\*&---------------------------------------------------------------------\*  
\*& Report ZPRASAD\_BAPI\_SALES\_DETAILS  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT zprasad\_bapi\_sales\_details.  
  
DATA : wa\_return   TYPE bapireturn,  
       wa\_tab TYPE bapisdstat,  
       lt\_tab      LIKE STANDARD TABLE OF wa\_tab.  
  
  
PARAMETERS p\_ebeln TYPE vbap-vbeln.  
  
TYPES : BEGIN OF ls\_tab,  
          matnr TYPE makt-matnr,  
          maktx TYPE makt-maktx,  
  
        END OF ls\_tab.  
  
DATA : wa\_tab2 TYPE ls\_tab,  
       it\_tab2 LIKE  STANDARD TABLE OF wa\_tab2 WITH KEY matnr.  
  
  
CALL FUNCTION 'BAPI\_SALESORDER\_GETSTATUS'  
  EXPORTING  
    salesdocument = p\_ebeln  
  IMPORTING  
    return        = wa\_return  
  TABLES  
    statusinfo    = lt\_tab.  
  
  
  
if wa\_return is INITIAL.  
  WRITE : /2    'DCO NO:'  ,  
  14   'DOC DATE:',  
  25  'PO NUMBER:',  
  40  'PROCESS STST:',  
  42  'PROCESS DEL STST:',  
  44  'REQ DEL DATE:' ,  
  55  'SALES DOC ITEM',  
  60  'MATERIAL NO:' ,  
  78  'MAT TEXT',  
  100 'NET VALUE'.  
  ULINE.  
loop at lt\_tab into wa\_tab.  
 READ TABLE IT\_TAB2 INTO WA\_TAB2  
    WITH KEY MATNR = WA\_TAB-MATERIAL.  
  
    WRITE : / WA\_TAB-DOC\_NUMBER, " Sales Document  
    WA\_TAB-DOC\_DATE,   " Document Date (Date Received/Sent)  
    WA\_TAB-PURCH\_NO,   " Customer purchase order number  
    WA\_TAB-PRC\_STAT\_H, " Overall processing status of document  
    WA\_TAB-DLV\_STAT\_H, " Overall processing status of document  
    WA\_TAB-REQ\_DATE\_H, " Requested delivery date  
    WA\_TAB-ITM\_NUMBER, " Sales Document Item  
    WA\_TAB-MATERIAL,   " Material Number  
    WA\_TAB-SHORT\_TEXT, " Short text for sales order item  
    WA\_TAB-NET\_VALUE.  " Net value of the order item in document currency  
  
    CLEAR WA\_TAB.  
  ENDLOOP.  
  REFRESH lT\_TAB[].  
ELSE.  
  WRITE WA\_RETURN-MESSAGE.  
ENDIF.  
  
\*cl\_demo\_output=>display(  
\*  data    =    lt\_tab              " Text or Data  
\*\*            name    =                  " Name  
\*\*            exclude =                  " Exclude structure components  
\*\*            include =                  " Include structure components  
\*).

\*&---------------------------------------------------------------------\*  
\*& Report ZPRASAD\_BAPI\_SERVER\_ITAB  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT zprasad\_bapi\_server\_itab.  
  
TABLES: bapimathead,  
        bapi\_mara,  
        bapi\_marax,  
        bapi\_marc,  
        bapi\_marcx,  
        bapiret2.  
  
TYPES : BEGIN OF ls\_tab,  
          matnr TYPE matnr,  
          mbrsh TYPE mbrsh,  
          mtart TYPE mtart,  
          maktx TYPE maktx,  
          meins TYPE meins,  
          matkl TYPE matkl,  
        END OF ls\_tab.  
  
TYPES :BEGIN OF ls\_tab2,  
         langu         TYPE lang,  
         langu\_iso(2),  
         matl\_desc(40),  
         del\_flag(1),  
       END OF ls\_tab2.  
  
DATA : lt\_tab  TYPE STANDARD TABLE OF ls\_tab,  
       wa\_tab  TYPE ls\_tab,  
       lt\_tab2 TYPE STANDARD TABLE OF ls\_tab2,  
       wa\_tab2 TYPE ls\_tab2.  
  
  
DATA : BEGIN OF lt\_return.  
         INCLUDE STRUCTURE bapiret2.  
DATA : END OF lt\_return.  
  
PARAMETERS : p\_file type localfile.  
  
at SELECTION-SCREEN on VALUE-REQUEST FOR p\_file.  
CALL FUNCTION 'F4\_FILENAME'  
 EXPORTING  
   PROGRAM\_NAME        = SYST-CPROG  
   DYNPRO\_NUMBER       = SYST-DYNNR  
   FIELD\_NAME          = ' '  
 IMPORTING  
   FILE\_NAME           = p\_file  
  
  
          .  
  
  
START-OF-SELECTION .  
data : lv\_file type string.  
lv\_file = p\_file.  
  
CALL FUNCTION 'GUI\_UPLOAD'  
  EXPORTING  
    filename                      = lv\_file  
   FILETYPE                      = 'ASC'  
   HAS\_FIELD\_SEPARATOR           = 'X'  
  tables  
    data\_tab                      = lt\_tab  
  
  
          .  
IF sy-subrc <> 0.  
\* Implement suitable error handling here  
ENDIF.  
  
cl\_demo\_output=>display(  
  data    =  lt\_tab                " Text or Data  
\*  name    =                  " Name  
\*  exclude =                  " Exclude structure components  
\*  include =                  " Include structure components  
).  
  
OPEN DATASET 'zprasadd\_ds' for OUTPUT IN TEXT MODE ENCODING DEFAULT.  
loop AT LT\_TAB INTO WA\_TAB.  
  TRANSFER  
  WA\_TAB TO 'zprasadd\_ds'.  
  CLEAR WA\_TAB.  
ENDLOOP.

\*&---------------------------------------------------------------------\*  
\*& Report ZPRASAD\_BAPI\_UPLOAD  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT zprasad\_bapi\_upload.  
  
TABLES :bapimathead,  
        bapi\_marax,  
        bapi\_mara,  
        bapi\_marc,  
        bapi\_marcx,  
        bapi\_makt,  
        bapiret2.  
  
  
TYPES : BEGIN OF ls\_tab,  
          matnr TYPE matnr,  
          mbrsh TYPE mbrsh,  
          mtart TYPE mtart,  
          maktx TYPE maktx,  
          meins TYPE meins,  
          matkl TYPE matkl,  
        END OF ls\_tab.  
  
TYPES:BEGIN OF ls\_tab2,  
        langu TYPE lang,  
        langu\_iso(2),  
        matl\_desc(40),  
        del\_flag(1),  
      END OF ls\_tab2.  
  
  
  
DATA : lt\_tab  TYPE STANDARD TABLE OF ls\_tab,  
       wa\_tab  TYPE ls\_tab,  
       lt\_tab2 TYPE STANDARD TABLE OF ls\_tab2,  
       wa\_tab2 TYPE ls\_tab2.  
  
DATA : BEGIN OF it\_return .  
  
         INCLUDE STRUCTURE bapiret2.  
DATA : END OF it\_return.  
  
  
PARAMETERS p\_file TYPE localfile.  
  
AT SELECTION-SCREEN ON VALUE-REQUEST FOR p\_file.  
  CALL FUNCTION 'F4\_FILENAME'  
    EXPORTING  
      program\_name  = syst-cprog  
      dynpro\_number = syst-dynnr  
    IMPORTING  
      file\_name     = p\_file.  
  
  
START-OF-SELECTION.  
  DATA : lv\_file TYPE string.  
  lv\_file = p\_file.  
  CALL FUNCTION 'GUI\_UPLOAD'  
    EXPORTING  
      filename            = lv\_file  
      filetype            = 'ASC'  
      has\_field\_separator = 'X'  
    TABLES  
      data\_tab            = lt\_tab.  
  IF sy-subrc <> 0.  
\* Implement suitable error handling here  
  ENDIF.  
  
\*  cl\_demo\_output=>display( lt\_tab ).  
  
  LOOP AT lt\_tab INTO wa\_tab.  
  
    bapimathead-material = wa\_tab-matnr.  
    bapimathead-ind\_sector = wa\_tab-mbrsh.  
    bapimathead-matl\_type = wa\_tab-mtart.  
    bapimathead-basic\_view  = 'X'.  
  
    bapi\_marax-matl\_group   = 'X'.  
    bapi\_marax-base\_uom     = 'X'.  
  
    bapi\_mara-matl\_group = wa\_tab-matkl.  
    bapi\_mara-base\_uom = wa\_tab-meins.  
  
    wa\_tab2-langu = 'E'.  
    wa\_tab2-matl\_desc = wa\_tab-maktx.  
  
    APPEND wa\_tab2 TO lt\_tab2.  
  
    CALL FUNCTION 'BAPI\_MATERIAL\_SAVEDATA'  
      EXPORTING  
        headdata            = bapimathead  
        clientdata          = bapi\_mara  
        clientdatax         = bapi\_marax  
        plantdata           = bapi\_marc  
        plantdatax          = bapi\_marcx  
      IMPORTING  
        return              = it\_return  
      TABLES  
        materialdescription = lT\_TAB2.  
  
    WRITE : / it\_return-type,  
      3  it\_return-id,  
     23 it\_return-number,  
     25 it\_return-message,  
     it\_return-log\_no,  
     it\_return-message\_v1,  
     it\_return-message\_v2,  
     it\_return-message\_v3,  
     it\_return-message\_v4,  
     it\_return-parameter,  
     it\_return-row,  
     it\_return-field.  
  
  
  ENDLOOP.  
  
  
  
  
  
  
  
  
  
  .

\*&---------------------------------------------------------------------\*  
\*& Report ZPRASAD\_BAPI\_WITHOUT\_SELECT  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ZPRASAD\_BAPI\_WITHOUT\_SELECT.  
  
data : BEGIN OF lt\_tab occurs 0  
  .  
  include STRUCTURE BAPIEKpO.   " for no header  
data : END OF lt\_tab.  
\*data : lt\_tab type STANDARD TABLE OF bapiekko. " new updated.  
  
  
PARAMETERS p\_ebeln TYPE ekko-ebeln.  
  
CALL FUNCTION 'BAPI\_PO\_GETDETAIL'  
  EXPORTING  
    purchaseorder                    = p\_ebeln  
\*   ITEMS                            = 'X'  
\*   ACCOUNT\_ASSIGNMENT               = ' '  
\*   SCHEDULES                        = ' '  
\*   HISTORY                          = ' '  
\*   ITEM\_TEXTS                       = ' '  
\*   HEADER\_TEXTS                     = ' '  
\*   SERVICES                         = ' '  
\*   CONFIRMATIONS                    = ' '  
\*   SERVICE\_TEXTS                    = ' '  
\*   EXTENSIONS                       = ' '  
\* IMPORTING  
\*   PO\_HEADER                        =  
\*   PO\_ADDRESS                       =  
 TABLES  
\*   PO\_HEADER\_TEXTS                  =  
   PO\_ITEMS                         = lt\_tab.  
\*   PO\_ITEM\_ACCOUNT\_ASSIGNMENT       =  
\*   PO\_ITEM\_SCHEDULES                =  
\*   PO\_ITEM\_CONFIRMATIONS            =  
\*   PO\_ITEM\_TEXTS                    =  
\*   PO\_ITEM\_HISTORY                  =  
\*   PO\_ITEM\_HISTORY\_TOTALS           =  
\*   PO\_ITEM\_LIMITS                   =  
\*   PO\_ITEM\_CONTRACT\_LIMITS          =  
\*   PO\_ITEM\_SERVICES                 =  
\*   PO\_ITEM\_SRV\_ACCASS\_VALUES        =  
\*   RETURN                           =  
\*   PO\_SERVICES\_TEXTS                =  
\*   EXTENSIONOUT                     =  
\*   NFMETALLITMS                     =  
          .  
  
\*cl\_demo\_output=>display( lt\_tab ).  
loop at lt\_tab.  
  WRITE : /  
  lt\_tab-acctasscat,  
  lt\_tab-ackn\_reqd,  
  lt\_tab-address,  
  lt\_tab-co\_code.  
ENDLOOP.

\*&---------------------------------------------------------------------\*  
\*& Report ZPRASAD\_EVENTS  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ZPRASAD\_EVENTS.  
\*class c1 DEFINITION.  
\*  PUBLIC SECTION.  
\*   EVENTS e1.  
\*  methods : m1 FOR EVENT e1 of c1.  
\*  METHODs t1.  
\*  
\*ENDCLASS.  
\*  
\*CLASS c1 IMPLEMENTATION.  
\*  METHOD m1 .  
\*    WRITE 'this is m1'.  
\*  
\*  ENDMETHOD.  
\*  METHOD t1.  
\*    WRITE 'This is triggering Event'.  
\*        raise EVENT e1.  
\*  ENDMETHOD.  
\*ENDCLASS.  
\*  
\*START-OF-SELECTION.  
\*data : i\_obj TYPE REF TO c1.  
\*CREATE OBJECT i\_obj.  
\*set HANDLER i\_obj->m1 for i\_obj.     "set handler i\_obj->m1 for i\_obj.  
\*i\_obj->t1( )..                       " call only trigger event it will automatically calls handler event  
  
  
  
\* #2\*  
\*  
\*CLASS c1 DEFINITION.  
\*  PUBLIC SECTION.  
\*  METHODs t1. " triggering event  
\*  EVENTS e1.  
\*ENDCLASS.  
\*  
\*CLASS c2 DEFINITION.  
\*  PUBLIC SECTION.  
\*  METHODS m1 for EVENT e1 of c1.  
\*ENDCLASS.  
\*  
\*  
\*CLASS c1 IMPLEMENTATION.  
\*  METHOD t1.  
\*    WRITE 'iam the triggering event'.  
\*     RAISE EVENT e1.  
\*   ENDMETHOD.  
\*ENDCLASS.  
\*  
\*CLASS c2 IMPLEMENTATION.  
\*  METHOD m1 .  
\*    WRITE 'event is handled'.  
\*  ENDMETHOD.  
\*ENDCLASS.  
\*  
\*START-OF-SELECTION.  
\*data : i\_obj TYPE REF TO c1.  
\*data : i\_obj1 TYPE REF TO c2.  
\*CREATE OBJECT i\_obj.  
\*CREATE OBJECT i\_obj1.  
\*SET HANDLER i\_obj1->m1 for i\_obj.  
\*i\_obj->t1( ).  
  
  
  
  
\*CLASS main DEFINITION.  
\*  PUBLIC SECTION.  
\*  data : i\_num TYPE i.  
\*  METHODs : process IMPORTING lv\_num type i.  
\*  EVENTS : cutoff.  
\*ENDCLASS.  
\*  
\*CLASS event\_handler DEFINITION.  
\*  PUBLIC SECTION.  
\*  METHODS event\_hendler\_method for EVENT cutoff of main.  
\* ENDCLASS.  
\*  
\* CLASS main IMPLEMENTATION.  
\*   METHOD process.  
\*     i\_num = lv\_num.  
\*     if i\_num ge 3.  
\*       RAISE EVENT cutoff.  
\*     ENDIF.  
\*  ENDMETHOd.  
\* ENDCLASS.  
\*  
\* CLASS event\_handler IMPLEMENTATION.  
\*   METHOD event\_hendler\_method.  
\*     WRITE 'event processed'.  
\*   ENDMETHOD.  
\* ENDCLASS.  
\*  
\* START-OF-SELECTION.  
\* data : i\_obj TYPE REF TO main,  
\*       i\_obj2 TYPE REF TO event\_handler.  
\* create OBJECT : i\_obj,i\_obj2.  
\* SET HANDLER : i\_obj2->event\_hendler\_method for i\_obj.  
\* i\_obj->process( lv\_num = 30 ).  
  
  
\*  
\*INITIALIZATION.  
\*WRITE 'hello'.  
\*at SELECTION-SCREEN.  
\*  WRITE 'hai'.  
\*START-OF-SELECTION.  
\*WRITE 'bye'.  
\*TOP-OF-PAGE.  
\*WRITE 'header'.  
\*at SELECTION-SCREEN OUTPUT.  
\*  WRITE 'output'.  
  
  
CLASS c1 DEFINITION.  
  PUBLIC SECTION.  
  EVENTS e1.  
  METHODS m1 FOR EVENT e1 of c1.  
  METHODS t1.  
ENDCLASS.  
  
CLASS c2 DEFINITION.  
  PUBLIC SECTION.  
  METHODS m2 FOR EVENT e1 of c1.  
ENDCLASS.  
  
CLASS c1 IMPLEMENTATION.  
  METHOD m1.  
    WRITE 'this is m1 in c1'.  
  ENDMETHOD.  
  METHOD t1 .  
    WRITE 'this is t1 in c1'.  
    RAISE EVENT e1.  
  ENDMETHOD.  
 ENDCLASS.  
  
 CLASS c2 IMPLEMENTATION.  
   METHOD m2 .  
     WRITE 'this is m2'.  
    ENDMETHOD.  
  ENDCLASS.  
  
  START-OF-SELECTION.  
  data : i\_obj TYPE REF TO c1,  
        i\_obj2 TYPE REF TO c2.  
 CREATE OBJECT : i\_obj , i\_obj2.  
 SET HANDLER:  i\_obj2->m2 for i\_obj,  
              i\_obj->m1 for i\_obj.  
  
 i\_obj->t1( ).

\*&---------------------------------------------------------------------\*  
\*& Report ZPRASAD\_FILE\_HANDLING1  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ZPRASAD\_FILE\_HANDLING1.  
TYPES: BEGIN OF LS\_TAB,  
         matnr TYPE matnr,  
         mbrsh TYPE mbrsh,  
         mtart TYPE mtart,  
         maktx TYPE maktx,  
         meins TYPE meins,  
         matkl TYPE matkl,  
  
       END OF LS\_tab.  
  
 DATA : LT\_TAB TYPE STANDARD TABLE OF LS\_TAB,  
        WA\_TAB TYPE LS\_TAB.  
  
 PARAMETERS P\_FILE TYPE LOCALFILE.  
  
AT SELECTION-SCREEN ON VALUE-REQUEST FOR P\_FILE.  
  CALL FUNCTION 'F4\_FILENAME'  
   EXPORTING  
     PROGRAM\_NAME        = SYST-CPROG  
     DYNPRO\_NUMBER       = SYST-DYNNR  
\*     FIELD\_NAME          = ' '  
   IMPORTING  
     FILE\_NAME           = P\_FILE  
.  
 DATA : LV\_FILE TYPE STRING.  
 LV\_FILE = P\_FILE.  
  
 CALL FUNCTION 'GUI\_UPLOAD'  
   EXPORTING  
     filename                      = LV\_FILE  
   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                      = LT\_TAB  
\*  CHANGING  
\*    ISSCANPERFORMED               = ' '  
\*  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.  
  
  
            .  
cl\_demo\_output=>display(  
  data    =   lt\_tab               " Text or Data  
\*  name    =                  " Name  
\*  exclude =                  " Exclude structure components  
\*  include =                  " Include structure components  
).

\*&---------------------------------------------------------------------\*  
\*& Report ZPRASAD\_FILE\_ITAB\_TO\_SERVER  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ZPRASAD\_FILE\_ITAB\_TO\_SERVER.  
  
TYPES: BEGIN OF LS\_TAB,  
         matnr TYPE matnr,  
         mbrsh TYPE mbrsh,  
         mtart TYPE mtart,  
         maktx TYPE maktx,  
         meins TYPE meins,  
         matkl TYPE matkl,  
  
       END OF LS\_tab.  
  
 DATA : LT\_TAB TYPE STANDARD TABLE OF LS\_TAB,  
        WA\_TAB TYPE LS\_TAB.  
  
 PARAMETERS P\_FILE TYPE LOCALFILE.  
  
AT SELECTION-SCREEN ON VALUE-REQUEST FOR P\_FILE.  
  CALL FUNCTION 'F4\_FILENAME'  
   EXPORTING  
     PROGRAM\_NAME        = SYST-CPROG  
     DYNPRO\_NUMBER       = SYST-DYNNR  
\*     FIELD\_NAME          = ' '  
   IMPORTING  
     FILE\_NAME           = P\_FILE  
.  
 DATA : LV\_FILE TYPE STRING.  
 LV\_FILE = P\_FILE.  
  
 CALL FUNCTION 'GUI\_UPLOAD'  
   EXPORTING  
     filename                      = LV\_FILE  
   FILETYPE                      = 'ASC'  
   HAS\_FIELD\_SEPARATOR           = 'X'  
  
   tables  
     data\_tab                      = LT\_TAB  
  
           .  
 IF sy-subrc <> 0.  
\* Implement suitable error handling here  
 ENDIF.  
  
  
            .  
cl\_demo\_output=>display(  
  data    =   lt\_tab               " Text or Data  
\*  name    =                  " Name  
\*  exclude =                  " Exclude structure components  
\*  include =                  " Include structure components  
).  
  
open DATASET 'prasad\_laptop\_info' FOR OUTPUT IN TEXT MODE ENCODING DEFAULT.  
loop at lt\_tab INTO wa\_tab.  
  TRANSFER wa\_tab to 'prasad\_laptop\_info'.  
  CLEAr wa\_tab.  
ENDLOOP.  
if sy-subrc eq 0.  
  MESSAGE i001(zprasad\_ust\_msg).  
ENDIF.

\*&---------------------------------------------------------------------\*  
\*& Report ZPRASAD\_FILE\_SERVER\_LOCAL  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ZPRASAD\_FILE\_SERVER\_LOCAL.  
  
TYPES: BEGIN OF LS\_TAB,  
         matnr TYPE matnr,  
         mbrsh TYPE mbrsh,  
         mtart TYPE mtart,  
         maktx TYPE maktx,  
         meins TYPE meins,  
         matkl TYPE matkl,  
  
       END OF LS\_tab.  
  
 DATA : LT\_TAB TYPE STANDARD TABLE OF LS\_TAB,  
        WA\_TAB TYPE LS\_TAB.  
  
 PARAMETERS P\_FILE TYPE LOCALFILE.  
  
AT SELECTION-SCREEN ON VALUE-REQUEST FOR P\_FILE.  
  CALL FUNCTION 'F4\_FILENAME'  
   EXPORTING  
     PROGRAM\_NAME        = SYST-CPROG  
     DYNPRO\_NUMBER       = SYST-DYNNR  
\*     FIELD\_NAME          = ' '  
   IMPORTING  
     FILE\_NAME           = P\_FILE  
.  
  START-OF-SELECTION.  
 DATA : LV\_FILE TYPE STRING.  
 LV\_FILE = P\_FILE.  
  
 OPEN DATASET 'prasad\_laptop\_info'  in TEXT MODE for INPUT ENCODING DEFAULT .  
 do.  
 READ DATASET 'prasad\_laptop\_info' INTO WA\_TAB.  
   IF SY-SUBRC EQ 0.  
      APPEND WA\_TAB TO LT\_TAB.  
      CLEAr wa\_tab.  
   ELSE.  
     EXIT.  
     ENDIF.  
 ENDDO.  
 CLOSE DATASET 'prasad\_laptop\_info' .  
  
  
 CALL FUNCTION 'GUI\_DOWNLOAD'  
   EXPORTING  
\*    BIN\_FILESIZE                    =  
     filename                        = LV\_FILE  
    FILETYPE                        = 'ASC'  
\*    APPEND                          = ' '  
    WRITE\_FIELD\_SEPARATOR           = 'X'  
  
   tables  
     data\_tab                        = LT\_TAB  
  
           .  
 IF sy-subrc <> 0.  
\* Implement suitable error handling here  
 ENDIF.  
 CL\_DEMO\_OUTPUT=>display( LT\_TAB ).

\*&---------------------------------------------------------------------\*  
\*& Report ZPRASAD\_FILE\_SERVER\_LOCAL\_2  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ZPRASAD\_FILE\_SERVER\_LOCAL\_2.  
\*&---------------------------------------------------------------------\*  
\*& Report ZPRASAD\_FILE\_SERVER\_LOCAL  
\*&------------------------------------------------  
  
TYPES: BEGIN OF LS\_TAB,  
         matnr TYPE matnr,  
         mbrsh TYPE mbrsh,  
         mtart TYPE mtart,  
         maktx TYPE maktx,  
         meins TYPE meins,  
         matkl TYPE matkl,  
  
       END OF LS\_tab.  
  
 DATA : LT\_TAB TYPE STANDARD TABLE OF LS\_TAB,  
        WA\_TAB TYPE LS\_TAB.  
  
 PARAMETERS P\_FILE TYPE LOCALFILE.  
  
   START-OF-SELECTION.  
 select a~matnr a~mbrsh a~mtart b~maktx a~meins a~matkl  
   into TABLE lt\_tab  
   from mara as a INNER  JOIN makt as b  
   on mara~matnr eq makt~matnr.  
\*AT SELECTION-SCREEN ON VALUE-REQUEST FOR P\_FILE.  
\*  CALL FUNCTION 'F4\_FILENAME'  
\*   EXPORTING  
\*     PROGRAM\_NAME        = SYST-CPROG  
\*     DYNPRO\_NUMBER       = SYST-DYNNR  
\*\*     FIELD\_NAME          = ' '  
\*   IMPORTING  
\*     FILE\_NAME           = P\_FILE  
\*.  
  
 DATA : LV\_FILE TYPE STRING.  
 LV\_FILE = P\_FILE.  
  
OPEN DATASET 'prasad\_laptop\_info' in TEXT MODE for APPENDING ENCODING DEFAULT.  
LOOP AT  lt\_tab INTO wa\_tab.  
  TRANSFER wa\_tab to 'prasad\_laptop\_info'.  
  clear wa\_tab.  
ENDLOOP.  
CLOSE DATASET 'prasad\_laptop\_info'.  
  
 OPEN DATASET 'prasad\_laptop\_info'  in TEXT MODE for INPUT ENCODING DEFAULT .  
 do.  
 READ DATASET 'prasad\_laptop\_info' INTO WA\_TAB.  
   IF SY-SUBRC EQ 0.  
      APPEND WA\_TAB TO LT\_TAB.  
      CLEAr wa\_tab.  
   ELSE.  
     EXIT.  
     ENDIF.  
 ENDDO.  
 CLOSE DATASET 'prasad\_laptop\_info' .  
  
  
 CALL FUNCTION 'GUI\_DOWNLOAD'  
   EXPORTING  
\*    BIN\_FILESIZE                    =  
     filename                        = LV\_FILE  
    FILETYPE                        = 'ASC'  
\*    APPEND                          = ' '  
    WRITE\_FIELD\_SEPARATOR           = 'X'  
  
   tables  
     data\_tab                        = LT\_TAB  
  
           .  
 IF sy-subrc <> 0.  
\* Implement suitable error handling here  
 ENDIF.  
 CL\_DEMO\_OUTPUT=>display( LT\_TAB ).

REPORT ZPRASAD\_INTERACTIVE\_ALV\_LINE.  
  
data : lt\_tab like table of vbak,  
       wa\_tab like LINE OF lt\_tab,  
       lt\_tab2 like table of vbaP,  
       wa\_tab2 like LINE OF lt\_tab2,  
  
       lv\_vbeln type vbeln,  
       it\_filedcat type slis\_t\_fieldcat\_alv.  
  
SELECTION-SCREEN BEGIN OF BLOCK a1 WITH FRAME TITLE text-100.  
  SELECT-OPTIONS S\_vbeln FOR lv\_vbeln.  
SELECTION-SCREEN END OF BLOCK a1.  
  
START-OF-SELECTION.  
  
  SELECT \*  
    into TABLE lt\_tab  
    from vbak  
    WHERE vbeln in s\_vbeln.  
  
    CALL FUNCTION 'REUSE\_ALV\_GRID\_DISPLAY'  
     EXPORTING  
\*       I\_INTERFACE\_CHECK                 = ' '  
\*       I\_BYPASSING\_BUFFER                = ' '  
\*       I\_BUFFER\_ACTIVE                   = ' '  
       I\_CALLBACK\_PROGRAM                = sy-repid  
\*       I\_CALLBACK\_PF\_STATUS\_SET          = ' '  
       I\_CALLBACK\_USER\_COMMAND           = 'PICK'  
\*       I\_CALLBACK\_TOP\_OF\_PAGE            = ' '  
\*       I\_CALLBACK\_HTML\_TOP\_OF\_PAGE       = ' '  
\*       I\_CALLBACK\_HTML\_END\_OF\_LIST       = ' '  
      I\_STRUCTURE\_NAME                  = 'VBAK'  
\*       I\_BACKGROUND\_ID                   = ' '  
\*       I\_GRID\_TITLE                      =  
\*       I\_GRID\_SETTINGS                   =  
\*       IS\_LAYOUT                         =  
\*       IT\_FIELDCAT                       = it\_filedcat  
\*       IT\_EXCLUDING                      =  
\*       IT\_SPECIAL\_GROUPS                 =  
      TABLES  
        t\_outtab                          = lt\_tab  
\*     EXCEPTIONS  
\*       PROGRAM\_ERROR                     = 1  
\*       OTHERS                            = 2  
              .  
    IF sy-subrc <> 0.  
\* Implement suitable error handling here  
    ENDIF.  
  
    FORM PICK USING P\_COMMAND TYPE SY-UCOMM  
               CHANGING P\_SCL TYPE  slis\_selfield  .  
  
      READ TABLE LT\_TAB INTO WA\_TAB INDEX P\_SCL-tabindex.  
      CASE  P\_COMMAND.  
        WHEN '&IC1'.  
              SELECT \*  
                INTO TABLE LT\_TAB2  
                FROM VBAP  
                WHERE VBELN IN S\_VBELN.  
  
  
  
          data: t\_fieldcat type slis\_t\_fieldcat\_alv,  
      t\_eventcat type slis\_t\_event,  
      w\_eventcat like line of t\_eventcat.  
  
  
  
  
  
                CALL FUNCTION 'REUSE\_ALV\_FIELDCATALOG\_MERGE'  
                 EXPORTING  
\*                   I\_PROGRAM\_NAME               =  
\*                   I\_INTERNAL\_TABNAME           =  
                   I\_STRUCTURE\_NAME             = 'VBAP'  
\*                   I\_CLIENT\_NEVER\_DISPLAY       = 'X'  
\*                   I\_INCLNAME                   =  
\*                   I\_BYPASSING\_BUFFER           =  
\*                   I\_BUFFER\_ACTIVE              =  
                  CHANGING  
                    ct\_fieldcat                  = t\_fieldcat  
\*                 EXCEPTIONS  
\*                   INCONSISTENT\_INTERFACE       = 1  
\*                   PROGRAM\_ERROR                = 2  
\*                   OTHERS                       = 3  
                          .  
                IF sy-subrc <> 0.  
\* Implement suitable error handling here  
                ENDIF.  
  
  
                CALL FUNCTION 'REUSE\_ALV\_GRID\_DISPLAY'  
                 EXPORTING  
\*                   I\_INTERFACE\_CHECK                 = ' '  
\*                   I\_BYPASSING\_BUFFER                = ' '  
\*                   I\_BUFFER\_ACTIVE                   = ' '  
                   I\_CALLBACK\_PROGRAM                = SY-REPID  
\*                   I\_CALLBACK\_PF\_STATUS\_SET          = ' '  
\*                   I\_CALLBACK\_USER\_COMMAND           = ' '  
\*                   I\_CALLBACK\_TOP\_OF\_PAGE            = ' '  
\*                   I\_CALLBACK\_HTML\_TOP\_OF\_PAGE       = ' '  
\*                   I\_CALLBACK\_HTML\_END\_OF\_LIST       = ' '  
                   I\_STRUCTURE\_NAME                  = 'VBAP'  
\*                   I\_BACKGROUND\_ID                   = ' '  
\*                   I\_GRID\_TITLE                      =  
\*                   I\_GRID\_SETTINGS                   =  
\*                   IS\_LAYOUT                         =  
\*                   IT\_FIELDCAT                       = I  
\*                   IT\_EXCLUDING                      =  
\*                   IT\_SPECIAL\_GROUPS                 =  
\*                   IT\_SORT                           =  
\*                   IT\_FILTER                         =  
\*                   IS\_SEL\_HIDE                       =  
\*                   I\_DEFAULT                         = 'X'  
\*                   I\_SAVE                            = ' '  
\*                   IS\_VARIANT                        =  
\*                   IT\_EVENTS                         =  
\*                   IT\_EVENT\_EXIT                     =  
\*                   IS\_PRINT                          =  
\*                   IS\_REPREP\_ID                      =  
\*                   I\_SCREEN\_START\_COLUMN             = 0  
\*                   I\_SCREEN\_START\_LINE               = 0  
\*                   I\_SCREEN\_END\_COLUMN               = 0  
\*                   I\_SCREEN\_END\_LINE                 = 0  
\*                   I\_HTML\_HEIGHT\_TOP                 = 0  
\*                   I\_HTML\_HEIGHT\_END                 = 0  
\*                   IT\_ALV\_GRAPHICS                   =  
\*                   IT\_HYPERLINK                      =  
\*                   IT\_ADD\_FIELDCAT                   =  
\*                   IT\_EXCEPT\_QINFO                   =  
\*                   IR\_SALV\_FULLSCREEN\_ADAPTER        =  
\*                   O\_PREVIOUS\_SRAL\_HANDLER           =  
\*                   O\_COMMON\_HUB                      =  
\*                 IMPORTING  
\*                   E\_EXIT\_CAUSED\_BY\_CALLER           =  
\*                   ES\_EXIT\_CAUSED\_BY\_USER            =  
                  TABLES  
                    t\_outtab                          = LT\_TAB2  
\*                 EXCEPTIONS  
\*                   PROGRAM\_ERROR                     = 1  
\*                   OTHERS                            = 2  
                          .  
                IF sy-subrc <> 0.  
\* Implement suitable error handling here  
                ENDIF.  
  
  
  
      ENDCASE.  
      ENDFORM.

\*&---------------------------------------------------------------------\*  
\*& Report ZPRASAD\_RFC\_CONN  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ZPRASAD\_RFC\_CONN.  
data : lv\_matnr TYPE matnr,  
      lt\_tab type  ZPRASAD\_MARA\_STR\_TT.  
select-OPTIONS s\_matnr for lv\_matnr.  
CALL FUNCTION 'ZPRASAD\_FUN\_RFC' DESTINATION 'zprasad\_rfc'  
  EXPORTING  
    lv\_matnr1       = s\_matnr-low  
    lv\_matnr2       = s\_matnr-high  
  TABLES  
    gt\_tab          = lt\_tab  
          .  
  
  
 cl\_demo\_output=>display( lt\_tab ).

Rfc..

Bapi…

Swo1

Al11