

```

*&-----*
*& Report ZBAPI_UPLOAD
*&-----*
*&
*&-----*
REPORT zbapi_upload.

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: ls_headdata TYPE bapimathead,
      ls_clientdata TYPE bapi_mara,
      ls_clientdatax TYPE bapi_marax.

DATA: lt_materialdescription TYPE TABLE OF bapi_makt,
      ls_materialdescription TYPE bapi_makt.

DATA: ls_return TYPE bapiret2,
      lt_return TYPE TABLE OF bapiret2.

DATA: lv_file 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 with
h help of Filemngn .

START-OF-SELECTION.

    lv_file = p_file. " type casting in this function filename is string however
p_file is char 128.

*****
* When you want to upload a Excel file, just one change - Instead of GUI_UP

```

LOAD, 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)
    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.

LOOP AT lt_data INTO ls_data.
  ls_headdata-material = ls_data-matnr.
  ls_headdata-ind_sector = ls_data-mbrsh.
  ls_headdata-matl_type = ls_data-mtart.
  ls_headdata-basic_view = 'X'.

  ls_clientdata-base_uom = ls_data-meins.
  ls_clientdatax-base_uom = 'X'.

  ls_materialdescription-langu = sy-langu.
  ls_materialdescription-matl_desc = ls_data-maktx.

APPEND ls_materialdescription TO lt_materialdescription.
CLEAR ls_materialdescription.
```

```

CALL FUNCTION 'BAPI_MATERIAL_SAVEDATA'
  EXPORTING
    headdata          = ls_headdata
    clientdata        = ls_clientdata
    clientdatax       = ls_clientdatax  "if dont pass the this SAP d
ont understand which of part of info should be updated
*    plantdata        = lv_plantdata
*    plantdatax       = lv_plantdatax
*    valuationdata    = lv_valuationdata
*    valuationdatax   = lv_valuationdatax
*    storagelocationdata = lv_storagelocationdata
*    storagelocationdatax = lv_storagelocationdatax
  IMPORTING
    return            = ls_return
  TABLES
    materialdescription = lt_materialdescription.
*    returnmessages    = it_returnmessages.

APPEND ls_return TO lt_return.
CLEAR :ls_return, ls_headdata, ls_clientdata, ls_clientdatax.

REFRESH lt_materialdescription.  "should be refreshed.
ENDLOOP.

*****
* show the results of the process
DATA: lo_alv TYPE REF TO cl_salv_table.

TRY.
  cl_salv_table=>factory(
    IMPORTING r_salv_table = lo_alv
    CHANGING  t_table      =  lt_return
  ).

  lo_alv->display( ).

CATCH cx_salv_msg INTO DATA(lx_msg).
  MESSAGE lx_msg->get_text( ) TYPE 'S'.
ENDTRY.

```