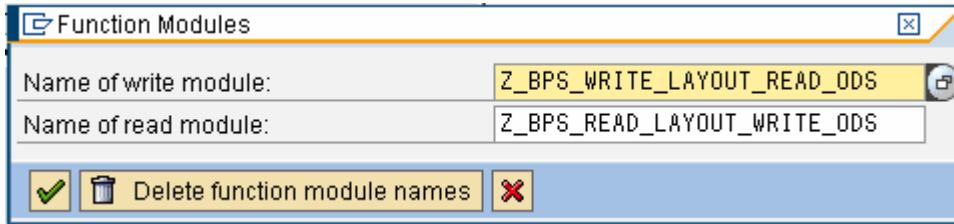
In planning workbench (TCODE - BPS0), please go into Change layout.



Put **lb_exit_fm** in the transaction code and press enter –



You get a pop up screen to asking for the exit function modules. Enter function module name in appropriate box –



ABAP Function Modules -

We require 2 ABAP function module. One for reading comments form planning layout and storing it in transactional ODS and second function module is for reading comments from transactional ODS and displaying in planning layout.

Include of function group–

Please define following variables in the TOP include of function group -

```

* "-----
type-pools: soi.

*Internal Tables
DATA : it_11_data      TYPE UPP_YT_LL,
       i_comments     TYPE STANDARD TABLE OF /BIC/ARA_BPS_100,
       i_comments_ods TYPE STANDARD TABLE OF /BIC/ARA_BPS_100,
       wa_comments    TYPE /BIC/ARA_BPS_100,
       i_contents     TYPE STANDARD TABLE OF soi_generic_item..

*Work Areas
DATA : wa_11          LIKE LINE OF it_11_data,
       wa_1101       LIKE LINE OF it_11_data,
       wa_comments_ods LIKE LINE OF i_comments_ods,
       wa_contents   LIKE LINE OF i_contents.

*Constants
DATA : l_lines        TYPE i.
* "-----
    
```

Read Module : To read comments from layout and store in ODS

This is the Read module of the planning layout. This will be used to read the comments data from the excel frontend and store in transactional ODS.

```

FUNCTION Z_BPS_READ_LAYOUT_WRITE_ODS.
* "-----
* " "Local Interface:
    
```

```

** IMPORTING
** REFERENCE(I_AREA) TYPE UPC_Y_AREA
** REFERENCE(I_PLEVEL) TYPE UPC_Y_PLEVEL
** REFERENCE(I_PARAM) TYPE UPC_Y_PARAM
** REFERENCE(ITO_HEAD) TYPE UPP_YTO_HEAD
** REFERENCE(IT_DIMENSION) TYPE SOI_DIMENSION_TABLE
** REFERENCE(IT_CONTENTS) TYPE SOI_GENERIC_TABLE
** EXPORTING
** REFERENCE(ET_DIMENSION) TYPE SOI_DIMENSION_TABLE
** REFERENCE(ET_CONTENTS) TYPE SOI_GENERIC_TABLE
** REFERENCE(ET_FORMAT) TYPE SOI_CELL_TABLE
** REFERENCE(ET_MESG) TYPE UPC_YT_MESG
** -----

```

*Read data records from the planning layout

```

i_contents[] = it_contents[].

DELETE i_contents where value IS INITIAL.

LOOP AT i_contents INTO wa_contents.
  IF wa_contents-value IS NOT INITIAL.

    wa_comments-/BIC/ZCOMMENTS = wa_contents-value.

    LOOP AT it_11_data INTO wa_11 WHERE
      ratio = 'Z' AND
      cont = 'A' AND
      col = '0001' AND
      row = wa_contents-row AND
      fcolumn = '0001'.

      wa_comments-/BIC/ZCHAR01 = wa_11-value.

      LOOP AT it_11_data INTO wa_1101 WHERE
        ratio = 'Z' AND
        cont = 'A' AND
        col = '0002' AND
        fcolumn = '0001' AND
        row = wa_11-row.

        wa_comments-CALMONTH = wa_1101-value.

      APPEND wa_comments TO i_comments.
    ENDL LOOP.

    Clear wa_1101.
  ENDL LOOP.

  Clear wa_11.
ENDIF.

Clear wa_comments.
ENDL LOOP.

```

*Store the data records in the transactional ODS using API 'RSDRI_ODSO_MODIFY'

```

IF NOT i_comments IS INITIAL.

  CALL FUNCTION 'RSDRI_ODSO_MODIFY'
  EXPORTING
    I_ODSOBJECT = 'RA_BPS_1'
    I_T_MODIFY = i_comments

```

* IMPORTING

```

*   E_RECORDS                                =
EXCEPTIONS
  DATA_TARGET_NOT_ODS                      = 1
  ODS_TYPE_NOT_TRANSACTIONAL                = 2
  ACTIVE_TABLE_NAME_NOT_FOUND              = 3
  RECORD_KEY_ALREADY_EXISTS                = 4
  ARRAY_MODIFY_FAILED                      = 5
  INTERNAL_ERROR                           = 6
  OTHERS                                    = 7

  .
  IF SY-SUBRC <> 0.
*   MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
*   WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
  ENDIF.
ENDIF.

ENDFUNCTION.

```

Write Module : To read comments from ODS and display in Planning Layout

This is the Write module of the planning layout. This will help in reading the saved comments from ODS and display in planning layout. You can make a copy of function UPF_EXCEL_AFTER_DATA_PUT_01, which will provide you with the correct interface specification already.

```

FUNCTION Z_BPS_WRITE_LAYOUT_READ_ODS.
*-----
*""Local Interface:
*  IMPORTING
*    REFERENCE(I_AREA) TYPE  UPC_Y_AREA
*    REFERENCE(I_PLEVEL) TYPE UPC_Y_PLEVEL
*    REFERENCE(I_PARAM) TYPE  UPC_Y_PARAM
*    REFERENCE(I_INIT) TYPE  UPP_Y_INITCALL
*    REFERENCE(ITO_HEAD) TYPE UPP_YTO_HEAD
*    REFERENCE(IT_LL) TYPE  UPP_YT_LL
*    REFERENCE(IT_MAP) TYPE  UPP_YT_MAP
*  EXPORTING
*    REFERENCE(ET_DIMENSION) TYPE SOI_DIMENSION_TABLE
*    REFERENCE(ET_CONTENTS) TYPE  SOI_GENERIC_TABLE
*    REFERENCE(ET_FORMAT) TYPE  SOI_CELL_TABLE
*    REFERENCE(ET_MESG) TYPE  UPC_YT_MESG
*-----
  DATA: range      TYPE soi_dimension_item,
        content     TYPE soi_generic_item.

*Read internal table to get Key values from layout.
*In the piece of code below we will get the values of
*key fields CHAR01 & CalmonthYear in internal table i_comments.

  it_ll_data = it_ll.

  LOOP AT it_ll_data INTO wa_ll WHERE
    ratio = 'Z' AND
    cont  = 'A' AND
    col   = '0001' AND
    fcolumn = '0001'.

    wa_comments-/BIC/ZCHAR01 = wa_ll-value.

  LOOP AT it_ll_data INTO wa_ll01 WHERE
    ratio = 'Z' AND
    cont  = 'A' AND
    col   = '0002' AND
    fcolumn = '0001' AND

```

```
row      = wa_11-row.  
  
wa_comments-CALMONTH = wa_1101-value.  
  
APPEND wa_comments TO i_comments.  
  
Clear wa_1101.  
ENDLOOP.  
  
Clear wa_11.  
ENDLOOP.
```

*Now after reading key values we will get the existing comments from the ODS.

```
IF NOT i_comments IS INITIAL.
```

*Read Comments ODS using SELECT Query. (You can use API 'BAPI_ODSO_READ_DATA' to read data from ODS.)

```
SELECT * INTO TABLE i_comments_ods  
FROM /BIC/ARA_BPS_100 FOR ALL ENTRIES IN i_comments  
WHERE  
/BIC/ZCHAR01      = i_comments-/BIC/ZCHAR01 AND  
CALMONTH         = i_comments-CALMONTH.  
  
SORT i_comments_ods.  
ENDIF.
```

*Now we modify internal table with key value(I_COMMENTS) with the comments from i_comments_ods.

```
IF NOT i_comments_ods IS INITIAL.  
LOOP AT i_comments INTO wa_comments.  
  READ TABLE i_comments_ods INTO wa_comments_ods WITH KEY  
  /BIC/ZCHAR01 = wa_comments-/BIC/ZCHAR01  
  CALMONTH     = wa_comments-CALMONTH BINARY SEARCH.  
  
IF sy-subrc = 0.  
  wa_comments-/BIC/ZCOMMENTS = wa_comments_ods-/BIC/ZCOMMENTS.  
  
  MODIFY i_comments FROM wa_comments TRANSPORTING /BIC/ZCOMMENTS.  
ENDIF.  
  
  CLEAR: wa_comments_ods,  
         wa_comments.  
ENDLOOP.  
ENDIF.  
  
SORT i_comments.  
DESCRIBE TABLE i_comments LINES l_lines.
```

*Write comments on the layout. As per our scenario we have comment 4th column
(Column D of excel) as comments column and data rows starts from row no 3.

* position of rectangle

```
range-row = 3.  
range-column = 4.
```

* size of rectangle

```
range-rows = l_lines .  
range-columns = 1.  
APPEND range TO et_dimension.
```


* add content.


```
LOOP AT i_comments INTO wa_comments.  
  content-column = 1.  
  content-row = sy-tabix.  
  content-value = wa_comments-/BIC/ZCOMMENTS.  
  APPEND content TO et_contents.
```

```
  Clear wa_comments.  
ENDLOOP.
```

```
ENDFUNCTION.
```

How does it works –

When you enter comments in the excel column and press the save button , then Read module executes and it reads the data from excel frontend and inserts/modifies the records in the transactional ODS.

After the system-save operation, system prepares the data to display it on planning layout. At this time Write module executes and it reads comments from ODS and puts it in the excel frontend.

On a planning layout opening event, only Write module executes and it picks already stored comments from the ODS and displays it on the planning layout.

Benefits –

Main benefit of this approach is, the user need not attach documents for the small comments. This approach is a totally customized solution. you can write ABAP code as per your requirements and put your own logic and validations.

Disadvantages –

Main disadvantage of this approach is, it can affect the performance of the layout opening and data saving. On every save-operation (while planning layout is open), Read and Write function module executes. These 2 exits can take few seconds to execute. Maintainability can be another disadvantage here. Changing in column positions may lead a change in the exit function modules.

Related Content

Please include at least three references to SDN documents or web pages.

- [Reference 1 - Reading data from Excel Frontend in BPS layout](#)
- [Reference 2 - UPF EXCEL AFTER DATA PUT 01](#)
- [Reference 3 - loading data into transactional ODS from BSP](#)

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.