ASSESSMENT DOC

\*&---------------------------------------------------------------------\*  
\*& Report Z280403\_ADOBE\_DRIVER\_PROG  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT z280403\_adobe\_driver\_prog.  
  
\*\*D A T A   D E C L A R A T I O N S  
INCLUDE z280403\_data\_delcaration\_top.  
  
\*\*I N P U T   I N I T I A L I Z A T I O N  
INCLUDE z280403\_initialization.  
  
\*\* U S E R   I N P U T   V A L I D A T I O N  
include z280403\_input\_validation.  
  
\*\*M A I N   S Q L  L O G I C S  
INCLUDE z280403\_get\_the\_data.  
  
\*\*C A L L I N G   F U N C T I O N   M O D U L E S / F O R M  
INCLUDE z280403\_call\_fms.

**INCLUDE FILES:**

\*&---------------------------------------------------------------------\*  
\*& Include          Z280403\_DATA\_DELCARATION\_TOP  
\*&---------------------------------------------------------------------\*  
##NEEDED  
TYPES: ty\_outparams TYPE sfpoutputparams,  
       BEGIN OF ls\_head,  
         kunnr TYPE  kunnr, " customer number  
         name1 TYPE  name1, " name  
         name2 TYPE  name2, " name2  
         ort01 TYPE  ort01, " city  
         pstlz TYPE  pstlz, " picode  
         land1 TYPE  land1, " Country/Region Key  
         vbeln TYPE  vbeln,  " Invoice Doc Num  
         netwr TYPE  netwr\_ak, " total cost  
         fkdat TYPE  fkdat,     " Invoice Document Date  
       END OF ls\_head.  
  
DATA: wa\_outparams TYPE sfpoutputparams, " to control the printer  
  
      gv\_fm\_name   TYPE rs38l\_fnam,      " to catch the fm name  
  
      wa\_vbrp2     TYPE z280403\_line\_struct,           " line items work area  
      it\_vbrp2     TYPE TABLE OF z280403\_line\_struct,   " line items internal table  
  
      lt\_head      TYPE TABLE OF ls\_head,  " header data internal table  
      wa\_head      TYPE ls\_head,           "header data work area  
  
      lv\_vbeln     TYPE vbeln, "Invoice Document number  
      lv\_fkdat     TYPE fkdat, "Invoice Document Date  
      lv\_kunnr     TYPE kunnr, "Customer number  
  
      ft\_final     TYPE z280403\_header\_tt, "Deep Structure to store all the data in a single table  
      wa\_final     TYPE z280403\_header\_struct,  
  
      index        TYPE i.  
  
SELECT-OPTIONS : s\_vbeln FOR lv\_vbeln,  
                 s\_fkdat FOR lv\_fkdat OBLIGATORY,  
                 s\_kunnr FOR lv\_kunnr.  
  
\*AT SELECTION-SCREEN. " user input validation  
\*  
\*  SELECT SINGLE vbeln INTO (lv\_vbeln)  
\*    FROM vbrk  
\*    WHERE vbeln IN s\_vbeln AND fkart = 'F2'.  
\*  IF sy-subrc NE 0.  
\*    MESSAGE e000(z280403\_messages).  
\*  ENDIF.  
\*  
\*  SELECT SINGLE fkdat INTO (lv\_fkdat)  
\*    FROM vbrk  
\*    WHERE fkdat IN s\_fkdat AND fkart = 'F2'.  
\*  IF sy-subrc NE 0.  
\*    MESSAGE e001(z280403\_messages).  
\*  ENDIF.  
\*  
\*  SELECT SINGLE kunnr INTO (lv\_kunnr)  
\*    FROM kna1  
\*    WHERE kunnr IN s\_kunnr.  
\*  IF sy-subrc NE 0.  
\*    MESSAGE e002(z280403\_messages).  
\*  ENDIF.

\*&---------------------------------------------------------------------\*  
\*& Include          Z280403\_INITIALIZATION  
\*&---------------------------------------------------------------------\*  
INITIALIZATION.  
  
s\_vbeln-low = '1'.  
s\_vbeln-high = '10000000000'.  
append s\_vbeln.  
  
s\_fkdat-low = '20140101'.  
s\_fkdat-high = '20240101'.  
append s\_fkdat.  
  
s\_kunnr-low = '1'.  
s\_kunnr-high = '10000000000'.  
append s\_kunnr.

\*&---------------------------------------------------------------------\*  
\*& Include          Z280403\_INPUT\_VALIDATION  
\*&---------------------------------------------------------------------\*  
AT SELECTION-SCREEN. " user input validation  
  
##WARN\_OK  
  SELECT SINGLE vbeln INTO (lv\_vbeln)  
    FROM vbrk  
    WHERE vbeln IN s\_vbeln AND fkart = 'F2'.  
  IF sy-subrc NE 0.  
    MESSAGE e000(z280403\_messages).  
  ENDIF.  
  
##WARN\_OK  
  SELECT SINGLE fkdat INTO (lv\_fkdat)  
    FROM vbrk  
    WHERE fkdat IN s\_fkdat AND fkart = 'F2'.  
  IF sy-subrc NE 0.  
    MESSAGE e001(z280403\_messages).  
  ENDIF.  
  
##WARN\_OK  
  SELECT SINGLE kunnr INTO (lv\_kunnr)  
    FROM kna1  
    WHERE kunnr IN s\_kunnr.  
  IF sy-subrc NE 0.  
    MESSAGE e002(z280403\_messages).  
  ENDIF.

\*&---------------------------------------------------------------------\*  
\*& Include          Z280403\_GET\_THE\_DATA  
\*&---------------------------------------------------------------------\*  
START-OF-SELECTION.  
  
  SELECT vbeln, fkdat INTO TABLE @DATA(it\_vbrk)  
    FROM vbrk WHERE vbeln IN @s\_vbeln AND  
    fkdat IN @s\_fkdat AND                     " fetching data based on user input and conditions  
    fkart = 'F2' AND gbstk = 'C'.  
  
  
  SELECT posnr,  
        vbeln,  
        aubel,  
        aupos,  
        vgbel,  
        vgpos,  
        matnr,  
        fkimg,  
        vrkme,  
        netwr,  
        waerk  
    INTO TABLE @DATA(it\_vbrp)  
    FROM vbrp  
    FOR ALL ENTRIES IN @it\_vbrk  
    WHERE vbeln EQ @it\_vbrk-vbeln.  
  
  
  SELECT kunnr, vbeln, netwr INTO TABLE @DATA(it\_vbak)  
    FROM vbak  
    FOR ALL ENTRIES IN @it\_vbrp  
    WHERE vbeln EQ @it\_vbrp-aubel AND kunnr IN @s\_kunnr.  " based on user input KUNNR  
  
  
  SELECT kunnr, name1, name2, ort01, pstlz, land1 INTO TABLE @DATA(it\_kna1)  
    FROM kna1  
    FOR ALL ENTRIES IN @it\_vbak  
    WHERE kunnr EQ @it\_vbak-kunnr.  
  
END-OF-SELECTION.  
  
  
  SORT it\_vbak BY kunnr.  
  SORT it\_vbrp BY aubel.  
  SORT it\_vbrk BY vbeln.  
  
\* combining all the data fetched using FOR ALL ENTRIES into single table  
##INTO\_OK  
  LOOP AT it\_kna1 INTO DATA(wa\_kna1).  
  
    READ TABLE it\_vbak INTO DATA(wa\_vbak) WITH KEY kunnr = wa\_kna1-kunnr BINARY SEARCH.  
    IF sy-subrc = 0.  
  
      READ TABLE it\_vbrp INTO DATA(wa\_vbrp) WITH KEY aubel = wa\_vbak-vbeln BINARY SEARCH.  
      IF sy-subrc EQ 0.  
        READ TABLE it\_vbrk INTO DATA(wa\_vbrk) WITH KEY vbeln = wa\_vbrp-vbeln BINARY SEARCH.  
  
        IF sy-subrc EQ 0.  
  
          MOVE : wa\_kna1-kunnr TO wa\_head-kunnr,  
               wa\_kna1-land1 TO wa\_head-land1,  
               wa\_kna1-name1 TO wa\_head-name1,  
               wa\_kna1-name2 TO wa\_head-name2,  
               wa\_kna1-ort01 TO wa\_head-ort01,  
               wa\_kna1-pstlz TO wa\_head-pstlz,  
  
               wa\_vbak-vbeln TO wa\_head-vbeln,  
               wa\_vbak-netwr TO wa\_head-netwr,  
  
               wa\_vbrk-fkdat TO wa\_head-fkdat.  
  
          APPEND wa\_head TO lt\_head.  
  
        ENDIF.  
  
      ENDIF.  
  
    ENDIF.  
  
  
  ENDLOOP.  
  
\* loading the header data and line items into final table (deep structure table)  
##INTO\_OK  
  LOOP AT it\_vbrp INTO wa\_vbrp.  
  
    MOVE-CORRESPONDING wa\_vbrp TO wa\_vbrp2.  
    APPEND wa\_vbrp2 TO it\_vbrp2.  
  
  ENDLOOP.  
  
  LOOP AT lt\_head INTO wa\_head.  
  
    MOVE-CORRESPONDING wa\_head TO wa\_final.  
  
    READ TABLE it\_vbrp2 TRANSPORTING NO FIELDS WITH KEY aubel = wa\_head-vbeln.  
    IF sy-subrc EQ 0.  
  
      index = sy-tabix.  
  
      LOOP AT it\_vbrp2 INTO wa\_vbrp2 FROM index.  
  
        IF wa\_vbrp2-aubel NE wa\_head-vbeln.  
          EXIT.  
        ENDIF.  
  
        APPEND wa\_vbrp2 TO wa\_final-item.  
        CLEAR wa\_vbrp2.  
      ENDLOOP.  
  
      APPEND wa\_final TO ft\_final.  
      CLEAR wa\_final.  
  
    ENDIF.  
    CLEAR wa\_head.  
  ENDLOOP.

\*&---------------------------------------------------------------------\*  
\*& Include          Z280403\_CALL\_FMS  
\*&---------------------------------------------------------------------\*  
  
  
  wa\_outparams-device = TEXT-000.  
  wa\_outparams-dest = TEXT-001.  
  wa\_outparams-nodialog = TEXT-002.  
  wa\_outparams-preview = TEXT-002.  
  
  CALL FUNCTION 'FP\_JOB\_OPEN'  
    CHANGING  
      ie\_outputparams = wa\_outparams  
\*   EXCEPTIONS  
\*     CANCEL          = 1  
\*     USAGE\_ERROR     = 2  
\*     SYSTEM\_ERROR    = 3  
\*     INTERNAL\_ERROR  = 4  
\*     OTHERS          = 5  
    .  
  
  
  CALL FUNCTION 'FP\_FUNCTION\_MODULE\_NAME'  
    EXPORTING  
      i\_name     = 'Z280403\_ADOBE\_FORM'  
    IMPORTING  
      e\_funcname = gv\_fm\_name.  
  
\*\* CALLING FORM  
  CALL FUNCTION gv\_fm\_name "'/1BCDWB/SM00000916'  
    EXPORTING  
      gt\_tab = ft\_final.  
  
  
  
  CALL FUNCTION 'FP\_JOB\_CLOSE'.

**OUTPUT: A screenshot of a computer

Description automatically generated**