

```

*&-----*
*& Report ZBDC_FILE_UPLOAD
*&-----*
*&
*&-----*
REPORT zbdc_call_transaction NO STANDARD PAGE HEADING LINE-SIZE 255.

```

```

TYPES: BEGIN OF lty_data,
        matnr TYPE matnr,
        mbrsh TYPE mbrsh,
        mtart TYPE mtart,
        maktx TYPE maktx,
        meins TYPE meins,
        END OF lty_data.

```

```

DATA: lt_data TYPE TABLE OF lty_data,
      ls_data TYPE lty_data.

```

```

DATA: lv_file TYPE string.

```

```

DATA: lt_bdcdata TYPE TABLE OF bdcdata,
      ls_bdcdata TYPE bdcdata.

```

```

*DATA: lt_bdcdata LIKE bdcdata OCCURS 0 WITH HEADER LINE.

```

```

*data: messtab like bdcmsgcoll occurs 0 WITH HEADER LINE. "outdated syntax

```

```

DATA: lt_messtab TYPE TABLE OF bdcmsgcoll,
      ls_messtab type bdcmsgcoll.

```

```

data: lv_message type string.

```

```

PARAMETERS: p_file TYPE localfile.

```

```

*****
* importing the data

```

```

AT SELECTION-SCREEN ON VALUE-REQUEST FOR p_file.

```

```

    CALL FUNCTION 'KD_GET_FILENAME_ON_F4'
      EXPORTING
        program_name = syst-
repid      " Modulpool program name for screen field
        dynpro_number = syst-
dynnr      " Dynpro number where F4 help is needed
        field_name   = space      " name of field where path is to be
entered        =                " Langer Pfadname
      CHANGING
        file_name    = p_file.      " Path name selected by user wit
h help of Filemngnr .

```

```

START-OF-SELECTION.

```

```

    lv_file = p_file. " type casting in this functin filename is string howe
ver p_file is char 128.

```

```
*****
* When you want to upload a Excel file, just one change - Instead of GUI_UPLOAD, use the FM : TEXT_CONVERT_XLS_TO_SAP.
```

```
CALL FUNCTION 'GUI_UPLOAD'
  EXPORTING
    filename          = lv_file          " Name der Datei
    * filetype        = 'ASC'
    has_field_separator = 'X'           " Spalten durch TAB getrennt bei ASCII Upload
  TABLES
    data_tab          = lt_data          " Übergabetabelle für Datei-Inhalt
  EXCEPTIONS
    file_open_error   = 1               " Datei nicht vorhanden, kann nicht geöffnet werden
    file_read_error   = 2               " Fehler beim Lesen der Datei
    no_batch          = 3               " Frontend-Funktion im Batch nicht ausführbar.
    gui_refuse_filetransfer = 4          " Falsches Frontend oder Fehler im Frontend
    invalid_type      = 5               " Falscher Parameter FILETYPE
    no_authority      = 6               " Keine Berechtigung für Upload
    unknown_error     = 7
    bad_data_format   = 8               " Daten in der Datei können nicht interpretiert werden.
    header_not_allowed = 9               " Header ist nicht zulässig.
    separator_not_allowed = 10          " Separator ist nicht zulässig.
    header_too_long   = 11              " Die Headerinformation ist zu groß (maximal 1023 Bytes bei Dataprovider)
    unknown_dp_error  = 12              " Fehler beim Aufruf des Dataproviders
    access_denied     = 13              " Zugriff auf Datei nicht erlaubt.
    dp_out_of_memory  = 14              " Nicht genug Speicher im Dataprovider
    disk_full         = 15              " Speichermedium ist voll.
    dp_timeout        = 16              " Timeout des Dataproviders
    OTHERS            = 17.

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

* INCLUDE bdcrcx1_s.

LOOP AT lt_data INTO ls_data.
  PERFORM bdc_dynpro USING 'SAPLMDMM' '0060'.
```

```

PERFORM bdc_field USING 'BDC_CURSOR'
                        'RMMG1-MTART'.

PERFORM bdc_field USING 'BDC_OKCODE'
                        '=ENTR'.

PERFORM bdc_field USING 'BRMMG1-MATNR'
                        ls_data-matnr.    ''P2Material'

PERFORM bdc_field USING 'RMMG1-MBRSH'
                        ls_data-mbrsh.    'P'

PERFORM bdc_field USING 'RMMG1-MTART'
                        ls_data-mtart.    ''TROH'

PERFORM bdc_dynpro USING 'SAPLMDMM' '0070'.
PERFORM bdc_field USING 'BDC_CURSOR'
                        'MSICHTAUSW-DYTXT(01)'.

PERFORM bdc_field USING 'BDC_OKCODE'
                        '=ENTR'.

PERFORM bdc_field USING 'SICHTAUSW-KZSEL(01)'
                        'X'.

PERFORM bdc_dynpro USING 'SAPLMDMM' '4004'.
PERFORM bdc_field USING 'BDC_OKCODE'
                        '=BU'.

PERFORM bdc_field USING 'MAKT-MAKTX'
                        ls_data-maktx.    ''Test Material'

PERFORM bdc_field USING 'BDC_CURSOR'
                        'MARA-MEINS'.

PERFORM bdc_field USING 'MARA-MEINS'
                        ls_data-meins.    'EA'

PERFORM bdc_field USING 'MARA-MTPOS_MARA'
                        'NORM'.

CALL TRANSACTION 'MM01' USING lt_bdcdata    "we collect all the data i
n the table lt_bdcdata via subroutine bdc_dynpro and bdc_field. "inside lo
op
                                MODE 'E'
                                UPDATE 'S'
                                MESSAGES INTO lt_messtab.    "if error collect
the error like log

REFRESH lt_bdcdata.    "otherwise only first entry in progress
ENDLOOP.

"info for User about messages from messages table via message text build
LOOP AT lt_messtab INTO ls_messtab.

call FUNCTION 'FORMAT_MESSAGE'    "instead of 'MESSAGE_TEXT_BUILD'
EXPORTING
    id            = ls_messtab-msgid    " Arbeitsgebiet

```

```

*   lang      = '-D'                " Sprache ('-' = Systemsprache)
   no         = ls_messtab-msgnr     " dreistellige Nummer
   v1         = ls_messtab-msgv1     " 1. Parameter
   v2         = ls_messtab-msgv2     " 2. Parameter
   v3         = ls_messtab-msgv3     " 3. Parameter
   v4         = ls_messtab-msgv4     " 4. Parameter
IMPORTING
   msg        = lv_message           " Aufbereiterer Meldungstext
EXCEPTIONS
   not_found  = 1                   " Nachricht in T100 nicht definiert oder l
eer
   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.

```

```

ENDLOOP.

```

```

FORM bdc_dynpro USING program dynpro.

```

```

   CLEAR ls_bdcdata.
   ls_bdcdata-program = program.
   ls_bdcdata-dynpro = dynpro.
   ls_bdcdata-dynbegin = 'X'.
   APPEND ls_bdcdata TO lt_bdcdata.
ENDFORM.

```

```

FORM bdc_field USING fnam fval.
* IF fval NE nodata.
   CLEAR ls_bdcdata.
   ls_bdcdata-fnam = fnam.
   ls_bdcdata-fval = fval.
   APPEND ls_bdcdata TO lt_bdcdata.
* ENDIF.
ENDFORM.

```

```

* importing the data
*****

```