

# Retraction through Bex Query (BW) into ECC



## Applies to:

SAP BI 7.0 and ECC R/3. For more information, visit the [EDW homepage](#)

## Summary

We can use the following program to fetch the data from BW (Bex query) to ECC side. After fetching data from BW Query (Retraction) we can use that data for further processing in ECC side.

**Author:** Pawan Kumar

**Company:** IBM.India

**Created on:** 30 September 2010

## Author Bio

Pawan kumar is working as SAP BI consultant at IBM, India. He has got a rich experience on various BW/ABAP Implementation/Support Projects.

## Table of Contents

Pre-requisite: .....	3
Scenario:.....	3
Steps: BW System Side.....	3
ECC System Side: .....	3
Related Content.....	12
Disclaimer and Liability Notice.....	13

## Pre-requisite:

There should be an RFC connection between BW and ECC system. RFC should be properly maintained for BW and ECC by Basis people.

## Scenario:

We have to fetch revenue planning numbers after BPS calculation has happened and results are stored in planning cube. On top of Planning Cube BEX Query made.

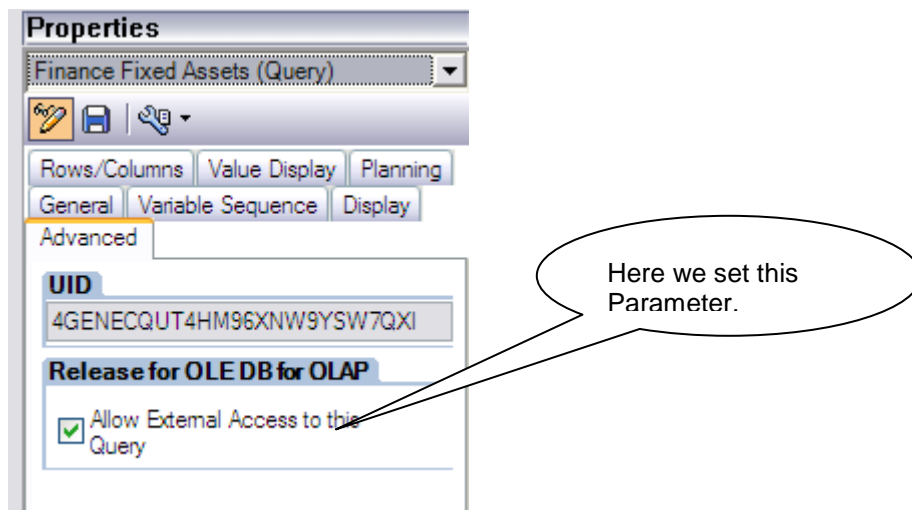
After taking revenue planning numbers we can create the memo records in ECC side and show the listing of memo records for retracted planning number.( for this I am not including the coding Part ,because every one has his own requirement).

After having data in ECC side from BW Query, we can play with that data according to requirement.

## Steps: BW System Side.

The Query from which we need to retract the data, that query should be enabled for external system access.

We have to check one checkbox "Allow External access to this query" through the properties of query in Advanced Tab.



## ECC System Side:

Here Using SE38 Tcode we have to create one ABAP programm , Following is the code :

**Note: I have commented some part of code which is not required for explaining the retraction .**

**I have wrote necessary comments wherever needed.**

```

TYPE-P00LS: slis.
DATA: g_name          TYPE uname.          "For Usernames
DATA: fieldcatalog    TYPE slis_fieldcat_alv,
      i_top_of_page   TYPE slis_t_listheader,
      st_top_of_page  TYPE slis_listheader.
DATA: st_layout       TYPE slis_layout_alv,
      gd_repid        LIKE sy-repid.
DATA: l_reportuid(72) TYPE c,              "#EC NEEDED
      l_varnt         TYPE variant,
      l_t_return      TYPE TABLE OF bapiret2,
      l_t_mdx         TYPE TABLE OF bapi6111mdx,
      l_t_variables   TYPE TABLE OF rscrmvar,

```

```

l_t_crmvariants    TYPE TABLE OF rscrmvariants,
l_s_crmvariant    TYPE rscrmvariants,           "#EC NEEDED*"
l_t_crmvar        TYPE TABLE OF rscrmvar,
l_t_crmvar_temp   TYPE TABLE OF rscrmvarh,
l_t_crmvarsel     TYPE TABLE OF rscrmvarp,
l_count(1)       TYPE n,
l_bdcfield       TYPE char16.
DATA: v_temp      TYPE string,
      v_lincnt    TYPE string.

*&-----
*&      CONSTANT DECLARATION
*&-----
CONSTANTS: c_update(1) TYPE c VALUE 'S',
           c_flag      TYPE flag      VALUE 'X'.

*&-----
*&      TYPES DECLARATION
*&-----
TYPES : BEGIN OF t_result,
        text      TYPE char250,
      END OF t_result.

TYPES: BEGIN OF t_bwdata,      "bw data
      *Memo record type (planning level): (based on liquidity item and z-
      table lookup)
        dsart TYPE fdes-dsart,
      *Value date: value date determined from retraction program or new R/3 z-
      table
        hzdat TYPE fdes-hzdat,
      *Account name: CASH
        bnkko TYPE fdes-bnkko ,
      *Amount: value date determined from retraction program
        dmshb TYPE fdes-dmshb,
      *Currency: USD
        dispw TYPE fdes-dispw,
      *Text: Liquidity item - Liquidity item text.
        sgtxt TYPE fdes-sgtxt,
      *Fund: <put in the reference field (FDES- REFER) in the memo record>
        refer TYPE fdes-refer,
      END OF t_bwdata.

*TYPES: BEGIN OF t_zcash_categories,
*      lqpos      TYPE zcash_categories-lqpos,
*      ebene      TYPE zcash_categories-ebene,
*      day_of_month TYPE zcash_categories-day_of_month,
*      percentage TYPE zcash_categories-percentage,
*      END OF t_zcash_categories.

*&-----*
*&      INTERNAL TABLE DECLARATION
*&-----*
DATA : i_result    TYPE STANDARD TABLE OF t_result WITH HEADER LINE,
      i_bwdata    TYPE TABLE OF fdes,
*      i_zcash_categories TYPE STANDARD TABLE OF t_zcash_categories,
      i_axis1     TYPE STANDARD TABLE OF rscrmaxis,
      i_cell11    TYPE STANDARD TABLE OF rscrmcell,
      i_axis      TYPE STANDARD TABLE OF rscrmaxis,

```

This t\_bwdata structure (internal table) we can customize according to our requirement in ECC which will be holding BW Query output.

```

i_cell      TYPE STANDARD TABLE OF rscrmcell,
i_days      TYPE STANDARD TABLE OF rke_dat.

*&-----*
*&      WORK AREA DECLARATION
*&-----*
DATA : wa_result TYPE t_result,
      wa_bwdata TYPE fdes,
*      wa_zcash_categories TYPE t_zcash_categories,
      wa_fcat      TYPE slis_t_fieldcat_alv,
      w_t_return   TYPE bapiret2,
      wa_t_crmvase1 TYPE rscrmvarp,
      wa_axis      TYPE rscrmaxis,
      wa_axis1     TYPE rscrmaxis,
      wa_cell      TYPE rscrmcell.

*&-----*
*&      SELECTION SCREEN DECLARATION
*&-----*
SELECTION-SCREEN BEGIN OF BLOCK blk1 WITH FRAME TITLE text-034.
PARAMETERS p_bw_sys TYPE fmbp_rfcdest OBLIGATORY.
      yid bw_sys
SELECTION-SCREEN BEGIN OF LINE.
* Query
SELECTION-SCREEN COMMENT 1(31) FOR FIELD p_query.
*SELECTION-SCREEN POSITION 25.
PARAMETERS p_query TYPE fmbp_query OBLIGATORY.
      RY ID query.
SELECTION-SCREEN END OF LINE.
PARAMETERS p_varnt TYPE fmbp_query_varnt.
SELECTION-SCREEN PUSHBUTTON /1(72) text-010 USER-COMMAND query VISIBLE LENGTH 35.
* Pushbutton for creating query variant
SELECTION-SCREEN END OF BLOCK blk1 .
SELECTION-SCREEN BEGIN OF BLOCK blk3 WITH FRAME TITLE text-034.
PARAMETER : p_test TYPE c AS CHECKBOX..
SELECTION-SCREEN END OF BLOCK blk3.

*&-----*
*&      INITIALIZATION DECLARATION
*&-----*

INITIALIZATION.
      gd_repid = sy-repid.
*      PERFORM z_seloption_restrict USING 'S_NAME'.

*&-----*
*&      AT SELECTION-SCREEN
*&-----*

AT SELECTION-SCREEN .
      IF sy-ucomm = 'QUERY'.
          PERFORM get_query_variant USING p_bw_sys p_query.
          uery variant
          ENDIF.

*&-----*
*&      START OF SELECTION
*&-----*

START-OF-SELECTION.
*      Get the Bw data

```

We have maintained selection text for selection screen Parameters.i.e BW System

This text we need to maintain in text symbol.i.e Source

Query

Query variant

Create Variant From BW Querr .

This Selection screen block is Optional, I have just put for my requirement.

"getting q

```

PERFORM get_bw_data.
* Arrange the BW data
* PERFORM prepare_data.
*&-----*
*&      END OF SELECTION
*&-----*
END-OF-SELECTION.
*Display the data .
* PERFORM display_data.
*&-----*
*&      Form  get_query_variant
*&-----*
*      text
*-----*
*      -->U_BW_SYS   text
*      -->U_QUERY   text
*-----*
FORM get_query_variant USING
                        u_bw_sys TYPE fmbp_rfcdest
                        u_query  TYPE fmbp_query.

DATA g_t_crmvar        TYPE TABLE OF rscrmvar.

CALL FUNCTION 'RSCRMBW_REPORT'                                "in
initialisation of report
  DESTINATION u_bw_sys
  EXPORTING
    i_mode = 'OPEN'
    i_reportuid = u_query
    i_repname = ' '
  IMPORTING
    e_reportuid = l_reportuid
  EXCEPTIONS
    communication_failure = 4
    system_failure = 5
    OTHERS = 7.
*!! check problem with mandatory variables -> e_t_meta is empty
IF sy-subrc <> 0.
  MESSAGE e000(fm_bp_bwret) WITH u_bw_sys.
ENDIF.

CALL FUNCTION 'RSCRMBW_VARIABLES_GETLIST'
  DESTINATION u_bw_sys
  EXPORTING
    i_reportuid = u_query
  TABLES
    e_t_crmvar = g_t_crmvar
  EXCEPTIONS
    communication_failure = 4
    system_failure = 5
    OTHERS = 7.
IF sy-subrc <> 0.
  MESSAGE e000(fm_bp_bwret) WITH u_bw_sys.
ENDIF.
IF g_t_crmvar[] IS INITIAL.
  MESSAGE e022(fm_bp_bwret) WITH u_query.

```

```

ENDIF.

CALL FUNCTION 'RSCRMBW_VARIABLE_POPUP'
  DESTINATION u_bw_sys
  EXPORTING
    i_reportuid      = u_query
*   I_POPUP          = 'X'
  EXCEPTIONS
    communication_failure = 4
    system_failure       = 5
    OTHERS                = 7.
IF sy-subrc <> 0.
  MESSAGE e000(fm_bp_bwret) WITH u_bw_sys.
ENDIF.

CALL FUNCTION 'RSCRMBW_REPORT'                                "initial
isatation of report
  DESTINATION u_bw_sys
  EXPORTING
    i_mode = 'CLOSE'
    i_reportuid = u_query
    i_repname = ' '
  IMPORTING
    e_reportuid = l_reportuid
  EXCEPTIONS
    communication_failure = 4
    system_failure       = 5
    OTHERS                = 7.
*!! check problem with mandatory variables -> e_t_meta is empty

IF sy-subrc <> 0.
  MESSAGE e000(fm_bp_bwret) WITH u_bw_sys.
ENDIF.

ENDFORM.                                                    "get_query_variant

*&-----*
*&      Form get_bw_data
*&-----*
*      text
*-----*

FORM get_bw_data .

  l_varnt = p_varnt.

CALL FUNCTION 'RSCRMBW_REPORT'                                "initia
lisatation of report
  DESTINATION p_bw_sys
  EXPORTING
    i_mode = 'OPEN'
    i_reportuid = p_query
    i_repname = ' '
  IMPORTING
    e_reportuid = l_reportuid
  TABLES
    e_t_axis     = i_axis1

```

```

        e_t_cell      = i_cell1
        e_t_return    = l_t_return
        e_t_mdx       = l_t_mdx
    EXCEPTIONS
        communication_failure = 4
        system_failure        = 5
        OTHERS                 = 7.
IF sy-subrc <> 0.
    MESSAGE e000(fm_bp_bwret) WITH p_bw_sys.
ELSEIF l_t_return[] IS NOT INITIAL.
    CLEAR w_t_return.
    READ TABLE l_t_return INTO w_t_return WITH KEY type = 'E'.
    IF sy-subrc EQ 0.
        MESSAGE e162(00) WITH w_t_return-message p_query.
    ENDIF.
ENDIF.

CALL FUNCTION 'RSCRMBW_FLATTEN_VIEW'
    DESTINATION p_bw_sys
    EXPORTING
        i_reportuid      = p_query
    TABLES
        e_t_mdx          = l_t_mdx
        e_t_return       = l_t_return
    EXCEPTIONS
        communication_failure = 4
        system_failure        = 5
        OTHERS                 = 7.
IF sy-subrc <> 0.
    MESSAGE e000(fm_bp_bwret) WITH p_bw_sys.
ENDIF.

IF NOT l_varnt IS INITIAL.
    REFRESH l_t_variables.
    CALL FUNCTION 'RSCRMBW_VARIANTS_GETLIST'
        DESTINATION p_bw_sys
        EXPORTING
            i_reportuid      = p_query
        TABLES
            e_t_crmvariants = l_t_crmvariants
            e_t_return       = l_t_return
        EXCEPTIONS
            communication_failure = 4
            system_failure        = 5
            OTHERS                 = 7.
    * check if problem with rfc-destination
    IF sy-subrc <> 0.
        ENDIF.
    READ TABLE l_t_crmvariants INTO l_s_crmvariant
        WITH KEY variant = l_varnt.
    IF sy-subrc NE 0.
        MESSAGE e032(fm_bp_bwret) WITH l_varnt p_query .
    ELSE.
        CALL FUNCTION 'RSCRMBW_VARIABLES_SET'
            DESTINATION p_bw_sys
            EXPORTING

```



```

        i_reportuid      = p_query
        i_variant        = l_s_crmvariant-variant
    TABLES
        i_t_crmvar        = l_t_variables
        i_t_crmvarp        = l_t_crmvar
    *
        i_t_crmvarsel      = l_t_crmvarsel
        e_t_return        = l_t_return
        e_t_mdx           = l_t_mdx
    EXCEPTIONS
        communication_failure = 4
        system_failure       = 5
        OTHERS               = 7.
    IF sy-subrc <> 0.
    ENDIF.

ENDIF.
ELSEIF l_varnt IS INITIAL.
    CALL FUNCTION 'RSCMBW_VARIABLES_GETLIST'
        DESTINATION p_bw_sys
    EXPORTING
        i_reportuid      = p_query
    TABLES
        e_t_crmvar        = l_t_crmvar_temp
        e_t_return        = l_t_return
    EXCEPTIONS
        communication_failure = 4
        system_failure       = 5
        OTHERS               = 7.

    IF sy-subrc <> 0.
    ENDIF.
    CALL FUNCTION 'RSCMBW_VARIABLE_POPUP'
        DESTINATION p_bw_sys
    EXPORTING
        i_reportuid      = p_query
    *
        I_POPUP          = 'X'
    TABLES
        e_t_crmvarsel    = l_t_crmvar
        e_t_mdx          = l_t_mdx
        e_t_return        = l_t_return
    EXCEPTIONS
        communication_failure = 4
        system_failure       = 5
        OTHERS               = 7.
    IF sy-subrc <> 0.
        MESSAGE e000(zf01) WITH 'Popup Failure'(t09).
    ENDIF.
    ENDIF.

    CALL FUNCTION 'RSCMBW_REPORT'
        DESTINATION p_bw_sys
    EXPORTING
        i_mode = 'START'
        i_reportuid = p_query
        i_repname = ' '
    IMPORTING

```

```

    e_reportuid = l_reportuid
TABLES
    e_t_axis      = i_axis1
    e_t_cell      = i_cell1
*
    e_t_meta      = e_t_meta
    e_t_return    = l_t_return
    e_t_mdx       = l_t_mdx
EXCEPTIONS
    communication_failure = 4
    system_failure       = 5
    OTHERS                = 7.
IF sy-subrc <> 0.
    MESSAGE e000(fm_bp_bwret) WITH p_bw_sys.
ELSE.
    i_axis[] = i_axis1[].
    i_cell[] = i_cell1[].
ENDIF.

CALL FUNCTION 'RSCRMBW_REPORT'
    DESTINATION p_bw_sys
EXPORTING
    i_mode = 'CLOSE'
    i_reportuid = p_query
    i_repname = ''
IMPORTING
    e_reportuid = l_reportuid
TABLES
    e_t_axis      = i_axis1
    e_t_cell      = i_cell1
*
    e_t_meta      = e_t_meta
    e_t_return    = l_t_return
    e_t_mdx       = l_t_mdx
EXCEPTIONS
    communication_failure = 4
    system_failure       = 5
    OTHERS                = 7.
IF sy-subrc <> 0.
    MESSAGE e000(fm_bp_bwret) WITH p_bw_sys.
ENDIF.

CLEAR wa_axis.
READ TABLE i_axis INTO wa_axis WITH KEY axis = '001'.

IF sy-subrc <> 0.
    MESSAGE e005(fm_bp_bwret) .
ENDIF.
ENDFORM.                "get_bw_data

*&-----*
*&      Form  prepare_data
*&-----*
*
*      text
*-----*
*FORM prepare_data.
*Logic needs to be changed according to the BEx layout.
*ENDFORM.                "prepare_data

```

i\_axis[] table  
having BW Query  
Output. Then  
Further according  
to the  
requirement we  
can access this  
table.

Here are Screen shots stepwise after executing the program.

**TR-LP Retraction Program**

Source

BW System  [Redacted]

Query  [Redacted]

Query Variant [Redacted]

Create Variant for BW Query

Option

Test Run

Screen Variants

This is optional, I have just put for my requirement.

BW System Name maintained by RFC connection.

BW Query Name

From this Button we can create Variant on BW Query. Then directly call that variant for query input

**TR-LP Retraction Program**

Source

BW System BCSBW

Query I\_ZTRLPIM01/ZTRLPIM01\_Q00001

Query Variant [Redacted]

Create Variant for BW Query

Option

Test Run

Execute Program now.

Retraction Query to Create Memo Records

Round (\*) 1 Round 1

Plan Year (\*) 2010 2010

Version (\*) WRK Working

Liquidity Item L320110\_FIRSTSTF HUT RELATED FIRST ST; HUT RELATED SECOND S; FASTER FEES; MISCELLANEOUS HUTF FE

Fiscal year/period (\*) 001/2010 To 012/2010

Execute Check [Icons] Cancel (F12)

Execute the query now.

Once you have Bex's Query data in ECC side (internal table `i_axis[]` as shown above), then we can process it further according to the particular requirement.

## Related Content

[www.help.sap.com](http://www.help.sap.com)

[www.sdn.sap.com](http://www.sdn.sap.com)

[www.service.sap.com](http://www.service.sap.com)

For more information, visit the [EDW 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.