Enhancing BMBC using BADI

Applies to:
SAP ECC 6.0 (Release 700, SP 12). For more information, visit the ABAP Homepage.

Summary
The article describes the enhancement of the standard transaction BMBC for displaying additional fields in the ALV output using BADI.

Author: Jayashree Thiyagarajan
Company: Applexus Technologies Pvt. Ltd.
Created on: 15 May 2011

Author Bio
Jayashree Thiyagarajan is working as a SAP Technology Consultant with Applexus Technologies (P) Ltd.
Introduction

The following are the steps for enhancing the standard transaction BMBC.
Initially we need to find the BADI which is available for BMBC.

Available BADI for BMBC

Goto se37 and provide the function module name SXV_GET_CLIF_BY_NAME. This function module is used for finding out the list of BADIs for the currently running transaction.

Set the BREAK-POINT.

Now execute the BMBC Transaction
Upon executing the transaction BMBC, the function module SXV_GET_CLIF_BY_NAME triggers and the exporting parameter NAME will provide the BADI Definition, BIC_ADDITIONAL_DATA which is used to add additional fields to BMBC.
Before Implementation

Now to copy the BADI Definition, BIC_ADDITIONAL_DATA goto to the transaction SE18 and provide the BADI name BIC_ADDITIONAL_DATA and click the display button.

Implementation of BADI
To implement this BADI drop down the Implementation option in the menu bar and click CREATE

Enter the name in Implementation Name Field.

Provide the Short description for the Implementation
To see the interface available for the BADI, click on the INTERFACE tab.

Save it in a package and activate it.

To add the custom code in the interface provided, goto SE19 and enter the implementation name ZAPG_IM_STOCK_AGEING and click the change button.
Goto the interface tab.

To add the additional fields to the ALV at the stock information level, we need to add those additional fields to the field catalogue.

Click on the method STOCK_COLUMNS_ADD to add the custom fields Vendor Batch, Posting Date, Age of GI and Age of GR.
Custom Code

```java
DATA:
  ls_fcat TYPE lvu_s_fcat.

* add column for Vendor Batch number at first position
CLEAR ls_fcat.
ls_fcat-col_pos = 1.
ls_fcat-fieldname = 'LVCA'.
ls_fcat-ref_field = 'LVCA'.
ls_fcat-ref_table = 'LVCA'.
APPEND ls_fcat.
TO ET_STOCK_FCAT_ADD.

CLEAR ls_fcat.
ls_fcat-col_pos = 2.
ls_fcat-fieldname = 'BUDAT'.
ls_fcat-ref_field = 'BUDAT'.
ls_fcat-ref_table = 'MDMB'.
APPEND ls_fcat.
TO ET_STOCK_FCAT_ADD.

CLEAR ls_fcat.
ls_fcat-col_pos = 3.
ls_fcat-fieldname = 'AGE_GR'.
ls_fcat-ref_field = 'XABLN'.
ls_fcat-ref_table = 'MBEP'.
ls_fcat-SCRTEXT = 'Age of GR[Days]'.
ls_fcat-SCRTEXT_N = 'Age of GR[Days]'.
ls_fcat-SKTEXT_b = 'Age of GR[Days]'.
ls_fcat-TEXT_FIELD = 'LT_STOCK_TAP'.
ls_fcat-COLTIDX = 'Age of GR',
ls_fcat-outputlen = '06'.
APPEND ls_fcat.
TO ET_STOCK_FCAT_ADD.

CLEAR ls_fcat.
ls_fcat-col_pos = 4.
ls_fcat-fieldname = 'AGE_GI'.
ls_fcat-ref_field = 'XABLN'.
ls_fcat-ref_table = 'MBEP'.
ls_fcat-SCRTEXT = 'Age of GI[Days]'.
ls_fcat-SCRTEXT_N = 'Age of GI[Days]'.
ls_fcat-SKTEXT_b = 'Age of GI[Days]'.
```

Store METHOD IF_EX_BIC_ADDITIONAL_DATA_STOCK_COLUMNS_ADD

Enhancing BMBC using BAD!
Fill the internal table with your additional fields.

For selecting the values from table, click on the method STOCK_ADD_DATA_SELECT.
METHOD if_ex_bic_additional_data~stock_add_data_select.

TYPES : BEGIN OF ty_mcha,
  matnr TYPE matnr,
  werks TYPE werks,
  charg TYPE charg_d,
  licha TYPE lichn,
  lwedt TYPE lwedt,
END OF ty_mcha,

BEGIN OF ty_mseg,
  mblnr TYPE mblnr,
  charg TYPE charg_d,
  matnr TYPE matnr,
END OF ty_mseg,

BEGIN OF ty_mkpf,
  mblnr TYPE mblnr,
  budat TYPE budat,
END OF ty_mkpf,

BEGIN OF ty_makt,
  matnr TYPE matnr,
  maktx TYPE maktx,
END OF ty_makt.

DATA : lt_mcha TYPE STANDARD TABLE OF ty_mcha,
  lt_mseg TYPE STANDARD TABLE OF ty_mseg,
  lt_mkpf TYPE STANDARD TABLE OF ty_mkpf,
  lt_makt TYPE STANDARD TABLE OF ty_makt.

DATA : lx_mcha TYPE ty_mcha,
  lx_mkpf TYPE ty_mkpf,
  lx_mseg TYPE ty_mseg,
  lx_makt TYPE ty_makt.

DATA : lx_master TYPE mchx_binco_stock,
  lx_stock TYPE ty_stock.

DATA : lv_sdate TYPE d,
  lv_edate TYPE d,
  lv_days TYPE p.

IF it_stock IS NOT INITIAL.

SELECT matnr
  werks
  charg
  licha
  lwedt
FROM mcha
INTO TABLE lt_mcha
FOR ALL ENTRIES IN it_stock
WHERE matnr EQ it_stock-matnr AND
  werks EQ it_stock-werks AND
  charg EQ it_stock-charg.
SELECT mblnr
    charg
    matnr
FROM mseg
INTO TABLE lt_mseg
FOR ALL ENTRIES IN it_stock
WHERE matnr EQ it_stock-matnr AND
    werks EQ it_stock-werks AND
    charg EQ it_stock-charg AND
    bwart EQ '601' AND
    lgort EQ it_stock-lgort.

IF lt_mseg IS NOT INITIAL.

SELECT mblnr
    budat
FROM mkpf
INTO TABLE lt_mkpf
FOR ALL ENTRIES IN lt_mseg
WHERE mblnr EQ lt_mseg-mblnr.
ENDIF.

SORT lt_mcha BY matnr.
SORT lt_mseg BY charg.
SORT lt_mkpf BY mblnr.

LOOP AT it_stock INTO lx_master.
    lx_stock-matnr = lx_master-matnr.
    lx_stock-werks = lx_master-werks.
    lx_stock-charg = lx_master-charg.

READ TABLE lt_mcha INTO lx_mcha WITH KEY matnr = lx_master-matnr
                charg = lx_master-charg.

IF sy-subrc EQ 0.

    lx_stock-licha = lx_mcha-licha.

    IF lx_mcha-lwedt IS INITIAL.
        lv_sdate = '99991231'.
    ELSE.

        lv_sdate = lx_mcha-lwedt.
    ENDIF.
    lv_edate = sy-datum.

    CALL FUNCTION '/SDF/CMO_DATETIME_DIFFERENCE'
        EXPORTING
            date1 = lv_sdate
            TIME1 =
            date2 = lv_edate
            TIME2 =
        IMPORTING
            lv_datbroken= sy-subrc
datediff = lv_days
TIMEDIFF =
EARLIEST =

* EXCEPTIONS
invalid_datetime = 1
OTHERS = 2

IF sy-subrc <> 0.
MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.

lx_stock-age_gr = lv_days.

CLEAR : lv_sdate,
        lv_edate,
        lv_days.

ELSE.
lx_stock-age_gr = '9999999999'.

ENDIF.

READ TABLE lt_mseg INTO lx_mseg WITH KEY charg = lx_master-charg
        matnr = lx_master-matnr
        BINARY SEARCH.

IF sy-subrc EQ 0.

READ TABLE lt_mkpf INTO lx_mkpf WITH KEY mblnr = lx_mseg-mblnr
        BINARY SEARCH.

IF sy-subrc EQ 0.

lx_stock-budat = lx_mkpf-budat.
lv_sdate = lx_stock-budat.
lv_edate = sy-datum.

CALL FUNCTION '/SDF/CMO_DATETIME_DIFFERENCE'
  EXPORTING
      date1 = lv_sdate
      TIME1 =
      date2 = lv_edate
      TIME2 =
  IMPORTING
      datendiff = lv_days
      TIMEDIFF =
      EARLIEST =

* EXCEPTIONS
invalid_datetime = 1
OTHERS = 2

IF sy-subrc <> 0.
MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.

lx_stock-age_gi = lv_days.
CLEAR : lv_sdate,
       lv_edate,
       lv_days.

ENDIF.
ELSE.
  CLEAR : lx_stock-budat .
          lx_stock-age_gi = '9999999999'.
ENDIF.
APPEND lx_stock TO lt_stock_tmp.
CLEAR lx_stock .
ENDLOOP.

if_ex_bic_additional_data~stock_add_data = lt_stock_tmp.

ENDIF.
ENDMETHOD.

Finally to add the custom fields to CS_STOCK, code it accordingly in the method STOCK_ADD_DATA_FILL_IN

---

**Business Add-In Builder: Change Implementation ZAPG**

<table>
<thead>
<tr>
<th>Implementation Name</th>
<th>ZAPG_IM_STOCK_AGEING</th>
<th>Active</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation Short Text</td>
<td>STOCK_AGEING</td>
<td></td>
</tr>
<tr>
<td>Definition Name</td>
<td>BIC_ADDITIONAL_DATA</td>
<td></td>
</tr>
</tbody>
</table>

**Attributes**

**Interface**

<table>
<thead>
<tr>
<th>Interface name</th>
<th>IF_EX_BIC_ADDITIONAL_DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of implementing class</td>
<td>ZCL_IM_RPG_IM_STOCK_AGEING</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Implement...</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MASTER_COLUMNS_ADD</td>
<td>ABAP Co.</td>
<td>Add Columns for Additional Data for S</td>
</tr>
<tr>
<td>STOCK_COLUMNS_ADD</td>
<td>ABAP Co.</td>
<td>Add Columns for Additional Data for S</td>
</tr>
<tr>
<td>MASTER_ADD_DATA_FILL_IN</td>
<td>ABAP Co.</td>
<td>Add Additional Data to Selection Res</td>
</tr>
<tr>
<td>STOCK_ADD_DATA_FILL_IN</td>
<td>ABAP Co.</td>
<td>Add Additional Data to Selection Res</td>
</tr>
<tr>
<td>MASTER_ADD_DATA_SELECT</td>
<td>ABAP Co.</td>
<td>Select Additional Data for Selection Res</td>
</tr>
<tr>
<td>STOCK_ADD_DATA_SELECT</td>
<td>ABAP Co.</td>
<td>Select Additional Data for Selection Res</td>
</tr>
</tbody>
</table>

Example implementation class: CL_EXM_IM_BIC_ADDITIONAL_DATA
DATA : lx_stock TYPE ty_stock.

FIELD-SYMBOLS:

<fs_matnr> TYPE matnr,
<fs_werks> TYPE werks_d,
<fs_charg> TYPE charg_d,
<fs_licha> TYPE lichn,
<fs_budat> TYPE budat,
<fs_age_gr> TYPE char10,
<fs_age_gi> TYPE char10.

DEFINE assign_comp_1_of_2_move_2_3.
  assign component &1
    of structure cs_stock
    to &2.
  if not &2 is initial.
    &3 = &2.
  endif.
END-OF-DEFINITION.

DEFINE assign_comp_1_of_2_move_3_2.
  assign component &1
    of structure cs_stock
    to &2.
  if not &3 is initial.
    &2 = &3.
  endif.
END-OF-DEFINITION.

lt_stock_tmp = if_ex_bic_additional_data-stock_add_data.

assign_comp_1_of_2_move_2_3 'MATNR' <fs_matnr> lx_stock-matnr.
assign_comp_1_of_2_move_2_3 'WERKS' <fs_werks> lx_stock-werks.
assign_comp_1_of_2_move_2_3 'CHARG' <fs_charg> lx_stock-charg.

READ TABLE lt_stock_tmp INTO lx_stock with KEY matnr = <fs_matnr>
  werks = <fs_werks>
  charg = <fs_charg>.

assign_comp_1_of_2_move_3_2 'LICHA' <fs_licha> lx_stock-licha.
assign_comp_1_of_2_move_3_2 'BUDAT' <fs_budat> lx_stock-budat.
assign_comp_1_of_2_move_3_2 'AGE_GR' <fs_age_gr> lx_stock-age_gr.
assign_comp_1_of_2_move_3_2 'AGE_GI' <fs_age_gi> lx_stock-age_gi.

CLEAR lx_stock.
After Implementation

Goto BMBC and Execute the Transaction.
Related Content

Adding a Field to BMBC
BADI Explanation
BADI for Batches

For more information, visit the ABAP 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.