\*&---------------------------------------------------------------------\*  
\*& Report  ZFILE\_UPLOAD\_10  
\*&  
\*&---------------------------------------------------------------------\*  
\*&  
\*&  
\*&---------------------------------------------------------------------\*  
  
REPORT  zfile\_upload\_10.  
" step1- what are the fields in the file  
" step2 - how is the format: field is seperated by what space, comma, etc  
" Location: Presentation server (GUI)  
" Actual program  
" 1- declare strutues  
TYPES: BEGIN OF t\_mara,  
  matnr TYPE matnr,  
  mtart TYPE mtart,  
  mbrsh TYPE mbrsh,  
END OF t\_mara,  
t\_mara\_tab TYPE STANDARD TABLE OF t\_mara.  
  
TYPES: BEGIN OF t\_data,  
      data TYPE c LENGTH 25,  
    END OF t\_data,  
    t\_data\_tab TYPE STANDARD TABLE OF t\_data.  
  
DATA: it\_data TYPE t\_data\_tab,  
      wa\_data TYPE t\_data.  
  
DATA:  
      it\_mara TYPE t\_mara\_tab,  
      wa\_mara TYPE t\_mara.  
  
" 2. choose file path F4\_FILENAME  
PARAMETERS p\_path TYPE ibipparms-path.  
  
  
AT SELECTION-SCREEN ON VALUE-REQUEST FOR p\_path.  
  PERFORM get\_path.  
  
  
START-OF-SELECTION.  
  " 3. Upload the file GUI\_UPLOAD  
  PERFORM upload\_data.  
  PERFORM convert\_data.  
  BREAK-POINT.  
\*&---------------------------------------------------------------------\*  
\*&      Form  GET\_PATH  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*  -->  p1        text  
\*  <--  p2        text  
\*----------------------------------------------------------------------\*  
FORM get\_path .  
  CALL FUNCTION 'F4\_FILENAME'  
\*   EXPORTING  
\*     PROGRAM\_NAME        = SYST-CPROG  
\*     DYNPRO\_NUMBER       = SYST-DYNNR  
\*     FIELD\_NAME          = ' '  
   IMPORTING  
     file\_name           = p\_path  
            .  
ENDFORM.                    " GET\_PATH  
\*&---------------------------------------------------------------------\*  
\*&      Form  UPLOAD\_DATA  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*  -->  p1        text  
\*  <--  p2        text  
\*----------------------------------------------------------------------\*  
FORM upload\_data .  
  DATA: g\_path TYPE string.  
  g\_path = p\_path.  
  CALL FUNCTION 'GUI\_UPLOAD'  
    EXPORTING  
      filename                      = g\_path  
\*   FILETYPE                      = 'ASC'  
\*   has\_field\_separator           = 'X'  
\*   HEADER\_LENGTH                 = 0  
\*   READ\_BY\_LINE                  = 'X'  
\*   DAT\_MODE                      = ' '  
\*   CODEPAGE                      = ' '  
\*   IGNORE\_CERR                   = ABAP\_TRUE  
\*   REPLACEMENT                   = '#'  
\*   CHECK\_BOM                     = ' '  
\*   VIRUS\_SCAN\_PROFILE            =  
\*   NO\_AUTH\_CHECK                 = ' '  
\* IMPORTING  
\*   FILELENGTH                    =  
\*   HEADER                        =  
    TABLES  
      data\_tab                      = it\_data  
\* EXCEPTIONS  
\*   FILE\_OPEN\_ERROR               = 1  
\*   FILE\_READ\_ERROR               = 2  
\*   NO\_BATCH                      = 3  
\*   GUI\_REFUSE\_FILETRANSFER       = 4  
\*   INVALID\_TYPE                  = 5  
\*   NO\_AUTHORITY                  = 6  
\*   UNKNOWN\_ERROR                 = 7  
\*   BAD\_DATA\_FORMAT               = 8  
\*   HEADER\_NOT\_ALLOWED            = 9  
\*   SEPARATOR\_NOT\_ALLOWED         = 10  
\*   HEADER\_TOO\_LONG               = 11  
\*   UNKNOWN\_DP\_ERROR              = 12  
\*   ACCESS\_DENIED                 = 13  
\*   DP\_OUT\_OF\_MEMORY              = 14  
\*   DISK\_FULL                     = 15  
\*   DP\_TIMEOUT                    = 16  
\*   OTHERS                        = 17  
            .  
  IF sy-subrc <> 0.  
\* Implement suitable error handling here  
  ENDIF.  
  
ENDFORM.                    " UPLOAD\_DATA  
\*&---------------------------------------------------------------------\*  
\*&      Form  CONVERT\_DATA  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*  -->  p1        text  
\*  <--  p2        text  
\*----------------------------------------------------------------------\*  
FORM convert\_data .  
  LOOP AT it\_data INTO wa\_data.  
    SPLIT  wa\_data-data AT ',' INTO wa\_mara-matnr wa\_mara-mtart wa\_mara-mbrsh.  
    APPEND wa\_mara TO it\_mara.  
  ENDLOOP.  
ENDFORM.                    " CONVERT\_DATA