

Display Objects in Transport



Applies to:

SAP Net Weaver BI 3.x and 7.0

Summary

The Purpose of this document is to implement a utility that will search and display Standard Objects and Bex Object that are contain in SAP Transports.

Author: Sandeep Dak

Company: Accenture

Created on: 03-September-2009

Author Bio



Sandeep Dak is currently working with Accenture. He is Business Intelligence Solution Architect With 6 years of SAP-BI Experience.

Table of Contents

Document Detail	3
Source Code	3
Defining Text Element.....	19
An Example	21
Related Content.....	22
Disclaimer and Liability Notice.....	23

Document Detail

In most cases, Business Information Warehouse development projects are not carried out in a productive system, rather in one or more development systems, depending on their scope. If your development projects are done in one development system you have to transport the developments to the target system (a test or productive system). However BI Transports are completely object dependable and there is no standard tool available which gives complete transport view at one place. This tool will help out in cutover phase and gives complete view of the object collected under transport.

Source Code

```

*-----*
* Program Name      : YBW_DOC_OBJECTS_IN_TRANSPORTS*
* Version
  : Version 1.0
* Last Update      : 03-September-
2009
* Specifier         : Sandeep Dak*
* Program Type     : ABAP
*-----*
* Description
*
* Program to Display the following in a selection of transports :-
*
*   - Standard SAP Objects
*   - Bex Elements and Workbooks
*-----*

```

PROGRAM YBW_DOC_OBJECTS_IN_TRANSPORTS.

```

*-----*
* Macros
*-----*
DEFINE insert_long_field.
  clear ls_fcat.
  ls_fcat-fieldname = &1.
  ls_fcat-seltext_1 = &2.
  append ls_fcat to lt_fieldcat.
END-OF-DEFINITION.

DEFINE insert_hotspot_field.
  clear ls_fcat.
  ls_fcat-fieldname = &1.
  ls_fcat-seltext_1 = &2.
  ls_fcat-hotspot = 'X'.
  append ls_fcat to lt_fieldcat.
END-OF-DEFINITION.

DEFINE add_order.
  gp_tkorr-low = &1.
  gp_tkorr-option = 'CP'.
  gp_tkorr-sign = 'I'.
  append gp_tkorr.
END-OF-DEFINITION.

```

```

*-----*
* Constants
*-----*

CONSTANTS: c_bex_elements TYPE trobjtype VALUE 'ELEM',
            c_workbook     TYPE trobjtype VALUE 'XLWB',
            c_comment      TYPE trobjtype VALUE 'RELE',
            c_transports   TYPE trkorr    VALUE 'WDGK9*'.

*-----*
* Global Types
*-----*

TABLES: e070, " Transport Header
        e071. " Transport Details

TYPE-POOLS: slis.
*
* Standard Objects ALV Display
*
TYPES: BEGIN OF t_std_tkorr,
        trkorr    TYPE trkorr,
        as4text   TYPE as4text,
        object    TYPE trobjtype,
        text      TYPE ddttext,
        obj_name  TYPE trobj_name,
        objfunc   TYPE objfunc,
        objfunct  TYPE ddttext,
        trstatus  TYPE trstatus,
        trstdesc  TYPE string,
    END OF t_std_tkorr.
*
* BEx Objects ALV Display
*
TYPES: BEGIN OF t_bex_tkorr,
        trkorr    TYPE trkorr,
        as4text   TYPE as4text,
        eltuid    TYPE sysuuid_25,
        compid    TYPE rszcompid,
        defftp    TYPE rszdefftp,
        deftpd    TYPE string,
        txtlg     TYPE rstxtlg,
        owner     TYPE rsowner,
        trstatus  TYPE trstatus,
        trstdesc  TYPE string,
    END OF t_bex_tkorr.
*
* Query Details
*
TYPES: BEGIN OF t_bex_elements,
        eltuid    TYPE sysuuid_25,
        defftp    TYPE rszdefftp,
        txtsh     TYPE rstxtsh,
        txtlg     TYPE rstxtlg,
        compid    TYPE rszcompid,
        owner     TYPE rsowner,
    END OF t_bex_elements.

```

```

*-----*
* Global Variables
*-----*

DATA: g_repid          TYPE syrepid.

DATA: gt_bdcdata      TYPE TABLE OF bdcdata      WITH HEADER LINE, " Messages
of call transaction
      gt_messtab     TYPE TABLE OF bdcmsgcoll   WITH HEADER LINE, " Returned
Error Messages
      gt_std_alv     TYPE TABLE OF t_std_tkorr   WITH HEADER LINE, " Standard
Object List Display
      gt_bex_alv     TYPE TABLE OF t_bex_tkorr   WITH HEADER LINE. " BEx Objec
t List Display

DATA: gt_e071b        TYPE TABLE OF e071        WITH HEADER LINE, " Transpor
ts with Standard Objects
      gt_e071s        TYPE TABLE OF e071        WITH HEADER LINE, " Transport
s with Bex Objects
      gt_bex_elements TYPE TABLE OF t_bex_elements WITH HEADER LINE. " Bex Eleme
nt Descriptions

RANGES: gx_queries    FOR rszelt-dir-eltuid.

*-----*
* Selection Screen Parameters
*-----*

SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME TITLE text-001.

SELECT-OPTIONS: gp_tkorr FOR e071-trkorr.
SELECT-OPTIONS: gp_ename FOR e071-obj_name.
PARAMETERS:    gp_top    AS CHECKBOX DEFAULT 'X'.

SELECTION-SCREEN END OF BLOCK b1.

SELECTION-SCREEN BEGIN OF BLOCK b2 WITH FRAME TITLE text-002.

PARAMETER:     gp_ostd    TYPE char1 RADIOBUTTON GROUP rg1 DEFAULT 'X',
               gp_obex    TYPE char1 RADIOBUTTON GROUP rg1.

SELECTION-SCREEN END OF BLOCK b2.

*-----*
* Initialization Section
*-----*

INITIALIZATION.

      g_repid = sy-repid.          " This ABAP Program
      add_order c_transports.     " Transport Orders

*-----*
* Start of Main Program
*-----*

START-OF-SELECTION.

```

```

TRY.
*
* Get Objects from Transports
*
    PERFORM get_transports.
*
* Display Standard Objects.
*
    IF gp_ostd = 'X'.
        PERFORM format_standard_alv.
        PERFORM display_standard_screen.
    ENDIF.
*
* Display BEx Objects.
*
    IF gp_obex = 'X'.
        PERFORM get_bex_elements.
        PERFORM get_workbooks.
        PERFORM format_bex_alv.
        PERFORM display_bex_screen.
    ENDIF.

CATCH cx_root.
    MESSAGE 'Exception raised. Restrict the Selection Criteria'(099) TYPE '
I'.

ENDTRY.

-----*
* End of Main Program
*-----*
END-OF-SELECTION.

=====*
FORM get_transports.
=====*

FIELD-SYMBOLS: <e071> TYPE e071.
*
* Get Transports
*
REFRESH: gt_e071b, gt_e071s, gx_queries.
*
* Find all BEx query objects for transport request entered
*
SELECT trkorr
       object
       obj_name
INTO   CORRESPONDING FIELDS OF TABLE gt_e071b
FROM   e071
WHERE  trkorr   IN gp_tkorr
AND    ( object = c_bex_elements
OR     object  = c_workbook ).
*
* Get a range of object names

```

```

*
CLEAR gx_queries.
gx_queries-option = 'EQ'.
gx_queries-sign   = 'I'.

LOOP AT gt_e071b ASSIGNING <e071>.

    gx_queries-low = <e071>-obj_name.
    APPEND gx_queries.

ENDLOOP.
*
* Find all standard objects for transport request entered
*
SELECT trkorr
       object
       obj_name
       objfunc
INTO   CORRESPONDING FIELDS OF TABLE gt_e071s
FROM   e071
WHERE  trkorr IN gp_tkorr
AND    object NE c_bex_elements
AND    object NE c_workbook.

*=====
ENDFORM. " get_transports
*=====

*=====
FORM get_bex_elements.
*=====

DATA: ls_rszcompdir TYPE rszcompdir.

FIELD-SYMBOLS:
    <bex_element> TYPE t_bex_elements.
*
* Get Names of Bex Element
*
SORT gx_queries.
DELETE ADJACENT DUPLICATES FROM gx_queries.

IF NOT gx_queries[] IS INITIAL.
*
* Get the Bex Elements
*
SELECT rszeltldir~eltuid
       rszeltldir~deftp
       rszelttxt~txtsh
       rszelttxt~txtlg
INTO   CORRESPONDING FIELDS OF TABLE gt_bex_elements
FROM   rszeltldir
JOIN   rszelttxt ON rszelttxt~eltuid = rszeltldir~eltuid
                AND rszelttxt~objvers = rszeltldir~objvers
WHERE  rszeltldir~eltuid IN gx_queries
AND    rszeltldir~objvers = 'A'

```

```

AND   rszelttxt~langu   = sy-langu
ORDER BY rszeltmdir~eltuid.
*
* Get the Element descriptions.
*
  LOOP AT gt_bex_elements ASSIGNING <bex_element>.

    SELECT SINGLE *
    INTO   ls_rszcompdir
    FROM   rszcompdir
    WHERE  compuid = <bex_element>-eltuid
    AND    objvers = 'A'.

    IF sy-subrc = 0.
      <bex_element>-compid = ls_rszcompdir-compid.
      <bex_element>-owner  = ls_rszcompdir-owner.
    ELSE.
      <bex_element>-compid = 'NOT FOUND'(010).
    ENDIF.

  ENDLOOP.

ENDIF.

*=====
ENDFORM. " get_bex_element_names
*=====

*=====
FORM get_workbooks.
*=====

  TYPES: BEGIN OF t_workbook,
          workbookid   TYPE rsguid25,
          tstpm        TYPE rststpm,
          title         TYPE rstxtlg,
        END OF t_workbook.

  DATA: ls_workbook   TYPE t_workbook,
         ls_bex_elements TYPE t_bex_elements.
*
* Get Workbooks that are in the transports.
*
  SORT gx_queries.
  DELETE ADJACENT DUPLICATES FROM gx_queries.

  IF NOT gx_queries[] IS INITIAL.
*
* Get the Workbook Details
*
    SELECT rsrwindex~workbookid
           rsrwindex~tstpm
           rsrwindex~title
    INTO CORRESPONDING FIELDS OF ls_workbook
    FROM rsrwindex
    JOIN rsrwindext ON rsrwindex~workbookid = rsrwindext~workbookid

```



```

                AND rsrwbindex~objvers      = rsrwbindext~objvers
WHERE rsrwbindex~workbookid IN gx_queries
AND   rsrwbindex~objvers   = 'A'
AND   rsrwbindext~langu   = sy-langu
ORDER BY rsrwbindex~workbookid.

    ls_bex_elements-eltuid = ls_workbook-workbookid.
    ls_bex_elements-deftp  = 'XLS'.
    ls_bex_elements-txtsh  = ls_workbook-title.
    ls_bex_elements-txtlg  = ls_workbook-title.
    ls_bex_elements-compid = ls_workbook-workbookid.
    ls_bex_elements-owner  = ls_workbook-tstpmn.

    APPEND ls_bex_elements TO gt_bex_elements.

ENDSELECT.

ENDIF.

*=====
ENDFORM. "get_workbooks.
*=====

*=====
FORM format_bex_alv.
*=====

DATA: lt_display_alv TYPE TABLE OF t_bex_tkorr,

      ls_display_alv TYPE t_bex_tkorr,
      ls_selection   TYPE rrrange,

      l_tkorr_top    TYPE strkorr.

FIELD-SYMBOLS:
  <display>      TYPE t_bex_tkorr,
  <e071>         TYPE e071,
  <bex_element>  TYPE t_bex_elements.

REFRESH: gt_bex_alv.
*
* Build Bex Element Details.
*
LOOP AT gt_bex_elements ASSIGNING <bex_element>.

    MOVE-CORRESPONDING <bex_element> TO ls_display_alv.
*
* Get Transports for the Bex Elements.
*
    LOOP AT gt_e071b ASSIGNING <e071>
      WHERE obj_name = <bex_element>-eltuid.
*
* Display only Top Level Transports ?
*
      IF gp_top = 'X'.

```

```

CLEAR l_tkorr_top.

SELECT SINGLE strkorr
FROM e070
INTO l_tkorr_top
WHERE trkorr = <e071>-trkorr.

IF l_tkorr_top IS NOT INITIAL .
    CONTINUE.
ENDIF.
ENDIF.

MOVE-CORRESPONDING <e071> TO ls_display_alv.
APPEND ls_display_alv TO gt_bex_alv.

ENDLOOP.

ENDLOOP.
*
* Remove non-selected items.
*
IF gp_otype IS NOT INITIAL.

LOOP AT gp_otype INTO ls_selection.

LOOP AT gt_bex_alv INTO ls_display_alv WHERE compid CP ls_selection-low.
APPEND ls_display_alv TO lt_display_alv.
ENDLOOP.

ENDLOOP.

gt_bex_alv[] = lt_display_alv.

ENDIF.
*
* Get Transport Header Details.
*
LOOP AT gt_bex_alv ASSIGNING <display>.

*
* Get Transport Status
*
SELECT SINGLE trstatus
FROM e070
INTO <display>-trstatus
WHERE trkorr = <display>-trkorr.
*
* Get Transport Description
*
SELECT SINGLE as4text
FROM e07t
INTO <display>-as4text
WHERE trkorr = <display>-trkorr
AND langu = sy-langu.
*
* Get Bex Element description.

```

```

*
CASE <display>-deftpd.
  WHEN 'NIL'.
    <display>-deftpd = 'No definition'(014).
  WHEN 'REP'.
    <display>-deftpd = 'Query'(015).
  WHEN 'SOB'.
    <display>-deftpd = 'Filter'(012).
  WHEN 'VAR'.
    <display>-deftpd = 'Variable'(018).
  WHEN 'STR'.
    <display>-deftpd = 'Structure'(017).
  WHEN 'SEL'.
    <display>-deftpd = 'Restricted Key Figure'(016).
  WHEN 'CKF'.
    <display>-deftpd = 'Calculated key figure'(011).
  WHEN 'FML'.
    <display>-deftpd = 'Formula'(013).
  WHEN 'XLS'.
    <display>-deftpd = 'Workbook'(031).

ENDCASE.
*
* Get Transport Status description.
*
CASE <display>-trstatus.
  WHEN 'D'.
    <display>-trstdesc = 'Modifiable'(020).
  WHEN 'L'.
    <display>-trstdesc = 'Modifiable, Protected'(019).
  WHEN 'O'.
    <display>-trstdesc = 'Release Started'(021).
  WHEN 'R'.
    <display>-trstdesc = 'Released'(023).
  WHEN 'N'.
    <display>-
trstdesc = 'Released (with Import Protection for Repaired Objects)'(022).
ENDCASE.

ENDLOOP.

*=====
ENDFORM. " format_bex_alv
*=====

*=====
FORM display_bex_screen.
*=====

DATA: lt_fieldcat  TYPE slis_t_fieldcat_alv,
      ls_fcat      TYPE slis_fieldcat_alv,
      ls_layout    TYPE slis_layout_alv,
      ls_variant   TYPE disvariant.

DATA: lt_sort      TYPE slis_t_sortinfo_alv,
      ls_sort      TYPE slis_sortinfo_alv.

```

```

*
* Define Layout Paramters.
*
ls_variant-report          = g_repid.
ls_layout-zebra           = 'X'.
ls_layout-colwidth_optimize = 'X'.
*
* Define ALV Columns and Title.
*
insert_hotspot_field 'TRKORR' 'Request/Task'(024).
insert_long_field   'AS4TEXT' 'Description'(025).
insert_long_field   'DEFTPD'  'Element Type'(027).
insert_long_field   'COMPID'  'Technical name'(026).
insert_long_field   'TXTLG'   'Long Description'(028).
insert_long_field   'OWNER'   'Owner'(029).
insert_long_field   'TRSTDESC' 'Transport Status'(030).
*
* Sort Parameters
*
ls_sort-fieldname = 'TRKORR'. APPEND ls_sort TO lt_sort.
ls_sort-fieldname = 'AS4TEXT'. APPEND ls_sort TO lt_sort.
ls_sort-fieldname = 'DEFTPD'. APPEND ls_sort TO lt_sort.
ls_sort-fieldname = 'COMPID'. APPEND ls_sort TO lt_sort.
*
* Call ALV Grid.
*
CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'
  EXPORTING
    i_callback_program      = g_repid           " This Program
    i_callback_user_command = 'USER_COMMAND'   " Hotspot Selection Execut
e Form....
    is_layout              = ls_layout         " Layout option
    it_sort                = lt_sort          " Sort Parameters
    it_fieldcat            = lt_fieldcat      " Fields
    i_save                 = 'X'              " Save Variant
    is_variant             = ls_variant       " Variant
  TABLES
    t_outtab              = gt_bex_alv
  EXCEPTIONS
    program_error         = 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.

*=====
ENDFORM. " display_bex_screen
*=====

*=====
FORM format_standard_alv.
*=====

DATA: lt_display_alv TYPE TABLE OF t_std_tkorr,

```

```

    lt_obj_tp_text LIKE TABLE OF ko100,

    ls_display_alv TYPE t_std_tkorr,
    ls_selection  TYPE rrrange,

    l_tkorr_top   TYPE strkorr.

FIELD-SYMBOLS:
    <display>    TYPE t_std_tkorr,
    <obj_type>    LIKE ko100,
    <e071>       TYPE e071.
*
* Get a list of Object Type Descriptions.
*
CALL FUNCTION 'TRINT_OBJECT_TABLE'
  EXPORTING
    iv_complete = 'X'
  TABLES
    tt_types_out = lt_obj_tp_text.

REFRESH: gt_std_alv.
*
* Get Transports for the Standard Elements.
*
LOOP AT gt_e071s ASSIGNING <e071>
  WHERE object NE c_bex_elements
  AND   object NE c_workbook
  AND   object NE c_comment.
*
* Display only Top Level Transports ?
*
  IF gp_top = 'X'.

    CLEAR l_tkorr_top.

    SELECT SINGLE strkorr
      FROM   e070
      INTO   l_tkorr_top
      WHERE  trkorr = <e071>-trkorr.

    IF l_tkorr_top IS NOT INITIAL .
      CONTINUE.
    ENDIF.
  ENDIF.

  MOVE-CORRESPONDING <e071> TO ls_display_alv.
  APPEND ls_display_alv    TO gt_std_alv.

ENDLOOP.
*
* Remove non-selected items.
*
  IF gp_otype IS NOT INITIAL.

    LOOP AT gp_otype INTO ls_selection.

```

```

        LOOP AT gt_std_alv INTO ls_display_alv WHERE obj_name CP ls_selection-low.
            APPEND ls_display_alv TO lt_display_alv.
        ENDLOOP.

    ENDLOOP.

    gt_std_alv[] = lt_display_alv.

ENDIF.
*
* Get Transport Header Details.
*
    LOOP AT gt_std_alv ASSIGNING <display>.

*
* Get Transport Status
*
        SELECT SINGLE trstatus
        FROM e070
        INTO <display>-trstatus
        WHERE trkorr = <display>-trkorr.

*
* Get Transport Description
*
        SELECT SINGLE as4text
        FROM e07t
        INTO <display>-as4text
        WHERE trkorr = <display>-trkorr
        AND langu = sy-langu.

*
* Get Object description.
*
        READ TABLE lt_obj_tp_text ASSIGNING <obj_type>
        WITH KEY object = <display>-object.

        IF sy-subrc = 0.
            <display>-text = <obj_type>-text.
        ENDIF.

*
* Get Object Function description.
*
        CASE <display>-objfunc.
            WHEN ' '.
                <display>-objfunc = 'Transported'(032).
            WHEN 'D'.
                <display>-objfunc = 'Deleted'(033).
            WHEN 'M'.
                <display>-objfunc = 'Recreated'(034).
            WHEN 'K'.
                <display>-objfunc = 'Object Keys'.
        ENDCASE.

*
* Get Transport Status description.
*
        CASE <display>-trstatus.
            WHEN 'D'.

```

```

    <display>-trstdesc = 'Modifiable'(020).
  WHEN 'L'.
    <display>-trstdesc = 'Modifiable, Protected'(019).
  WHEN 'O'.
    <display>-trstdesc = 'Release Started'(021).
  WHEN 'R'.
    <display>-trstdesc = 'Released'(023).
  WHEN 'N'.
    <display>-
trstdesc = 'Released (with Import Protection for Repaired Objects)'(022).
  ENDCASE.
ENDLOOP.

```

```

*=====
ENDFORM. " format_standard_alv
*=====

```

```

*=====
FORM display_standard_screen.
*=====

```

```

DATA: lt_fieldcat  TYPE slis_t_fieldcat_alv,
      ls_fcat      TYPE slis_fieldcat_alv,
      ls_layout    TYPE slis_layout_alv,
      ls_variant   TYPE disvariant.

```

```

DATA: lt_sort      TYPE slis_t_sortinfo_alv,
      ls_sort      TYPE slis_sortinfo_alv.

```

*

* Define Layout Paramters.

*

```

ls_variant-report      = g_repid.
ls_layout-zebra        = 'X'.
ls_layout-colwidth_optimize = 'X'.

```

*

* Define ALV Columns and Title.

*

```

insert_hotspot_field 'TRKORR'  'Request/Task'(024).
insert_long_field   'AS4TEXT'  'Description'(025).
insert_long_field   'OBJECT'   'Object Type'(036).
insert_long_field   'TEXT'     'Technical name'(026).
insert_long_field   'OBJ_NAME' 'Object Name'(037).
insert_long_field   'OBJFUNCT' 'Function'(038).
insert_long_field   'TRSTDDESC' 'Transport Status'(030).

```

*

* Sort Parameters

*

```

ls_sort-fieldname = 'TRKORR'. APPEND ls_sort TO lt_sort.
ls_sort-fieldname = 'AS4TEXT'. APPEND ls_sort TO lt_sort.
ls_sort-fieldname = 'OBJECT'. APPEND ls_sort TO lt_sort.
ls_sort-fieldname = 'TEXT'. APPEND ls_sort TO lt_sort.

```

*

* Call ALV Grid.

*

```

CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'

```

```

EXPORTING
  i_callback_program      = g_repdi          " This Program
  i_callback_user_command = 'USER_COMMAND'  " Hotspot Selection Execut
e Form....
  is_layout               = ls_layout       " Layout option
  it_sort                 = lt_sort         " Sort Parameters
  it_fieldcat             = lt_fieldcat     " Fields
  i_save                  = 'X'            " Save Variant
  is_variant              = ls_variant      " Variant
TABLES
  t_outtab                = gt_std_alv
EXCEPTIONS
  program_error           = 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.

*=====
ENDFORM. " display_standard_screen
*=====

*=====
FORM user_command USING p_ucomm      TYPE syucomm
                  ps_selfield TYPE slis_selfield.
*=====

*
* Hotspot call.
*
CASE ps_selfield-fieldname.

  WHEN 'TRKORR'.
    PERFORM call_se10 USING ps_selfield.

ENDCASE.

*=====
ENDFORM. " user_command
*=====

*=====
FORM call_se10 USING ps_selfield TYPE slis_selfield.
*=====

PERFORM bdc_dynpro      USING 'RDDM0001'      '0100'.
PERFORM bdc_field      USING 'BDC_OKCODE'    '=SINGL'.
PERFORM bdc_field      USING 'BDC_CURSOR'    'TRDYSE01CM-USERNAME'.
PERFORM bdc_dynpro      USING 'SAPLSTRH'     '1200'.
PERFORM bdc_field      USING 'BDC_CURSOR'    'GV_1200_TRKORR'.
PERFORM bdc_field      USING 'BDC_OKCODE'    '=TAKE'.
PERFORM bdc_field      USING 'GV_1200_TRKORR' ps_selfield-value.
PERFORM bdc_transaction USING 'SE10'.

```



```

*=====*
ENDFORM.                                " call_se10
*=====*

*=====*
FORM bdc_transaction USING p_tcode.
*=====*

DATA: l_mstring    TYPE string,
      l_ctu_params TYPE ctu_params,
      ls_t100      TYPE t100.
*
* Call BDC Transaction.
*
l_ctu_params-dismode = 'E'.
l_ctu_params-updmode = 'L'.
l_ctu_params-nobinpt = 'X'.
l_ctu_params-nobiend = 'X'.

REFRESH gt_messtab.

CALL TRANSACTION p_tcode USING gt_bdcdata
                MESSAGES INTO gt_messtab
                OPTIONS FROM l_ctu_params.
*
* Display Any Error Messages
*
LOOP AT gt_messtab.

  SELECT SINGLE *
  INTO   ls_t100
  FROM   t100
  WHERE  sprsl = gt_messtab-msgspra
  AND    arbgb = gt_messtab-msgid
  AND    msgnr = gt_messtab-msgnr.

  IF sy-subrc = 0.

    l_mstring = ls_t100-text.

    IF l_mstring CS '&1'.
      REPLACE '&1' WITH gt_messtab-msgv1 INTO l_mstring.
      REPLACE '&2' WITH gt_messtab-msgv2 INTO l_mstring.
      REPLACE '&3' WITH gt_messtab-msgv3 INTO l_mstring.
      REPLACE '&4' WITH gt_messtab-msgv4 INTO l_mstring.
    ELSE.
      REPLACE '&' WITH gt_messtab-msgv1 INTO l_mstring.
      REPLACE '&' WITH gt_messtab-msgv2 INTO l_mstring.
      REPLACE '&' WITH gt_messtab-msgv3 INTO l_mstring.
      REPLACE '&' WITH gt_messtab-msgv4 INTO l_mstring.
    ENDIF.

    CONDENSE l_mstring.
    WRITE: / gt_messtab-msgtyp, l_mstring.

  ELSE.

```

```

        WRITE: / gt_messtab.
      ENDIF.
    ENDLOOP.

    REFRESH gt_bdcdata.

*=====*
ENDFORM. "bdc_transaction
*=====*

*=====*
FORM bdc_dynpro USING p_program p_dynpro.
*=====*

    CLEAR gt_bdcdata.
    gt_bdcdata-program = p_program.
    gt_bdcdata-dynpro  = p_dynpro.
    gt_bdcdata-dynbegin = 'X'.
    APPEND gt_bdcdata.

*=====*
ENDFORM. "bdc_dynpro
*=====*

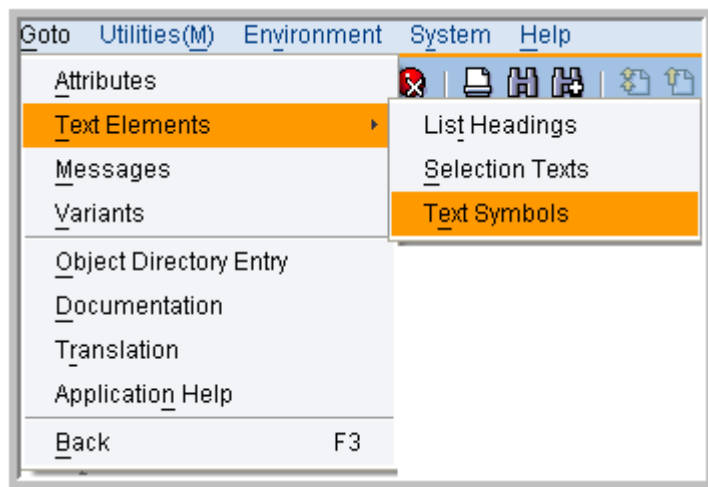
*=====*
FORM bdc_field USING p_fnam p_fval.
*=====*

    IF p_fval <> '/'.
      CLEAR gt_bdcdata.
      gt_bdcdata-fnam = p_fnam.
      gt_bdcdata-fval = p_fval.
      APPEND gt_bdcdata.
    ENDIF.

*=====*
ENDFORM. "bdc_field
*=====*

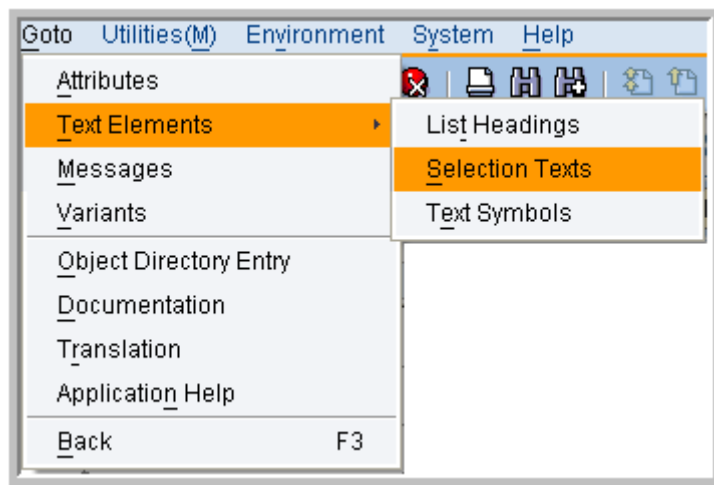
```

Defining Text Element



- 001 Transport Range Selection
- 002 Object Types
- 010 NOT FOUND
- 011 Calculated key figure
- 012 Filter
- 013 Formula
- 014 No definition
- 015 Query
- 016 Restricted Key Figure
- 017 Structure
- 018 Variable
- 019 Modifiable, Protected
- 020 Modifiable
- 021 Release Started
- 022 Released (with Import Protection for Repaired Objects)
- 023 Released
- 024 Request/Task
- 025 Description
- 026 Technical name
- 027 Element Type
- 028 Long Description
- 029 Owner
- 030 Transport Status
- 031 Workbook
- 032 Transported
- 033 Deleted
- 034 Recreated

- 035 Object Keys
- 036 Object Type
- 037 Object Name
- 038 Function
- 099 Exception raised. Restrict the Selection Criteria



- GP_OBEX Bex Objects
- GP_ONAME Object Name
- GP_OSTD Standard Objects
- GP_TKORR Request/Task
- GP_TOP Top Level Transports Only

An Example

Objects in Transport Requests

🔍
📄

Transport Range Selection

Request/Task 🔍

Object Name to 🔍

Top Level Transports Only

Object Types

Standard Objects

Bex Objects

Objects in Transport Requests

🔍
📄

Request/Task	Description	Object Type	Technical name	Object Name	Function
WDGK902508	T2: Including all ITS templates & queries in IT&S Input Role	ACGR	Role	ZSBWM01_TC_T2_ITSTEMPLATES	Transported
		TABU	Table Contents	AGR_TIMEB	Object Keys
				SPERS_OBJ	Object Keys
				UCUW001	Object Keys
				UCUW011	Object Keys
				USR12	Object Keys
				USR13	Object Keys
				UST12	Object Keys
		TDAT	Customizing: Table Contents	SUSPR	Object Keys
WDGK902518	T2: IT&S Actuals Workbooks and Queries	PLSE	Planning Service	ZTCTCA211_PFO2	Transported
				ZTCTCA211_PFO4	Transported
WDGK902547	T2: IT&S Variables	CLSD	Class Definition (ABAP Objects)	ZCL_BEVARS_EXIT_ZTC_FNACT	Transported
		CPRI	Private Header (ABAP Objects)	ZCL_BEVARS_EXIT_ZTC_FNACT	Transported
		CPUB	Public Header (ABAP Objects)	ZCL_BEVARS_EXIT_ZTC_FNACT	Transported
		METH	Method (ABAP Objects)	ZCL_BEVARS_EXIT_ZTC_FNACT VARIABLE_ZMSXFA03	Transported
WDGK902549	T2: ZCA_OBJECTS	REPS	Report Source Code	ZCA_OBJECTS	Transported
WDGK902555	TC2: Currency Conversion Fix	PLSE	Planning Service	ZTCTCA208FUNC08	Transported

Related Content

[Display objects exists in a transport request](#)

[Table for transport request](#)

[Transport Guide Tips for SAP Objects](#)

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.