**MAIN PROGRAM**

\*&---------------------------------------------------------------------\*  
\*& Report Z280416\_REP\_ADOBE  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
report z280416\_rep\_adobe MESSAGE-ID zvish\_msg.  
  
\*-------------------------------------------------\*  
\*  TOP INCLUDE FOR THE DECLERATIONS  
\*-------------------------------------------------\*  
include z280416\_include\_top.  
  
at selection-screen.  
\*-------------------------------------------------\*  
\*  VALIDATIONS FOR THE INPUTS  
\*-------------------------------------------------\*  
  perform validating\_vbeln\_input.  
  perform validating\_fkdat\_input.  
  perform validating\_kunnr\_input.  
  
  
start-of-selection.  
  
\*-------------------------------------------------------\*  
\*  GETTING DATA FROM DICTIONARY INTO INTERNAL TABLES  
\*-------------------------------------------------------\*  
INCLUDE ZVISH\_GETDATA.  
  
  
  
\*-------------------------------------------------------------\*  
\*  APPENDING INTERNAL TABLES DATA INTO SINGLE AND FINAL TABLE  
\*-------------------------------------------------------------\*  
INCLUDE ZVISH\_FORMATTING\_DATA.  
  
  
  
\*---------------------------------------------------------------------------------------------------\*  
\*  GIVING OUTPUT PARAMETERS TO AVOID DYNAMICALLY GIVING PRINTER DETAILS AFTER EXECUTING THE REPORT  
\*---------------------------------------------------------------------------------------------------\*  
wa\_outputparams-device = 'PRINTER'.  
wa\_outputparams-dest = 'LP01'.  
wa\_outputparams-nodialog = 'X'.  
wa\_outputparams-preview = 'X'.  
  
  
  
\*-------------------------------------------------------\*  
\*  CALLING REQUIRED FUNCTION MODULES  
\*-------------------------------------------------------\*  
INCLUDE ZVISH\_CALLING\_FMS.  
  
  
\*-------------------------------------------------------\*  
\*  FORM IMPLEMENTATIONS  
\*-------------------------------------------------------\*  
form validating\_vbeln\_input .  
 SELECT SINGLE vbeln  
    INTO lv\_vbeln1  
     FROM vbrk  
  WHERE vbeln IN s\_vbeln.  
  
  IF sy-subrc EQ 0.  
    MESSAGE s000.  
  ELSE.  
    MESSAGE e007.  
  ENDIF.  
endform.  
  
form validating\_fkdat\_input .  
SELECT SINGLE vbeln  
    INTO lv\_vbeln1  
     FROM vbrk  
  WHERE fkdat IN s\_fkdat.  
  
  IF sy-subrc EQ 0.  
    MESSAGE s000.  
  ELSE.  
    MESSAGE e006.  
  ENDIF.  
endform.  
  
form validating\_kunnr\_input .  
 SELECT SINGLE vbeln  
    INTO lv\_vbeln1  
     FROM vbrk  
  WHERE kunrg IN s\_kunrg.  
  
  IF sy-subrc EQ 0.  
    MESSAGE s000.  
  ELSE.  
    MESSAGE e005.  
  ENDIF.  
endform.

ALL INCLUDES:

\*&---------------------------------------------------------------------\*  
\*& Include          Z280416\_INCLUDE\_TOP  
\*&---------------------------------------------------------------------\*  
  
data : lv\_vbeln        type vbeln\_vf, " DOCUMENT NUMBER FOR SELECT OPTIONS  
       lv\_fkdat        type fkdat,  
       lv\_kunrg        type kunrg,  
       lv\_fmname       type funcname,  
       lt\_final        type z280416\_tt\_vbrk,  
       wa\_outputparams type sfpoutputparams,  
       lv\_vbeln1       type vbeln\_vf,  
       wa\_docparams    type sfpdocparams.  
  
select-options : s\_vbeln for lv\_vbeln,  
                  s\_fkdat for lv\_fkdat obligatory,  
                  s\_kunrg for lv\_kunrg.

\*&---------------------------------------------------------------------\*  
\*& Include          ZVISH\_GETDATA  
\*&---------------------------------------------------------------------\*  
select vbeln, fkdat, kunrg, netwr  
    from vbrk  
    into table @data(lt\_vbrk)  
  where vbeln in @s\_vbeln and fkdat in @s\_fkdat and fkart = 'F2' and gbstk = 'C' and kunrg in @s\_kunrg.  
  
  
  select vbeln,posnr, aubel ,aupos , vgbel ,vgpos , matnr,fkimg, vrkme,netwr,waerk  
    from vbrp  
     into table @data(lt\_vbrp)  
    for all entries in @lt\_vbrk  
  where vbeln = @lt\_vbrk-vbeln.  
  
  
  select kunnr,name1, name2, ort01,pstlz ,land1 from kna1  
   into table @data(lt\_tab3)  
     for all entries in @lt\_vbrk  
  where kunnr = @lt\_vbrk-kunrg.

\*&---------------------------------------------------------------------\*  
\*& Include          ZVISH\_FORMATTING\_DATA  
\*&---------------------------------------------------------------------\*  
  
 if lt\_vbrp is not initial.  
    sort lt\_vbrk by vbeln.  
    sort lt\_vbrp by vbeln.  
    data(gv\_index) = value syst\_tabix( ).  
    loop at lt\_vbrk assigning field-symbol(<lfs\_vbrk>).  
      append initial line to lt\_final assigning field-symbol(<gfs\_final>).  
      move-corresponding <lfs\_vbrk> to <gfs\_final>."MOVING HEADER DETAILS FROM INRTNAL TABLE TO FINAL TABLE  
  
      read table lt\_vbrp transporting no fields  
      with key vbeln = <lfs\_vbrk>-vbeln binary search.  
      if sy-subrc is initial.  
        gv\_index = sy-tabix.  
  
        loop at lt\_vbrp assigning field-symbol(<lfs\_vbrp>) from gv\_index.  
          if <lfs\_vbrp>-vbeln ne <lfs\_vbrk>-vbeln.  
            exit.  
          endif.  
          append initial line to <gfs\_final>-lineitems assigning field-symbol(<lfs\_items>).  
          move-corresponding <lfs\_vbrp> to <lfs\_items>."MOVING LINEITEMS DETAILS FROM INRTNAL TABLE TO FINAL TABLE  
        endloop.  
      endif.  
    endloop.  
  endif.  
  
\*MOVING CUSTOMER DETAILS FROM INRTNAL TABLE TO FINAL TABLE  
  if lt\_tab3 is not initial.  
    sort lt\_final by kunrg.  
    sort lt\_tab3 by kunnr.  
    loop at lt\_final assigning field-symbol(<lfs\_final2>).  
      loop at lt\_tab3 assigning field-symbol(<lfs\_kna1>).  
        if <lfs\_final2>-kunrg eq <lfs\_kna1>-kunnr.  
          move-corresponding <lfs\_kna1> to <lfs\_final2>.  
        endif.  
      endloop.  
    endloop.  
    endif.  
sort lt\_final by vbeln." FINAL SORTING BEFORE PRINTING

\*&---------------------------------------------------------------------\*  
\*& Include          ZVISH\_CALLING\_FMS  
\*&---------------------------------------------------------------------\*  
  
  
call function 'FP\_JOB\_OPEN'  
  changing  
    ie\_outputparams       = wa\_outputparams  
          .  
if sy-subrc <> 0.  
\* Implement suitable error handling here  
endif.  
  
  
  
 call function 'FP\_FUNCTION\_MODULE\_NAME'  
   exporting  
    i\_name     = 'Z280416\_ADOBE\_FORM'  
   importing  
    e\_funcname = lv\_fmname  "IMPORTING FUNCTION MODULE NAME INTO LOCAL VARIABLE  
    .  
 call function lv\_fmname "'/1BCDWB/SM00000927' "AVOIDING STATICALLY GIVING FUNCTION MODULE NAME BECAUSE IT LEADS TO DUMPS WHEN THE GENERATED FUNCTION MODULE NAME CHANGES  
   exporting  
        gt\_final = lt\_final  
    .  
  if sy-subrc <> 0.  
\* Implement suitable error handling here  
  endif.  
  
call function 'FP\_JOB\_CLOSE'  
\* IMPORTING  
\*   E\_RESULT             =  
\* EXCEPTIONS  
\*   USAGE\_ERROR          = 1  
\*   SYSTEM\_ERROR         = 2  
\*   INTERNAL\_ERROR       = 3  
\*   OTHERS               = 4  
          .  
if sy-subrc <> 0.  
\* Implement suitable error handling here  
endif.

OUTPUT ADOBE FORM

A screenshot of a computer

Description automatically generated

FORM SCREENSHOTS:

A screenshot of a document

Description automatically generated

A screenshot of a document

Description automatically generated

A screenshot of a computer

Description automatically generated