Step By Step Process to Run BEX Report in Background

Applies to:

SAP BW 3.x.

Summary

This paper describes step by step process of implementing of a generic tool and how to run BEX query in background using that tool. It is accomplished through a Custom ABAP Program, which can be used to run any BEX query restricted by selection variant in background. The output is stored in a delimited text File in a specified application server directory.

Author: Dhiraj Kumar Thakur

Company: Accenture Service Pvt. Ltd

Created on: 6th October 2011

Author Bio

Dhiraj Kumar Thakur is currently with Accenture.He is a Business Intelligence Solution Consultant (NetWeaver 2004) and is mainly involved in Development, Support and Enhancement work.

Table of Contents

Requirement

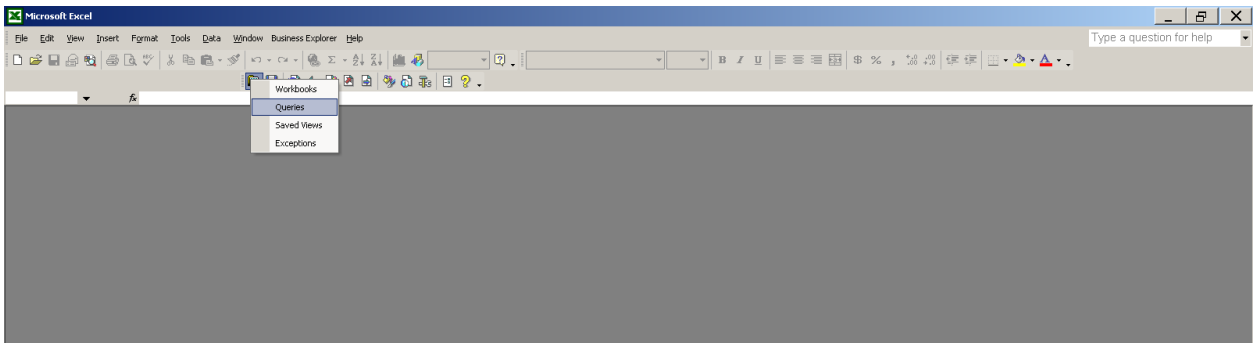
BEX Query Analyzer is the tool used to analyze ad hoc query results in MS Excel. BEX Query Analyzer can be used to run ad hoc queries and open BEX Query Designer to create and modify ad hoc queries. BEX Query Analyzer opens in MS Excel and provides its own toolbar and Business Explorer menu to perform query analysis.

Proposed Tool can be used in below situation:-

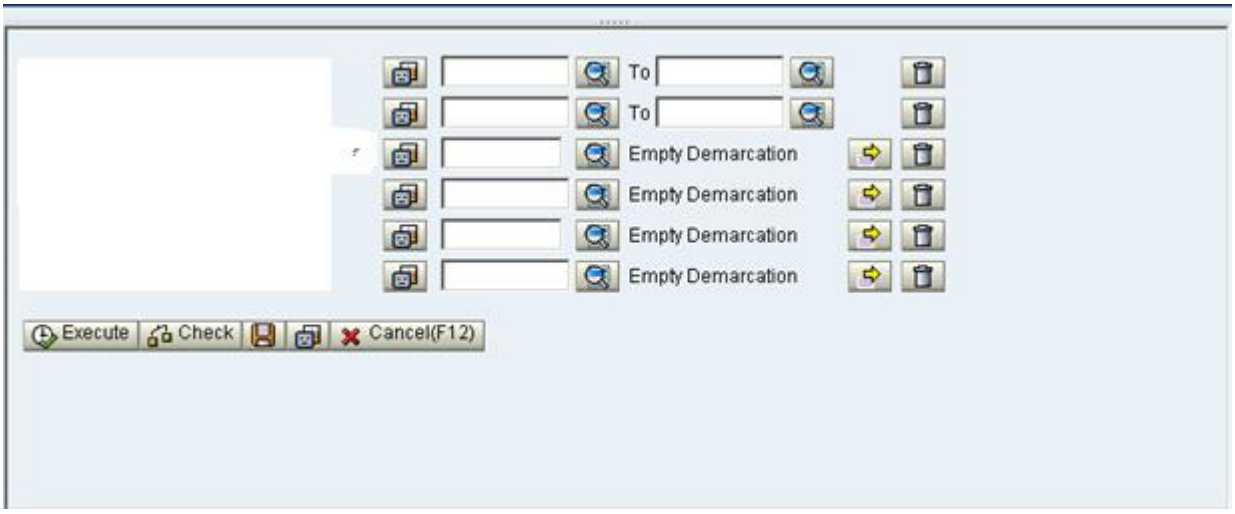
- Those environments, where Excel 2003 or below is installed for BEX reporting, which is having a limitation to display only 65 K rows.
- Run a BEX-Query in Batch-Mode (periodically and automatically) and download output in a directory, which can be read form another system.
- If the query is performed on detail level data and no further drill down is required on the output. Only Static information is required.
- As an alternate of Open Hub.
- If the query gives shortage of memory dump or exceeding the time limit dump due to high volume of data.

How BEX Query Run in Foreground

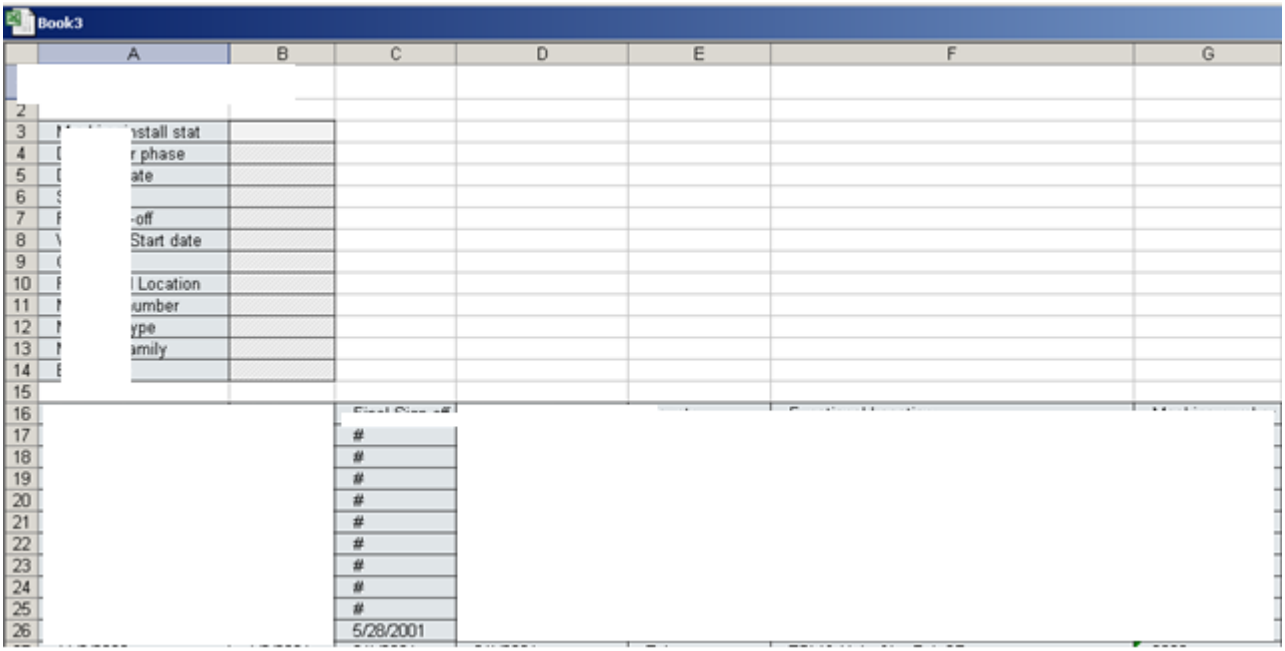
Run Tcode RRMX and click on open button. Click on query menu and select the query you want to run



Give the value for the selection field in the selection screen



Click on the execute button. Output is displayed in the Excel



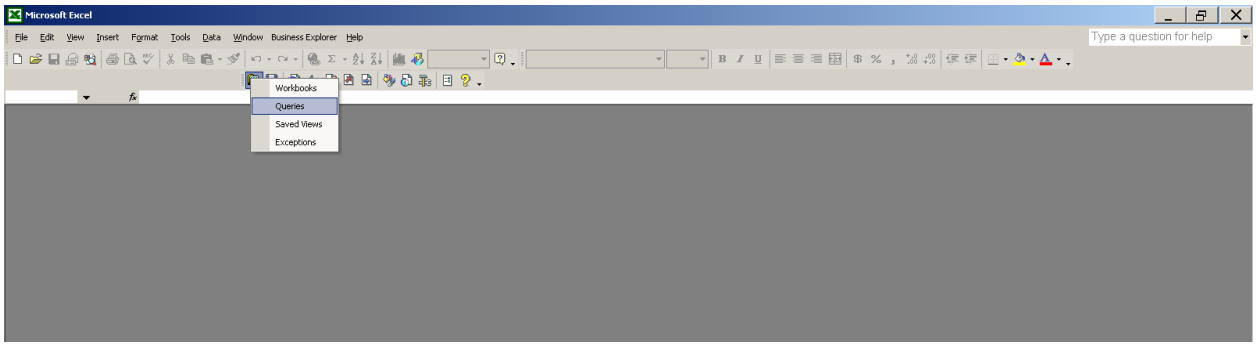
How this tool works

The program ZBWR_BEX_QUERY can be used to run any BEX query as a background job. User will be able to run the query on predefined selection values which are saved as a query variant.

Step by step process

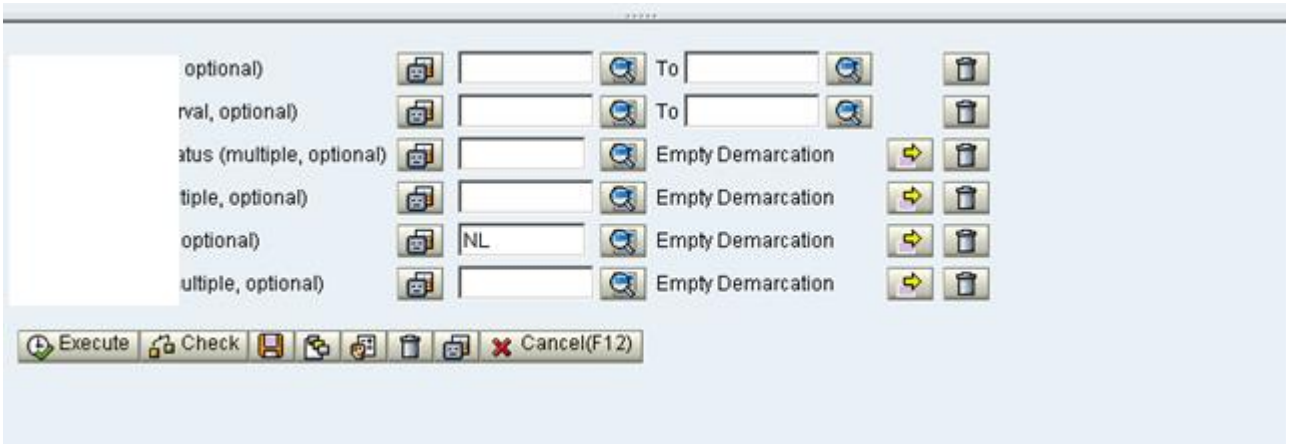
Step 1: Run BEX Query

Run Tcode RRMX and click on open button. It will be found on the Business Explorer toolbar. Click on query and select the query you want to run



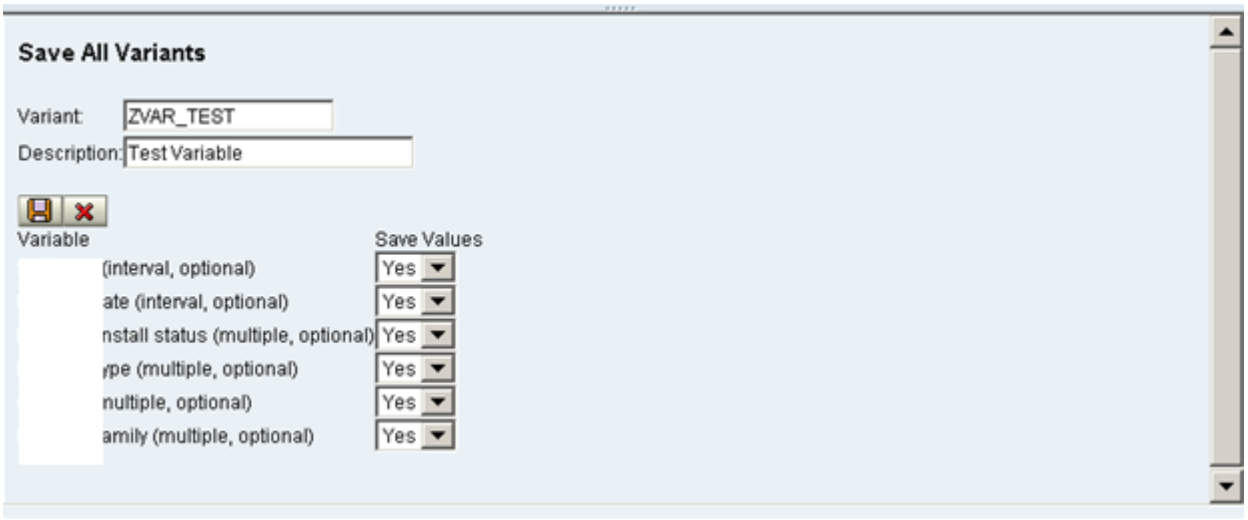
Step 2: Give the value in Selection screen

Give the value in the selection fields in input screen

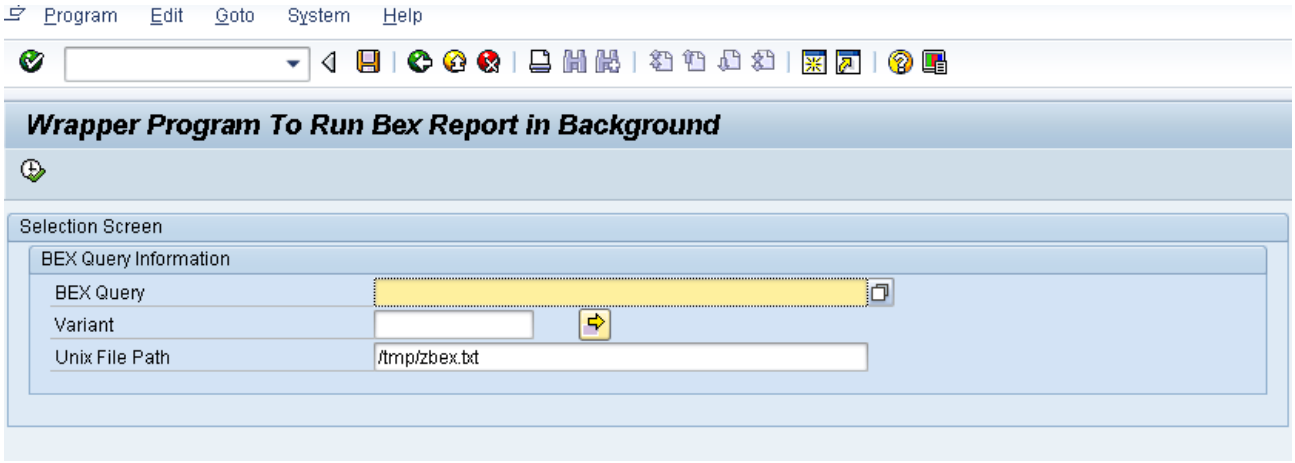


Step 3: Save the Variant

Click on the save button to save the selection value

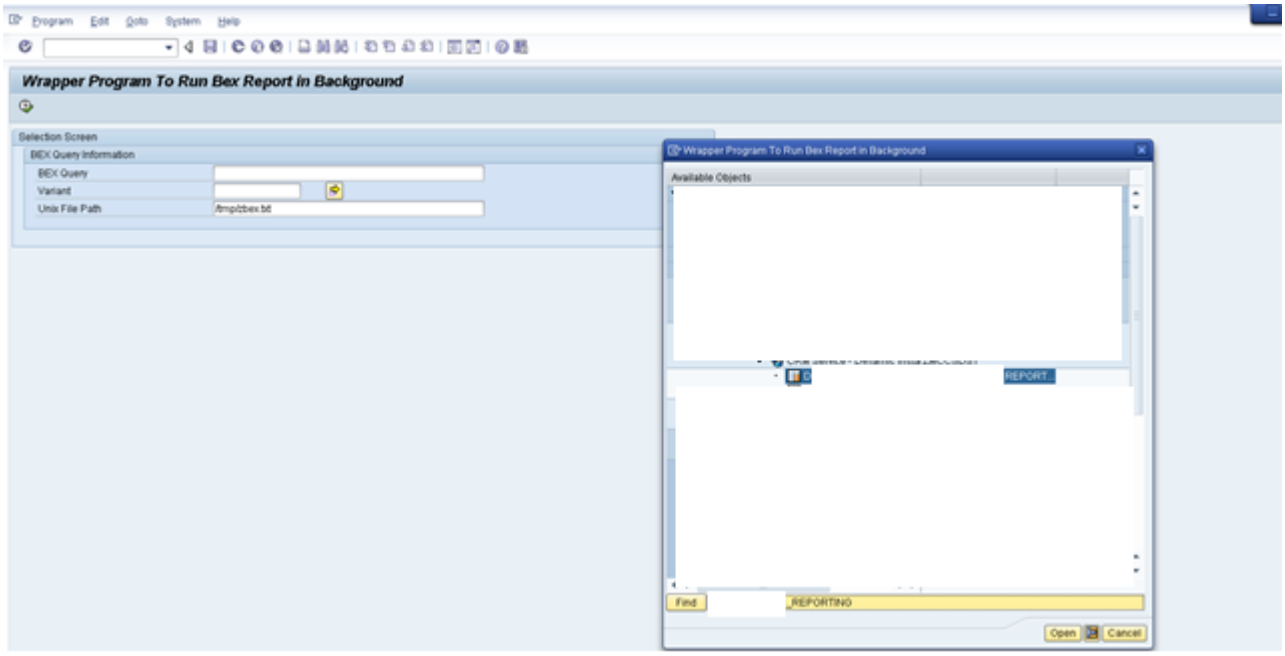


Step 4: Run the program ZBWR_BEX_QUERY (Refer Appendix for Source Code)



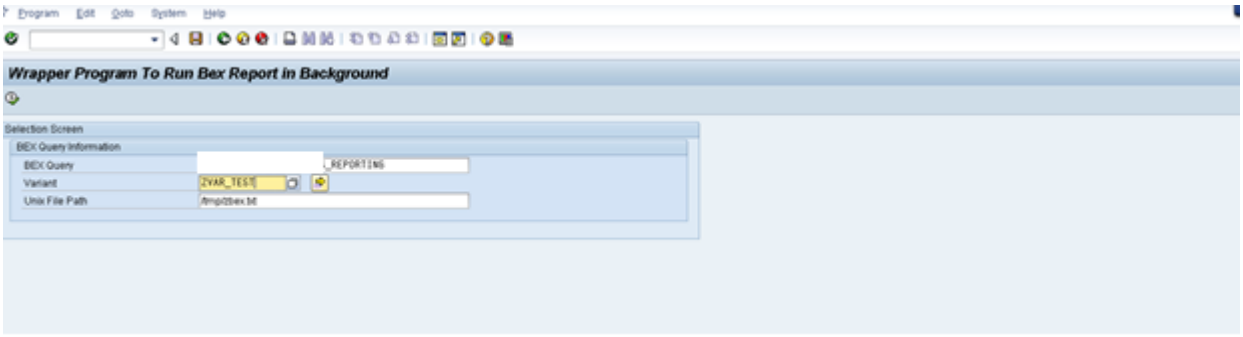
Step 5: Give Query Name

Search Query Name you want to Execute



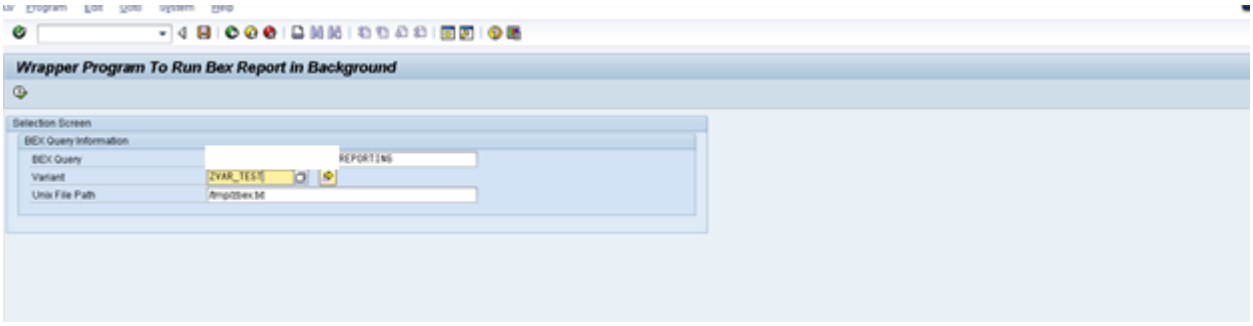
Step 6: Give Variant Name

Give the variant you saved in the previous step



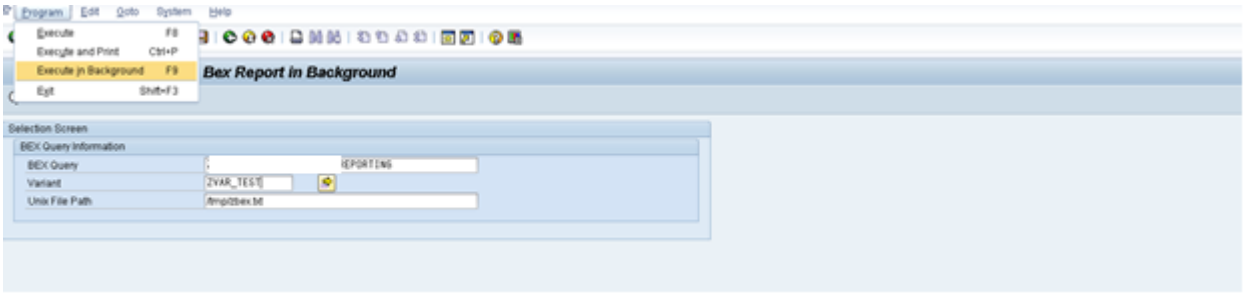
Step 7: Give File Path

Give the valid application server directory where you want to save the output file

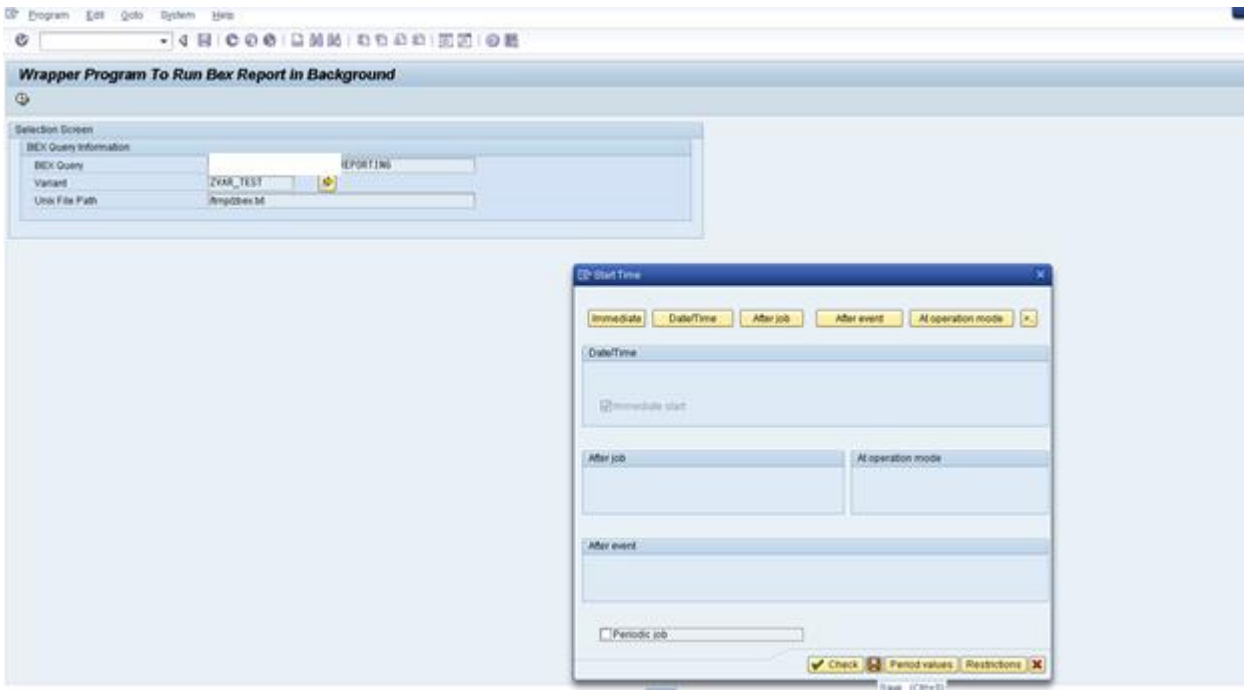


Step 8: Execute the Program in Background

Press F9 or click on program->Execute in Background

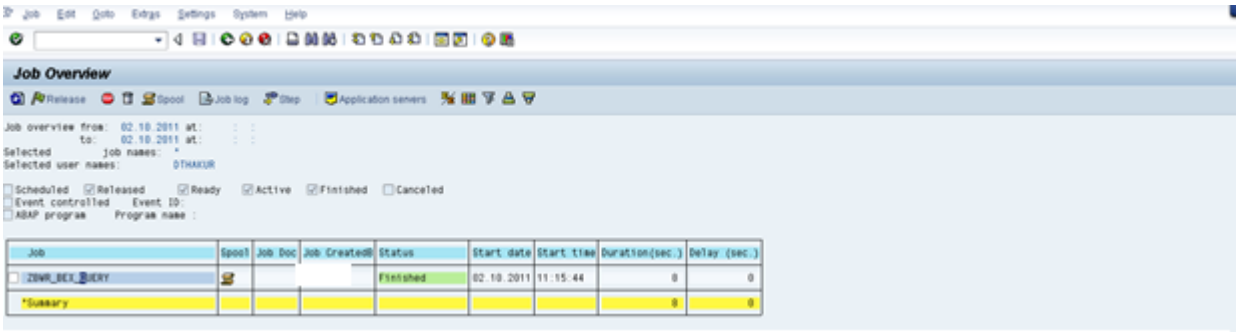


Click on Immediate or specify the Date/Time to execute the report



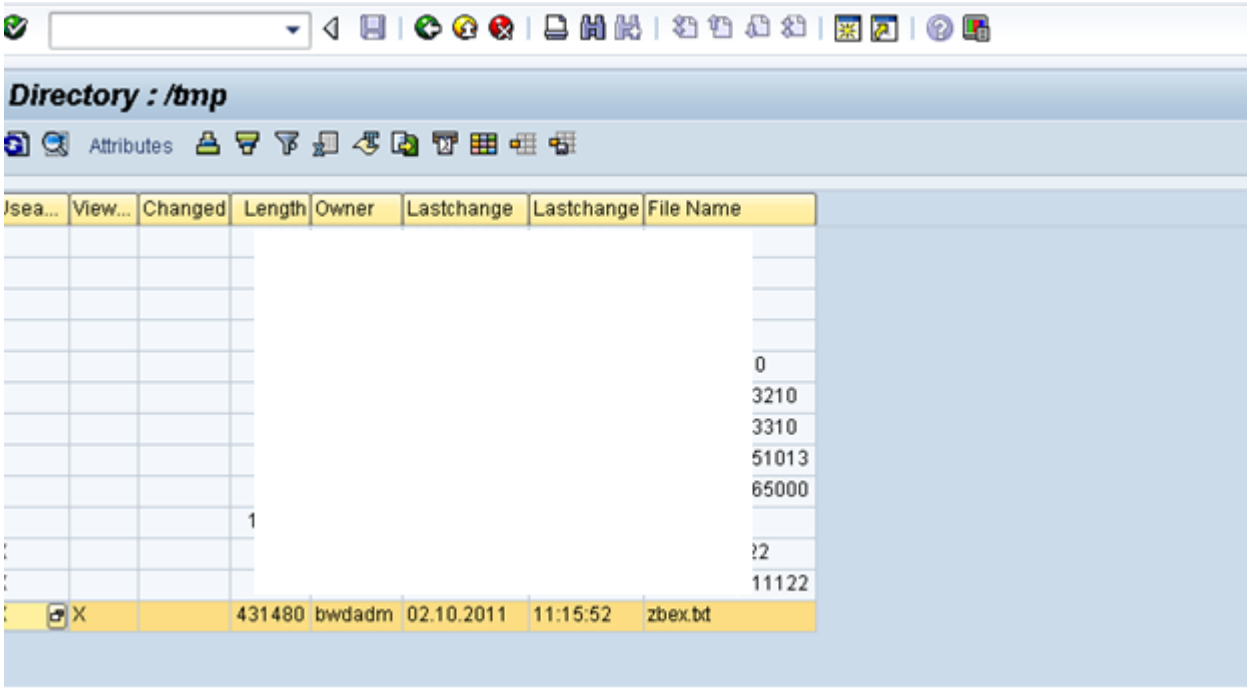
Step 9 : Check the Log

Run Tcode SM37 and check the job log and see if it is completed.



Step 10: Download the File

Run Tcode AL11 and check the file in the specified directory. You can download the file directly from here or OS level



Limitation:-

The only supported file type is delimited Text file. However, program can be twisted to save the output in other format.

APPENDIX:-

```
REPORT  ZBWR_BEX_QUERY      line-size 1023  message-id ZBW_ALL.
************************************************************************
*               Types Declarations
************************************************************************
TYPE-POOLS: rsr, rzi0, rrx1, rrms, rsbbs, rrkh, rro01, rsdd,
          rrx2, rsdm4, vrm.
TYPES: BEGIN OF g_s_col_width,
       column TYPE rrx1_s_grid-x,
       width  TYPE i,
      END OF g_s_col_width,
      g_t_col_widths TYPE g_s_col_width OCCURS 0.
 tables : varid.
************************************************************************
*               Constants Decalaration
************************************************************************
Constants:    c_x  type c value 'X',
              c_m  type c value 'M',
              c_w  type c value 'W',
              c_t  type c value 'T',
              c_n  type c value 'R',
              c_r  type c value 'R',
              c_s  type c value 'S'.
************************************************************************
*               Variable Declarations
************************************************************************
data: v_string(65000) type c.
DATA: g_th_dat_n       TYPE rrx1_th_dat_n,
      g_th_txt_n       TYPE rrx1_th_txt_n,
      g_gentime        TYPE rsr_s_rkb1d-gentime,
      g_s_grid_layout  TYPE rrx1_s_grid_layout,
      g_t_grid         TYPE rrx1_t_grid,
      g_t_ranges       TYPE rrx1_t_ranges,
      g_t_hry_types    TYPE rrx1_t_hry_types,
      g_t_ranges_out   TYPE rrx1_t_ranges,
      g_s_ranges       TYPE rrx1_s_ranges,
      g_dflt_req_lines TYPE i,
      g_t_mesg         TYPE rrx1_t_mesg,
      g_t_menu         TYPE rrx1_t_menu,
      g_fieldname(20)  TYPE c,
      g_t_repdir       LIKE rsrrepdir OCCURS 10,
      g_s_repkey       LIKE rszcompkey,
      g_handle         LIKE rrx_misc-handle,
      g_handle2        LIKE rrx_misc-handle,
      g_handle_int     LIKE rrx_grid-x,
      g_t_var          LIKE rrx_var OCCURS 0,
      g_t_var2         LIKE rrx_var OCCURS 0,
      g_t_msg          LIKE rrx_mesg OCCURS 0,
      g_t_msg2         LIKE rrx_mesg OCCURS 0,
      g_x              LIKE rrx_grid-x,
      g_y              LIKE rrx_grid-y,
      g_subrc          LIKE sy-subrc,
      g_ucomm          LIKE sy-ucomm,
      g_cmdid          TYPE rrx_menu-cmdid
                       value 'STRT', "ID of Excel context menu
      g_t_dim          TYPE rrx1_t_dim,
```

```
      g_t_atr         TYPE rrx1_t_atr,
      g_t_mem         TYPE rrx1_t_mem,
      g_t_con         TYPE rrx1_t_con,
      g_t_fac         TYPE rrx1_t_fac,
      g_t_cel         TYPE rrx1_t_cel,
      g_s_prptys      TYPE rrx1_s_prptys,
      g_t_prptys      TYPE rrx1_t_prptys,
      g_s_debugflags  TYPE rsr_s_debugflags,
      g_t_col_widths  TYPE g_t_col_widths,
      g_bbs_receiver  TYPE rsbbsreceiver,
      g_t_genuniid    TYPE w3html OCCURS 0,
      g_iobjnm        TYPE rs_char30,
      g_row           TYPE rsint4,
      g_row_hide      TYPE rsint4,
      g_start_row     TYPE rsint4,
      g_col           TYPE rsint4,
      g_cancel        TYPE rs_bool,
      g_ok(20)        TYPE c,
      g_chavl         TYPE rsd_chavl,
      g_r_e           TYPE REF TO cx_root,
      g_cmd_count     TYPE i.

DATA: g_txt1(50)   TYPE c,
      g_txt2(50)   TYPE c,
      g_txt3(50)   TYPE c,
      g_txt4(50)   TYPE c,
      g_txt5(50)   TYPE c,
      g_txt6(50)   TYPE c,
      g_txt7(50)   TYPE c,
      g_txt8(50)   TYPE c,
      g_txt9(50)   TYPE c,
      g_txt10(50)  TYPE c,
      g_txt11(50)  TYPE c,
      g_txt12(50)  TYPE c,
      g_txt13(50)  TYPE c,
      g_txt14(50)  TYPE c,
      g_txt15(50)  TYPE c,
      g_txt16(50)  TYPE c,
      g_txt17(50)  TYPE c,
      g_txt18(50)  TYPE c,
      g_txt19(50)  TYPE c,
      g_txt20(50)  TYPE c,
      g_txt21(50)  TYPE c,
      g_txt22(50)  TYPE c,
      g_txt23(50)  TYPE c,
      g_txt24(50)  TYPE c,
      g_txt25(50)  TYPE c,
      g_txt26(50)  TYPE c,
      g_txt27(50)  TYPE c,
      g_txt28(50)  TYPE c,
      g_txt29(50)  TYPE c,
      g_txt30(50)  TYPE c,
      g_txt31(50)  TYPE c,
      g_txt32(50)  TYPE c,
      g_txt33(50)  TYPE c,
      g_txt34(50)  TYPE c,
      g_txt35(50)  TYPE c,
      g_txt36(50)  TYPE c,
```

```
      g_txt37(50)  TYPE c,
      g_txt38(50)  TYPE c,
      g_txt39(50)  TYPE c,
      g_txt40(50)  TYPE c,
      g_txt41(50)  TYPE c,
      g_txt42(50)  TYPE c,
      g_txt43(50)  TYPE c,
      g_txt44(50)  TYPE c,
      g_txt45(50)  TYPE c,
      g_txt46(50)  TYPE c,
      g_txt47(50)  TYPE c,
      g_txt48(50)  TYPE c,
      g_txt49(50)  TYPE c,
      g_txt50(50)  TYPE c,
      g_txt51(50)  TYPE c,
      g_txt52(50)  TYPE c,
      g_txt53(50)  TYPE c,
      g_txt54(50)  TYPE c,
      g_txt55(50)  TYPE c,
      g_txt56(50)  TYPE c,
      g_txt57(50)  TYPE c,
      g_txt58(50)  TYPE c,
      g_txt59(50)  TYPE c,
      g_txt60(50)  TYPE c,
      g_txt61(50)  TYPE c,
      g_txt62(50)  TYPE c,
      g_txt63(50)  TYPE c,
      g_txt64(50)  TYPE c.

data : G_compid like RSZCOMPKEY-compid.

DATA: v_genuniid TYPE rscrmrepuid        ,
      v_reportuid   TYPE rscrmrepuid      ,
      v_num type i                        ,
      v_EVENT like RSMPE-FUNC             ,
     v_CUBENAME like BAPI6110CUB-CUBE_NAM ,
      v_CATALOG like  BAPI6110CAT-CAT_NAM  ,
      v_REPNAME like RSCRMREPDEF-REPNAME   ,
      v_REPTXT like RSCRMREPDEF-REPTXT     ,
      v_varient type RSVAR-VARIANT         ,
      v_report TYPE RSVAR-REPORT           ,
      v_RSR_S_RKB1D like RSR_S_RKB1D      .


************************************************************************
*              Declaration of work areas
************************************************************************
DATA: BEGIN OF w_session,
       reportuid TYPE rscrmrepuid,
       sessionid TYPE rscrmsession,
      END OF w_session.

data : wa_infocube like RSZCOMPKEY-infocube.

************************************************************************
*          Declaration of Selection Screen
************************************************************************
SELECTION-SCREEN BEGIN OF BLOCK BLK1 WITH FRAME TITLE TEXT-000.
```

```
  SELECTION-SCREEN BEGIN OF BLOCK B1 WITH FRAME TITLE TEXT-001.
    PARAMETERS: p_genuid LIKE rsrrepdir-genuniid.
   select-options: s_varnt for VARID-VARIANT  no intervals.
    PARAMETERS: P_OFILE(80)  DEFAULT '/tmp/zbex.txt'
                LOWER CASE.         "Down load file

  SELECTION-SCREEN END OF BLOCK B1.

SELECTION-SCREEN END OF BLOCK BLK1.

************************************************************************
*           Initializaion event
************************************************************************
Initialization.

************************************************************************
*          At selection screen event
************************************************************************
AT SELECTION-SCREEN  ON VALUE-REQUEST FOR p_genuid.
*get Bex query list
*by using this FM, we get the infocube and Internal display of the
*report identifier, UUID in compressed form and the bex report name
*the table RSRREPDIR contains all bex report details
  CALL FUNCTION 'RSZ_COMPONENT_TREE_GET'
    EXPORTING
      i_title     =  sy-title
    IMPORTING
      E_INFOCUBE  =  wa_infocube
      E_COMPID    =  g_compid
      e_genuniid  =  p_genuid
      e_infocube_txt = v_REPTXT
    EXCEPTIONS
      OTHERS      = 1.

*AT SELECTION-SCREEN  ON VALUE-REQUEST FOR s_varnt-low.
*** variant of the selected query can be picked up from this FM.
*  CALL FUNCTION 'RRS_VARIANT_F4'
*    EXPORTING
*      I_GENUNIID           = p_genuid
*      I_INFOCUBE           = wa_infocube
*      I_COMPID             = g_compid
*    IMPORTING
*      E_VARIANT            = p_varnt
*    EXCEPTIONS
*      REPORT_NOT_FOUND     = 1
*      NO_VARIANTS_POSSIBLE = 2
*      VARIANT_NOT_FOUND    = 3
*      CANCELED             = 4
*      OTHERS               = 5.

AT SELECTION-SCREEN.
*validating the file name and file path
  OPEN DATASET P_OFILE FOR OUTPUT IN  TEXT MODE ENCODING DEFAULT.
  IF SY-SUBRC <> 0.
    MESSAGE E000 WITH 'File Could not be Open'(002).
  else.
    close dataset p_ofile.
  endif.
```

```
* validate if the user enters the range instead single values
if not s_varnt[] is initial.
 read table s_varnt with key option = 'BT'.  "option = BT for range vals
  if sy-subrc = 0.
message e000 with 'Enter single values for variant instead ranges'(005).
  endif.
endif.

************************************************************************
*          START-OF-SELECTION EVENT
************************************************************************
START-OF-SELECTION.
* Call to function module to open the session
  perform sub_getdata.
************************************************************************
* END-OF-SELECTION EVENT
************************************************************************
END-OF-SELECTION.


*&---------------------------------------------------------------------*
*&      Form  sub_opensession
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
FORM sub_getdata .
* open the output file for data download, if not raise error
OPEN DATASET P_OFILE FOR OUTPUT IN  TEXT MODE ENCODING DEFAULT.
  IF SY-SUBRC <> 0.
    MESSAGE E000(zz) WITH 'File Could not be Open'.
  endif.
* generate the report for multiple variant values.
loop at s_varnt.

      refresh: g_t_ranges,g_t_dim, g_t_mem, g_t_cel,g_t_prptys,
      g_t_atr,g_t_grid,g_t_con,g_t_fac,g_t_var.

   clear v_RSR_S_RKB1D.
*getting the handle
  CALL FUNCTION 'RRX_REPORT_OPEN'
    EXPORTING
      i_genuniid       = p_genuid
    IMPORTING
      e_handle         = g_handle
    EXCEPTIONS
      open_failed      = 1
      invalid_genuniid = 2
      msg_init_failed  = 3
      inherited_error  = 4
      x_message        = 5
      OTHERS           = 6.

*To hide the popup of selection screen of the selected query
  CALL FUNCTION 'RRC_VARIABLES_START'
    EXPORTING
      I_HANDLE              = g_handle
      I_SELSCRN              = RSR_C_SELSCRN-INACT
*     I_VARIANT             = p_varnt
```

```
      I_VARIANT             = s_varnt-low " low value of variant
      I_PMODE               = ' '
      I_NEW_VARIANT         = RS_C_FALSE
    IMPORTING
      E_S_RKB1D             = v_RSR_S_RKB1D.

  IF SY-SUBRC <> 0.
    MESSAGE E000 WITH 'Problem in getting value of variant'(003).
  ENDIF.

*getting the  BEX query results
  CALL FUNCTION 'RRX_GRID_CMD_PROCESS'
    EXPORTING
      i_handle          = g_handle
      i_cmdid           = g_cmdid
      i_iobjnm          = g_iobjnm
    IMPORTING
      e_max_x           = g_x
      e_max_y           = g_y
    TABLES
      i_t_ranges        = g_t_ranges
      e_t_dim           = g_t_dim
      e_t_mem           = g_t_mem
      e_t_cel           = g_t_cel
      c_t_prptys        = g_t_prptys
      e_t_atr           = g_t_atr
      e_t_grid          = g_t_grid
      e_t_ranges        = g_t_ranges
      e_t_con           = g_t_con
      e_t_fac           = g_t_fac
      e_t_var           = g_t_var
    EXCEPTIONS
      inherited_error    = 1
      no_record_found    = 2
      terminated_by_user = 3
      no_processing      = 4
      no_change          = 5
      dbcl_nosupport     = 6
      no_authorization   = 7
      x_message          = 8
      screen_canceled    = 9
      launch_url         = 10
      OTHERS             = 11.
  .
  IF SY-SUBRC <> 0.
     MESSAGE E000 WITH 'Problem in getting BEx query results'(004).
  ENDIF.
  perform sub_report_genarate.   " generate report

CALL FUNCTION 'RRX_REPORT_CLOSE'
  EXPORTING
    I_HANDLE          = g_handle.
         .
   IF SY-SUBRC <> 0.
* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
*        WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
  ENDIF.
 endloop.
```

```
* close the output file
close dataset p_ofile.  " close the output file
ENDFORM.                    " sub_getdata
*&---------------------------------------------------------------------*
*&      Form  sub_report_genarate
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
FORM sub_report_genarate .
*determined the column windth of the report
  PERFORM sub_determine_col_widths CHANGING g_t_grid
                                   g_t_col_widths.

*write the output in the file/list
  PERFORM sub_write_grid CHANGING g_t_grid.
ENDFORM.                    " sub_report_genarate



*&---------------------------------------------------------------------*
*&      Form  sub_determine_col_widths
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*      -->C_T_GRID   text
*      -->C_T_COL_WIDtext
*----------------------------------------------------------------------*
FORM sub_determine_col_widths
             CHANGING c_t_grid       TYPE rrx1_t_grid
                      c_t_col_widths TYPE g_t_col_widths.

  DATA: l_s_grid      TYPE rrx1_s_grid,
        l_s_grid_last TYPE rrx1_s_grid,
        l_s_col_width TYPE g_s_col_width,
        l_len         TYPE i,
        l_maxwidth    TYPE i VALUE 25.                    "MST

  SORT c_t_grid BY x.

  LOOP AT c_t_grid INTO l_s_grid.
    AT NEW x.
      CLEAR: l_s_grid_last, l_s_col_width.
      l_s_col_width-column = l_s_grid-x.
    ENDAT.
    l_len = strlen( l_s_grid-data ).
* Länge wird abhängig vom mwkz erweitert                MST
    CASE l_s_grid-mwkz.
      WHEN c_m.
        IF l_s_grid-y NE 1.           "Keine Zahlen in Zeile 1
          l_len = l_len + 5.
        ENDIF.
      WHEN C_w.
        IF l_s_grid-y NE 1.           "Keine Zahlen in Zeile 1
          l_len = l_len + 4.
        ENDIF.
```

```
      WHEN OTHERS.
        l_len = l_len + 0.
    ENDCASE.
* Ende einfügen
    IF l_len GT l_s_col_width-width.
      IF l_len GE l_maxwidth.
        l_s_col_width-width = l_maxwidth.
      ELSE.
        l_s_col_width-width = l_len.
      ENDIF.
    ENDIF.
    AT END OF x.
      APPEND l_s_col_width TO c_t_col_widths.
    ENDAT.
    l_s_grid_last = l_s_grid.
  ENDLOOP.

ENDFORM.                              " sub_DETERMINE_COL_WIDTHS

*&---------------------------------------------------------------------*
*&      Form  WRITE_GRID
*&---------------------------------------------------------------------*
FORM sub_write_grid CHANGING c_t_grid TYPE rrx1_t_grid.

  DATA: l_s_grid      TYPE rrx1_s_grid,
        l_s_grid2     TYPE rrx1_s_grid,
        l_s_grid_last TYPE rrx1_s_grid,
        l_tabix       TYPE i.

  DATA: l_s_col_width  TYPE g_s_col_width,
        l_width        TYPE i,
        l_position     TYPE i,
        l_x            TYPE i,
        l_y            TYPE i,
        l_max_x        TYPE i,
        l_max_y        TYPE i,
        l_help_position     TYPE i,
        l_cell_type    TYPE c,
        l_cell_text(50) TYPE c,       " MST Länge unschön
        l_cell_add(4) TYPE c.         " MST

  g_start_row = sy-linno.

  CLEAR: g_t_ranges[].

  SORT c_t_grid BY y x.

  LOOP AT c_t_grid INTO l_s_grid.
    IF l_s_grid-y > l_max_y.
      l_max_y = l_s_grid-y.
    ENDIF.
    IF l_s_grid-x > l_max_x.
      l_max_x = l_s_grid-x.
    ENDIF.
  ENDLOOP.

  l_tabix = 1.
  READ TABLE c_t_grid INTO l_s_grid2 INDEX 1.
```

```
  DO l_max_y TIMES.
    l_y = sy-index.
    DO l_max_x TIMES.
      l_x = sy-index.

      WHILE    l_y > l_s_grid2-y
          OR ( l_y = l_s_grid2-y
           AND l_x > l_s_grid2-x ).
        ADD 1 TO l_tabix.
        READ TABLE c_t_grid INTO l_s_grid2 INDEX l_tabix.
        IF sy-subrc NE 0.
          EXIT.
        ENDIF.
      ENDWHILE.

      IF l_s_grid2-y = l_y AND l_s_grid2-x = l_x.
        l_s_grid = l_s_grid2.
      ELSE.
        CLEAR: l_s_grid.
        l_s_grid-x = l_x.
        l_s_grid-y = l_y.
        l_s_grid-content = 'NN'.
      ENDIF.

      IF l_s_grid_last-y < l_s_grid-y.
        g_row_hide = l_s_grid_last-y.
        HIDE g_row_hide.
        CLEAR l_position.
        NEW-LINE.
      ENDIF.

      READ TABLE g_t_col_widths INTO l_s_col_width INDEX l_s_grid-x.
      l_width = l_s_col_width-width.
      l_position = l_position + l_width.
* format
* Ausgabeattribute in Abhängigkeit vom Cell-style setzen.
      l_help_position = l_position + 20.
*     CHECK l_help_position LT sy-linsz.

      IF l_s_grid-content NE l_s_grid_last-content.
        PERFORM sub_set_format_attributes USING l_s_grid-content.
      ENDIF.

      l_cell_type = l_s_grid-content+1(1).
* Eingefügt zur Anzeige von Währung/Menge.
      CLEAR l_cell_add.
      CASE l_s_grid-mwkz.
        WHEN c_m.
          IF l_s_grid-y NE 1.           "keine Zahlen in Zeile 1
            l_cell_add = l_s_grid-unit.
          ENDIF.
        WHEN c_w.
          IF l_s_grid-y NE 1.           "keine Zahlen in Zeile 1
            l_cell_add = l_s_grid-currency.
          ENDIF.
        WHEN OTHERS.
          CLEAR l_cell_add.
      ENDCASE.
```

```
      CONCATENATE l_s_grid-data l_cell_add INTO l_cell_text
                SEPARATED BY ' '.
* Ende Eingefügt
      IF l_s_grid-x EQ 1.
        WRITE l_s_grid-drillstate.
      ENDIF.



      PERFORM sub_write_cell_col USING l_width
                                 l_s_grid-x
                                 l_cell_text.
*    WRITE  AT (l_width) l_cell_text.                    "MST

      data : l_temp(1000) type c,
             l_len type i.

      if l_cell_text = space.
        l_temp = '|'.

      else.
        CONCATENATE l_cell_text '|' into l_temp.
      endif.

      CONCATENATE  v_string l_temp  into v_string.

      l_s_grid_last = l_s_grid.
    ENDDO.
    l_len = strlen( v_string ).
    transfer v_string to p_ofile length l_len.
    clear v_string.
  ENDDO.

ENDFORM.                              " sub_WRITE_GRID

*&---------------------------------------------------------------------*
*&      Form  set_format_attributes
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*      -->I_CONTENT  text
*----------------------------------------------------------------------*
FORM sub_set_format_attributes
              USING    i_content      TYPE rrx1_s_grid-content.

  DATA: l_color        TYPE c,
        l_intensified  TYPE rs_bool,
        l_inverse      TYPE rs_bool,
        l_linetype     TYPE c,
        l_itemtype     TYPE c.

  l_linetype = i_content+1(1).
  l_itemtype = i_content+0(1).

  CASE l_itemtype.
    WHEN c_n.
      CASE l_linetype.
        WHEN c_t. l_color = 3.
```

```
      WHEN c_n. l_color = 2.
      WHEN c_r. l_color = 2.
      WHEN c_s. l_color = 2.
    ENDCASE.
  WHEN c_t.
    CASE l_linetype.
      WHEN c_t. l_color = 3.
      WHEN c_n. l_color = 4.
      WHEN c_r. l_color = 4.
      WHEN c_s. l_color = 4.
    ENDCASE.
ENDCASE.


CASE l_inverse.
  WHEN rs_c_true.
  WHEN rs_c_false.
ENDCASE.

CASE l_intensified.
  WHEN rs_c_true.
  WHEN rs_c_false.
ENDCASE.

CASE i_content+2(2).
  WHEN '01'. l_color = 5.
  WHEN '02'. l_color = 5.
  WHEN '03'. l_color = 5.
  WHEN '04'. l_color = 7.
  WHEN '05'. l_color = 7.
  WHEN '06'. l_color = 7.
  WHEN '07'. l_color = 6.
  WHEN '08'. l_color = 6.
  WHEN '09'. l_color = 6.
ENDCASE.

CASE l_color.
  WHEN 0. FORMAT COLOR COL_BACKGROUND.
  WHEN 1. FORMAT COLOR COL_HEADING.
  WHEN 2. FORMAT COLOR COL_NORMAL.
  WHEN 3. FORMAT COLOR COL_TOTAL.
  WHEN 4. FORMAT COLOR COL_KEY.
  WHEN 5. FORMAT COLOR COL_POSITIVE.
  WHEN 6. FORMAT COLOR COL_NEGATIVE.
  WHEN 7. FORMAT COLOR COL_GROUP.
ENDCASE.

ENDFORM.                    " sub_GET_FORMAT_ATTRIBUTES

*---------------------------------------------------------------------*
*       FORM sub_write_cell_col
*---------------------------------------------------------------------*
*       .....                                                         *
*---------------------------------------------------------------------*
*  -->  I_WIDTH                                                       *
*  -->  I_COL                                                         *
*  -->  L_TEXT                                                        *
*---------------------------------------------------------------------*
```

```
FORM sub_write_cell_col USING i_width TYPE i
                      i_col   TYPE i
                      l_text  TYPE c.

  FORMAT INTENSIFIED OFF.

  CASE i_col.
    WHEN 1.  g_txt1  = l_text. WRITE AT (i_width) g_txt1.
    WHEN 2.  g_txt2  = l_text. WRITE AT (i_width) g_txt2.
    WHEN 3.  g_txt3  = l_text. WRITE AT (i_width) g_txt3.
    WHEN 4.  g_txt4  = l_text. WRITE AT (i_width) g_txt4.
    WHEN 5.  g_txt5  = l_text. WRITE AT (i_width) g_txt5.
    WHEN 6.  g_txt6  = l_text. WRITE AT (i_width) g_txt6.
    WHEN 7.  g_txt7  = l_text. WRITE AT (i_width) g_txt7.
    WHEN 8.  g_txt8  = l_text. WRITE AT (i_width) g_txt8.
    WHEN 9.  g_txt9  = l_text. WRITE AT (i_width) g_txt9.
    WHEN 10. g_txt10 = l_text. WRITE AT (i_width) g_txt10.
    WHEN 11. g_txt11 = l_text. WRITE AT (i_width) g_txt11.
    WHEN 12. g_txt12 = l_text. WRITE AT (i_width) g_txt12.
    WHEN 13. g_txt13 = l_text. WRITE AT (i_width) g_txt13.
    WHEN 14. g_txt14 = l_text. WRITE AT (i_width) g_txt14.
    WHEN 15. g_txt15 = l_text. WRITE AT (i_width) g_txt15.
    WHEN 16. g_txt16 = l_text. WRITE AT (i_width) g_txt16.
    WHEN 17. g_txt17 = l_text. WRITE AT (i_width) g_txt17.
    WHEN 18. g_txt18 = l_text. WRITE AT (i_width) g_txt18.
    WHEN 19. g_txt19 = l_text. WRITE AT (i_width) g_txt19.
    WHEN 20. g_txt20 = l_text. WRITE AT (i_width) g_txt20.
    WHEN 21. g_txt21 = l_text. WRITE AT (i_width) g_txt21.
    WHEN 22. g_txt22 = l_text. WRITE AT (i_width) g_txt22.
    WHEN 23. g_txt23 = l_text. WRITE AT (i_width) g_txt23.
    WHEN 24. g_txt24 = l_text. WRITE AT (i_width) g_txt24.
    WHEN 25. g_txt25 = l_text. WRITE AT (i_width) g_txt25.
    WHEN 26. g_txt26 = l_text. WRITE AT (i_width) g_txt26.
    WHEN 27. g_txt27 = l_text. WRITE AT (i_width) g_txt27.
    WHEN 28. g_txt28 = l_text. WRITE AT (i_width) g_txt28.
    WHEN 29. g_txt29 = l_text. WRITE AT (i_width) g_txt29.
    WHEN 30. g_txt30 = l_text. WRITE AT (i_width) g_txt30.
    WHEN 31. g_txt31 = l_text. WRITE AT (i_width) g_txt31.
    WHEN 32. g_txt32 = l_text. WRITE AT (i_width) g_txt32.
    WHEN 33. g_txt33 = l_text. WRITE AT (i_width) g_txt33.
    WHEN 34. g_txt34 = l_text. WRITE AT (i_width) g_txt34.
    WHEN 35. g_txt35 = l_text. WRITE AT (i_width) g_txt35.
    WHEN 36. g_txt36 = l_text. WRITE AT (i_width) g_txt36.
    WHEN 37. g_txt37 = l_text. WRITE AT (i_width) g_txt37.
    WHEN 38. g_txt38 = l_text. WRITE AT (i_width) g_txt38.
    WHEN 39. g_txt39 = l_text. WRITE AT (i_width) g_txt39.
    WHEN 40. g_txt30 = l_text. WRITE AT (i_width) g_txt40.
    WHEN 41. g_txt41 = l_text. WRITE AT (i_width) g_txt41.
    WHEN 42. g_txt42 = l_text. WRITE AT (i_width) g_txt42.
    WHEN 43. g_txt43 = l_text. WRITE AT (i_width) g_txt43.
    WHEN 44. g_txt44 = l_text. WRITE AT (i_width) g_txt44.
    WHEN 45. g_txt45 = l_text. WRITE AT (i_width) g_txt45.
    WHEN 46. g_txt46 = l_text. WRITE AT (i_width) g_txt46.
    WHEN 47. g_txt47 = l_text. WRITE AT (i_width) g_txt47.
    WHEN 48. g_txt48 = l_text. WRITE AT (i_width) g_txt48.
    WHEN 49. g_txt49 = l_text. WRITE AT (i_width) g_txt49.
    WHEN 50. g_txt50 = l_text. WRITE AT (i_width) g_txt50.
    WHEN 51. g_txt51 = l_text. WRITE AT (i_width) g_txt51.
```

```
    WHEN 52. g_txt52 = l_text. WRITE AT (i_width) g_txt52.
    WHEN 53. g_txt53 = l_text. WRITE AT (i_width) g_txt53.
    WHEN 54. g_txt54 = l_text. WRITE AT (i_width) g_txt54.
    WHEN 55. g_txt55 = l_text. WRITE AT (i_width) g_txt55.
    WHEN 56. g_txt56 = l_text. WRITE AT (i_width) g_txt56.
    WHEN 57. g_txt57 = l_text. WRITE AT (i_width) g_txt57.
    WHEN 58. g_txt58 = l_text. WRITE AT (i_width) g_txt58.
    WHEN 59. g_txt59 = l_text. WRITE AT (i_width) g_txt59.
    WHEN 60. g_txt60 = l_text. WRITE AT (i_width) g_txt60.
    WHEN 61. g_txt61 = l_text. WRITE AT (i_width) g_txt61.
    WHEN 62. g_txt62 = l_text. WRITE AT (i_width) g_txt62.
    WHEN 63. g_txt63 = l_text. WRITE AT (i_width) g_txt63.
    WHEN 64. g_txt64 = l_text. WRITE AT (i_width) g_txt64.
    WHEN OTHERS. WRITE AT (i_width) l_text.
  ENDCASE.

ENDFORM.                    "sub_write_cell_col
```

Related Content

RUN BEX QUERY

BEX In Background

Run BEX queries in background or batch

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.