**DATA TYPES**

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_DATA*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_DATA.  
  
DATA X(10) type c value 'UST'.  
WRITE X.  
  
DATA: lv\_x type i value 123,  
      lv\_y type i value 123,  
      lv\_z type i.  
  
lv\_z = lv\_x + lv\_y.  
  
WRITE lv\_z.  
  
  
PARAMETERS : par\_a type i DEFAULT 10,  
        par\_b type i DEFAULT 20.  
  
DATA l\_var type i.  
l\_var = par\_a + par\_b.  
  
WRITE l\_var.

==========================================================================

**STRINGS**

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_STRINGS*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_STRINGS.  
  
*\*\*\*\*\*TRANSLATE\*\*\*\*\*\**  
*\*PARAMETERS : p\_a(10), p\_b(10), p\_c(50) OBLIGATORY.*  
*\*TRANSLATE p\_a to UPPER CASE.*  
*\*TRANSLATE p\_b to LOWER CASE.*  
*\*WRITE / : p\_a, p\_b, p\_c.*  
  
  
*\*\*\*CONCATENATE\*\*\*\**  
*\*PARAMETERS : p\_a(10), p\_b(10), del(10).*  
*\*DATA : p\_c(50),p\_d(20),p\_e(30).*  
*\*CONCATENATE p\_a p\_b into p\_c.*  
*\*CONCATENATE p\_a p\_b into p\_d SEPARATED BY ','.*  
*\*CONCATENATE p\_a p\_b into p\_e SEPARATED BY del.*  
*\*WRITE:/ p\_c, p\_d, p\_e.*  
*\**  
*\**  
  
*\*\*\*\*Split\*\*\*\*\*\*\**  
*\*PARAMETERS: p\_a(20), del(10).*  
*\*DATA : lv\_a(20), lv\_b(20), lv\_p(20), lv\_q(20).*  
*\*SPLIT p\_a at '@' into lv\_a lv\_b.*  
*\**  
*\*\*SPLIT p\_b at del into lv\_p lv\_q.*  
*\*WRITE:/ lv\_a , lv\_b, lv\_p, lv\_q.*  
  
*\*\*\*\*\*\*OFFSETTING\*\*\*\*\*\**  
*\*PARAMETERS p\_a(10).*  
*\*data lv\_dest(10).*  
*\**  
*\*lv\_dest = p\_a+3(4).*  
*\*WRITE lv\_dest.*  
*\*lv\_dest = p\_a+4.*  
*\*WRITE lv\_dest.*  
  
*\*\*\*\*STRING LENGTH\*\*\*\*\**  
*\*PARAMETERS p\_a(40).*  
*\*DATA lv\_len type i.*  
*\*lv\_len = strlen( p\_a ).*  
*\*WRITE:/ p\_a, lv\_len.*  
*\**  
*\*\*\*\*\*\*REPLACE\*\*\*\*\*\**  
*\*PARAMETERS p\_src(20).*  
*\*if p\_src ca ','.*  
*\*  REPLACE ',' with '@' into p\_src.*  
*\*else.*  
*\*  exit.*  
*\*ENDIF.*  
*\*WRITE p\_src.*  
*\**  
*\**  
*\*\*\*\*\*\*CONDENSE(NO GAPS)\*\*\*\*\**  
*\*DATA :  string(50) value '      one two three four       ',*  
*\*      len type i.*  
*\*len = strlen( string ).*  
*\*write: 'len of given string',len,*  
*\*        'given string is',string.*  
*\*CONDENSE string.*  
*\*len = strlen( string ).*  
*\*write :/  'string - ',string,*  
*\*        'len -',len.*  
*\*CONDENSE string NO-GAPS.*  
*\*len = strlen( string ).*  
*\*write :/  'after string with no gaps - ',string,*  
*\*        'and len -',len.*  
*\**  
*\*\*\*\*\*\*\*\*\*\*\*SHIFT\*\*\*\*\*\*\*\*\**  
DATA: str1(50) value 'ABAP PROGRAM',  
      str2(50) value 'ABAP PROGRAM'.  
WRITE: / 'before left shift:', str1.  
         shift str1 left by 10 places.  
 WRITE:/ 'after left shift:', str1.  
  
 WRITE: / 'before right shift:', str2.  
         shift str2 right by 3 places.  
 WRITE:/ 'after right shift:', str2.

==========================================================================

**Select statement**

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_SELECT\_STMT*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_SELECT\_STMT.  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*LFA1\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\*TABLES lfa1.*  
*\**  
*\*DATA : lv\_lifnr type lfa1-lifnr,*  
*\*      lv\_land1 type lfa1-land1,*  
*\*      lv\_ort01 type lfa1-ort01,*  
*\*      lv\_name1 type lfa1-name1,*  
*\*      lv\_pfach type lfa1-pfach,*  
*\*      lv\_regio type lfa1-regio.*  
*\**  
*\*Select mandt land1 ort01 name1 pfach regio*  
*\*  into ( lv\_lifnr, lv\_land1, lv\_ort01, lv\_name1, lv\_pfach, lv\_regio )*  
*\*  from lfa1 where lifnr = '0000000006'.*  
*\**  
*\*WRITE : / lv\_lifnr, lv\_land1, lv\_ort01, lv\_name1, lv\_pfach, lv\_regio.*  
*\*ENDSELECT.*  
  
  
TABLES lfa1.  
  
DATA : lv\_lifnr type lfa1-lifnr,  
      lv\_land1 type lfa1-land1,  
      lv\_ort01 type lfa1-ort01,  
      lv\_name1 type lfa1-name1,  
      lv\_pfach type lfa1-pfach,  
      lv\_regio type lfa1-regio.  
  
PARAMETERS p\_lifnr TYPE lfa1-lifnr.  
  
Select mandt land1 ort01 name1 pfach regio  
  into ( lv\_lifnr, lv\_land1, lv\_ort01, lv\_name1, lv\_pfach, lv\_regio )  
  from lfa1 where lifnr = p\_lifnr.  
  
WRITE : / lv\_lifnr, lv\_land1, lv\_ort01, lv\_name1, lv\_pfach, lv\_regio.  
ENDSELECT.  
  
  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*VBAK\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\*TABLES vbak.*  
*\**  
*\*DATA : lv\_vbeln type vbak-vbeln,*  
*\*      lv\_vbtyp type vbak-vbtyp,*  
*\*      lv\_trvog type vbak-trvog,*  
*\*      lv\_auart type vbak-auart,*  
*\*      lv\_augru type vbak-augru.*  
*\**  
*\*Select vbeln vbtyp trvog auart augru*  
*\*  into ( lv\_vbeln, lv\_vbtyp, lv\_trvog, lv\_auart, lv\_augru )*  
*\*  from vbak where vbeln = '0000000004'.*  
*\**  
*\*WRITE : / lv\_vbeln, lv\_vbtyp, lv\_trvog, lv\_auart, lv\_augru.*  
*\*ENDSELECT.*  
  
*\*TABLES vbak.*  
*\**  
*\*DATA : lv\_vbeln type vbak-vbeln,*  
*\*      lv\_vbtyp type vbak-vbtyp,*  
*\*      lv\_trvog type vbak-trvog,*  
*\*      lv\_auart type vbak-auart,*  
*\*      lv\_augru type vbak-augru.*  
*\**  
*\*PARAMETERS p\_vbeln type vbak-vbeln.*  
*\**  
*\*Select vbeln vbtyp trvog auart augru*  
*\*  into ( lv\_vbeln, lv\_vbtyp, lv\_trvog, lv\_auart, lv\_augru )*  
*\*  from vbak where vbeln = p\_vbeln.*  
*\**  
*\*WRITE : / lv\_vbeln, lv\_vbtyp, lv\_trvog, lv\_auart, lv\_augru.*  
*\*ENDSELECT.*  
  
  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*EKKO\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\*TABLES ekko.*  
*\**  
*\*DATA : lv\_ebeln type ekko-ebeln,*  
*\*      lv\_kufix type ekko-kufix,*  
*\*      lv\_unsez type ekko-unsez,*  
*\*      lv\_bukrs type ekko-bukrs,*  
*\*      lv\_stako type ekko-stako.*  
*\**  
*\*Select ebeln kufix unsez bukrs stako*  
*\*  into ( lv\_ebeln, lv\_kufix, lv\_unsez, lv\_bukrs, lv\_stako )*  
*\*  from ekko where ebeln = '4500000011'.*  
*\**  
*\*WRITE : / lv\_ebeln, lv\_kufix, lv\_unsez, lv\_bukrs, lv\_stako.*  
*\*ENDSELECT.*  
  
  
*\*TABLES ekko.*  
*\**  
*\*DATA : lv\_ebeln type ekko-ebeln,*  
*\*      lv\_kufix type ekko-kufix,*  
*\*      lv\_unsez type ekko-unsez,*  
*\*      lv\_bukrs type ekko-bukrs,*  
*\*      lv\_stako type ekko-stako.*  
*\**  
*\*PARAMETERS p\_ebeln type ekko-ebeln.*  
*\**  
*\*Select ebeln kufix unsez bukrs stako*  
*\*  into ( lv\_ebeln, lv\_kufix, lv\_unsez, lv\_bukrs, lv\_stako )*  
*\*  from ekko where ebeln = p\_ebeln.*  
*\**  
*\*WRITE : / lv\_ebeln, lv\_kufix, lv\_unsez, lv\_bukrs, lv\_stako.*  
*\*ENDSELECT.*

=========================================================================

**Control statements**

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_CNTRLSTMTS*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_CNTRLSTMTS.  
  
*\*DO 5 times.*  
*\*  WRITE 'SAP'.*  
*\*ENDDO.*  
*\**  
*\*DATA a type i value 4.*  
*\*while a < 10.*  
*\*  WRITE / a.*  
*\*  a = a + 1.*  
*\*ENDWHILE.*  
*\**  
*\*\*DATA x TYPE i VALUE 4.*  
*\*\*if x eq 4.*  
*\*\*  WRITE / 'x is 4'.*  
*\*\*ENDIF.*  
*\**  
*\*DATA x type i value 10.*  
*\*if x eq 4.*  
*\*  Write ' x is 4 '.*  
*\*ELSEIF x eq 10.*  
*\*  WRITE ' x is 10 '.*  
*\*ENDIF.*  
*\**  
*\*DATA: k(20) type c value 'ABAP'.*  
*\*PARAMETERS p\_k(20).*  
*\*CASE p\_k.*  
*\*  When 'SAP'.*  
*\*    WRITE ' not sap '.*  
*\*  When 'ERP'.*  
*\*    WRITE ' not ERP '.*  
*\*  When Others.*  
*\*    WRITE ' IT is a ABAP '.*  
*\*ENDCASE.*  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*LOOP TERMINATING STMTS\*\*\*\*\*\*\**  
WRITE / 'using continue'.  
DO 5 times.  
  if sy-index = 3.  
    CONTINUE.  
  ENDIF.  
  write /  sy-index.  
ENDDO.  
  
WRITE / 'using check'.  
DO 5 times.  
CHECK sy-index BETWEEN 3 and 4.  
write / sy-index.  
ENDDO.  
  
WRITE / 'using exit'.  
DO 5 times.  
  if sy-index = 3.  
    EXIT.  
  ENDIF.  
  write / sy-index.  
ENDDO.

=====================================================================

Formatting:

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_FORMAT*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_FORMAT.  
  
TABLES lfa1.  
TYPES: BEGIN OF ls\_tab,               *""""local structures*  
      lifnr type lifnr,  
      land1 type land1,  
      ort01 type ort01,  
      name1 type name1\_gp,  
      end of ls\_tab.  
  
DATA: LT\_tab type STANDARD TABLE OF ls\_tab,  
      wa\_tab type ls\_tab,  
      lv\_lifnr TYPE lifnr.  
  
SELECT-OPTIONS s\_lifnr FOR lv\_lifnr.  
  
PARAMETERS p\_land1 TYPE lfa1-land1.  
  
START-OF-SELECTION.  
select lifnr land1 ort01 name1  
  INTO TABLE LT\_Tab FROM lfa1  
   WHERE lifnr IN s\_lifnr  
  AND land1 EQ p\_land1.  
  
IF sy-subrc eq 0.  
  write:/ 'records are found in lfa1', sy-dbcnt.  
ELSE.  
  WRITE:/ 'records are not found', sy-dbcnt.  
ENDIF.  
  
SKIP 2.  
  
FORMAT COLOR COL\_NEGATIVE.  
loop at lt\_Tab into wa\_tab.  
  WRITE :/ 'row no',  
  10 wa\_tab-lifnr,  
  30 wa\_tab-land1,  
  50 wa\_tab-ort01,  
  65 wa\_tab-name1.  
  
ENDLOOP.  
  
FORMAT COLOR Off.  
  
*\*BAck.*  
*\*SKIP 2.*  
*\*write /120 'ust hyd'.*  
*\*back.*  
*\*skip 4.*  
*\*write /130 'ust usa'.*

======================================================================

**Joins**

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_JOINS*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_JOINS.  
  
TYPES : BEGIN OF ls\_tab,  *""""EKKO*  
        ebeln type ebeln,  
        pincr type pincr,  
        kufix type kufix,  
        ausnr type ausch,  
  
        abueb type abueb, *""""""EKPO*  
        fabkz type fabkz,  
        matkl type matkl,  
  
    end of ls\_tab.  
DATA: lt\_tab type STANDARD TABLE OF ls\_tab,  
      wa\_tab type ls\_tab.  
  
SELECT-OPTIONS s\_ebeln for wa\_tab-ebeln.  
  
START-OF-SELECTION.  
  
SELECT ekko~ebeln  
        ekko~pincr  
        ekko~kufix  
        ekko~ausnr  
  
        ekpo~abueb  
        ekpo~fabkz  
        ekpo~matkl  
  INTO TABLE lt\_Tab from ekko  
  inner join ekpo on ekko~ebeln = ekpo~ebeln  
  where ekko~ebeln IN s\_ebeln.  
  
  IF sy-subrc eq 0.  
    WRITE : ' PURCHASING ORDER INFO ', sy-dbcnt.  
  else.  
    WRITE: ' PURCHASING ORDER RECORDS ARE NOT FOUND ', sy-dbcnt.  
  ENDIF.  
  
end-of-SELECTION.  
  
  loop at lt\_tab into wa\_tab.  
  
    Write:/ wa\_tab-ebeln, wa\_tab-pincr, wa\_tab-kufix,  
        wa\_tab-ausnr,  wa\_tab-abueb,wa\_tab-matkl, wa\_tab-fabkz.  
  
  ENDLOOP.

=========================================================================

FOR all Entries:

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_FORALL\_ENTRIES*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_FORALL\_ENTRIES NO STANDARD PAGE HEADING LINE-COUNT 30(2) LINE-SIZE 400.  
  
INCLUDE ZPRAC\_FORALLENTRIES\_TOP.  
  
TABLES ekko.  
  
select-OPTIONS s\_ebeln for ekko-ebeln.  
  
START-OF-SELECTION.  
select ebeln pincr kufix ausnr into table lt\_tab1 from ekko  
  where ebeln in s\_ebeln.  
  
if not lt\_tab1[] is INITIAL.                *"""""""""""""# 1*  
  select ebeln bednr insmk matkl into table lt\_tab2 from ekpo  
    FOR ALL ENTRIES IN lt\_tab1 where ebeln eq lt\_tab1-ebeln.    *"""""""""'# 2*  
  
DELETE ADJACENT DUPLICATES FROM lt\_tab1 COMPARING ebeln.        *""""""""""""#3*  
  
sort lt\_tab2 by ebeln.            *""""""""""#4*  
  
loop at lt\_tab1 into wa\_tab1.  
  
READ TABLE lt\_tab2 into wa\_tab2 with key ebeln = wa\_tab1-ebeln BINARY SEARCH.   *"""""""""""#5*  
  
if sy-subrc = 0.  
  
MOVE: wa\_tab1-ebeln to wa\_final-ebeln,  
      wa\_tab1-pincr to wa\_final-pincr,  
      wa\_tab1-kufix to wa\_final-kufix,  
      wa\_tab1-ausnr to wa\_final-ausnr,  
  
      wa\_tab2-bednr to wa\_final-bednr,  
      wa\_tab2-insmk to wa\_final-insmk,  
      wa\_tab2-matkl to wa\_final-matkl.  
append wa\_final to lt\_final.  
  
else.  
  continue.  
ENDIF.  
ENDLOOP.  
  
LOOP AT lt\_final into wa\_final.  
  WRITE:/ wa\_final-ebeln,  
          wa\_final-pincr,  
          wa\_final-kufix,  
          wa\_final-ausnr,  
  
          wa\_final-bednr,  
          wa\_final-insmk,  
          wa\_final-matkl.  
ENDLOOP.  
else.  
Write 'Records are NOT FOUND'.  
  
ENDIF.

----------------------------------------------------------

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_FORALL\_JOINS\_3TABS*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_FORALL\_JOINS\_3TABS NO STANDARD PAGE HEADING LINE-COUNT 30(5) LINE-SIZE 600.  
  
INCLUDE ZPRAC\_FORALL\_3TABS\_TOP.  
  
TABLES ekko.  
  
select-OPTIONS s\_ebeln for ekko-ebeln.  
  
START-OF-SELECTION.  
select ekko~ebeln  
       ekko~pincr  
       ekko~kufix  
       ekko~ausnr  
  
       ekpo~bednr  
       ekpo~insmk  
       ekpo~matkl into table lt\_tab1 from ekko inner JOIN ekpo  
       on ekko~ebeln = ekpo~ebeln where ekko~ebeln in s\_ebeln.  
  
if not lt\_tab1[] is INITIAL.                *"""""""""""""# 1*  
  select ebeln ameng wemng wamng  into table lt\_tab2 from eket  
    FOR ALL ENTRIES IN lt\_tab1 where ebeln eq lt\_tab1-ebeln.    *"""""""""'# 2*  
  
DELETE ADJACENT DUPLICATES FROM lt\_tab1 COMPARING ebeln.        *""""""""""""#3*  
  
sort lt\_tab2 by ebeln.            *""""""""""#4*  
  
loop at lt\_tab1 into wa\_tab1.  
  
READ TABLE lt\_tab2 into wa\_tab2 with key ebeln = wa\_tab1-ebeln BINARY SEARCH.   *"""""""""""#5*  
  
if sy-subrc = 0.  
  
MOVE: wa\_tab1-ebeln to wa\_final-ebeln,  
      wa\_tab1-pincr to wa\_final-pincr,  
      wa\_tab1-kufix to wa\_final-kufix,  
      wa\_tab1-ausnr to wa\_final-ausnr,  
      wa\_tab1-bednr to wa\_final-bednr,  
      wa\_tab1-insmk to wa\_final-insmk,  
      wa\_tab1-matkl to wa\_final-matkl,  
  
      wa\_tab2-ebeln to wa\_final-ebeln,  
      wa\_tab2-ameng to wa\_final-ameng,  
      wa\_tab2-wemng to wa\_final-wemng,  
      wa\_tab2-wamng to wa\_final-wamng.  
  
append wa\_final to lt\_final.  
  
else.  
  continue.  
ENDIF.  
ENDLOOP.  
  
LOOP AT lt\_final into wa\_final.  
  WRITE:/ wa\_final-ebeln,  
          wa\_final-pincr,  
          wa\_final-kufix,  
          wa\_final-ausnr,  
  
          wa\_final-bednr,  
          wa\_final-insmk,  
          wa\_final-matkl,  
  
          wa\_final-ameng,  
          wa\_final-wemng,  
          wa\_final-wamng.  
ENDLOOP.  
else.  
Write 'Records are NOT FOUND'.  
  
ENDIF.

INCLUDE:

*\*&---------------------------------------------------------------------\**  
*\*& Include          ZPRAC\_FORALL\_3TABS\_TOP*  
*\*&---------------------------------------------------------------------\**  
  
TYPES: BEGIN OF ls\_tab1,  
         ebeln TYPE ebeln,     *"purchasing doc no.*  
         pincr TYPE pincr,     *"item no. interval*  
         kufix TYPE kufix,     *"fixed exchange rate indicator*  
         ausnr TYPE ausch,     *"bid invitation no.*  
  
         bednr TYPE bednr,      *"req tracking no.*  
         insmk TYPE insmk,        *"stock type*  
         matkl TYPE matkl,        *"material group*  
       END OF ls\_tab1.  
DATA: lt\_tab1 TYPE STANDARD TABLE OF ls\_tab1,  
      wa\_tab1 TYPE ls\_tab1.  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*EKET*  
TYPES: BEGIN OF ls\_tab2,  
         ebeln TYPE ebeln,         *"purchasing doc no.*  
         ameng TYPE  vomng,        *"Previous quan*  
         wemng TYPE  weemg,          *"recieved goods*  
         wamng TYPE  wamng ,         *"issued quan*  
       END OF ls\_tab2.  
DATA: lt\_tab2 TYPE STANDARD TABLE OF ls\_tab2,  
      wa\_tab2 TYPE ls\_tab2.  
  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*final*  
TYPES: BEGIN OF ls\_final,  
         ebeln TYPE ebeln,     *"purchasing doc no.*  
         pincr TYPE pincr,     *"item no. interval*  
         kufix TYPE kufix,     *"fixed exchange rate indicator*  
         ausnr TYPE ausch,     *"bid invitation no.*  
  
         bednr TYPE bednr,      *"req tracking no.*  
         insmk TYPE insmk,        *"stock type*  
         matkl TYPE matkl,        *"material group*  
  
         ameng TYPE  vomng,        *"Previous quan*  
         wemng TYPE  weemg,          *"recieved goods*  
         wamng TYPE  wamng ,         *"issued quan*  
       END OF ls\_final.  
  
DATA: lt\_final TYPE STANDARD TABLE OF ls\_final,  
      wa\_final TYPE ls\_final.

----------------------------------------------------------------------

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_FORALL\_JOINS\_3TABS2*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT zprac\_forall\_joins\_3tabs2 NO STANDARD PAGE HEADING.  
  
TABLES: ZGLPCT, ZGLPCO, ZGLPCC.  
  
  INCLUDE zprac\_forall\_joins\_top.  
  
  SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME TITLE TEXT-001.  
    SELECT-OPTIONS :s\_rldnr FOR lv\_rldnr.  
  SELECTION-SCREEN END OF BLOCK b1.  
  
  SELECTION-SCREEN BEGIN OF BLOCK b2 WITH FRAME TITLE TEXT-002.  
    SELECT-OPTIONS: s\_bukrs FOR lv\_bukrs.  
  SELECTION-SCREEN END OF BLOCK b2.  
  
  SELECTION-SCREEN BEGIN OF BLOCK b3 WITH FRAME TITLE TEXT-003.  
    SELECT-OPTIONS:s\_kokrs FOR lv\_kokrs.  
  SELECTION-SCREEN END OF BLOCK b3.  
  
  SELECTION-SCREEN BEGIN OF BLOCK b4 WITH FRAME TITLE TEXT-004.  
    SELECT-OPTIONS:s\_racct FOR lv\_racct.  
  SELECTION-SCREEN END OF BLOCK b4.  
  
INITIALIZATION.  
  CLEAR: wa\_tab1,  
  wa\_tab2,  
  wa\_final,  
  lt\_tab1,  
  lt\_tab2,  
  lt\_final.  
  
START-OF-SELECTION.  
  SELECT zglpct~rldnr  
         zglpct~rrcty  
    zglpct~rvers  
    zglpct~ryear  
    zglpct~robjnr  
    zglpct~cobjnr  
    zglpct~sobjnr  
    zglpct~rtcur  
  
    zglpco~bukrs  
    zglpco~hoart  
    INTO CORRESPONDING FIELDS OF TABLE lt\_tab1  
    FROM zglpct INNER JOIN zglpco ON zglpct~robjnr = zglpco~objnr  
    WHERE rldnr IN s\_RLDNR AND bukrs IN s\_bukrs.  
  
  IF lt\_tab1[] IS NOT INITIAL.                *"""""""""""""# 1*  
    SELECT objnr racct activ kokrs INTO CORRESPONDING FIELDS OF TABLE lt\_tab2 FROM zglpcc  
       FOR ALL ENTRIES IN lt\_tab1 WHERE objnr EQ lt\_tab1-cobjnr AND  
      racct IN s\_racct AND kokrs IN s\_kokrs.    *"""""""""'# 2*  
  
    DELETE ADJACENT DUPLICATES FROM lt\_tab1 COMPARING robjnr.  
  
    SORT lt\_tab2 BY objnr.  
  
    LOOP AT lt\_tab1 INTO wa\_tab1.  
  
      READ TABLE lt\_tab2 INTO wa\_tab2 WITH KEY objnr = wa\_tab1-cobjnr BINARY SEARCH.  
  
      IF sy-subrc = 0.  
  
        MOVE: wa\_tab1-rldnr TO wa\_final-rldnr,  
              wa\_tab1-rrcty TO wa\_final-rrcty,  
              wa\_tab1-ryear TO wa\_final-ryear,  
              wa\_tab1-rvers TO wa\_final-rvers,  
              wa\_tab1-robjnr TO wa\_final-robjnr,  
              wa\_tab1-cobjnr TO wa\_final-cobjnr,  
              wa\_tab1-sobjnr TO wa\_final-sobjnr,  
              wa\_tab1-rtcur TO wa\_final-rtcur,  
  
  
              wa\_tab1-bukrs TO wa\_final-bukrs,  
              wa\_tab1-hoart TO wa\_final-hoart,  
  
  
        wa\_tab2-racct TO wa\_final-racct,  
        wa\_tab2-activ TO wa\_final-activ,  
        wa\_tab2-kokrs TO wa\_final-kokrs.  
        APPEND wa\_final TO lt\_final.  
  
      ELSE.  
        CONTINUE.  
      ENDIF.  
    ENDLOOP.  
  
    LOOP AT lt\_final INTO wa\_final.  
      WRITE:/ wa\_final-rldnr,  
              wa\_final-rrcty,  
              wa\_final-ryear,  
              wa\_final-rvers,  
              wa\_final-robjnr,  
              wa\_final-cobjnr,  
              wa\_final-sobjnr,  
              wa\_final-rtcur,  
  
              wa\_final-bukrs,  
              wa\_final-hoart,  
  
            wa\_final-racct,  
            wa\_final-activ,  
            wa\_final-kokrs.  
    ENDLOOP.  
  ELSE.  
    WRITE 'Records are NOT FOUND'.  
  ENDIF.  
  
*\*\*\*  ENDIF.*  
*\*\**  
*\*\*\*END-OF-SELECTION.*  
*\*\*\**  
*\*\*\*  LOOP AT lt\_tab1 INTO wa\_tab1.*  
*\*\*\*    WRITE:/ wa\_tab1-rldnr,*  
*\*\*\*                wa\_tab1-rrcty,*  
*\*\*\*                wa\_tab1-ryear,*  
*\*\*\*                wa\_tab1-robjnr,*  
*\*\*\*                wa\_tab1-cobjnr,*  
*\*\*\*                wa\_tab1-sobjnr,*  
*\*\*\*                wa\_tab1-rtcur,*  
*\*\*\**  
*\*\*\*                wa\_tab1-robjnr,*  
*\*\*\*                wa\_tab1-bukrs,*  
*\*\*\*                wa\_tab1-hoart.*  
*\*\*\*  ENDLOOP.*

*INCLUDE:*

*\*&---------------------------------------------------------------------\**  
*\*& Include          ZPRAC\_FORALL\_JOINS\_TOP*  
*\*&---------------------------------------------------------------------\**  
  
  
TYPES: BEGIN OF ls\_tab,                                 *"""structure for ZGLPCT and ZGLPCO tables*  
         rldnr  TYPE  zg08\_rldnr,  
         rrcty  TYPE  zg08\_rrcty,  
         rvers  TYPE zg08\_rvers,  
         ryear  TYPE  zg08\_ryear,  
         robjnr TYPE  zg08\_objnr,  
         cobjnr TYPE  zg08\_cobjnr,  
         sobjnr TYPE  zg08\_objnr,  
         rtcur  TYPE  zg08\_rtcur,  
  
         objnr  TYPE  zg08\_objnr,  
         bukrs  TYPE  zg08\_bukrs,  
         hoart  TYPE  zg08\_hoart,  
  
       END OF ls\_tab.  
  
DATA: wa\_tab1 TYPE ls\_tab,  
      lt\_tab1 TYPE STANDARD TABLE OF ls\_tab.  
  
TYPES: BEGIN OF ls\_ZGLPCC,  
         objnr TYPE  zg08\_cobjnr,  
         racct TYPE  zg08\_racct,  
         activ TYPE  zg08\_activ,  
         kokrs TYPE  zg08\_kokrs,  
       END OF ls\_ZGLPCC.  
  
DATA: wa\_tab2 TYPE zglpcc,  
      lt\_tab2 TYPE STANDARD TABLE OF zglpcc.  
  
TYPES: BEGIN OF ls\_FINAL,  
         rldnr  TYPE  zg08\_rldnr,  
         rrcty  TYPE  zg08\_rrcty,  
         rvers  TYPE zg08\_rvers,  
         ryear  TYPE  zg08\_ryear,  
         robjnr TYPE  zg08\_objnr,  
         cobjnr TYPE  zg08\_cobjnr,  
         sobjnr TYPE  zg08\_objnr,  
         rtcur  TYPE  zg08\_rtcur,  
  
         bukrs  TYPE  zg08\_bukrs,  
         hoart  TYPE  zg08\_hoart,  
  
         racct  TYPE  zg08\_racct,  
         activ  TYPE  zg08\_activ,  
         kokrs  TYPE  zg08\_kokrs,  
  
       END OF ls\_FINAL.  
  
DATA: wa\_final TYPE ls\_FINAL,  
      lt\_final TYPE TABLE OF ls\_FINAL.  
  
  
  DATA : lv\_rldnr TYPE zg08\_rldnr,  
         lv\_bukrs TYPE zg08\_bukrs,  
         lv\_kokrs TYPE zg08\_kokrs,  
         lv\_racct TYPE zg08\_racct.

==========================================================================

Subroutines:

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_SUBROUTINES*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT zprac\_subroutines.  
  
  
*\*\*\*\*\*Call by value(insert value otherwise it takes reference by default)*  
  
*\*DATA x TYPE i VALUE 20.         """o/p - 20*  
*\*PERFORM sub USING x.*  
*\*\*&---------------------------------------------------------------------\**  
*\*\*& Form SUB*  
*\*\*&---------------------------------------------------------------------\**  
*\*\*& text*  
*\*\*&---------------------------------------------------------------------\**  
*\*\*&      --> X*  
*\*\*&---------------------------------------------------------------------\**  
*\*WRITE x.*  
*\*FORM sub  USING VALUE(p\_x).*  
*\**  
*\*  p\_x = 100.*  
*\*ENDFORM.*  
  
  
*\*\*\*\*\*\*\*\*\*call by reference(default)*  
*\**  
*\*DATA x TYPE i VALUE 20.             """O/P - 100*  
*\**  
*\*PERFORM sub USING x.*  
*\*\*&---------------------------------------------------------------------\**  
*\*\*& Form SUB*  
*\*\*&---------------------------------------------------------------------\**  
*\*\*& text*  
*\*\*&---------------------------------------------------------------------\**  
*\*\*&      --> X*  
*\*\*&---------------------------------------------------------------------\**  
*\*WRITE x.*  
*\*FORM sub  USING p\_x.*  
*\**  
*\*  p\_x = 100.*  
*\**  
*\*ENDFORM.*  
  
*\*DATA x TYPE i VALUE 100.              """"o/p - 600*  
*\**  
*\*PERFORM sub CHANGING x.*  
*\*\*&---------------------------------------------------------------------\**  
*\*\*& Form SUB*  
*\*\*&---------------------------------------------------------------------\**  
*\*\*& text*  
*\*\*&---------------------------------------------------------------------\**  
*\*\*&      --> X*  
*\*\*&---------------------------------------------------------------------\**  
*\*WRITE x.*  
*\*FORM sub  CHANGING p\_x.*  
*\**  
*\*  p\_x = 600.*  
*\**  
*\*ENDFORM.*  
  
*\*\*\*\*\*\*call by value and return*  
DATA x TYPE i VALUE 100.               *""o/p - 1000*  
  
PERFORM sub CHANGING x.  
*\*&---------------------------------------------------------------------\**  
*\*& Form SUB*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*&      --> X*  
*\*&---------------------------------------------------------------------\**  
WRITE x.  
FORM sub  CHANGING VALUE(p\_x).  
  
  p\_x = 1000.  
  
ENDFORM.

========================================================================

Lock objects:

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_LOCKOBJE*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_LOCKOBJE.  
  
PARAMETERS p\_matnr type matnr.  
CALL FUNCTION 'ENQUEUE\_EZPRAC\_LOCKOBJ'  
EXPORTING  
MODE\_MARA            = 'E'  
MANDT                = SY-MANDT  
MATNR                = 'p\_matnr' *"ust\_381*  
*\*   X\_MATNR              = ' '*  
*\*   \_SCOPE               = '2'*  
*\*   \_WAIT                = ' '*  
*\*   \_COLLECT             = ' '*  
*\* EXCEPTIONS*  
*\*   FOREIGN\_LOCK         = 1*  
*\*   SYSTEM\_FAILURE       = 2*  
*\*   OTHERS               = 3*  
          .  
IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
ENDIF.  
  
BREAK-POINT.  
  
CALL FUNCTION 'DEQUEUE\_EZPRAC\_LOCKOBJ'  
EXPORTING  
 MODE\_MARA       = 'E'  
   MANDT           = SY-MANDT  
   MATNR           = 'p\_matnr' *"ust\_381*  
*\*   X\_MATNR         = ' '*  
*\*   \_SCOPE          = '3'*  
*\*   \_SYNCHRON       = ' '*  
*\*   \_COLLECT        = ' '*  
          .

===========================================================================

Internal tables

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_INTERNALTABS*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT zprac\_internaltabs.  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*NORMAL SELECT STAT\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\*TABLES lfa1.*  
*\*DATA : lv\_lifnr type lfa1-lifnr,*  
*\*      lv\_land1 type lfa1-land1,*  
*\*      lv\_ort01 type lfa1-ort01,*  
*\*      lv\_name1 type lfa1-name1,*  
*\*      lv\_pfach type lfa1-pfach,*  
*\*      lv\_regio type lfa1-regio.*  
*\**  
*\*SELECT-OPTIONS s\_lifnr for lv\_lifnr.*  
*\*PARAMETERS p\_land1 TYPE lfa1-land1.*  
*\**  
*\*Select mandt land1 ort01 name1 pfach regio*  
*\*  into ( lv\_lifnr, lv\_land1, lv\_ort01, lv\_name1, lv\_pfach, lv\_regio )*  
*\*  from lfa1 where land1 = p\_land1.*  
*\**  
*\*WRITE : / lv\_lifnr, lv\_land1, lv\_ort01, lv\_name1, lv\_pfach, lv\_regio.*  
*\*ENDSELECT.*  
  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ITABS WITH HEADER\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
  
*\*TABLES lfa1.*  
*\*DATA: lt\_tab TYPE lfa1 OCCURS 0 WITH HEADER LINE.*  
*\*DATA lv\_lifnr TYPE lifnr.*  
*\*SELECT-OPTIONS s\_lifnr FOR lv\_lifnr.*  
*\*PARAMETERS p\_land1 TYPE lfa1-land1.*  
*\*SELECT \* INTO lt\_Tab FROM lfa1 WHERE lifnr IN s\_lifnr AND land1 EQ p\_land1.*  
*\**  
*\*APPEND lt\_tab to lt\_tab.*  
*\*ENDSELECT.*  
*\**  
*\*loop at lt\_Tab into lt\_tab.*  
*\*  WRITE:/ lt\_tab-lifnr, lt\_tab-land1, lt\_tab-ort01,*  
*\*          lt\_tab-name1,lt\_tab-pfach, lt\_tab-regio.*  
*\*ENDLOOP.*  
  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ITABS WITHOUT HEADER\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\*TABLES lfa1.*  
*\**  
*\*DATA: lt\_tab TYPE lfa1 OCCURS 0 WITH HEADER LINE.*  
*\*DATA lv\_lifnr TYPE lifnr.*  
*\*SELECT-OPTIONS s\_lifnr FOR lv\_lifnr.*  
*\*PARAMETERS p\_land1 TYPE lfa1-land1.*  
*\**  
*\*select \**  
*\*  INTO TABLE lt\_Tab FROM lfa1*  
*\*   WHERE lifnr IN s\_lifnr*  
*\*  AND land1 EQ p\_land1.*  
*\**  
*\*loop at lt\_Tab into lt\_tab.*  
*\*  WRITE:/ lt\_tab-lifnr, lt\_tab-land1, lt\_tab-ort01,*  
*\*          lt\_tab-name1,lt\_tab-pfach, lt\_tab-regio.*  
*\*ENDLOOP.*  
  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ITABS with work area\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\*TABLES lfa1.*  
*\**  
*\*TYPES: BEGIN OF ls\_tab,               """"local structures*  
*\*      lifnr type lifnr,*  
*\*      land1 type land1,*  
*\*      ort01 type ort01,*  
*\*      name1 type name1\_gp,*  
*\*      pfach type pfach,*  
*\*      regio type regio,*  
*\*      end of ls\_tab.*  
*\**  
*\*DATA: lt\_tab type STANDARD TABLE OF ls\_tab,*  
*\*      wa\_tab type ls\_tab.*  
*\*DATA lv\_lifnr TYPE lifnr.*  
*\*SELECT-OPTIONS s\_lifnr FOR lv\_lifnr.*  
*\*PARAMETERS p\_land1 TYPE lfa1-land1.*  
*\**  
*\*select lifnr land1 ort01 name1 pfach regio*  
*\*  INTO TABLE lt\_Tab FROM lfa1*  
*\*   WHERE lifnr IN s\_lifnr*  
*\*  AND land1 EQ p\_land1.*  
*\**  
*\*loop at lt\_Tab into lt\_tab.*  
*\*  WRITE :/ lt\_tab-lifnr, lt\_tab-land1, lt\_tab-ort01,*  
*\*          lt\_tab-name1,lt\_tab-pfach, lt\_tab-regio.*  
*\*ENDLOOP.*  
  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*KNA1\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\*TABLES kna1.*  
*\*TYPES: BEGIN OF ls\_tab,*  
*\*         kunnr TYPE kunnr, " V TYPE DTL.*  
*\*         name1 TYPE name1\_gp,*  
*\*         ort01 TYPE ort01\_gp,*  
*\*         land1 TYPE land1\_gp,*  
*\**  
*\*       END OF ls\_tab.*  
*\*DATA : lt\_tab type STANDARD TABLE OF ls\_tab,*  
*\* wa\_tab type ls\_tab.*  
*\**  
*\*DATA lv\_kunnr TYPE kunnr.*  
*\*SELECT-OPTIONS s\_kunnr FOR lv\_kunnr.*  
*\*PARAMETERS p\_land1 TYPE kna1-land1.*  
*\**  
*\*SELECT kunnr name1 ort01 land1*  
*\**  
*\*  INTO TABLE LT\_tAB*  
*\*        FROM kna1*  
*\*          WHERE Kunnr IN s\_kunnr AND*  
*\*                land1 EQ p\_land1.*  
*\**  
*\*LOOP AT lt\_Tab INTO WA\_Tab.*  
*\**  
*\*  WRITE: / WA\_tab-kunnr,*  
*\*           WA\_tab-name1,*  
*\*           WA\_tab-ort01,*  
*\*           WA\_tab-land1.*  
*\*ENDLOOP.*  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*FINAL CODE\*\*\*\*\*\*\*\*\*\*\*\**  
TABLES lfa1.  
TYPES: BEGIN OF ls\_tab,               *""""local structures*  
      lifnr type lifnr,  
      land1 type land1,  
      ort01 type ort01,  
      name1 type name1\_gp,  
      end of ls\_tab.  
DATA: LT\_tab type STANDARD TABLE OF ls\_tab,  
      wa\_tab type ls\_tab.  
DATA lv\_lifnr TYPE lifnr.  
SELECT-OPTIONS s\_lifnr FOR lv\_lifnr.  
PARAMETERS p\_land1 TYPE lfa1-land1.  
  
select lifnr land1 ort01 name1  
  INTO TABLE LT\_Tab FROM lfa1  
   WHERE lifnr IN s\_lifnr  
  AND land1 EQ p\_land1.  
  
loop at lt\_Tab into wa\_tab.  
  WRITE :/ wa\_tab-lifnr, wa\_tab-land1, wa\_tab-ort01,  
          wa\_tab-name1.  
ENDLOOP.

===========================================================================

Internal table operations:

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_INTERNALTABS\_OPS*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_INTERNALTABS\_OPS.  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*clear&refresh*  
*\*DATA: BEGIN of datas,   ""field string*  
*\*  col1(20) type c,*  
*\*  col2(20) type c,*  
*\*  end of datas.*  
*\**  
*\*  DATA itab like TABLE OF datas.*  
*\*  datas-col1 = 'SAP'.*  
*\*  datas-col2 = 'ABAP'.*  
*\*  APPEND datas to itab.*  
*\*  clear datas.*  
*\**  
*\*refresh itab.*  
*\*  IF itab is INITIAL.*  
*\*    write ' itab is empty '.*  
*\*   free itab.*  
*\*  ENDIF.*  
  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*inserting with 'insert'*  
*\*  DATA : BEGIN OF datas,*  
*\*    name(20) type c,*  
*\*    gender(8) type c,*  
*\*    age(5) type i,*  
*\*    end of datas.*  
*\**  
*\*  DATA itab like SORTED TABLE OF datas WITH NON-UNIQUE KEY name gender age.*  
*\*\*DATA itab like Standard TABLE OF datas WITH NON-UNIQUE KEY name gender age.*  
*\*    datas-name = 'SAP'.*  
*\*    datas-gender = 'Male'.*  
*\*    datas-age = 99.*  
*\*    Insert datas into table itab.*  
*\**  
*\*    datas-name = 'ABAP'.*  
*\*    datas-gender = 'Female'.*  
*\*    datas-age = 98.*  
*\*    Insert datas into table itab.*  
*\**  
*\*    datas-name = 'PROGRAMMING'.*  
*\*    datas-gender = 'Male'.*  
*\*    datas-age = 99.*  
*\*    Insert datas into table itab.*  
*\**  
*\*\*sort itab by name.*  
*\**  
*\*    loop at itab into datas.*  
*\*      write:/ datas-name, datas-age , datas-gender.*  
*\*    ENDLOOP.*  
  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*index*  
*\*DATA: BEGIN of datas,   ""field string*  
*\*  col1(20) type i,*  
*\*  col2(20) type i,*  
*\*  end of datas.*  
*\**  
*\*data itab like TABLE OF datas.*  
*\**  
*\*do 2 times.*  
*\*  datas-col1 = sy-index.*  
*\*  datas-col2 = sy-index \*\* 2.*  
*\*  append datas to itab.*  
*\*ENDDO.*  
*\**  
*\*loop at itab into datas.*  
*\*  datas-col1 = 3 \* sy-tabix.*  
*\*  datas-col2 = 5 \* sy-tabix.*  
*\*  insert datas into itab.*  
*\*endloop.*  
*\**  
*\**  
*\*loop at itab into datas.*  
*\*  write:/ datas-col1, datas-col2.*  
*\*endloop.*  
  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*MODIFY*  
*\*DATA: BEGIN of datas,   ""field string*  
*\*  col1(20) type i,*  
*\*  col2(20) type i,*  
*\*  end of datas.*  
*\**  
*\*data itab like TABLE OF datas.*  
*\**  
*\*do 2 times.*  
*\*  datas-col1 = sy-index.*  
*\*  datas-col2 = sy-index \*\* 2.*  
*\*  append datas to itab.*  
*\*ENDDO.*  
*\**  
*\*loop at itab into datas.*  
*\*  if sy-tabix = 2.*  
*\*  datas-col1 = 3 \* ( sy-tabix \* 2 ).*  
*\*  datas-col2 = 5 \* ( sy-tabix \* 3 ).*  
*\*  modify itab from datas.*  
*\*  ENDIF.*  
*\*endloop.*  
*\**  
*\**  
*\*loop at itab into datas.*  
*\*  write:/ datas-col1, datas-col2, sy-tabix.*  
*\*endloop.*  
  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DELETE*  
*\*DATA: BEGIN of datas,   ""field string*  
*\*  col1(20) type i,*  
*\*  col2(20) type i,*  
*\*  end of datas.*  
*\**  
*\*data itab like HASHED TABLE OF datas with UNIQUE key col1.*  
*\**  
*\*do 2 times.*  
*\*  datas-col1 = sy-index.*  
*\*  datas-col2 = sy-index \*\* 2.*  
*\*  insert datas into table itab.*  
*\*ENDDO.*  
*\**  
*\*datas-col1 = 1.*  
*\**  
*\*DELETE TABLE itab : from datas,*  
*\*WITH TABLE KEY col1 = 3.*  
*\**  
*\**  
*\*loop at itab into datas.*  
*\*  write:/ datas-col1, datas-col2, sy-tabix.*  
*\*endloop.*  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*COLLECT*  
DATA: BEGIN of datas,   *""field string*  
  col1(20) type i,  
  col2(20) type i,  
  end of datas.  
  
data itab like HASHED TABLE OF datas with UNIQUE key col1.  
  
do 2 times.  
  datas-col1 = sy-index.  
  datas-col2 = sy-index \*\* 2.  
  insert datas into table itab.  
ENDDO.  
  
datas-col1 = 1.  
  
DELETE TABLE itab : from datas,  
WITH TABLE KEY col1 = 3.  
  
  
loop at itab into datas.  
  write:/ datas-col1, datas-col2, sy-tabix.  
endloop.

=================================================================

DATABASE VIEW PROG,

*\*&---------------------------------------------------------------------\**  
*\*& Report Z0381\_DBVIEW\_DISP*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT z0381\_dbview\_disp.  
  
TABLES z0381\_dbview.  
  
DATA : lv\_vbeln TYPE z0381\_DBVIEW-vbeln,  
       lv\_fkart TYPE z0381\_dbview-fkart,  
       lv\_fktyp TYPE z0381\_dbview-fktyp,  
       lv\_fkimg TYPE Z0381\_dbview-fkimg,  
       lv\_vrkme TYPE z0381\_dbview-vrkme.  
  
PARAMETERS p\_vbeln TYPE z0381\_DBVIEW-vbeln.  
  
SELECT  
vbeln  
fkart  
fktyp  
fkimg  
vrkme  
  INTO ( lv\_vbeln, lv\_fkart, lv\_fktyp, lv\_fkimg, lv\_vrkme )  
  FROM z0381\_dbview WHERE vbeln = p\_vbeln.  
  
  WRITE : / lv\_vbeln, lv\_fkart, lv\_fktyp, lv\_fkimg, lv\_vrkme.  
ENDSELECT.

===========================================================================

Function modules,

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_FUNCMODS*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_FUNCMODS.  
  
PARAMETERS: p\_a type I,  
      p\_b type I.  
DATA lv\_add type i.  
CALL FUNCTION 'ZPRAC\_FUNCMOD'  
 EXPORTING  
   I\_X           = p\_a  
   I\_Y           = p\_b  
 IMPORTING  
   I\_ADD         = lv\_add  
  
   .  
*\*lv\_add = p\_a + p\_b.*  
  
WRITE lv\_add.

==============================================

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_FUNCMODSSTR*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_FUNCMODSSTR.  
  
DATA : lt\_tab TYPE ZPRAC\_FUNCSTRUCT1,  
      wa\_tab TYPE ZPRAC\_FUNCSTRUCT.  
  
  
SELECT-OPTIONS s\_lifnr for wa\_tab-lifnr.  
  
START-OF-SELECTION.  
  
CALL FUNCTION 'ZPRAC\_FUNCMODSTR'  
EXPORTING  
  I\_LIFNR1       = S\_LIFNR-LOW *"'1'*  
  I\_LIFNR2       = S\_LIFNR-HIGH *"'100'*  
  TABLES  
    gt\_tab         = LT\_TAB  
          .  
  
perform disp\_lifnr\_data.  
*\*&---------------------------------------------------------------------\**  
*\*& Form disp\_lifnr\_data*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM disp\_lifnr\_data .  
  loop at lt\_tab into wa\_tab.  
    WRITE :/ wa\_tab-lifnr,  
              wa\_tab-land1,  
              wa\_tab-name1,  
              wa\_tab-pstl2,  
              wa\_tab-pstlz,  
              wa\_tab-regio.  
    ENDLOOP.  
ENDFORM.

==========================================================================

Classical Report,

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_CLASSICALREPO*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT zprac\_classicalrepo NO STANDARD PAGE HEADING LINE-COUNT 20(3) LINE-SIZE 500.  
  
LOAD-OF-PROGRAM.  
  
  INCLUDE zprac\_clasrepo\_include.  
  
  
  
  SELECT-OPTIONS s\_lifnr FOR wa\_tab-lifnr.  
  
INITIALIZATION. *""to clear, default one, append*  
  
  CLEAR: wa\_tab, lt\_Tab.  
  
  s\_lifnr-low ='1'.  
  s\_lifnr-high = '100'.  
  APPEND s\_lifnr.  
  
AT SELECTION-SCREEN ON s\_lifnr.  
  
  SELECT SINGLE lifnr INTO lv\_lifnr FROM lfa1 WHERE lifnr IN s\_lifnr.  
  
  IF sy-subrc EQ 0.  
    MESSAGE s001(ZPRAC\_message).  
  ELSE.  
    MESSAGE e000(zprac\_message).  
  ENDIF.  
  
START-OF-SELECTION.  
  PERFORM get\_data.  
  
END-OF-SELECTION.  
  PERFORM disp\_data.  
  
data lv\_coun type i. *"""""TO SKIP IF THERE IS ANY SPACE FOR PAGE LIMIT THEN SKIP*  
lv\_coun = sy-linct - sy-linno.  
skip lv\_coun.  
  
TOP-OF-PAGE.  
ULINE.  
write:/ 'sno',  
      10 'city',  
      20 'name',  
      30 'pstl2',  
      60 'pstlz',  
      70 'regio'.  
uline.  
  
end-of-PAGE.  
write:/  sy-pagno,  
          sy-datum,  
          sy-uzeit.  
  
  
*\*&---------------------------------------------------------------------\**  
*\*& Form get\_data*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM get\_data .  
  SELECT lifnr land1 name1 pstl2 pstlz regio  
     INTO TABLE lt\_Tab  
    FROM lfa1  
    WHERE lifnr IN s\_lifnr.  
  IF sy-subrc EQ 0.  
      WRITE:/'RECORDS ARE FOUND'.  
    ELSE.  
      WRITE:/'RECORDS ARE NOT FOUND'.  
    ENDIF.  
ENDFORM.  
*\*&---------------------------------------------------------------------\**  
*\*& Form disp\_data*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM disp\_data .  
  LOOP AT lt\_Tab INTO wa\_tab.  
    WRITE :/  wa\_tab-lifnr,  
              wa\_tab-land1,  
              wa\_tab-name1,  
              wa\_tab-pstl2,  
              wa\_tab-pstlz,  
              wa\_tab-regio.  
  ENDLOOP.  
ENDFORM.

========================================================================

Interactive report,

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_INTERACTREPO1*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT zprac\_classicalrepo2 NO STANDARD PAGE HEADING LINE-COUNT 20(3) LINE-SIZE 500.  
  
LOAD-OF-PROGRAM.  
  
  INCLUDE zprac\_clasrepo2\_include.  
  
  SELECT-OPTIONS s\_lifnr FOR wa\_tab-lifnr OBLIGATORY.  
  
INITIALIZATION. *""to clear, default one, append*  
  
  CLEAR: wa\_Tab, lt\_Tab.  
  
  s\_lifnr-low ='1'.  
  s\_lifnr-high = '100'.  
  APPEND s\_lifnr.  
  
AT SELECTION-SCREEN ON s\_lifnr.  
  
  SELECT SINGLE lifnr INTO lv\_lifnr FROM lfa1 WHERE lifnr IN s\_lifnr.  
  
  IF sy-subrc EQ 0.  
    MESSAGE s001(ZPRAC\_message).  
  ELSE.  
    MESSAGE e000(zprac\_message).  
  ENDIF.  
  
START-OF-SELECTION.  
  PERFORM get\_data\_tab.  
  
END-OF-SELECTION.  
  PERFORM disp\_data\_tab.  
  
  DATA lv\_coun TYPE i. *"""""TO SKIP IF THERE IS ANY SPACE FOR PAGE LIMIT THEN SKIP*  
  lv\_coun = sy-linct - sy-linno.  
  SKIP lv\_coun.  
  
TOP-OF-PAGE.  
  ULINE.  
  WRITE:/ 'sno',  
        10 'city',  
        20 'name',  
        30 'pstl2',  
        60 'pstlz',  
        70 'regio'.  
  ULINE.  
  
END-OF-PAGE.  
  WRITE:/  sy-pagno,  
            sy-datum,  
            sy-uzeit.  
  
AT LINE-SELECTION.  
  
  CASE sy-lsind.  
    WHEN 1.  
  
      SELECT lifnr begru vzskz zwels xverr  
     INTO TABLE lt\_Tab2  
    FROM lfb1  
    WHERE lifnr eq wa\_tab-lifnr.  
  
      LOOP AT lt\_Tab2 INTO wa\_tab2.  
        WRITE : /  wa\_tab2-lifnr,  
                  wa\_tab2-begru,  
                  wa\_tab2-vzskz,  
                  wa\_tab2-zwels,  
                  wa\_tab2-xverr.  
      ENDLOOP.  
  ENDCASE.  
  
TOP-OF-PAGE DURING LINE-SELECTION.  
  CASE sy-lsind.  
      WRITE: / 'records are found'.  
      ULINE.  
      WRITE:/ 'sno','city','regio','land','name'.  
      ULINE.  
  ENDCASE.  
  
  
*\*&---------------------------------------------------------------------\**  
*\*& Form get\_data\_tab*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM get\_data\_tab .  
  SELECT lifnr land1 name1 pstl2 pstlz regio  
     INTO TABLE lt\_Tab  
    FROM lfa1  
    WHERE lifnr IN s\_lifnr.  
  IF sy-subrc EQ 0.  
    WRITE:/'RECORDS ARE FOUND'.  
  ELSE.  
    WRITE:/'RECORDS ARE NOT FOUND'.  
  ENDIF.  
ENDFORM.  
*\*&---------------------------------------------------------------------\**  
*\*& Form disp\_data\_tab*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM disp\_data\_tab .  
  LOOP AT lt\_Tab INTO wa\_tab.  
    WRITE :/  wa\_tab-lifnr HOTSPOT,  
              wa\_tab-land1,  
              wa\_tab-name1,  
              wa\_tab-pstl2,  
              wa\_tab-pstlz,  
              wa\_tab-regio.  
  
    HIDE wa\_tab-lifnr.  
  
  ENDLOOP.  
ENDFORM.

==========================================================================

Interactive at user,

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_INTERACTIVEREPO\_ATUSER*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_INTERACTIVEREPO\_ATUSER NO STANDARD PAGE HEADING LINE-COUNT 20(3) LINE-SIZE 500.  
  
LOAD-OF-PROGRAM.  
  
  INCLUDE zprac\_interuserepo\_include.  
  
  SELECT-OPTIONS s\_lifnr FOR wa\_tab-lifnr OBLIGATORY.  
  
INITIALIZATION. *""to clear, default one, append*  
  
  CLEAR: wa\_Tab, lt\_Tab.  
  
  s\_lifnr-low ='1'.  
  s\_lifnr-high = '100'.  
  APPEND s\_lifnr.  
  
AT SELECTION-SCREEN ON s\_lifnr.  
  
  SELECT SINGLE lifnr INTO lv\_lifnr FROM lfa1 WHERE lifnr IN s\_lifnr.  
  
  IF sy-subrc EQ 0.  
    MESSAGE s001(ZPRAC\_message).  
  ELSE.  
    MESSAGE e000(zprac\_message).  
  ENDIF.  
  
START-OF-SELECTION.  
  set pf-status 'ZPRAC\_PFS'.  
  
  PERFORM get\_report\_data.  
  
END-OF-SELECTION.  
  PERFORM disp\_report\_data.  
  
  DATA lv\_coun TYPE i. *"""""TO SKIP IF THERE IS ANY SPACE FOR PAGE LIMIT THEN SKIP*  
  lv\_coun = sy-linct - sy-linno.  
  SKIP lv\_coun.  
  
TOP-OF-PAGE.  
  ULINE.  
  WRITE:/ 'sno',  
        10 'city',  
        20 'name',  
        30 'pstl2',  
        60 'pstlz',  
        70 'regio'.  
  ULINE.  
  
END-OF-PAGE.  
  WRITE:/  sy-pagno,  
            sy-datum,  
            sy-uzeit.  
  
at user-command.  
  
  case sy-ucomm.  
    when 'INFO'.  
      write ' UST INFO '.  
    when 'VIEW'.  
      write ' UST VIEW INFO'.  
ENDCASE.  
  
  
*\*&---------------------------------------------------------------------\**  
*\*& Form get\_report\_data*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM get\_report\_data .  
  SELECT lifnr land1 name1 pstl2 pstlz regio  
     INTO TABLE lt\_Tab  
    FROM lfa1  
    WHERE lifnr IN s\_lifnr.  
  IF sy-subrc EQ 0.  
    WRITE:/'RECORDS ARE FOUND'.  
  ELSE.  
    WRITE:/'RECORDS ARE NOT FOUND'.  
  ENDIF.  
ENDFORM.  
*\*&---------------------------------------------------------------------\**  
*\*& Form disp\_report\_data*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM disp\_report\_data .  
  LOOP AT lt\_Tab INTO wa\_tab.  
    WRITE :/  wa\_tab-lifnr HOTSPOT,  
              wa\_tab-land1,  
              wa\_tab-name1,  
              wa\_tab-pstl2,  
              wa\_tab-pstlz,  
              wa\_tab-regio.  
  
    HIDE wa\_tab-lifnr.  
  
    ENDLOOP.  
ENDFORM.

=======================================================================

*\*&---------------------------------------------------------------------\**  
*\*& Report Z0381\_INTERACTREPOR*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT z0381\_interactrepor NO STANDARD PAGE HEADING LINE-COUNT 20(3) LINE-SIZE 500.  
  
LOAD-OF-PROGRAM.  
  
  TYPES: BEGIN OF ls\_tab,  
           vbeln TYPE  z0381vbeln,  
           fkart TYPE  fkart,  
           fktyp TYPE  fktyp,  
           vbtyp TYPE  vbtyp,  
         END OF ls\_tab,  
  
         BEGIN OF ls\_tab2,  
           vbeln TYPE  z0381vbeln,  
           posnr TYPE  z0381posnr,  
           fkimg TYPE  fkimg,  
           vrkme  TYPE vrkme,  
           meins type meins,  
         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 lv\_vbeln TYPE vbeln.  
  SELECT-OPTIONS s\_vbeln FOR wa\_tab-vbeln OBLIGATORY.  
  
INITIALIZATION. *""to clear, default one, append*  
  
  CLEAR: wa\_Tab, lt\_Tab.  
  
  APPEND s\_vbeln.  
  
AT SELECTION-SCREEN ON s\_vbeln.  
  
  SELECT SINGLE vbeln INTO lv\_vbeln FROM z0381\_vbrk WHERE vbeln IN s\_vbeln.  
  
  IF sy-subrc EQ 0.  
    MESSAGE s001(ZPRAC\_message).  
  ELSE.  
    MESSAGE e000(zprac\_message).  
  ENDIF.  
  
START-OF-SELECTION.  
  PERFORM get\_tab\_data.  
  
END-OF-SELECTION.  
  PERFORM disp\_tab\_data.  
  
  DATA lv\_coun TYPE i. *"""""TO SKIP IF THERE IS ANY SPACE FOR PAGE LIMIT THEN SKIP*  
  lv\_coun = sy-linct - sy-linno.  
  SKIP lv\_coun.  
  
TOP-OF-PAGE.  
  ULINE.  
  WRITE:/ 'billing doc',  
        10 'billing type',  
        20 'billing category',  
        30 'sd doc',  
        50 'sales'.  
  ULINE.  
  
END-OF-PAGE.  
  WRITE:/  sy-pagno,  
            sy-datum,  
            sy-uzeit.  
  
AT LINE-SELECTION.  
  
  CASE sy-lsind.  
    WHEN 1.  
  
      SELECT vbeln posnr fkimg vrkme meins  
     INTO TABLE lt\_Tab2  
      FROM z0381\_vbrp  
      WHERE vbeln EQ wa\_tab-vbeln.  
  
      LOOP AT lt\_Tab2 INTO wa\_tab2.  
        WRITE : /  wa\_tab2-vbeln,  
                  wa\_tab2-posnr,  
                  wa\_tab2-fkimg,  
                  wa\_tab2-vrkme,  
                  wa\_tab2-meins.  
      ENDLOOP.  
  ENDCASE.  
  TOP-OF-PAGE DURING LINE-SELECTION.  
  CASE sy-lsind.  
      WRITE: / 'Records are found'.  
      skip.  
      WRITE: /10'item',  
      20'invoice', 30'sales', 50'units'.  
      ULINE.  
  ENDCASE.  
*\*&---------------------------------------------------------------------\**  
*\*& Form get\_tab\_data*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM get\_tab\_data .  
  SELECT vbeln fkart fktyp vbtyp  
     INTO TABLE lt\_Tab  
    FROM z0381\_vbrk  
    WHERE vbeln IN s\_vbeln.  
  IF sy-subrc EQ 0.  
    WRITE:/'RECORDS ARE FOUND'.  
  ELSE.  
    WRITE:/'RECORDS ARE NOT FOUND'.  
  ENDIF.  
ENDFORM.  
*\*&---------------------------------------------------------------------\**  
*\*& Form disp\_tab\_data*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM disp\_tab\_data .  
   LOOP AT lt\_Tab INTO wa\_tab.  
    WRITE :/  wa\_tab-vbeln HOTSPOT,  
              wa\_tab-fkart,  
              wa\_tab-fktyp,  
              wa\_tab-vbtyp.  
  
    HIDE wa\_tab-vbeln.  
  
  ENDLOOP.  
ENDFORM.

=========================================================================

Interactive reports with joins,

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_INTERACTREPO\_JOINS*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT zprac\_interactrepo\_joins NO STANDARD PAGE HEADING LINE-COUNT 30(5) LINE-SIZE 400.  
  
LOAD-OF-PROGRAM.  
  
  INCLUDE zprac\_interactrepo\_joins\_top.  
  
  SELECT-OPTIONS s\_matnr FOR lv\_matnr OBLIGATORY.  
  
INITIALIZATION.  
  CLEAR: wa\_tab, lt\_tab.  
  
  s\_matnr-low = '1'.  
  s\_matnr-high = '100'.  
  APPEND s\_matnr.  
  
AT SELECTION-SCREEN ON s\_matnr.  
  SELECT SINGLE matnr INTO lv\_matnr FROM mara WHERE  
    matnr IN s\_matnr.  
  IF sy-subrc EQ 0.  
    WRITE ' records are found '.  
  ELSE.  
    WRITE ' records are not found '.  
  ENDIF.  
  
START-OF-SELECTION.  
  PERFORM get\_mat\_tab.  
  
END-OF-SELECTION.  
  PERFORM disp\_mat\_tab.  
  
TOP-OF-PAGE.  
  ULINE.  
  WRITE:/5 'mat num',  
        20 'mat type',  
        40 'status',  
        45 'flag'.  
  ULINE.  
  
END-OF-PAGE.  
  WRITE:/ sy-datum, sy-uzeit.  
  
AT LINE-SELECTION.  
  CASE sy-lsind.  
    WHEN 1.  
      SELECT marc~matnr  
             marc~ekgrp  
             marc~ausme  
  
             makt~spras  
             makt~maktx  
      INTO TABLE lt\_tab2 FROM marc INNER JOIN makt ON  
        marc~matnr = makt~matnr  
      WHERE marc~matnr in s\_matnr.  
           LOOP AT lt\_tab2 into wa\_tab2.  
             write:/ wa\_tab2-matnr,  
             wa\_tab2-ekgrp,  
             wa\_tab2-ausme,  
             wa\_tab2-spras,  
             wa\_tab2-maktx.  
           ENDLOOP.  
  ENDCASE.  
 TOP-OF-PAGE DURING LINE-SELECTION.  
  CASE sy-lsind.  
      WRITE: / 'records are found'.  
      ULINE.  
      WRITE:/ 'mat num',  
          10 'group',  
          30 'outputunit',  
          50 'lang key',  
          60 'mat desc'.  
      ULINE.  
  ENDCASE.  
  
  
*\*&---------------------------------------------------------------------\**  
*\*& Form get\_mat\_tab*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM get\_mat\_tab .  
  SELECT matnr mtart pstat lvorm INTO TABLE lt\_tab FROM mara  
  WHERE matnr IN s\_matnr.  
  IF sy-subrc EQ 0.  
    WRITE:/ ' records are found '.  
  ELSE.  
    WRITE:/ ' records are not found '.  
  ENDIF.  
ENDFORM.  
*\*&---------------------------------------------------------------------\**  
*\*& Form disp\_mat\_tab*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM disp\_mat\_tab .  
  LOOP AT lt\_tab into wa\_tab.  
    write:/ wa\_tab-matnr HOTSPOT,  
            wa\_tab-mtart,  
            wa\_tab-pstat,  
            wa\_tab-lvorm.  
  
    Hide wa\_tab-matnr.  
  ENDLOOP.  
ENDFORM.

========================================================================

Intetractive at get cursor,

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_INTERACTREPO\_GCURSOR*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_INTERACTREPO\_GCURSOR NO STANDARD PAGE HEADING LINE-COUNT 20(3) LINE-SIZE 600.  
  
data lv\_vbeln type vbeln\_va.  
TYPES: BEGIN OF ls\_tab,  
  vbeln type vbeln\_va,  
  kunnr type kunnr,  
  erzet type erzet,  
  erdat type erdat,  
  end of ls\_tab.  
  
DATA: lt\_tab type table of ls\_tab,  
      wa\_tab type ls\_tab.  
  
SELECT-OPTIONS s\_vbeln for lv\_vbeln.  
  
START-OF-SELECTION.  
select vbeln kunnr erzet erdat into table lt\_tab from vbak  
  where vbeln in s\_vbeln.  
  
end-of-SELECTION.  
LOOP AT lt\_tab into wa\_tab.  
  
  FORMAT HOTSPOT on.  
  WRITE:/ wa\_tab-vbeln,  
          wa\_tab-kunnr,  
          wa\_tab-erzet.  
  FORMAT HOTSPOT off.  
  WRITE  wa\_tab-erdat.  
ENDLOOP.  
  
TOP-OF-PAGE.  
WRITE:/ 'Sales doc no',  
        'cust no',  
        'Time',  
        'date'.  
  
AT LINE-SELECTION.  
  
DATA: fnam(15), fval(15).  
GET CURSOR FIELD fnam VALUE fval.  
if fnam = 'wa\_tab-vbeln'.  
  set PARAMETER ID 'AUN' Field fval.  
  call TRANSACTION 'VA03' AND SKIP FIRST SCREEN.  
  
ELSEIF fnam = 'wa\_tab-kunnr'.  
  set PARAMETER ID 'KUN' Field fval.  
  call TRANSACTION 'XD02' AND SKIP FIRST SCREEN.  
ENDIF.

==========================================================================

For all entries,

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_FORALL\_ENTRIES*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_FORALL\_ENTRIES NO STANDARD PAGE HEADING LINE-COUNT 30(2) LINE-SIZE 400.  
  
INCLUDE ZPRAC\_FORALLENTRIES\_TOP.  
  
TABLES ekko.  
  
select-OPTIONS s\_ebeln for ekko-ebeln.  
  
START-OF-SELECTION.  
select ebeln pincr kufix ausnr into table lt\_tab1 from ekko  
  where ebeln in s\_ebeln.  
  
if not lt\_tab1[] is INITIAL.                *"""""""""""""# 1*  
  select ebeln bednr insmk matkl into table lt\_tab2 from ekpo  
    FOR ALL ENTRIES IN lt\_tab1 where ebeln eq lt\_tab1-ebeln.    *"""""""""'# 2*  
  
DELETE ADJACENT DUPLICATES FROM lt\_tab1 COMPARING ebeln.        *""""""""""""#3*  
  
sort lt\_tab2 by ebeln.            *""""""""""#4*  
  
loop at lt\_tab1 into wa\_tab1.  
  
READ TABLE lt\_tab2 into wa\_tab2 with key ebeln = wa\_tab1-ebeln BINARY SEARCH.   *"""""""""""#5*  
  
if sy-subrc = 0.  
  
MOVE: wa\_tab1-ebeln to wa\_final-ebeln,  
      wa\_tab1-pincr to wa\_final-pincr,  
      wa\_tab1-kufix to wa\_final-kufix,  
      wa\_tab1-ausnr to wa\_final-ausnr,  
  
      wa\_tab2-bednr to wa\_final-bednr,  
      wa\_tab2-insmk to wa\_final-insmk,  
      wa\_tab2-matkl to wa\_final-matkl.  
append wa\_final to lt\_final.  
  
else.  
  continue.  
ENDIF.  
ENDLOOP.  
  
LOOP AT lt\_final into wa\_final.  
  WRITE:/ wa\_final-ebeln,  
          wa\_final-pincr,  
          wa\_final-kufix,  
          wa\_final-ausnr,  
  
          wa\_final-bednr,  
          wa\_final-insmk,  
          wa\_final-matkl.  
ENDLOOP.  
else.  
Write 'Records are NOT FOUND'.  
  
ENDIF.

Include prog,

*\*&---------------------------------------------------------------------\**  
*\*& Include          ZPRAC\_FORALLENTRIES\_TOP*  
*\*&---------------------------------------------------------------------\**  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*EKKO*  
TYPES: BEGIN OF ls\_tab1,  
    ebeln type ebeln,     *"purchasing doc no.*  
    pincr type pincr,     *"item no. interval*  
    kufix type kufix,     *"fixed exchange rate indicator*  
    ausnr type ausch,     *"bid invitation no.*  
    end of ls\_tab1.  
DATA: lt\_tab1 type STANDARD TABLE OF ls\_tab1,  
      wa\_tab1 type ls\_tab1.  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*EKPO*  
TYPES: BEGIN OF ls\_tab2,  
    ebeln type ebeln,      *"purchase doc no.*  
    bednr type bednr,      *"req tracking no.*  
  insmk type insmk,        *"stock type*  
  matkl type matkl,        *"material group*  
    end of ls\_tab2.  
DATA: lt\_tab2 type STANDARD TABLE OF ls\_tab2,  
      wa\_tab2 type ls\_tab2.  
  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*final*  
TYPES: BEGIN OF ls\_final,  
    ebeln type ebeln,     *"purchasing doc no.*  
    pincr type pincr,     *"item no. interval*  
    kufix type kufix,     *"fixed exchange rate indicator*  
    ausnr type ausch,     *"bid invitation no.*  
  
    bednr type bednr,      *"req tracking no.*  
  insmk type insmk,        *"stock type*  
  matkl type matkl,        *"material group*  
    end of ls\_final.  
  
DATA: lt\_final type STANDARD TABLE OF ls\_final,  
      wa\_final type ls\_final.

=========================================================================

For all entries for 3 tables,

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_FORALL\_JOINS\_3TABS*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_FORALL\_JOINS\_3TABS NO STANDARD PAGE HEADING LINE-COUNT 30(5) LINE-SIZE 600.  
  
INCLUDE ZPRAC\_FORALL\_3TABS\_TOP.  
  
TABLES ekko.  
  
select-OPTIONS s\_ebeln for ekko-ebeln.  
  
START-OF-SELECTION.  
select ekko~ebeln  
       ekko~pincr  
       ekko~kufix  
       ekko~ausnr  
  
       ekpo~bednr  
       ekpo~insmk  
       ekpo~matkl into table lt\_tab1 from ekko inner JOIN ekpo  
       on ekko~ebeln = ekpo~ebeln where ekko~ebeln in s\_ebeln.  
  
if not lt\_tab1[] is INITIAL.                *"""""""""""""# 1*  
  select ebeln ameng wemng wamng  into table lt\_tab2 from eket  
    FOR ALL ENTRIES IN lt\_tab1 where ebeln eq lt\_tab1-ebeln.    *"""""""""'# 2*  
  
DELETE ADJACENT DUPLICATES FROM lt\_tab1 COMPARING ebeln.        *""""""""""""#3*  
  
sort lt\_tab2 by ebeln.            *""""""""""#4*  
  
loop at lt\_tab1 into wa\_tab1.  
  
READ TABLE lt\_tab2 into wa\_tab2 with key ebeln = wa\_tab1-ebeln BINARY SEARCH.   *"""""""""""#5*  
  
if sy-subrc = 0.  
  
MOVE: wa\_tab1-ebeln to wa\_final-ebeln,  
      wa\_tab1-pincr to wa\_final-pincr,  
      wa\_tab1-kufix to wa\_final-kufix,  
      wa\_tab1-ausnr to wa\_final-ausnr,  
      wa\_tab1-bednr to wa\_final-bednr,  
      wa\_tab1-insmk to wa\_final-insmk,  
      wa\_tab1-matkl to wa\_final-matkl,  
  
      wa\_tab2-ebeln to wa\_final-ebeln,  
      wa\_tab2-ameng to wa\_final-ameng,  
      wa\_tab2-wemng to wa\_final-wemng,  
      wa\_tab2-wamng to wa\_final-wamng.  
  
append wa\_final to lt\_final.  
  
else.  
  continue.  
ENDIF.  
ENDLOOP.  
  
LOOP AT lt\_final into wa\_final.  
  WRITE:/ wa\_final-ebeln,  
          wa\_final-pincr,  
          wa\_final-kufix,  
          wa\_final-ausnr,  
  
          wa\_final-bednr,  
          wa\_final-insmk,  
          wa\_final-matkl,  
  
          wa\_final-ameng,  
          wa\_final-wemng,  
          wa\_final-wamng.  
ENDLOOP.  
else.  
Write 'Records are NOT FOUND'.  
  
ENDIF.

Include prog,

*\*&---------------------------------------------------------------------\**  
*\*& Include          ZPRAC\_FORALL\_3TABS\_TOP*  
*\*&---------------------------------------------------------------------\**  
  
TYPES: BEGIN OF ls\_tab1,  
         ebeln TYPE ebeln,     *"purchasing doc no.*  
         pincr TYPE pincr,     *"item no. interval*  
         kufix TYPE kufix,     *"fixed exchange rate indicator*  
         ausnr TYPE ausch,     *"bid invitation no.*  
  
         bednr TYPE bednr,      *"req tracking no.*  
         insmk TYPE insmk,        *"stock type*  
         matkl TYPE matkl,        *"material group*  
       END OF ls\_tab1.  
DATA: lt\_tab1 TYPE STANDARD TABLE OF ls\_tab1,  
      wa\_tab1 TYPE ls\_tab1.  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*EKET*  
TYPES: BEGIN OF ls\_tab2,  
         ebeln TYPE ebeln,         *"purchasing doc no.*  
         ameng TYPE  vomng,        *"Previous quan*  
         wemng TYPE  weemg,          *"recieved goods*  
         wamng TYPE  wamng ,         *"issued quan*  
       END OF ls\_tab2.  
DATA: lt\_tab2 TYPE STANDARD TABLE OF ls\_tab2,  
      wa\_tab2 TYPE ls\_tab2.  
  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*final*  
TYPES: BEGIN OF ls\_final,  
         ebeln TYPE ebeln,     *"purchasing doc no.*  
         pincr TYPE pincr,     *"item no. interval*  
         kufix TYPE kufix,     *"fixed exchange rate indicator*  
         ausnr TYPE ausch,     *"bid invitation no.*  
  
         bednr TYPE bednr,      *"req tracking no.*  
         insmk TYPE insmk,        *"stock type*  
         matkl TYPE matkl,        *"material group*  
  
         ameng TYPE  vomng,        *"Previous quan*  
         wemng TYPE  weemg,          *"recieved goods*  
         wamng TYPE  wamng ,         *"issued quan*  
       END OF ls\_final.  
  
DATA: lt\_final TYPE STANDARD TABLE OF ls\_final,  
      wa\_final TYPE ls\_final.

=========================================================================

ALV report, LIST

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_ALV\_REPO\_LIST*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_ALV\_REPO\_LIST NO STANDARD PAGE HEADING LINE-COUNT 30(2) LINE-SIZE 500.  
  
INCLUDE ZPRAC\_ALVREPO\_TOP.  
  
select-options s\_vbeln for lv\_vbeln OBLIGATORY.  
  
INITIALIZATION.  
  
clear: lt\_tab, wa\_fieldcat, lt\_fieldcat.  
s\_vbeln-low = '0000000023'.  
s\_vbeln-high = '0000000032'.  
append s\_vbeln.  
  
at SELECTION-SCREEN on s\_vbeln.  
  
  select single vbeln into lv\_vbeln from vbak where vbeln in s\_vbeln.  
    IF sy-subrc EQ 0.  
    MESSAGE s001(ZPRAC\_message).  
  ELSE.  
    MESSAGE e000(zprac\_message).  
  ENDIF.  
  
  START-OF-SELECTION.  
  PERFORM get\_data.  
  
  PERFORM get\_alv\_fact.  
  
  END-OF-SELECTION.  
  perform disp\_data.  
*\*&---------------------------------------------------------------------\**  
*\*& Form get-alv-fact*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM get\_alv\_fact .  
  
  *"""""VBELN""""*  
  wa\_fieldcat-col\_pos = '1'.  
  wa\_fieldcat-seltext\_l = 'sale doc'.  
  wa\_fieldcat-fieldname = 'vbeln'.  
  wa\_fieldcat-tabname = 'lt-tab'.  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  *"""""""ERDAT"""""""*  
  wa\_fieldcat-col\_pos = '2'.  
  wa\_fieldcat-seltext\_l = 'record created on'.  
  wa\_fieldcat-fieldname = 'erdat'.  
  wa\_fieldcat-tabname = 'lt-tab'.  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  *"""""""erzet""""""""*  
  wa\_fieldcat-col\_pos = '3'.  
  wa\_fieldcat-seltext\_l = 'entry time'.  
  wa\_fieldcat-fieldname = 'erzet'.  
  wa\_fieldcat-tabname = 'lt-tab'.  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  *"""""""ernam""""""""""*  
  wa\_fieldcat-col\_pos = '4'.  
  wa\_fieldcat-seltext\_l = 'person name'.  
  wa\_fieldcat-fieldname = 'ernam'.  
  wa\_fieldcat-tabname = 'lt-tab'.  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
ENDFORM.  
*\*&---------------------------------------------------------------------\**  
*\*& Form get\_data*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM get\_data .  
  select vbeln erdat erzet ernam into table lt\_tab from vbak where vbeln in s\_vbeln.  
    IF sy-subrc EQ 0.  
    MESSAGE s001(ZPRAC\_message).  
  ELSE.  
    MESSAGE e000(zprac\_message).  
  ENDIF.  
  
ENDFORM.  
*\*&---------------------------------------------------------------------\**  
*\*& Form disp\_data*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM disp\_data .  
  CALL FUNCTION 'REUSE\_ALV\_LIST\_DISPLAY'  
EXPORTING  
  I\_CALLBACK\_PROGRAM             = 'ZPRAC\_ALV\_REPO\_LIST '  
  IT\_FIELDCAT                    = lt\_fieldcat  
    TABLES  
      t\_outtab                       = lt\_tab  
   EXCEPTIONS  
     PROGRAM\_ERROR                  = 1  
     OTHERS                         = 2  
            .  
  IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
  ENDIF.  
  
ENDFORM.

Include prog,

*\*&---------------------------------------------------------------------\**  
*\*& Include          ZPRAC\_ALVREPO\_TOP*  
*\*&---------------------------------------------------------------------\**  
  
  
TYPES: BEGIN OF ls\_tab,  
         vbeln TYPE vbeln\_va,    *"SALES DOC*  
         erdat TYPE  erdat,      *"Record created on*  
         erzet TYPE  erzet,        *"entry time*  
         ernam TYPE  ernam,        *"nameof person*  
       END OF ls\_tab.  
  
DATA : lt\_tab      TYPE STANDARD TABLE OF ls\_tab,  
       wa\_tab      TYPE ls\_tab,  
  
       lv\_vbeln    TYPE vbeln,  
  
       lt\_fieldcat TYPE slis\_T\_fieldcat\_alv,  
       wa\_fieldcat TYPE slis\_fieldcat\_alv.  
  
CONSTANTS : cs\_tab(10) type c value 'lt-tab',  
            cs\_vbeln(10) type c value 'vbeln',  
            cs\_erdat(10) type c value 'erdat',  
            cs\_erzet(10) type c value 'erzet',  
            cs\_ernam(10) type c value 'ernam'.

==========================================================================

ALV reports – GRID,

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_ALV\_REPO\_GRID*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_ALV\_REPO\_GRID NO STANDARD PAGE HEADING LINE-COUNT 30(2) LINE-SIZE 500.  
  
INCLUDE ZPRAC\_ALVREPO\_TOP.  
  
select-options s\_vbeln for lv\_vbeln OBLIGATORY.  
  
INITIALIZATION.  
  
clear: lt\_tab, wa\_fieldcat, lt\_fieldcat.  
s\_vbeln-low = '0000000023'.  
s\_vbeln-high = '0000000032'.  
append s\_vbeln.  
  
at SELECTION-SCREEN on s\_vbeln.  
  
  select single vbeln into lv\_vbeln from vbak where vbeln in s\_vbeln.  
    IF sy-subrc EQ 0.  
    MESSAGE s001(ZPRAC\_message).  
  ELSE.  
    MESSAGE e000(zprac\_message).  
  ENDIF.  
  
  START-OF-SELECTION.  
  PERFORM get\_data.  
  
  PERFORM get\_alv\_fact.  
  
  END-OF-SELECTION.  
  perform disp\_data.  
*\*&---------------------------------------------------------------------\**  
*\*& Form get-alv-fact*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM get\_alv\_fact .  
  
  *"""""VBELN""""*  
  wa\_fieldcat-col\_pos = '1'.  
  wa\_fieldcat-seltext\_l = 'sale doc'.  
  wa\_fieldcat-fieldname = 'vbeln'.  
  wa\_fieldcat-tabname = 'lt-tab'.  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  *"""""""ERDAT"""""""*  
  wa\_fieldcat-col\_pos = '2'.  
  wa\_fieldcat-seltext\_l = 'record created on'.  
  wa\_fieldcat-fieldname = 'erdat'.  
  wa\_fieldcat-tabname = 'lt-tab'.  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  *"""""""erzet""""""""*  
  wa\_fieldcat-col\_pos = '3'.  
  wa\_fieldcat-seltext\_l = 'entry time'.  
  wa\_fieldcat-fieldname = 'erzet'.  
  wa\_fieldcat-tabname = 'lt-tab'.  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  *"""""""ernam""""""""""*  
  wa\_fieldcat-col\_pos = '4'.  
  wa\_fieldcat-seltext\_l = 'person name'.  
  wa\_fieldcat-fieldname = 'ernam'.  
  wa\_fieldcat-tabname = 'lt-tab'.  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
ENDFORM.  
*\*&---------------------------------------------------------------------\**  
*\*& Form get\_data*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM get\_data .  
  select vbeln erdat erzet ernam into table lt\_tab from vbak where vbeln in s\_vbeln.  
    IF sy-subrc EQ 0.  
    MESSAGE s001(ZPRAC\_message).  
  ELSE.  
    MESSAGE e000(zprac\_message).  
  ENDIF.  
  
ENDFORM.  
*\*&---------------------------------------------------------------------\**  
*\*& Form disp\_data*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM disp\_data .  
  CALL FUNCTION 'REUSE\_ALV\_GRID\_DISPLAY'  
   EXPORTING  
      I\_CALLBACK\_PROGRAM                = 'ZPRAC\_ALV\_REPO\_GRID '    *"""or sy-repid*  
   I\_GRID\_TITLE                      = 'SALES DOCUMENT HEADER'  
    IT\_FIELDCAT                       = lt\_fieldcat  
    TABLES  
      t\_outtab                          = lt\_tab    .  
  IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
  ENDIF.  
  
  
ENDFORM.

Include prog,

*\*&---------------------------------------------------------------------\**  
*\*& Include          ZPRAC\_ALVREPO\_TOP*  
*\*&---------------------------------------------------------------------\**  
  
  
TYPES: BEGIN OF ls\_tab,  
         vbeln TYPE vbeln\_va,    *"SALES DOC*  
         erdat TYPE  erdat,      *"Record created on*  
         erzet TYPE  erzet,        *"entry time*  
         ernam TYPE  ernam,        *"nameof person*  
       END OF ls\_tab.  
  
DATA : lt\_tab      TYPE STANDARD TABLE OF ls\_tab,  
       wa\_tab      TYPE ls\_tab,  
  
       lv\_vbeln    TYPE vbeln,  
  
       lt\_fieldcat TYPE slis\_T\_fieldcat\_alv,  
       wa\_fieldcat TYPE slis\_fieldcat\_alv.  
  
CONSTANTS : cs\_tab(10) type c value 'lt-tab',  
            cs\_vbeln(10) type c value 'vbeln',  
            cs\_erdat(10) type c value 'erdat',  
            cs\_erzet(10) type c value 'erzet',  
            cs\_ernam(10) type c value 'ernam'.

=========================================================================

ALV reports – JOINS

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_ALV\_REPO\_GRID\_JOINS*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_ALV\_REPO\_GRID\_JOINS NO STANDARD PAGE HEADING LINE-COUNT 30(2) LINE-SIZE 500.  
  
INCLUDE ZPRAC\_ALVREPOSJOINS\_TOP.  
  
select-options s\_vbeln for lv\_vbeln OBLIGATORY.  
  
INITIALIZATION.  
  
clear: lt\_tab, wa\_fieldcat, lt\_fieldcat.  
s\_vbeln-low = '0000000023'.  
s\_vbeln-high = '0000000032'.  
append s\_vbeln.  
  
at SELECTION-SCREEN on s\_vbeln.  
  
  select single vbeln into lv\_vbeln from vbak where vbeln in s\_vbeln.  
    IF sy-subrc EQ 0.  
    MESSAGE s001(ZPRAC\_message).  
  ELSE.  
    MESSAGE e000(zprac\_message).  
  ENDIF.  
  
  START-OF-SELECTION.  
  PERFORM get\_data.  
  
  PERFORM get\_alv\_fact.  
  
  END-OF-SELECTION.  
  perform disp\_data.  
*\*&---------------------------------------------------------------------\**  
*\*& Form get-alv-fact*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM get\_alv\_fact .  
  
  *"""""VBELN""""*  
  wa\_fieldcat-col\_pos = '1'.  
  wa\_fieldcat-seltext\_l = 'sale doc'.  
  wa\_fieldcat-fieldname = 'vbeln'.  
  wa\_fieldcat-tabname = 'lt-tab'.  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  *"""""""ERDAT"""""""*  
  wa\_fieldcat-col\_pos = '6'.  
  wa\_fieldcat-seltext\_l = 'record created on'.  
  wa\_fieldcat-fieldname = 'erdat'.  
  wa\_fieldcat-tabname = 'lt-tab'.  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  *"""""""erzet""""""""*  
  wa\_fieldcat-col\_pos = '3'.  
  wa\_fieldcat-seltext\_l = 'entry time'.  
  wa\_fieldcat-fieldname = 'erzet'.  
  wa\_fieldcat-tabname = 'lt-tab'.  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  *"""""""ernam""""""""""*  
  wa\_fieldcat-col\_pos = '4'.  
  wa\_fieldcat-seltext\_l = 'person name'.  
  wa\_fieldcat-fieldname = 'ernam'.  
  wa\_fieldcat-tabname = 'lt-tab'.  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  *""""""""""""POSNR""""""""*  
   wa\_fieldcat-col\_pos = '2'.  
  wa\_fieldcat-seltext\_l = 'sales document item'.  
  wa\_fieldcat-fieldname = 'posnr'.  
  wa\_fieldcat-tabname = 'lt-tab'.  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  *"""""""""""""""""""""""""MATNR*  
   wa\_fieldcat-col\_pos = '5'.  
  wa\_fieldcat-seltext\_l = 'Material number'.  
  wa\_fieldcat-fieldname = 'matnr'.  
  wa\_fieldcat-tabname = 'lt-tab'.  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
  *"""""""""""""MATWA"""""""""*  
   wa\_fieldcat-col\_pos = '4'.  
  wa\_fieldcat-seltext\_l = 'Material Entered'.  
  wa\_fieldcat-fieldname = 'matwa'.  
  wa\_fieldcat-tabname = 'lt-tab'.  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
ENDFORM.  
*\*&---------------------------------------------------------------------\**  
*\*& Form get\_data*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM get\_data .  
  select vbak~vbeln vbak~erdat vbak~erzet vbak~ernam  
    vbap~posnr vbap~matnr vbap~matwa  
    into table lt\_tab  
    from vbak inner join vbap on  
    vbak~vbeln = vbap~vbeln where vbak~vbeln in s\_vbeln.  
  
    IF sy-subrc EQ 0.  
    MESSAGE s001(ZPRAC\_message).  
  ELSE.  
    MESSAGE e000(zprac\_message).  
  ENDIF.  
  
ENDFORM.  
*\*&---------------------------------------------------------------------\**  
*\*& Form disp\_data*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM disp\_data .  
  CALL FUNCTION 'REUSE\_ALV\_GRID\_DISPLAY'  
   EXPORTING  
      I\_CALLBACK\_PROGRAM                = 'ZPRAC\_ALV\_REPO\_GRID '    *"""or sy-repid*  
   I\_GRID\_TITLE                      = 'SALES DOCUMENT HEADER'  
    IT\_FIELDCAT                       = lt\_fieldcat  
    TABLES  
      t\_outtab                          = lt\_tab    .  
  IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
  ENDIF.  
  
  
ENDFORM.

Include prog,

*\*&---------------------------------------------------------------------\**  
*\*& Include          ZPRAC\_ALVREPOSJOINS\_TOP*  
*\*&---------------------------------------------------------------------\**  
  
  
TYPES: BEGIN OF ls\_tab,  
         vbeln TYPE vbeln\_va,    *"SALES DOC*  
         erdat TYPE  erdat,      *"Record created on*  
         erzet TYPE  erzet,        *"entry time*  
         ernam TYPE  ernam,        *"nameof person*  
  
  
         posnr TYPE  posnr\_va,  
         matnr TYPE  matnr,  
         matwa TYPE  matwa,  
         pmatn TYPE   pmatn,  
       END OF ls\_tab.  
  
DATA : lt\_tab      TYPE STANDARD TABLE OF ls\_tab,  
       wa\_tab      TYPE ls\_tab,  
  
       lv\_vbeln    TYPE vbeln,  
  
       lt\_fieldcat TYPE slis\_T\_fieldcat\_alv,  
       wa\_fieldcat TYPE slis\_fieldcat\_alv.

==========================================================================

ALV reports – LOGO

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_ALV\_REPO\_GRID\_LOGO*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_ALV\_REPO\_GRID\_LOGO NO STANDARD PAGE HEADING LINE-COUNT 30(2) LINE-SIZE 500.  
  
INCLUDE ZPRAC\_ALVREPO\_TOP.  
  
select-options s\_vbeln for lv\_vbeln OBLIGATORY.  
  
INITIALIZATION.  
  
clear: lt\_tab, wa\_fieldcat, lt\_fieldcat.  
s\_vbeln-low = '0000000023'.  
s\_vbeln-high = '0000000032'.  
append s\_vbeln.  
  
at SELECTION-SCREEN on s\_vbeln.  
  
  select single vbeln into lv\_vbeln from vbak where vbeln in s\_vbeln.  
    IF sy-subrc EQ 0.  
    MESSAGE s001(ZPRAC\_message).  
  ELSE.  
    MESSAGE e000(zprac\_message).  
  ENDIF.  
  
  START-OF-SELECTION.  
  PERFORM get\_data.  
  
*\*  PERFORM get\_alv\_fact.*  
  
PERFORM get\_alv\_fact USING text-001 'VBELN' text-006 .  
PERFORM get\_alv\_fact USING text-002 'ERDAT' text-006.  
PERFORM get\_alv\_fact USING text-003 'ERZET' text-006.  
PERFORM get\_alv\_fact USING text-004 'ERNAM' text-006.  
  
  
  END-OF-SELECTION.  
  perform disp\_data.  
*\*&---------------------------------------------------------------------\**  
*\*& Form get-alv-fact*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM get\_alv\_fact USING P\_A p\_B P\_C .  
  
  
*\*  wa\_fieldcat-col\_pos = '1'.*  
  wa\_fieldcat-seltext\_l = P\_A.  
  wa\_fieldcat-fieldname = P\_B.  
  wa\_fieldcat-tabname = P\_C.  
*\*  wa\_fieldcat-emphasize = 'C611'.       ""color*  
*\*  wa\_fieldcat-no\_zero = 'X'.*  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
*\*  """""VBELN""""*  
*\*  wa\_fieldcat-col\_pos = '1'.*  
*\*  wa\_fieldcat-seltext\_l = text-001.*  
*\*  wa\_fieldcat-fieldname = cs\_vbeln.*  
*\*  wa\_fieldcat-tabname = cs\_tab.*  
*\*  wa\_fieldcat-emphasize = 'C611'.       ""color*  
*\*\*  wa\_fieldcat-no\_zero = 'X'.*  
*\*  append wa\_fieldcat to lt\_fieldcat.*  
*\*  clear wa\_fieldcat.*  
*\**  
*\*  """""""ERDAT"""""""*  
*\*  wa\_fieldcat-col\_pos = '2'.*  
*\*  wa\_fieldcat-seltext\_l = text-002.*  
*\*  wa\_fieldcat-fieldname = cs\_erdat.*  
*\*  wa\_fieldcat-tabname = cs\_tab.*  
*\*  wa\_fieldcat-emphasize = 'C711'.*  
*\*  append wa\_fieldcat to lt\_fieldcat.*  
*\*  clear wa\_fieldcat.*  
*\**  
*\*  """""""erzet""""""""*  
*\*  wa\_fieldcat-col\_pos = '3'.*  
*\*  wa\_fieldcat-seltext\_l = text-003.*  
*\*  wa\_fieldcat-fieldname = cs\_erzet.*  
*\*  wa\_fieldcat-tabname = cs\_tab.*  
*\*  wa\_fieldcat-emphasize = 'C511'.*  
*\*  append wa\_fieldcat to lt\_fieldcat.*  
*\*  clear wa\_fieldcat.*  
*\**  
*\*  """""""ernam""""""""""*  
*\*  wa\_fieldcat-col\_pos = '4'.*  
*\*  wa\_fieldcat-seltext\_l = text-004.*  
*\*  wa\_fieldcat-fieldname = cs\_ernam.*  
*\*  wa\_fieldcat-tabname = cs\_tab.*  
*\*  wa\_fieldcat-emphasize = 'C411'.*  
*\**  
*\*  append wa\_fieldcat to lt\_fieldcat.*  
*\*  clear wa\_fieldcat.*  
  
ENDFORM.  
*\*&---------------------------------------------------------------------\**  
*\*& Form get\_data*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM get\_data .  
  select vbeln erdat erzet ernam into table lt\_tab from vbak where vbeln in s\_vbeln.  
    IF sy-subrc EQ 0.  
    MESSAGE s001(ZPRAC\_message).  
  ELSE.  
    MESSAGE e000(zprac\_message).  
  ENDIF.  
  
ENDFORM.  
*\*&---------------------------------------------------------------------\**  
*\*& Form disp\_data*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM disp\_data .  
  
  ty\_layout-no\_colhead = 'X'. *""remove colheads,vlines nd hlines for grid*  
  ty\_layout-no\_vline = 'X'.  
  ty\_layout-no\_hline = 'X'.  
  
 CALL FUNCTION 'REUSE\_ALV\_GRID\_DISPLAY'  
 EXPORTING  
  I\_CALLBACK\_PROGRAM                = sy-repid  
  I\_CALLBACK\_TOP\_OF\_PAGE            = 'UST\_LOGO' *""subroutine*  
   I\_GRID\_TITLE                      = text-005  
   IS\_LAYOUT                         = TY\_LAYOUT   *"""remove colheads,vlines nd hlines for grid*  
   IT\_FIELDCAT                       = lt\_fieldcat  
   TABLES  
     t\_outtab                          = lt\_tab  
  EXCEPTIONS  
    PROGRAM\_ERROR                     = 1  
    OTHERS                            = 2  
           .  
 IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
 ENDIF.  
  
ENDFORM.  
  
form ust\_logo.  
  data: lt\_header type slis\_t\_listheader,  
        wa\_header type slis\_listheader.  
  
  wa\_header-typ = 'H'.  
  
  wa\_header-info = 'SAP ABAP'.  
  wa\_header-key = 'TRAINING'.  
  append wa\_header to lt\_header.  
  
  CALL FUNCTION 'REUSE\_ALV\_COMMENTARY\_WRITE'  
    EXPORTING  
      it\_list\_commentary       = lt\_header  
    I\_LOGO                   = 'ZPRAC\_OBJKEY'  
*\*     I\_END\_OF\_LIST\_GRID       =*  
*\*     I\_ALV\_FORM               =*  
            .  
  ENDFORM.

Include prog,

*\*&---------------------------------------------------------------------\**  
*\*& Include          ZPRAC\_ALVREPO\_TOP*  
*\*&---------------------------------------------------------------------\**  
  
  
TYPES: BEGIN OF ls\_tab,  
         vbeln TYPE vbeln\_va,    *"SALES DOC*  
         erdat TYPE  erdat,      *"Record created on*  
         erzet TYPE  erzet,        *"entry time*  
         ernam TYPE  ernam,        *"nameof person*  
       END OF ls\_tab.  
  
DATA : lt\_tab      TYPE STANDARD TABLE OF ls\_tab,  
       wa\_tab      TYPE ls\_tab,  
  
       lv\_vbeln    TYPE vbeln,  
  
       lt\_fieldcat TYPE slis\_T\_fieldcat\_alv,  
       wa\_fieldcat TYPE slis\_fieldcat\_alv.  
  
CONSTANTS : cs\_tab(10) type c value 'lt-tab',  
            cs\_vbeln(10) type c value 'vbeln',  
            cs\_erdat(10) type c value 'erdat',  
            cs\_erzet(10) type c value 'erzet',  
            cs\_ernam(10) type c value 'ernam'.

===========================================================================

ALV reports – Merge

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_ALV\_REPO\_GRID\_MERGE*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_ALV\_REPO\_GRID\_MERGE NO STANDARD PAGE HEADING LINE-COUNT 30(4) LINE-SIZE 400.  
  
include ZPRAC\_ALVMERGE\_TOP.  
  
SELECT-OPTIONS s\_vbeln for lv\_vbeln OBLIGATORY.  
  
INITIALIZATION.  
clear: lt\_tab, wa\_fieldcat, lt\_fieldcat.  
  
s\_vbeln-low = '0000000004'.  
s\_vbeln-high = '0000000123'.  
append s\_vbeln.  
  
at SELECTION-SCREEN on s\_vbeln.  
  select SINGLE vbeln into lv\_vbeln from vbak where vbeln in s\_vbeln.  
  
   START-OF-SELECTION.  
   PERFORM get\_merge\_data.  
  
   PERFORM get\_alv\_fcat.  
  
   END-OF-SELECTION.  
   PERFORM disp\_merge\_data.  
  
*\*&---------------------------------------------------------------------\**  
*\*& Form get\_merge\_data*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM get\_merge\_data .  
  select \* into table lt\_tab from vbak where vbeln in s\_vbeln.  
ENDFORM.  
*\*&---------------------------------------------------------------------\**  
*\*& Form get\_alv\_fcat*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM get\_alv\_fcat .  
  CALL FUNCTION 'REUSE\_ALV\_FIELDCATALOG\_MERGE'  
 EXPORTING  
   I\_PROGRAM\_NAME               = sy-repid  
     I\_INTERNAL\_TABNAME           = 'LT\_TAB'  
    I\_STRUCTURE\_NAME             = 'VBAK'  
*\*     I\_CLIENT\_NEVER\_DISPLAY       = 'X'*  
*\*     I\_INCLNAME                   =*  
*\*     I\_BYPASSING\_BUFFER           =*  
*\*     I\_BUFFER\_ACTIVE              =*  
    CHANGING  
      ct\_fieldcat                  = LT\_FIELDCAT  
   EXCEPTIONS  
     INCONSISTENT\_INTERFACE       = 1  
     PROGRAM\_ERROR                = 2  
     OTHERS                       = 3  
            .  
  IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
  ENDIF.  
  
ENDFORM.  
*\*&---------------------------------------------------------------------\**  
*\*& Form disp\_merge\_data*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM disp\_merge\_data .  
    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                  =*  
*\*       I\_BACKGROUND\_ID                   = ' '*  
*\*       I\_GRID\_TITLE                      =*  
*\*       I\_GRID\_SETTINGS                   =*  
*\*       IS\_LAYOUT                         =*  
     IT\_FIELDCAT                       = LT\_FIELDCAT  
*\*       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\_TAB  
*\*     EXCEPTIONS*  
*\*       PROGRAM\_ERROR                     = 1*  
*\*       OTHERS                            = 2*  
              .  
    IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
    ENDIF.  
  
ENDFORM.

Include prog,

*\*&---------------------------------------------------------------------\**  
*\*& Include          ZPRAC\_ALVMERGE\_TOP*  
*\*&---------------------------------------------------------------------\**  
  
  
DATA : lt\_tab      TYPE STANDARD TABLE OF VBAK,  
       lv\_vbeln    TYPE vbeln,  
       lt\_fieldcat TYPE slis\_T\_fieldcat\_alv,  
       wa\_fieldcat TYPE slis\_fieldcat\_alv.

==========================================================================

ALV reports – POPUP,

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_ALV\_REPO\_GRID\_POPUP*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_ALV\_REPO\_GRID\_POPUP NO STANDARD PAGE HEADING LINE-COUNT 30(2) LINE-SIZE 500.  
  
INCLUDE ZPRAC\_ALVREPOPUP\_TOP.  
  
select-options s\_vbeln for lv\_vbeln OBLIGATORY.  
  
INITIALIZATION.  
  
clear: lt\_tab, wa\_fieldcat, lt\_fieldcat.  
s\_vbeln-low = '0000000023'.  
s\_vbeln-high = '0000000032'.  
append s\_vbeln.  
  
at SELECTION-SCREEN on s\_vbeln.  
  
  select single vbeln into lv\_vbeln from vbak where vbeln in s\_vbeln.  
    IF sy-subrc EQ 0.  
    MESSAGE s001(ZPRAC\_message).  
  ELSE.  
    MESSAGE e000(zprac\_message).  
  ENDIF.  
  
  START-OF-SELECTION.  
  PERFORM get\_data.  
  
*\*  PERFORM get\_alv\_fact.*  
  
PERFORM get\_alv\_fact USING text-001 'VBELN' text-005 .  
PERFORM get\_alv\_fact USING text-002 'ERDAT' text-005.  
PERFORM get\_alv\_fact USING text-003 'ERZET' text-005.  
PERFORM get\_alv\_fact USING text-004 'ERNAM' text-005.  
  
  
  END-OF-SELECTION.  
  perform disp\_data.  
*\*&---------------------------------------------------------------------\**  
*\*& Form get-alv-fact*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM get\_alv\_fact USING P\_A p\_B P\_C .  
  
  
*\*  wa\_fieldcat-col\_pos = '1'.*  
  wa\_fieldcat-seltext\_l = P\_A.  
  wa\_fieldcat-fieldname = P\_B.  
  wa\_fieldcat-tabname = P\_C.  
*\*  wa\_fieldcat-emphasize = 'C611'.       ""color*  
*\*  wa\_fieldcat-no\_zero = 'X'.*  
  append wa\_fieldcat to lt\_fieldcat.  
  clear wa\_fieldcat.  
  
*\*  """""VBELN""""*  
*\*  wa\_fieldcat-col\_pos = '1'.*  
*\*  wa\_fieldcat-seltext\_l = text-001.*  
*\*  wa\_fieldcat-fieldname = cs\_vbeln.*  
*\*  wa\_fieldcat-tabname = cs\_tab.*  
*\*  wa\_fieldcat-emphasize = 'C611'.       ""color*  
*\*\*  wa\_fieldcat-no\_zero = 'X'.*  
*\*  append wa\_fieldcat to lt\_fieldcat.*  
*\*  clear wa\_fieldcat.*  
*\**  
*\*  """""""ERDAT"""""""*  
*\*  wa\_fieldcat-col\_pos = '2'.*  
*\*  wa\_fieldcat-seltext\_l = text-002.*  
*\*  wa\_fieldcat-fieldname = cs\_erdat.*  
*\*  wa\_fieldcat-tabname = cs\_tab.*  
*\*  wa\_fieldcat-emphasize = 'C711'.*  
*\*  append wa\_fieldcat to lt\_fieldcat.*  
*\*  clear wa\_fieldcat.*  
*\**  
*\*  """""""erzet""""""""*  
*\*  wa\_fieldcat-col\_pos = '3'.*  
*\*  wa\_fieldcat-seltext\_l = text-003.*  
*\*  wa\_fieldcat-fieldname = cs\_erzet.*  
*\*  wa\_fieldcat-tabname = cs\_tab.*  
*\*  wa\_fieldcat-emphasize = 'C511'.*  
*\*  append wa\_fieldcat to lt\_fieldcat.*  
*\*  clear wa\_fieldcat.*  
*\**  
*\*  """""""ernam""""""""""*  
*\*  wa\_fieldcat-col\_pos = '4'.*  
*\*  wa\_fieldcat-seltext\_l = text-004.*  
*\*  wa\_fieldcat-fieldname = cs\_ernam.*  
*\*  wa\_fieldcat-tabname = cs\_tab.*  
*\*  wa\_fieldcat-emphasize = 'C411'.*  
*\**  
*\*  append wa\_fieldcat to lt\_fieldcat.*  
*\*  clear wa\_fieldcat.*  
  
ENDFORM.  
*\*&---------------------------------------------------------------------\**  
*\*& Form get\_data*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM get\_data .  
  select vbeln erdat erzet ernam into table lt\_tab from vbak where vbeln in s\_vbeln.  
    IF sy-subrc EQ 0.  
    MESSAGE s001(ZPRAC\_message).  
  ELSE.  
    MESSAGE e000(zprac\_message).  
  ENDIF.  
  
ENDFORM.  
*\*&---------------------------------------------------------------------\**  
*\*& Form disp\_data*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM disp\_data .  
  
 CALL FUNCTION 'REUSE\_ALV\_POPUP\_TO\_SELECT'  
   EXPORTING  
   I\_TITLE                       = 'SALES DOCUMENT'  
*\*    I\_SELECTION                   = 'X'*  
*\*    I\_ALLOW\_NO\_SELECTION          =*  
  I\_ZEBRA                       = 'X'  
*\*    I\_SCREEN\_START\_COLUMN         = 0*  
*\*    I\_SCREEN\_START\_LINE           = 0*  
*\*    I\_SCREEN\_END\_COLUMN           = 0*  
*\*    I\_SCREEN\_END\_LINE             = 0*  
*\*    I\_CHECKBOX\_FIELDNAME          =*  
*\*    I\_LINEMARK\_FIELDNAME          =*  
*\*    I\_SCROLL\_TO\_SEL\_LINE          = 'X'*  
     i\_tabname                     = 'LT\_TAB'  
   I\_STRUCTURE\_NAME              = 'LS\_TAB'  
  IT\_FIELDCAT                   = LT\_FIELDCAT  
*\*    IT\_EXCLUDING                  =*  
   I\_CALLBACK\_PROGRAM            = SY-REPID  
*\*    I\_CALLBACK\_USER\_COMMAND       =*  
*\*    IS\_PRIVATE                    =*  
*\*  IMPORTING*  
*\*    ES\_SELFIELD                   =*  
*\*    E\_EXIT                        =*  
   tables  
     t\_outtab                      = LT\_TAB  
  EXCEPTIONS  
    PROGRAM\_ERROR                 = 1  
    OTHERS                        = 2  
           .  
 IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
 ENDIF.  
  
  
ENDFORM.

Include prog,

*\*&---------------------------------------------------------------------\**  
*\*& Include          ZPRAC\_ALVREPOPUP\_TOP*  
*\*&---------------------------------------------------------------------\**  
  
TYPES: BEGIN OF ls\_tab,  
         vbeln TYPE vbeln\_va,    *"SALES DOC*  
         erdat TYPE  erdat,      *"Record created on*  
         erzet TYPE  erzet,        *"entry time*  
         ernam TYPE  ernam,        *"nameof person*  
       END OF ls\_tab.  
  
DATA : lt\_tab      TYPE STANDARD TABLE OF ls\_tab,  
       wa\_tab      TYPE ls\_tab,  
  
       lv\_vbeln    TYPE vbeln,  
  
       lt\_fieldcat TYPE slis\_T\_fieldcat\_alv,  
       wa\_fieldcat TYPE slis\_fieldcat\_alv.  
  
CONSTANTS : cs\_tab(10) type c value 'lt-tab',  
            cs\_vbeln(10) type c value 'vbeln',  
            cs\_erdat(10) type c value 'erdat',  
            cs\_erzet(10) type c value 'erzet',  
            cs\_ernam(10) type c value 'ernam'.

=========================================================================

ALV reports – BLOCK,

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_ALV\_BLOCK\_REPO*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_ALV\_BLOCK\_REPO NO STANDARD PAGE HEADING LINE-COUNT 30(5) LINE-SIZE 400.  
  
INCLUDE zprac\_alv\_block\_repo\_top.  
  
data lv\_vbeln type vbeln\_va.  
select-OPTIONS s\_vbeln for lv\_vbeln.  
  
START-OF-SELECTION.  
perform get\_data\_tab.  
  
perform create\_fcat\_vbak.  
perform create\_fcat\_vbap.  
  
perform block\_list\_init.  
perform block\_list\_append\_vbak.  
perform block\_list\_append\_vbap.  
perform block\_list\_display.  
  
*\*&---------------------------------------------------------------------\**  
*\*& Form get\_data\_tab*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM get\_data\_tab .  
  select vbeln erdat erzet ernam from vbak into table lt\_tab  
    where vbeln in s\_vbeln.  
  
  if not lt\_tab[] is INITIAL.  
  
    select vbeln posnr matnr matwa from vbap into table lt\_tab2 where  
      vbeln in s\_vbeln.  
   ENDIF.  
  
ENDFORM.  
*\*&---------------------------------------------------------------------\**  
*\*& Form create\_fcat\_vbak*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM create\_fcat\_vbak .  
  
  w\_fcat\_vbak-col\_pos = '1'.  
  w\_fcat\_vbak-fieldname = 'vbeln'.  
  w\_fcat\_vbak-ref\_tabname = 'vbak'.  
  w\_fcat\_vbak-ref\_fieldname = 'vbeln'.  
  append w\_fcat\_vbak to t\_fcat\_vbak.  
  clear w\_fcat\_vbak.  
  
  w\_fcat\_vbak-col\_pos = '2'.  
  w\_fcat\_vbak-fieldname = 'erdat'.  
  w\_fcat\_vbak-ref\_tabname = 'vbak'.  
  w\_fcat\_vbak-ref\_fieldname = 'erdat'.  
  append w\_fcat\_vbak to t\_fcat\_vbak.  
  clear w\_fcat\_vbak.  
  
  w\_fcat\_vbak-col\_pos = '3'.  
  w\_fcat\_vbak-fieldname = 'erzet'.  
  w\_fcat\_vbak-ref\_tabname = 'vbak'.  
  w\_fcat\_vbak-ref\_fieldname = 'erzet'.  
  append w\_fcat\_vbak to t\_fcat\_vbak.  
  clear w\_fcat\_vbak.  
  
  w\_fcat\_vbak-col\_pos = '4'.  
  w\_fcat\_vbak-fieldname = 'ernam'.  
  w\_fcat\_vbak-ref\_tabname = 'vbak'.  
  w\_fcat\_vbak-ref\_fieldname = 'ernam'.  
  append w\_fcat\_vbak to t\_fcat\_vbak.  
  clear w\_fcat\_vbak.  
  
ENDFORM.  
*\*&---------------------------------------------------------------------\**  
*\*& Form create\_fcat\_vbap*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM create\_fcat\_vbap .  
  
  w\_fcat\_vbap-col\_pos = '1'.  
  w\_fcat\_vbap-fieldname = 'vbeln'.  
  w\_fcat\_vbap-ref\_tabname = 'vbap'.  
  w\_fcat\_vbap-ref\_fieldname = 'vbeln'.  
  append w\_fcat\_vbap to t\_fcat\_vbap.  
  clear w\_fcat\_vbap.  
  
  w\_fcat\_vbap-col\_pos = '2'.  
  w\_fcat\_vbap-fieldname = 'posnr'.  
  w\_fcat\_vbap-ref\_tabname = 'vbap'.  
  w\_fcat\_vbap-ref\_fieldname = 'posnr'.  
  append w\_fcat\_vbap to t\_fcat\_vbap.  
  clear w\_fcat\_vbap.  
  
  w\_fcat\_vbap-col\_pos = '3'.  
  w\_fcat\_vbap-fieldname = 'matnr'.  
  w\_fcat\_vbap-ref\_tabname = 'vbap'.  
  w\_fcat\_vbap-ref\_fieldname = 'matnr'.  
  append w\_fcat\_vbap to t\_fcat\_vbap.  
  clear w\_fcat\_vbap.  
  
  w\_fcat\_vbap-col\_pos = '4'.  
  w\_fcat\_vbap-fieldname = 'matwa'.  
  w\_fcat\_vbap-ref\_tabname = 'vbap'.  
  w\_fcat\_vbap-ref\_fieldname = 'matwa'.  
  append w\_fcat\_vbap to t\_fcat\_vbap.  
  clear w\_fcat\_vbap.  
  
ENDFORM.  
*\*&---------------------------------------------------------------------\**  
*\*& Form block\_list\_init*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM block\_list\_init .  
  CALL FUNCTION 'REUSE\_ALV\_BLOCK\_LIST\_INIT'  
    EXPORTING  
      i\_callback\_program             = sy-repid  
*\*     I\_CALLBACK\_PF\_STATUS\_SET       = ' '*  
*\*     I\_CALLBACK\_USER\_COMMAND        = ' '*  
*\*     IT\_EXCLUDING                   =*  
            .  
  
ENDFORM.  
*\*&---------------------------------------------------------------------\**  
*\*& Form block\_list\_append\_vbak*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM block\_list\_append\_vbak .  
  CALL FUNCTION 'REUSE\_ALV\_BLOCK\_LIST\_APPEND'  
    EXPORTING  
      is\_layout                        = s\_layout  
      it\_fieldcat                      = t\_fcat\_vbak  
      i\_tabname                        = 'lt\_tab'  
      it\_events                        =  s\_events  
*\*     IT\_SORT                          =*  
*\*     I\_TEXT                           = ' '*  
    TABLES  
      t\_outtab                         =  lt\_tab  
*\*   EXCEPTIONS*  
*\*     PROGRAM\_ERROR                    = 1*  
*\*     MAXIMUM\_OF\_APPENDS\_REACHED       = 2*  
*\*     OTHERS                           = 3*  
            .  
  IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
  ENDIF.  
  
ENDFORM.  
*\*&---------------------------------------------------------------------\**  
*\*& Form block\_list\_append\_vbap*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM block\_list\_append\_vbap .  
  
  CALL FUNCTION 'REUSE\_ALV\_BLOCK\_LIST\_APPEND'  
    EXPORTING  
      is\_layout                        = s\_layout  
      it\_fieldcat                      = t\_fcat\_vbap  
      i\_tabname                        = 'lt\_tab2'  
      it\_events                        = s\_events  
*\*     IT\_SORT                          =*  
*\*     I\_TEXT                           = ' '*  
    TABLES  
      t\_outtab                         = lt\_tab2  
*\*   EXCEPTIONS*  
*\*     PROGRAM\_ERROR                    = 1*  
*\*     MAXIMUM\_OF\_APPENDS\_REACHED       = 2*  
*\*     OTHERS                           = 3*  
            .  
  IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
  ENDIF.  
  
ENDFORM.  
*\*&---------------------------------------------------------------------\**  
*\*& Form block\_list\_display*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM block\_list\_display .  
  CALL FUNCTION 'REUSE\_ALV\_BLOCK\_LIST\_DISPLAY'  
*\*   EXPORTING*  
*\*     I\_INTERFACE\_CHECK             = ' '*  
*\*     IS\_PRINT                      =*  
*\*     I\_SCREEN\_START\_COLUMN         = 0*  
*\*     I\_SCREEN\_START\_LINE           = 0*  
*\*     I\_SCREEN\_END\_COLUMN           = 0*  
*\*     I\_SCREEN\_END\_LINE             = 0*  
*\*   IMPORTING*  
*\*     E\_EXIT\_CAUSED\_BY\_CALLER       =*  
*\*     ES\_EXIT\_CAUSED\_BY\_USER        =*  
*\*   EXCEPTIONS*  
*\*     PROGRAM\_ERROR                 = 1*  
*\*     OTHERS                        = 2*  
            .  
  IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
  ENDIF.  
  
ENDFORM.

Include prog,

*\*&---------------------------------------------------------------------\**  
*\*& Include          ZPRAC\_ALV\_BLOCK\_REPO\_TOP*  
*\*&---------------------------------------------------------------------\**  
  
TYPES: BEGIN OF ls\_tab,  
         vbeln TYPE vbeln\_va,    *"SALES DOC*  
         erdat TYPE  erdat,      *"Record created on*  
         erzet TYPE  erzet,        *"entry time*  
         ernam TYPE  ernam,        *"nameof person*  
       END OF ls\_tab.  
  
DATA: lt\_tab type STANDARD TABLE OF ls\_tab,  
      wa\_tab type ls\_tab,  
      t\_fcat\_vbak type slis\_t\_fieldcat\_alv,  
      w\_fcat\_vbak type slis\_fieldcat\_alv.  
  
TYPES: BEGIN OF ls\_tab2,  
  VBELN type  VBELN\_VA,  
POSNR type  POSNR\_VA,  
MATNR type  MATNR,  
MATWA type  MATWA,  
  end of ls\_tab2.  
  
DATA: lt\_tab2 type STANDARD TABLE OF ls\_tab2,  
      wa\_tab2 type ls\_tab2,  
      t\_fcat\_vbap type slis\_t\_fieldcat\_alv,  
      w\_fcat\_vbap type slis\_fieldcat\_alv.  
  
DATA: s\_events type slis\_t\_event,  
      s\_layout type slis\_layout\_alv.

======================================================================

ALV hireseq repport,

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_ALV\_REPO\_HIERSEQ*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT zprac\_alv\_repo\_hierseq NO STANDARD PAGE HEADING LINE-COUNT 30(5) LINE-SIZE 400.  
  
TABLES vbak.  
  
INCLUDE zprac\_alv\_hireseq\_repo\_top.  
  
SELECT-OPTIONS s\_vbeln FOR vbak-vbeln.  
  
START-OF-SELECTION.  
  PERFORM get\_data.  
  
  PERFORM create\_fcat.  
  
  PERFORM hierseq\_display.  
  
  
*\*&---------------------------------------------------------------------\**  
*\*& Form create\_fcat\_vbak*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM create\_fcat .  
  
  w\_fcat-col\_pos = '1'.  
  w\_fcat-fieldname = 'vbeln'.  
  w\_fcat-tabname = 'lt\_tab'.  
  w\_fcat-ref\_tabname = 'vbak'.  
  w\_fcat-ref\_fieldname = 'vbeln'.  
  APPEND w\_fcat TO t\_fcat.  
  CLEAR w\_fcat.  
  
  w\_fcat-col\_pos = '2'.  
  w\_fcat-fieldname = 'erdat'.  
  w\_fcat-tabname = 'lt\_tab'.  
  w\_fcat-ref\_tabname = 'vbak'.  
  w\_fcat-ref\_fieldname = 'erdat'.  
  APPEND w\_fcat TO t\_fcat.  
  CLEAR w\_fcat.  
  
  w\_fcat-col\_pos = '3'.  
  w\_fcat-fieldname = 'erzet'.  
  w\_fcat-tabname = 'lt\_tab'.  
  w\_fcat-ref\_tabname = 'vbak'.  
  w\_fcat-ref\_fieldname = 'erzet'.  
  APPEND w\_fcat TO t\_fcat.  
  CLEAR w\_fcat.  
  
  w\_fcat-col\_pos = '4'.  
  w\_fcat-fieldname = 'ernam'.  
  w\_fcat-tabname = 'lt\_tab'.  
  w\_fcat-ref\_tabname = 'vbak'.  
  w\_fcat-ref\_fieldname = 'ernam'.  
  APPEND w\_fcat TO t\_fcat.  
  CLEAR w\_fcat.  
  
  
  w\_fcat-col\_pos = '1'.  
  w\_fcat-fieldname = 'vbeln'.  
  w\_fcat-tabname = 'lt\_tab2'.  
  w\_fcat-ref\_tabname = 'vbap'.  
  w\_fcat-ref\_fieldname = 'vbeln'.  
  APPEND w\_fcat TO t\_fcat.  
  CLEAR w\_fcat.  
  
  w\_fcat-col\_pos = '2'.  
  w\_fcat-fieldname = 'posnr'.  
  w\_fcat-tabname = 'lt\_tab2'.  
  w\_fcat-ref\_tabname = 'vbap'.  
  w\_fcat-ref\_fieldname = 'posnr'.  
  APPEND w\_fcat TO t\_fcat.  
  CLEAR w\_fcat.  
  
  w\_fcat-col\_pos = '3'.  
  w\_fcat-fieldname = 'matnr'.  
  w\_fcat-tabname = 'lt\_tab2'.  
  w\_fcat-ref\_tabname = 'vbap'.  
  w\_fcat-ref\_fieldname = 'matnr'.  
  APPEND w\_fcat TO t\_fcat.  
  CLEAR w\_fcat.  
  
  w\_fcat-col\_pos = '4'.  
  w\_fcat-fieldname = 'matwa'.  
  w\_fcat-tabname = 'lt\_tab2'.  
  w\_fcat-ref\_tabname = 'vbap'.  
  w\_fcat-ref\_fieldname = 'matwa'.  
  APPEND w\_fcat TO t\_fcat.  
  CLEAR w\_fcat.  
  
ENDFORM.  
*\*&---------------------------------------------------------------------\**  
*\*& Form hierseq\_display*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM hierseq\_display .  
  lt\_key-header01 = 'vbeln'.  
  s\_layout-expand\_fieldname = 'UST'.  
  IF sy-subrc EQ 0.  
    CALL FUNCTION 'REUSE\_ALV\_HIERSEQ\_LIST\_DISPLAY'  
      EXPORTING  
*\*       I\_INTERFACE\_CHECK  = ' '*  
        i\_callback\_program = sy-repid  
*\*       I\_CALLBACK\_PF\_STATUS\_SET       = ' '*  
*\*       I\_CALLBACK\_USER\_COMMAND        = ' '*  
        is\_layout          = s\_layout  
       IT\_FIELDCAT        = t\_fcat  
*\*       IT\_EXCLUDING       =*  
*\*       IT\_SPECIAL\_GROUPS  =*  
*\*       IT\_SORT            =*  
*\*       IT\_FILTER          =*  
*\*       IS\_SEL\_HIDE        =*  
*\*       I\_SCREEN\_START\_COLUMN          = 0*  
*\*       I\_SCREEN\_START\_LINE            = 0*  
*\*       I\_SCREEN\_END\_COLUMN            = 0*  
*\*       I\_SCREEN\_END\_LINE  = 0*  
*\*       I\_DEFAULT          = 'X'*  
*\*       I\_SAVE             = ' '*  
*\*       IS\_VARIANT         =*  
*\*       IT\_EVENTS          =*  
*\*       IT\_EVENT\_EXIT      =*  
        i\_tabname\_header   = 'lt\_tab'  
        i\_tabname\_item     = 'lt\_tab2'  
*\*       I\_STRUCTURE\_NAME\_HEADER        =*  
*\*       I\_STRUCTURE\_NAME\_ITEM          =*  
        is\_keyinfo         = lt\_key  
*\*       IS\_PRINT           =*  
*\*       IS\_REPREP\_ID       =*  
*\*       IS\_HIERSEQ\_SETTINGS            =*  
*\*       I\_BYPASSING\_BUFFER =*  
*\*       I\_BUFFER\_ACTIVE    =*  
*\*       IR\_SALV\_HIERSEQ\_ADAPTER        =*  
*\*       IT\_EXCEPT\_QINFO    =*  
*\*       I\_SUPPRESS\_EMPTY\_DATA          = ABAP\_FALSE*  
*\*       I\_LIST\_DISPLAY\_ONLY            = ABAP\_FALSE*  
*\*       O\_COMMON\_HUB       =*  
*\*     IMPORTING*  
*\*       E\_EXIT\_CAUSED\_BY\_CALLER        =*  
*\*       ES\_EXIT\_CAUSED\_BY\_USER         =*  
      TABLES  
        t\_outtab\_header    = lt\_tab  
        t\_outtab\_item      = lt\_tab2  
      EXCEPTIONS  
        program\_error      = 1  
        OTHERS             = 2.  
    IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
    ENDIF.  
  ELSE.  
    WRITE ' no records found '.  
  ENDIF.  
ENDFORM.  
*\*&---------------------------------------------------------------------\**  
*\*& Form get\_data*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM get\_data .  
  SELECT vbeln  
    erdat erzet ernam FROM vbak  
    INTO CORRESPONDING FIELDS OF TABLE lt\_tab  
    WHERE vbeln IN s\_vbeln.  
  
  SELECT vbeln posnr matnr matwa FROM vbap  
    INTO TABLE lt\_tab2 WHERE  
    vbeln IN s\_vbeln.  
ENDFORM.

Include prog,

*\*&---------------------------------------------------------------------\**  
*\*& Include          ZPRAC\_ALV\_HIRESEQ\_REPO\_TOP*  
*\*&---------------------------------------------------------------------\**  
  
TYPES: BEGIN OF ls\_tab,  
  UST(1),  
  vbeln type vbeln\_va,  
  erdat type erdat,  
  erzet type erzet,  
  ernam type ernam,  
  END OF ls\_tab.  
DAta: lt\_tab type table of ls\_tab,  
      wa\_tab type ls\_tab.  
  
TYPES: BEGIN OF ls\_tab2,  
  vbeln type vbeln\_va,  
  posnr type posnr\_va,  
  matnr type matnr,  
  matwa type matwa,  
  END OF ls\_tab2.  
DAta: lt\_tab2 type table of ls\_tab2,  
      wa\_tab2 type ls\_tab2.  
  
DATA: t\_fcat type slis\_t\_fieldcat\_alv,  
      w\_fcat LIKE LINE OF t\_fcat,  
  
      lt\_key type slis\_keyinfo\_alv,  
  
      s\_layout type slis\_layout\_alv.

=========================================================================

Selection screen,

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_SELECTIONSCREENS*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_SELECTIONSCREENS NO STANDARD PAGE HEADING LINE-COUNT 20(3) LINE-SIZE 500.  
  
LOAD-OF-PROGRAM.  
  
  INCLUDE zprac\_interuserepo\_include.  
  
selection-screen : begin of block b1 with frame TITLE text-001.  
  
  SELECT-OPTIONS s\_lifnr FOR wa\_tab-lifnr OBLIGATORY.  
  
SELECTION-SCREEN end of block b1.  
  
SELECTION-SCREEN : begin of block b2 with frame title text-001.  
  PARAMETERS : r1 radiobutton group rad1,  
               r2 RADIOBUTTON GROUP rad1,  
               r3 RADIOBUTTON GROUP rad1.  
  
*\*               r4 RADIOBUTTON GROUP rad2,*  
*\*               r5 RADIOBUTTON GROUP rad2.*  
  
SELECTION-SCREEN uline.  
SELECTION-SCREEN skip 2 .  
SELECTION-SCREEN COMMENT /79(40) lv\_commm.  
SELECTION-SCREEN skip 3.  
SELECTION-SCREEN uline.  
SELECTION-SCREEN end of block b2.  
  
 SELECTION-SCREEN : begin of block b3 with frame title text-001.  
  PARAMETERS : c1 AS CHECKBOX,  
            c2 AS CHECKBOX.  
SELECTION-SCREEN end of block b3.  
  
INITIALIZATION. *""to clear, default one, append*  
  
lv\_commm = 'UST HYD'.  
  CLEAR: wa\_Tab, lt\_Tab.  
  
  s\_lifnr-low ='1'.  
  s\_lifnr-high = '100'.  
  APPEND s\_lifnr.  
  
AT SELECTION-SCREEN ON s\_lifnr.  
  
  SELECT SINGLE lifnr INTO lv\_lifnr FROM lfa1 WHERE lifnr IN s\_lifnr.  
  
  IF sy-subrc EQ 0.  
    MESSAGE s001(ZPRAC\_message).  
  ELSE.  
    MESSAGE e000(zprac\_message).  
  ENDIF.  
  
START-OF-SELECTION.  
  set pf-status 'ZPRAC\_PFS'.  
  
  PERFORM get\_report\_data.  
  
END-OF-SELECTION.  
  PERFORM disp\_report\_data.  
  
  DATA lv\_coun TYPE i. *"""""TO SKIP IF THERE IS ANY SPACE FOR PAGE LIMIT THEN SKIP*  
  lv\_coun = sy-linct - sy-linno.  
  SKIP lv\_coun.  
  
TOP-OF-PAGE.  
  ULINE.  
  WRITE:/ 'sno',  
        10 'city',  
        20 'name',  
        30 'pstl2',  
        40 'pstlz',  
        50 'regio'.  
  ULINE.  
  
END-OF-PAGE.  
  WRITE:/  sy-pagno,  
            sy-datum,  
            sy-uzeit.  
  
at user-command.  
  
  case sy-ucomm.  
    when 'INFO'.  
      write ' UST INFO '.  
    when 'VIEW'.  
      write ' UST VIEW INFO'.  
ENDCASE.  
  
  
*\*&---------------------------------------------------------------------\**  
*\*& Form get\_report\_data*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM get\_report\_data .  
  SELECT lifnr land1 name1 pstl2 pstlz regio  
     INTO TABLE lt\_Tab  
    FROM lfa1  
    WHERE lifnr IN s\_lifnr.  
  IF sy-subrc EQ 0.  
    WRITE:/'RECORDS ARE FOUND'.  
  ELSE.  
    WRITE:/'RECORDS ARE NOT FOUND'.  
  ENDIF.  
ENDFORM.  
*\*&---------------------------------------------------------------------\**  
*\*& Form disp\_report\_data*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM disp\_report\_data .  
  LOOP AT lt\_Tab INTO wa\_tab.  
    WRITE :/  wa\_tab-lifnr HOTSPOT,  
              wa\_tab-land1,  
              wa\_tab-name1,  
              wa\_tab-pstl2,  
              wa\_tab-pstlz,  
              wa\_tab-regio.  
  
    HIDE wa\_tab-lifnr.  
  
    ENDLOOP.  
ENDFORM.

Include prog,

*\*&---------------------------------------------------------------------\**  
*\*& Include          ZPRAC\_INTERUSEREPO\_INCLUDE*  
*\*&---------------------------------------------------------------------\**  
TYPES: BEGIN OF ls\_tab,  
         lifnr TYPE lifnr,  
         land1 TYPE land1,  
         name1 TYPE name1\_gp,  
         pstl2 TYPE pstl2,  
         pstlz TYPE pstlz,  
         regio TYPE regio,  
       END OF ls\_tab.  
  
DATA : lt\_tab  TYPE STANDARD TABLE OF ls\_tab,  
       wa\_tab  TYPE ls\_tab.  
  
  
DATA lv\_lifnr TYPE lifnr.

========================================================================

Selection screen output,

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_SELECTIONSCREEN\_OP*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_SELECTIONSCREEN\_OP  
NO STANDARD PAGE HEADING LINE-COUNT 20(3) LINE-SIZE 500.  
  
LOAD-OF-PROGRAM.  
  
  INCLUDE zprac\_interuserepo\_include.  
  
  selection-screen : begin of block b1 with frame TITLE text-001.  
  
  SELECT-OPTIONS s\_lifnr FOR wa\_tab-lifnr OBLIGATORY.  
  
SELECTION-SCREEN end of block b1.  
  
SELECTION-SCREEN : begin of block b2 with frame title text-001.  
  PARAMETERS : r1 radiobutton group rad1,  
               r2 RADIOBUTTON GROUP rad1,  
               r3 RADIOBUTTON GROUP rad1.  
  
*\*               r4 RADIOBUTTON GROUP rad2,*  
*\*               r5 RADIOBUTTON GROUP rad2.*  
  
SELECTION-SCREEN uline.  
SELECTION-SCREEN skip 2 .  
SELECTION-SCREEN COMMENT /79(40) lv\_commm.  
SELECTION-SCREEN skip 3.  
SELECTION-SCREEN uline.  
SELECTION-SCREEN end of block b2.  
  
 SELECTION-SCREEN : begin of block b3 with frame title text-001.  
  PARAMETERS : c1 AS CHECKBOX,  
            c2 AS CHECKBOX.  
SELECTION-SCREEN end of block b3.  
  
  
INITIALIZATION. *""to clear, default one, append*  
  
  CLEAR: wa\_Tab, lt\_Tab.  
  
  s\_lifnr-low ='1'.  
  s\_lifnr-high = '100'.  
  APPEND s\_lifnr.  
  
AT SELECTION-SCREEN ON s\_lifnr.  
  
  SELECT SINGLE lifnr INTO lv\_lifnr FROM lfa1 WHERE lifnr IN s\_lifnr.  
  
  IF sy-subrc EQ 0.  
    MESSAGE s001(ZPRAC\_message).  
  ELSE.  
    MESSAGE e000(zprac\_message).  
  ENDIF.  
  
START-OF-SELECTION.  
  set pf-status 'ZPRAC\_PFS'.  
  
  PERFORM get\_report\_data.  
  
END-OF-SELECTION.  
  PERFORM disp\_report\_data.  
  
  DATA lv\_coun TYPE i. *"""""TO SKIP IF THERE IS ANY SPACE FOR PAGE LIMIT THEN SKIP*  
  lv\_coun = sy-linct - sy-linno.  
  SKIP lv\_coun.  
  
TOP-OF-PAGE.  
  ULINE.  
  WRITE:/ 'sno',  
        10 'city',  
        20 'name',  
        30 'pstl2',  
        40 'pstlz',  
        50 'regio'.  
  ULINE.  
  
END-OF-PAGE.  
  WRITE:/  sy-pagno,  
            sy-datum,  
            sy-uzeit.  
  
at user-command.  
  
  case sy-ucomm.  
    when 'INFO'.  
      write ' UST INFO '.  
    when 'VIEW'.  
      write ' UST VIEW INFO'.  
ENDCASE.  
  
  
*\*&---------------------------------------------------------------------\**  
*\*& Form get\_report\_data*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM get\_report\_data .  
  SELECT lifnr land1 name1 pstl2 pstlz regio  
     INTO TABLE lt\_Tab  
    FROM lfa1  
    WHERE lifnr IN s\_lifnr.  
  IF sy-subrc EQ 0.  
    WRITE:/'RECORDS ARE FOUND'.  
  ELSE.  
    WRITE:/'RECORDS ARE NOT FOUND'.  
  ENDIF.  
ENDFORM.  
*\*&---------------------------------------------------------------------\**  
*\*& Form disp\_report\_data*  
*\*&---------------------------------------------------------------------\**  
*\*& text*  
*\*&---------------------------------------------------------------------\**  
*\*& -->  p1        text*  
*\*& <--  p2        text*  
*\*&---------------------------------------------------------------------\**  
FORM disp\_report\_data .  
  LOOP AT lt\_Tab INTO wa\_tab.  
    WRITE :/  wa\_tab-lifnr HOTSPOT,  
              wa\_tab-land1,  
              wa\_tab-name1,  
              wa\_tab-pstl2,  
              wa\_tab-pstlz,  
              wa\_tab-regio.  
  
    HIDE wa\_tab-lifnr.  
  
    ENDLOOP.  
ENDFORM.

Include prog,

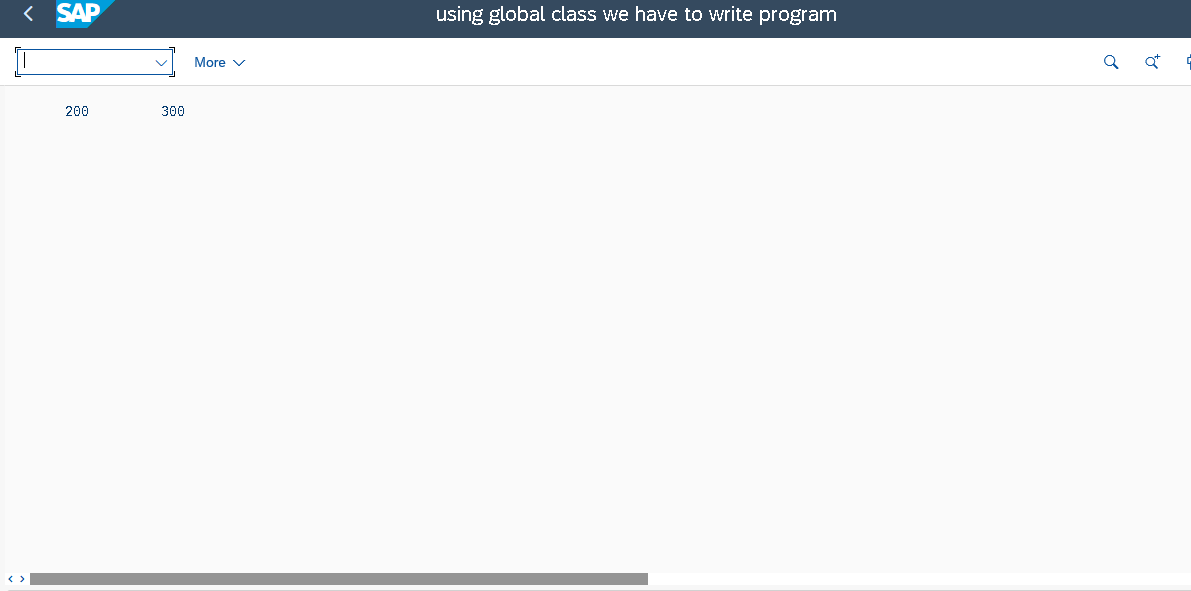
*\*&---------------------------------------------------------------------\**  
*\*& Include          ZPRAC\_INTERUSEREPO\_INCLUDE*  
*\*&---------------------------------------------------------------------\**  
TYPES: BEGIN OF ls\_tab,  
         lifnr TYPE lifnr,  
         land1 TYPE land1,  
         name1 TYPE name1\_gp,  
         pstl2 TYPE pstl2,  
         pstlz TYPE pstlz,  
         regio TYPE regio,  
       END OF ls\_tab.  
  
DATA : lt\_tab  TYPE STANDARD TABLE OF ls\_tab,  
       wa\_tab  TYPE ls\_tab.  
  
  
DATA lv\_lifnr TYPE lifnr.

==========================================================================

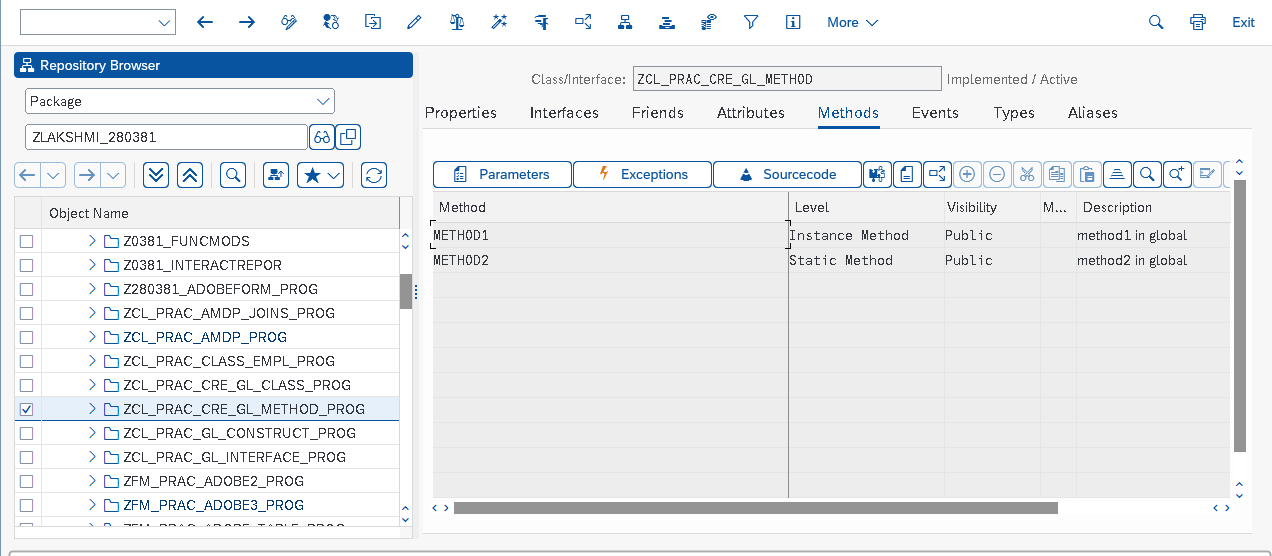
LOCAL CLASSES

*\*&---------------------------------------------------------------------\**  
*\*& Report ZCL\_PRAC\_CRE\_GL\_CLASS\_PROG*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZCL\_PRAC\_CRE\_GL\_CLASS\_PROG NO STANDARD PAGE HEADING LINE-COUNT 20(3) LINE-SIZE 300.  
  
*\*DATA obj TYPE REF TO ZCL\_PRAC\_CRE\_GL\_CLASS.         """""""OBJECT CREATION*  
*\*CREATE OBJECT obj.*  
*\*START-OF-SELECTION.*  
*\*WRITE:/ obj->A,                                     """"""""INSTANCE VARIABLE*  
*\*        ZCL\_PRAC\_CRE\_GL\_CLASS=>B.                   """"""""STATIC VARIABLE*  
  
  
CLASS cl\_prac DEFINITION.  
  PUBLIC SECTION.  
    DATA v1 TYPE i value 200.  
    class-DATA v2 type i value 300.  
ENDCLASS.  
  
DATA obj type REF TO cl\_prac.  
CREATE OBJECT obj.  
  
START-OF-SELECTION.  
WRITE:/ obj->v1,  
        cl\_prac=>v2.

Output:



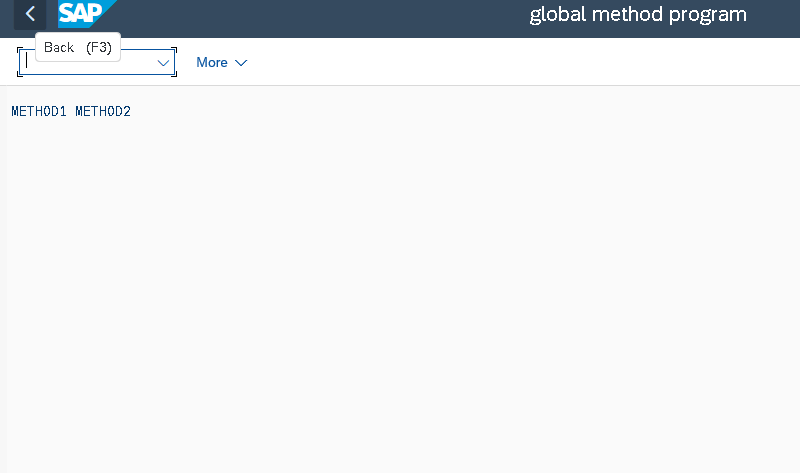
GLOBAL METHODS:

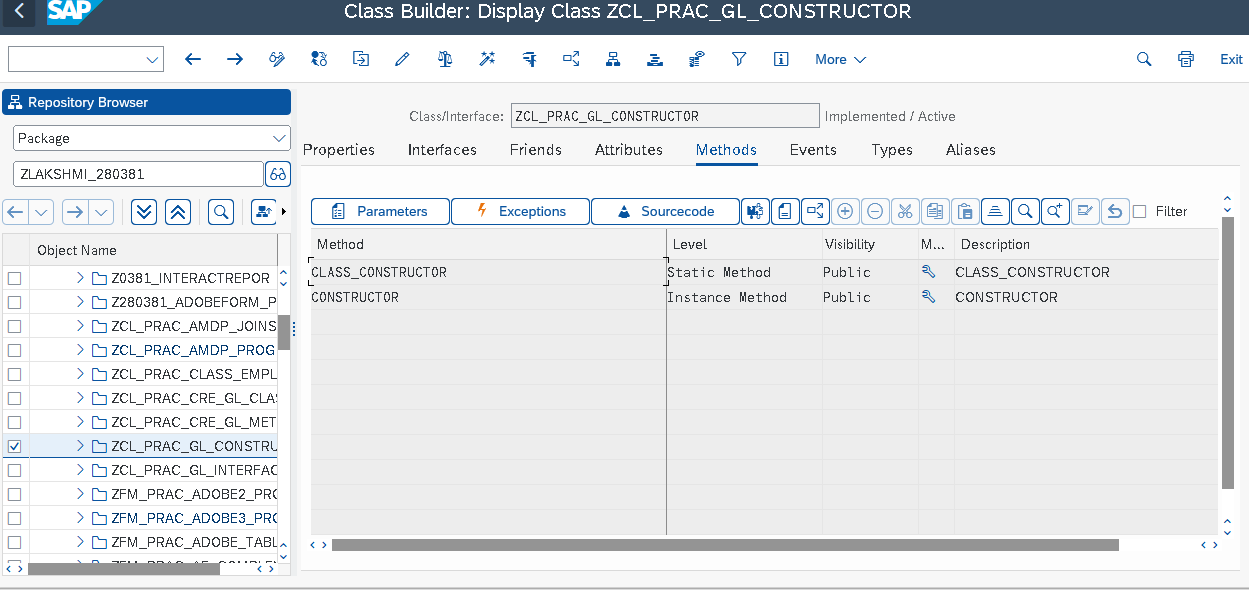


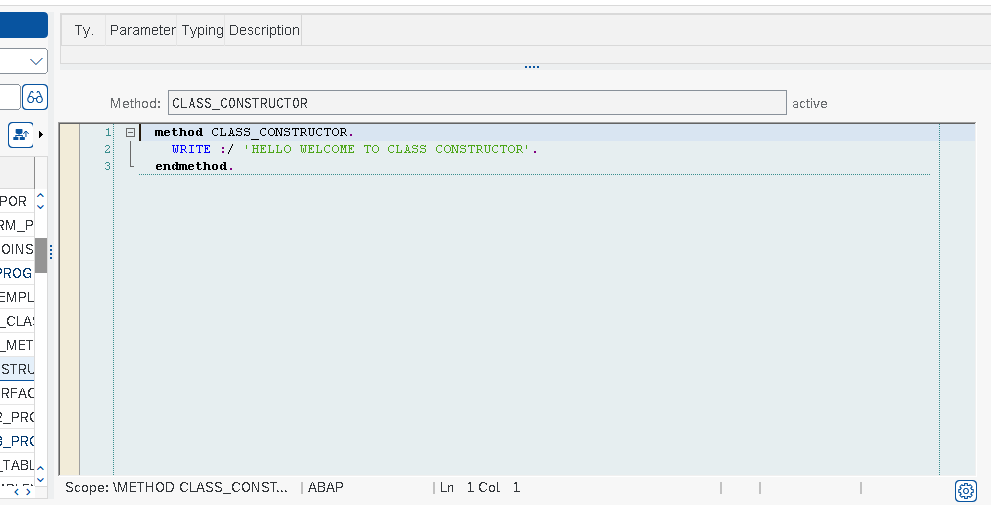
  method METHOD1. “write in method1 and method2 source code.  
    WRITE 'METHOD1'.  
  endmethod.

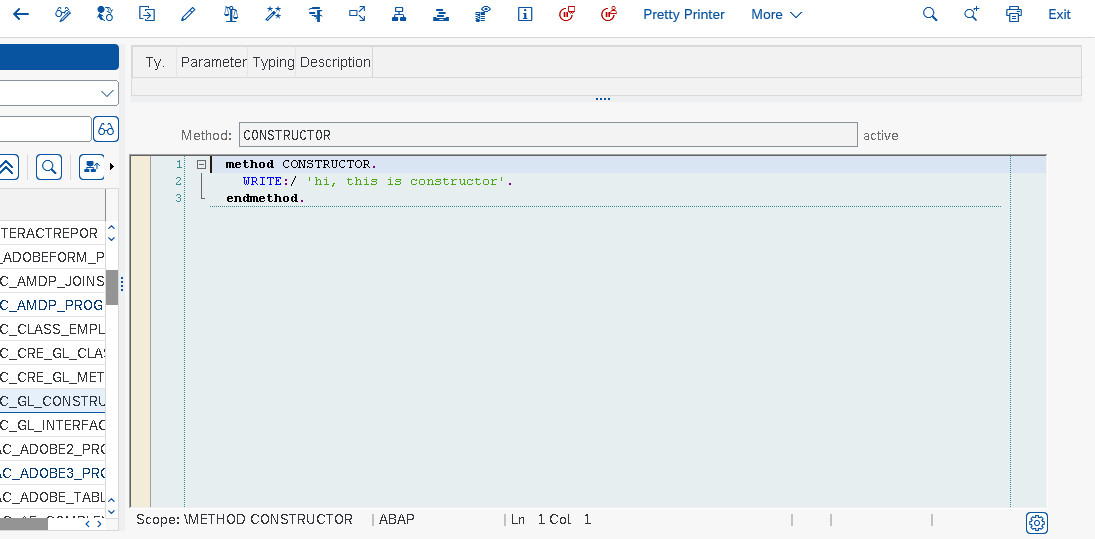
  method METHOD2.  
    WRITE 'METHOD2'.  
  endmethod.

*\*&---------------------------------------------------------------------\**  
*\*& Report ZCL\_PRAC\_CRE\_GL\_METHOD\_PROG*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZCL\_PRAC\_CRE\_GL\_METHOD\_PROG NO STANDARD PAGE HEADING LINE-COUNT 30(5) LINE-SIZE 800.  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* creating obj for GLOBAL METHODS*  
  
*\*data obj TYPE REF TO ZCL\_PRAC\_CRE\_GL\_METHOD.                 """"""""""object for Method*  
*\*create OBJECT obj.*  
*\**  
*\*START-OF-SELECTION.*  
*\*Call method obj->method1.                                     """""""""calling INSTANCE method*  
*\**  
*\*skip 2.*  
*\*call METHOD ZCL\_PRAC\_CRE\_GL\_METHOD=>method2.                   """""""""calling STATIC method.*  
  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*creation of LOCAL METHODS*  
  
class cl\_prac DEFINITION.  
  PUBLIC SECTION.  
  methods method1.                                            *"""""""""INSTANCE method*  
  CLASS-METHODS method2.                                      *""""""""STATIC method*  
ENDCLASS.  
  
class cl\_prac IMPLEMENTATION.  
  method method1.                                             *""""""""IMPLEMENTING method1*  
    WRITE 'Method1'.  
  ENDMETHOD.  
  
  method method2.                                            *""""""""IMPLEMENTING method2*  
    WRITE 'Method2'.  
  ENDMEthod.  
ENDCLASS.  
  
DATA obj TYPE REF TO ZCL\_PRAC\_CRE\_GL\_METHOD.  
  
START-OF-SELECTION.  
CREATE OBJECT obj.  
call method obj->method1.  
  
Call method ZCL\_PRAC\_CRE\_GL\_METHOD=>method2.

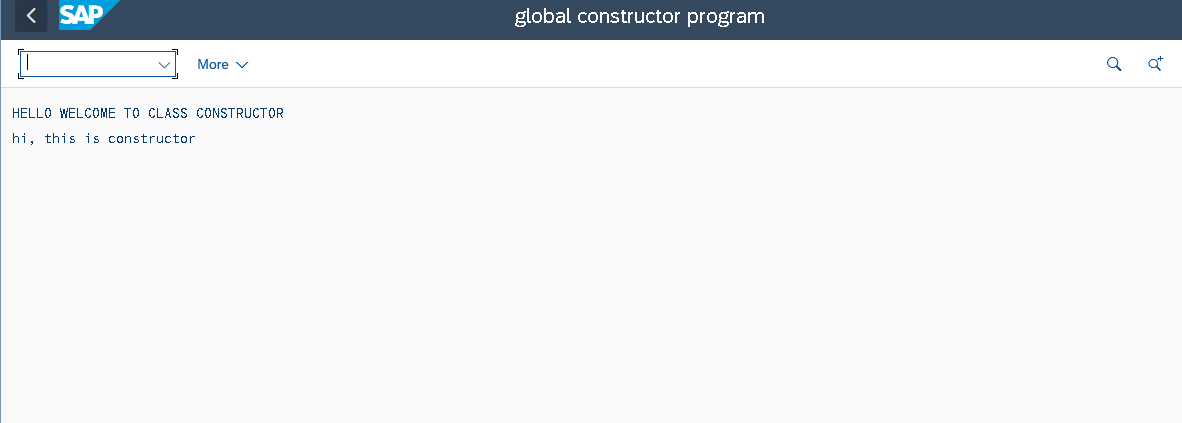


GLOBAL CONSTRUCTOR:  


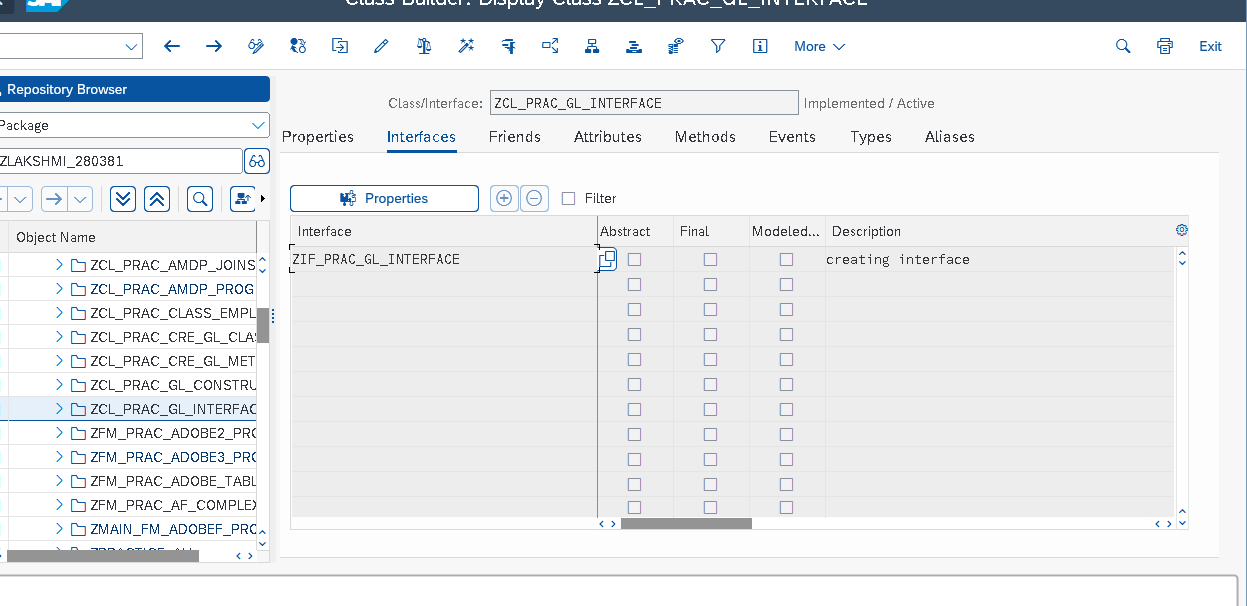
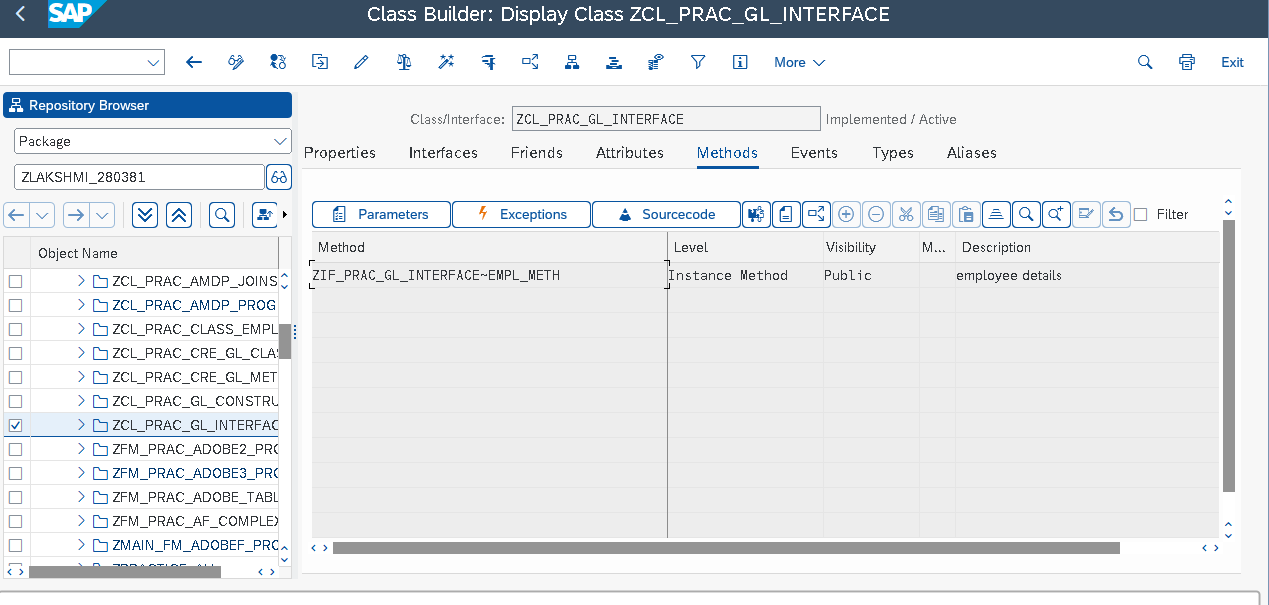


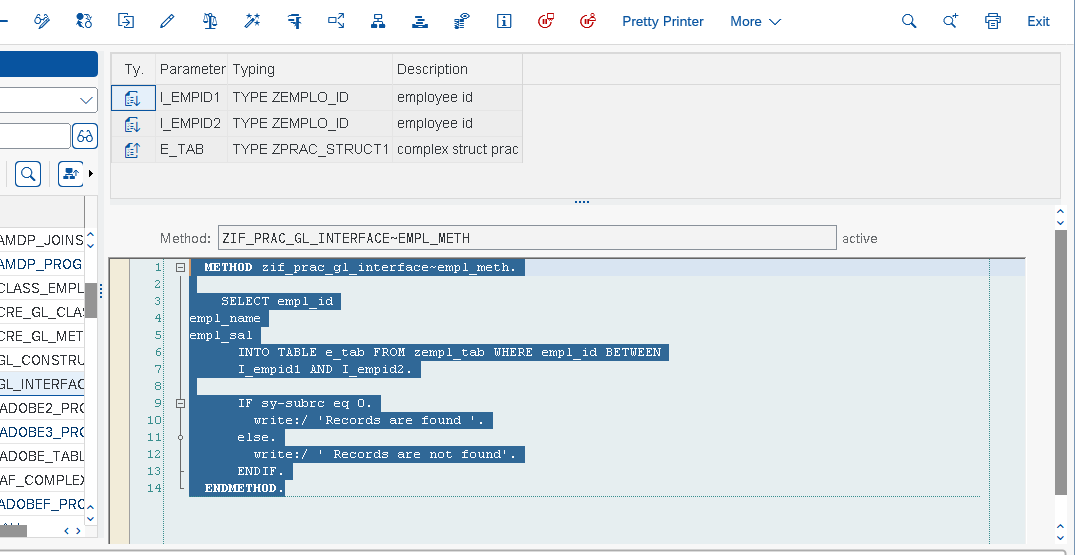


*\*&---------------------------------------------------------------------\**  
*\*& Report ZCL\_PRAC\_GL\_CONSTRUCT\_PROG*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZCL\_PRAC\_GL\_CONSTRUCT\_PROG NO STANDARD PAGE HEADING LINE-COUNT 30(3) LINE-SIZE 400.  
  
data: gl\_obj type REF TO ZCL\_PRAC\_GL\_CONSTRUCTOR.  
 create object gl\_obj.

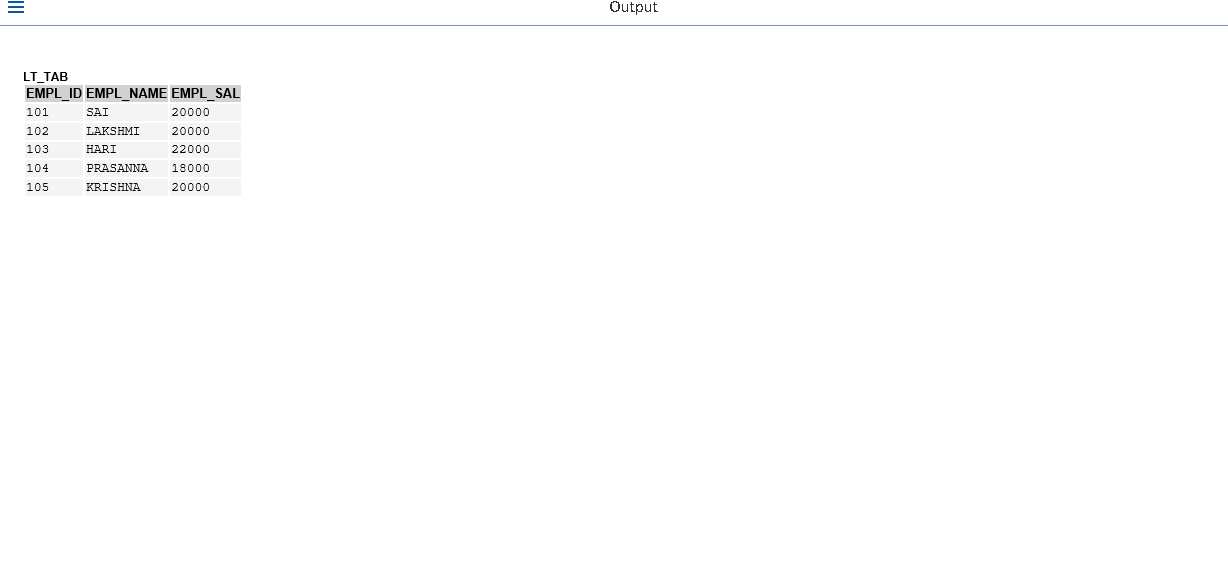


GLOBAL INTERFACE:



*\*&---------------------------------------------------------------------\**  
*\*& Report ZCL\_PRAC\_GL\_INTERFACE\_PROG*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZCL\_PRAC\_GL\_INTERFACE\_PROG NO STANDARD PAGE HEADING LINE-COUNT 30(5) LINE-SIZE 400.  
  
data lv\_empid type ZEMPLO\_ID.  
select-options s\_empid for lv\_empid.  
  
DATA lt\_tab type ZPRAC\_STRUCT1.  
  
START-OF-SELECTION.  
data obj type ref to ZCL\_PRAC\_GL\_INTERFACE.  
create object obj.  
  
obj->zif\_prac\_gl\_interface~empl\_meth(  
  EXPORTING  
    i\_empid1 = s\_empid-low  
    i\_empid2 = s\_empid-high  
  IMPORTING  
    e\_tab    = lt\_tab  
    ).  
  
CL\_DEMO\_OUTPUT=>display( lt\_tab ).



INHERITANCE:

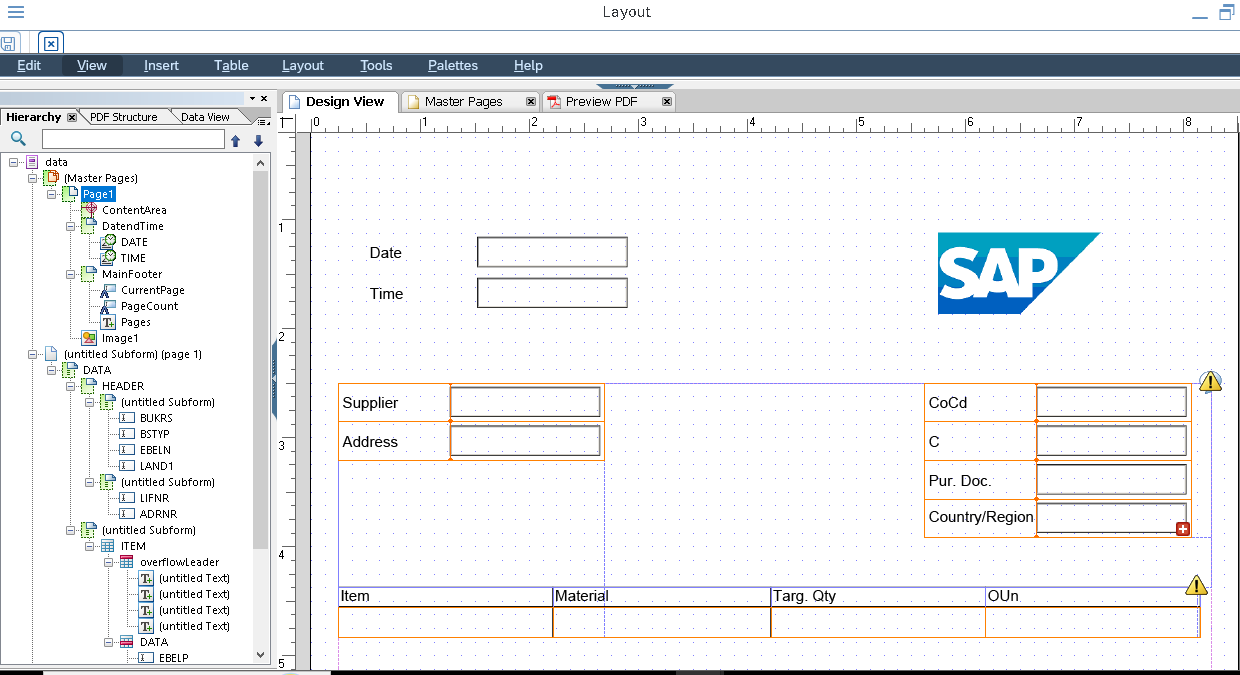
*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_INHERITANCE*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_INHERITANCE NO STANDARD PAGE HEADING LINE-COUNT 20(3) LINE-SIZE 300.  
  
*\*class cl\_prac DEFINITION.*  
*\*  PUBLIC SECTION.*  
*\*  METHODS: m1.*  
*\*  data: n1 type i value 9.*  
*\**  
*\*  PROTECTED SECTION.*  
*\*  data: n2 type i value 7.*  
*\*  METHODS: m2.*  
*\*ENDCLASS.*  
*\**  
*\*class cl\_prac IMPLEMENTATION.*  
*\*  method: m1.*  
*\*    write:/5 n1.*  
*\*  ENDMETHOD.*  
*\*  METHOD: m2.*  
*\*    write:/5 n2.*  
*\*  ENDMETHOD.*  
*\*ENDCLASS.*  
*\**  
*\*class cl\_prac2 DEFINITION INHERITING FROM cl\_prac.*  
*\*  PUBLIC SECTION.*  
*\*  METHODS: me1.*  
*\*ENDCLASS.*  
*\**  
*\*class cl\_prac2 IMPLEMENTATION.*  
*\*  METHOD me1.                                   ""calling method inside ,ethod.*  
*\*    call method : m1, m2.*  
*\*  ENDMETHOD.*  
*\*ENDCLASS.*  
*\**  
*\*START-OF-SELECTION.*  
*\*DATA obj type REF TO cl\_prac2.                   ""creating obj for sub class*  
*\*CREATE OBJECT obj.*  
*\**  
*\*obj->me1( ).*  
  
  
*\*class cl\_prac DEFINITION.*  
*\*  PUBLIC SECTION.*  
*\*  methods: m1.*  
*\*  PROTECTED SECTION.*  
*\*  methods: m2.*  
*\*ENDCLASS.*  
*\*class cl\_prac IMPLEMENTATION.*  
*\*  method: m1.*  
*\*    WRITE:/5 ' i am m1 in class1 '.*  
*\*  ENDMETHOD.*  
*\*  method: m2.*  
*\*    WRITE:/5 ' i am m2 in class1 '.*  
*\*  ENDMETHOD.*  
*\*ENDCLASS.*  
*\**  
*\*class cl\_prac2 DEFINITION INHERITING FROM cl\_prac.*  
*\*  PUBLIC SECTION.*  
*\*  methods: m1 REDEFINITION.*  
*\*  PROTECTED SECTION.*  
*\*  methods: m2 REDEFINITION.*  
*\*ENDCLASS.*  
*\**  
*\*class cl\_prac2 IMPLEMENTATION.*  
*\*  method: m1.*  
*\*    write:/5 ' i am m1 in class2 '.*  
*\*    call method m2.*  
*\*  ENDMETHOD.*  
*\*  method: m2.*  
*\*    write:/5 ' i am m1 in class2 '.*  
*\*  ENDMETHOD.*  
*\*ENDCLASS.*  
*\**  
*\*START-OF-SELECTION.*  
*\*data: obj type ref to cl\_prac2.*  
*\*create OBJECT obj.*  
*\**  
*\*obj->m1( ).*  
*\*\*obj->m2( ).                                       """cannot able to call the protected methods*  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*abstarct class( objects cannot create )*  
*\*class cl\_prac DEFINITION ABSTRACT.*  
*\*  PUBLIC SECTION.*  
*\*  methods m1.*  
*\*ENDCLASS.*  
*\**  
*\*class cl\_prac IMPLEMENTATION.*  
*\**  
*\*ENDCLASS.*  
*\**  
*\*class cl\_prac2 DEFINITION INHERITING FROM cl\_prac.*  
*\**  
*\*ENDCLASS.*  
*\*class cl\_prac2 IMPLEMENTATION.*  
*\*  method: m1.*  
*\**  
*\*  ENDMETHOD.*  
*\*ENDCLASS.*  
*\**  
*\*START-OF-SELECTION.*  
*\*data:  obj2 type ref to cl\_prac2,*  
*\*      obj type ref to cl\_prac.*  
*\*create object: obj2, obj.*  
*\*obj2->m1( ).*  
  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*final classes connot have any subclass*  
*\**  
*\*class cl\_prac DEFINITION FINAL.*  
*\*ENDCLASS.*  
*\**  
*\*class cl\_prac IMPLEMENTATION.*  
*\*ENDCLASS.*  
*\**  
*\*class cl\_prac2 DEFINITION INHERITING FROM cl\_prac.*  
*\*ENDCLASS.*  
*\**  
*\*class cl\_prac2 IMPLEMENTATION.*  
*\*ENDCLASS.*  
*\**  
*\*START-OF-SELECTION.*  
*\*Data obj type ref to cl\_prac2.*  
*\*create object obj.*  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*FINAL methods cannot be redefined.*  
*\**  
*\*class cl\_prac DEFINITION.*  
*\*  PUBLIC SECTION.*  
*\*  METHODS: m1, m2 final.*  
*\*ENDCLASS.*  
*\**  
*\*class cl\_prac IMPLEMENTATION.*  
*\*  method m1.*  
*\*    write:/5 ' I am method1 '.*  
*\*  ENDMETHOD.*  
*\*   method m2.*  
*\*    write:/5 ' I am method2 '.*  
*\*  ENDMETHOD.*  
*\*ENDCLASS.*  
*\**  
*\*class cl\_prac2 DEFINITION INHERITING FROM cl\_prac.*  
*\*  PUBLIC SECTION.*  
*\*  methods: m1 redefinition.*  
*\*ENDCLASS.*  
*\**  
*\*class cl\_prac2 IMPLEMENTATION.*  
*\*  method: m1.*  
*\*    WRITE:/ ' I am method1, in subclass'.*  
*\*  ENDMETHOD.*  
*\*ENDCLASS.*  
*\**  
*\*START-OF-SELECTION.*  
*\*Data obj type ref to cl\_prac2.*  
*\*create object obj.*  
*\*obj->m2( ).*

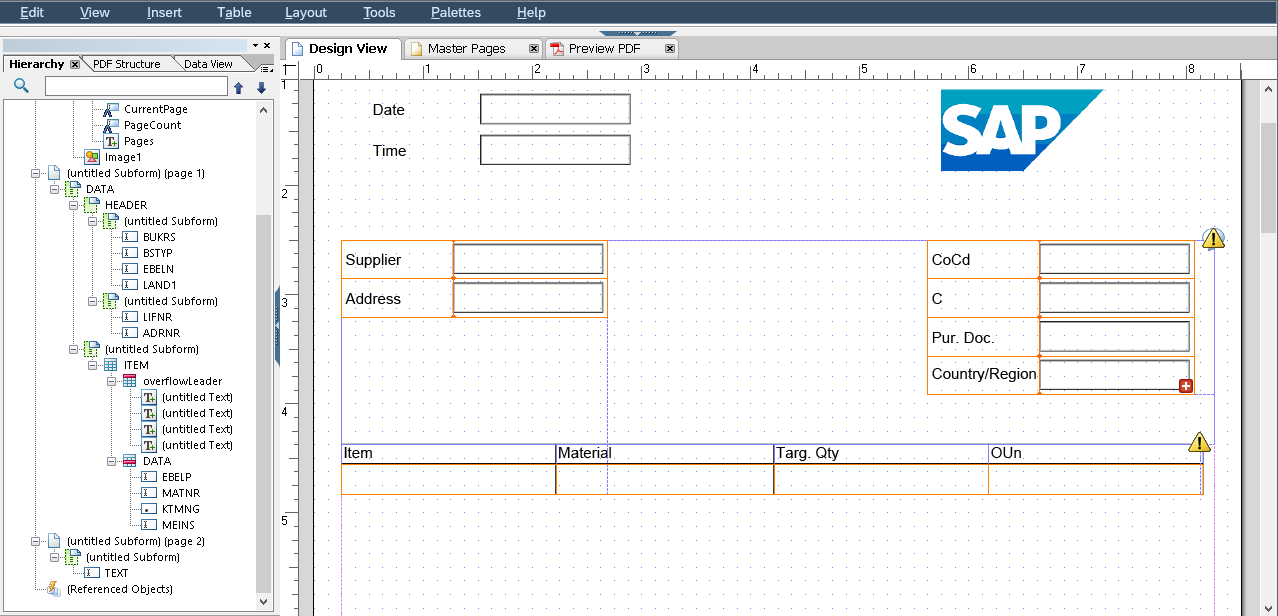
FORMATTING: *\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_FORMAT*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_FORMAT.  
  
TABLES lfa1.  
TYPES: BEGIN OF ls\_tab,               *""""local structures*  
      lifnr type lifnr,  
      land1 type land1,  
      ort01 type ort01,  
      name1 type name1\_gp,  
      end of ls\_tab.  
  
DATA: LT\_tab type STANDARD TABLE OF ls\_tab,  
      wa\_tab type ls\_tab,  
      lv\_lifnr TYPE lifnr.  
  
SELECT-OPTIONS s\_lifnr FOR lv\_lifnr.  
  
PARAMETERS p\_land1 TYPE lfa1-land1.  
  
START-OF-SELECTION.  
select lifnr land1 ort01 name1  
  INTO TABLE LT\_Tab FROM lfa1  
   WHERE lifnr IN s\_lifnr  
  AND land1 EQ p\_land1.  
  
IF sy-subrc eq 0.  
  write:/ 'records are found in lfa1', sy-dbcnt.  
ELSE.  
  WRITE:/ 'records are not found', sy-dbcnt.  
ENDIF.  
  
SKIP 2.  
  
FORMAT COLOR COL\_NEGATIVE.  
loop at lt\_Tab into wa\_tab.  
  WRITE :/ 'row no',  
  10 wa\_tab-lifnr,  
  30 wa\_tab-land1,  
  50 wa\_tab-ort01,  
  65 wa\_tab-name1.  
  
ENDLOOP.  
  
FORMAT COLOR Off.  
  
*\*BAck.*  
*\*SKIP 2.*  
*\*write /120 'ust hyd'.*  
*\*back.*  
*\*skip 4.*  
*\*write /130 'ust usa'.*

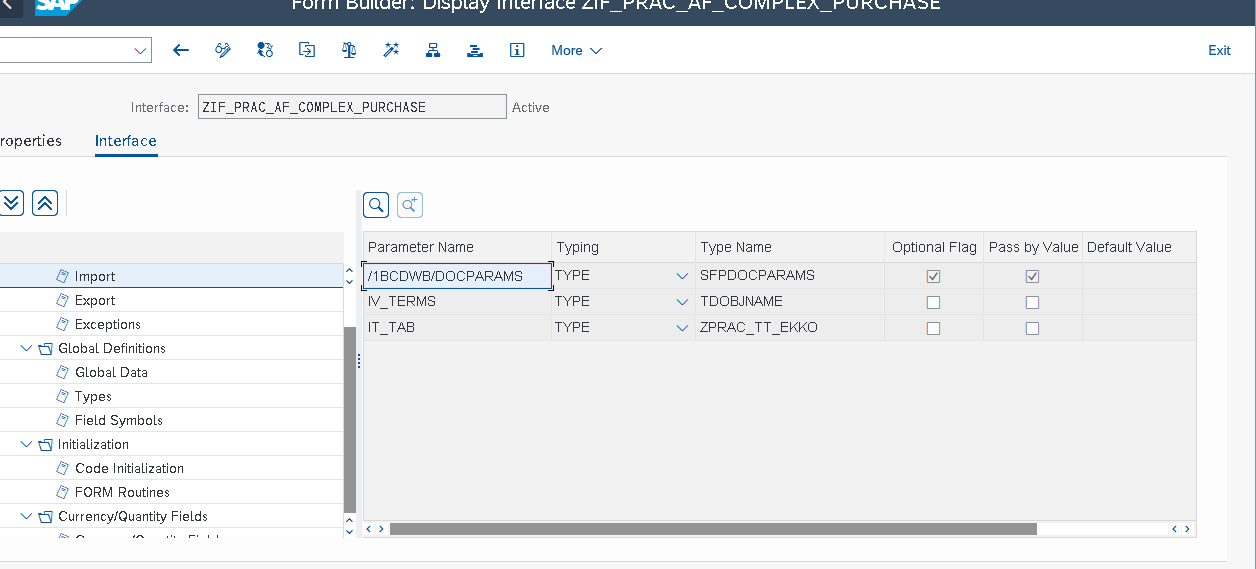
EXCEPTIONS:

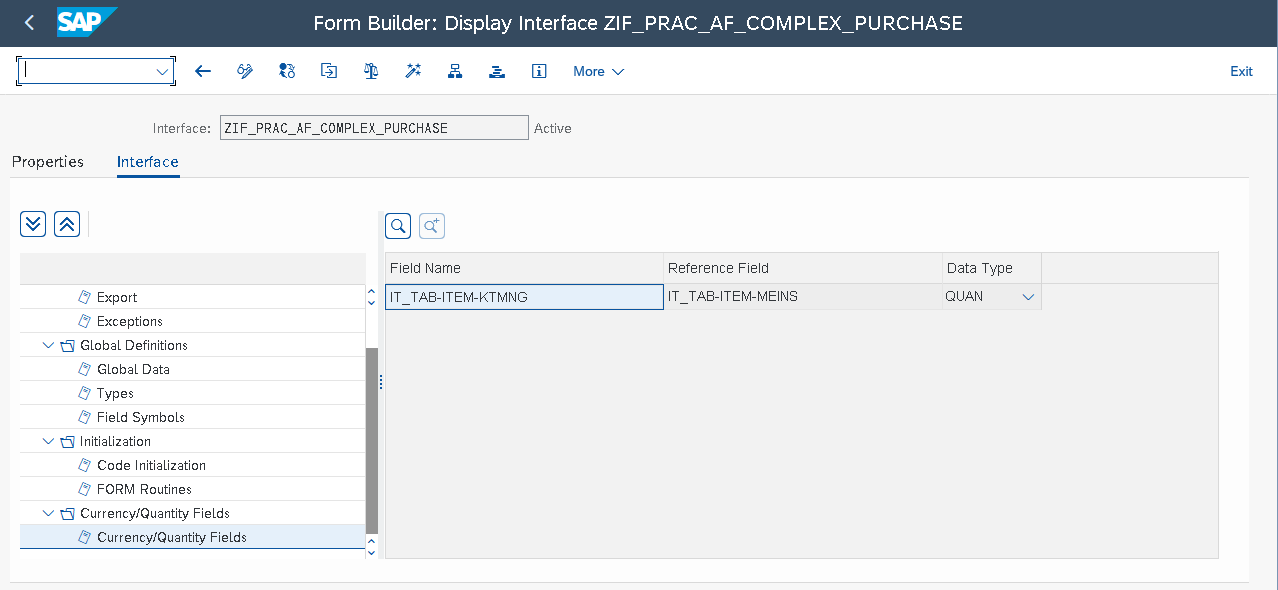
*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_EXCEPTIONS*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_EXCEPTIONS NO STANDARD PAGE HEADING LINE-COUNT 30(5) LINE-SIZE 400.  
  
*\*\*PARAMETERS n type i.*  
*\*\*DATA: lv\_i type i value 1.*  
*\*\**  
*\*\*START-OF-SELECTION.*  
*\*\**  
*\*\*lv\_i = lv\_i / 0.*  
  
*\*DATA: a type i value 10.*  
*\*START-OF-SELECTION.*  
*\**  
*\*CATCH SYSTEM-EXCEPTIONS COMPUTE\_INT\_ZERODIVIDE = 4.*  
*\**  
*\*a = a / 0.*  
*\*endCAtch.*  
*\**  
*\*IF sy-subrc = 4.*  
*\*  write:/ ' division by zero!!!!CHECK'.*  
*\*ENDIF.*

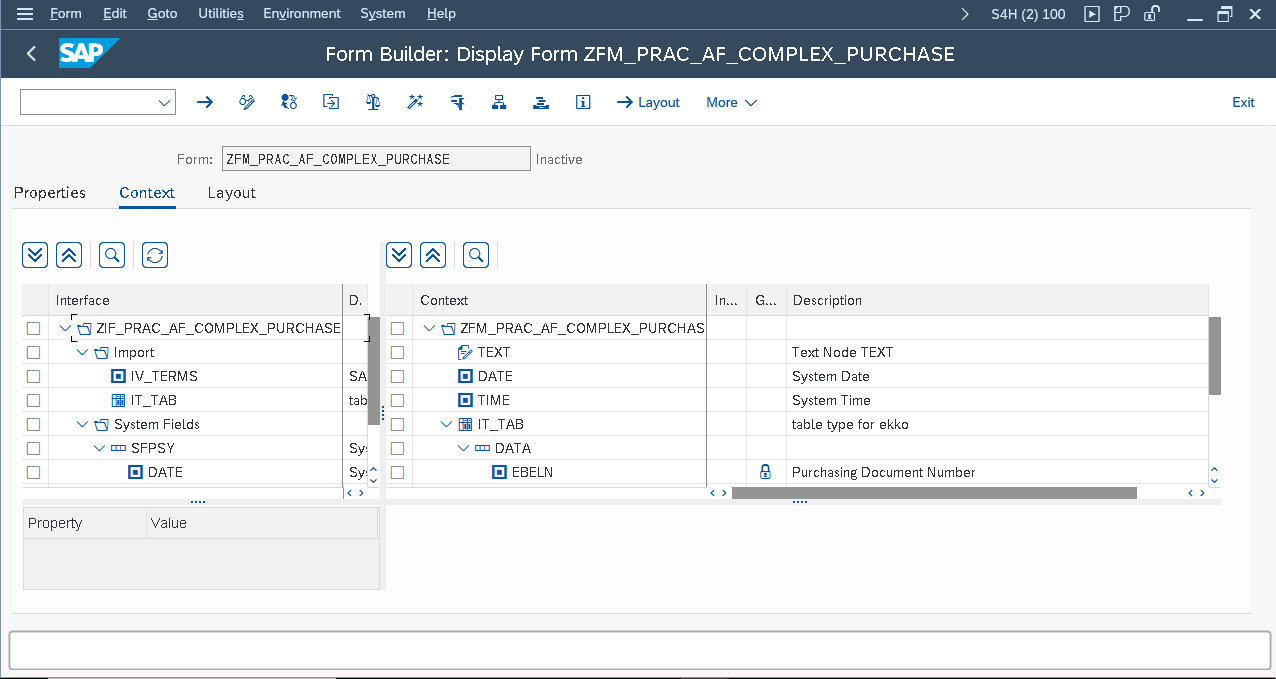
*ADOBE FORMS – COMPLEX PROGRAM*

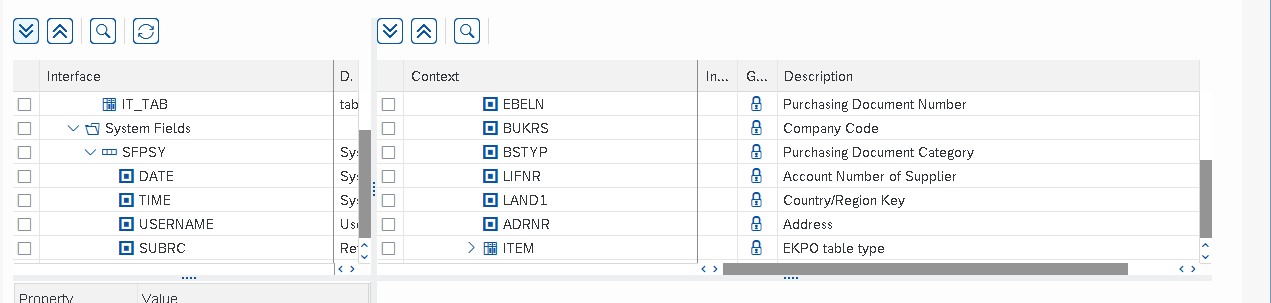


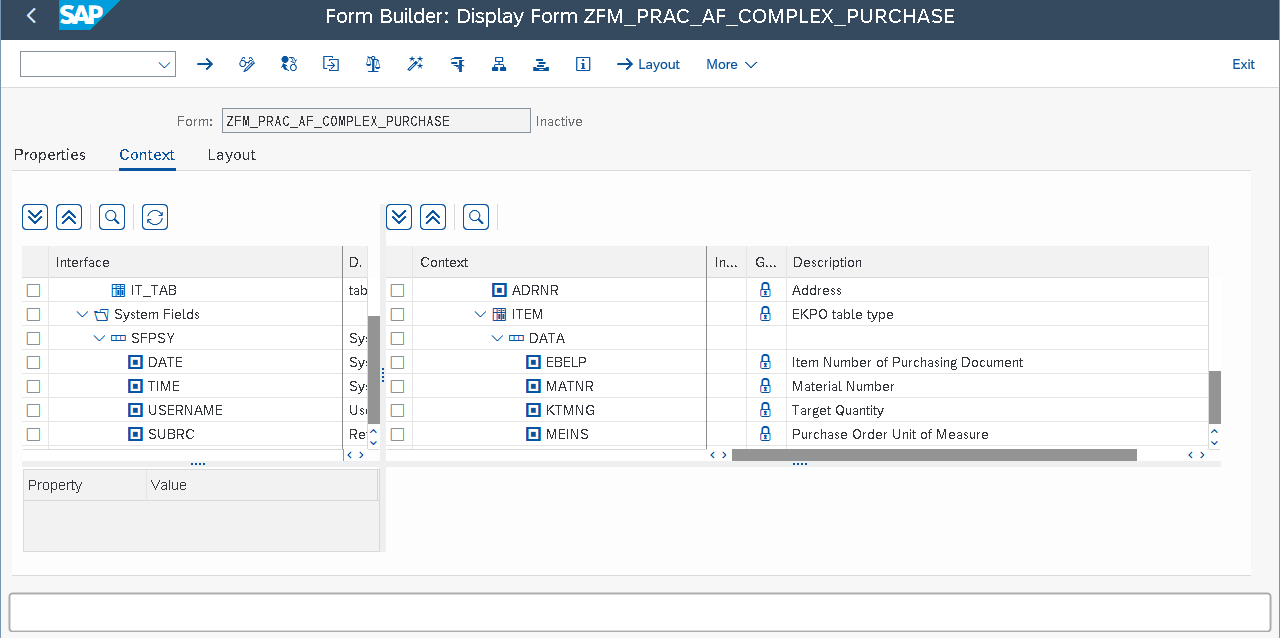






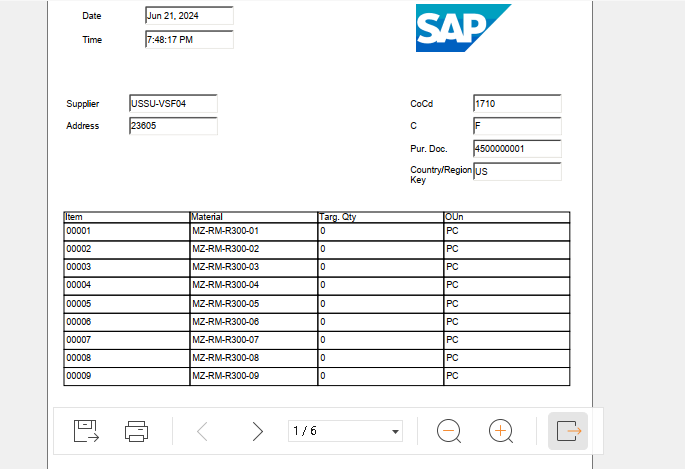




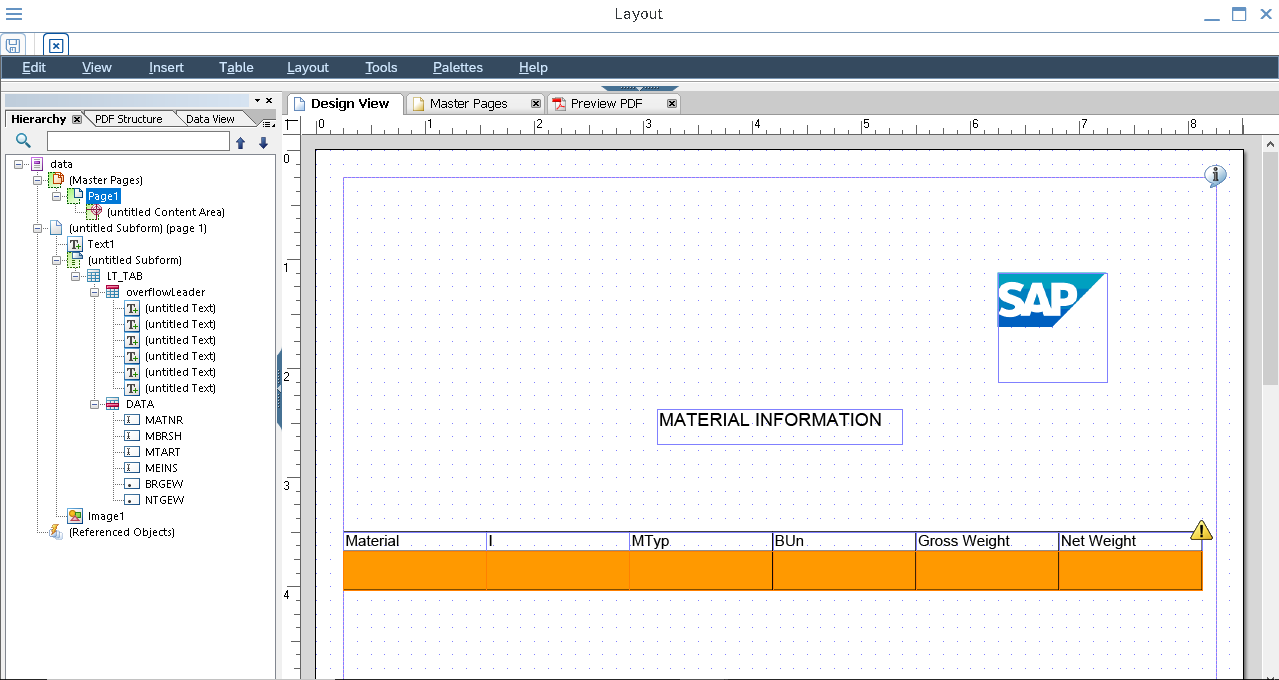


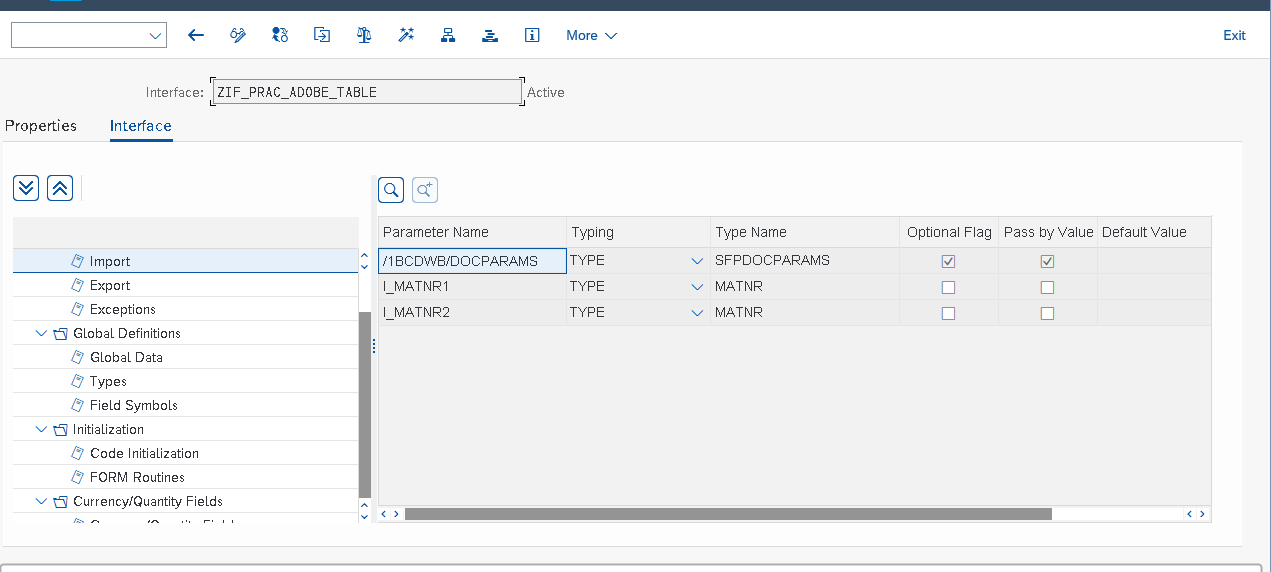
*\*&---------------------------------------------------------------------\**  
*\*& Report ZTRUPEN\_UST\_SM\_PAGE\_PROG*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZFM\_PRAC\_AF\_COMPLEX\_PROG.  
  
  
DATA: gv\_ebeln TYPE ekko-ebeln,  
      gv\_func\_name  TYPE funcname,  
      gs\_vendor   TYPE ZPRAC\_LT\_LFA1,  
      gs\_outp\_param TYPE sfpoutputparams,  
      gs\_output type SSFCTRLOP,  
      gt\_final  TYPE ZPRAC\_TT\_EKKO.  
  
SELECT-OPTIONS:s\_ebeln FOR gv\_ebeln OBLIGATORY.  
  
*\*SELECTION-SCREEN: BEGIN OF BLOCK b1 WITH FRAME TITLE TEXT-000.*  
*\*PARAMETERS: p\_adobe RADIOBUTTON GROUP form,*  
*\*            p\_smart RADIOBUTTON GROUP form.*  
*\*SELECTION-SCREEN: END OF BLOCK b1.*  
  
AT SELECTION-SCREEN.  
*\* -- Validate Sales orders*  
  
  SELECT SINGLE ebeln  
  FROM ekko  
  INTO gv\_ebeln  
  WHERE ebeln IN s\_ebeln.  
  
  IF sy-subrc IS NOT INITIAL.  
    MESSAGE e017(pord\_messages).  
  ENDIF.  
  
START-OF-SELECTION.  
  SELECT ebeln, bukrs, bstyp, ekko~lifnr, land1, lfa1~adrnr  
    FROM ekko  
    INNER JOIN lfa1 ON lfa1~lifnr = ekko~lifnr  
    INTO TABLE @DATA(gt\_ekko)  
    WHERE ebeln IN @s\_ebeln.  
  
  IF sy-subrc IS INITIAL.  
    SELECT EBELN ,ebelp, matnr, ktmng, meins  
      FROM ekpo  
      INTO TABLE @DATA(gt\_ekpo)  
      FOR ALL ENTRIES IN @gt\_ekko  
            WHERE ebeln = @gt\_ekko-ebeln.  
  
    IF sy-subrc IS NOT INITIAL.  
      CLEAR: gt\_ekko, gt\_ekpo.  
      MESSAGE e017(prod\_messages).  
    ENDIF.  
  ENDIF.  
  
  IF gt\_ekko IS NOT INITIAL.  
*\* -- Parallel Cursor*  
*\* -- Pre-requisite to sort with common fields used in Parallel cursor*  
    SORT gt\_ekko BY ebeln.  
    SORT gt\_ekpo BY ebeln.  
  
    DATA(gv\_index) = VALUE syst\_tabix( ).  
  
    LOOP AT gt\_ekko ASSIGNING FIELD-SYMBOL(<lfs\_ekko>).  
      APPEND INITIAL LINE TO gt\_final ASSIGNING FIELD-SYMBOL(<gfs\_final>).  
      MOVE-CORRESPONDING <lfs\_ekko> TO <gfs\_final>.  
  
      READ TABLE gt\_ekpo TRANSPORTING NO FIELDS  
      WITH KEY ebeln = <lfs\_ekko>-ebeln BINARY SEARCH.  
      IF sy-subrc IS INITIAL.  
        gv\_index = sy-tabix.  
        LOOP AT gt\_ekpo ASSIGNING FIELD-SYMBOL(<lfs\_ekpo>) FROM gv\_index.  
          IF <lfs\_ekpo>-ebeln NE <lfs\_ekko>-ebeln.  
            EXIT.  
          ENDIF.  
          APPEND INITIAL LINE TO <gfs\_final>-item ASSIGNING FIELD-SYMBOL(<lfs\_items>).  
          MOVE-CORRESPONDING <lfs\_ekpo> TO <lfs\_items>.  
        ENDLOOP.  
      ENDIF.  
    ENDLOOP.  
  
    CALL FUNCTION 'FP\_JOB\_OPEN'  
      CHANGING  
        ie\_outputparams       = gs\_outp\_param  
*\*     EXCEPTIONS*  
*\*       CANCEL                = 1*  
*\*       USAGE\_ERROR           = 2*  
*\*       SYSTEM\_ERROR          = 3*  
*\*       INTERNAL\_ERROR        = 4*  
*\*       OTHERS                = 5*  
              .  
    IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
    ENDIF.  
  
    CALL FUNCTION 'FP\_FUNCTION\_MODULE\_NAME'  
      EXPORTING  
        i\_name                     = 'ZFM\_PRAC\_AF\_COMPLEX\_PURCHASE'  
     IMPORTING  
       E\_FUNCNAME                 = gv\_func\_name  
*\*       E\_INTERFACE\_TYPE           =*  
*\*       EV\_FUNCNAME\_INBOUND        =*  
              .  
  
      CALL FUNCTION gv\_func\_name *"'/1BCDWB/SM00000693'*  
        EXPORTING  
*\*         /1BCDWB/DOCPARAMS        =*  
          iv\_terms                 = 'ZPRAC\_TEXTMOD'  
          it\_tab                   = gt\_final  
*\*       IMPORTING*  
*\*         /1BCDWB/FORMOUTPUT       =*  
*\*       EXCEPTIONS*  
*\*         USAGE\_ERROR              = 1*  
*\*         SYSTEM\_ERROR             = 2*  
*\*         INTERNAL\_ERROR           = 3*  
*\*         OTHERS                   = 4*  
                .  
      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.  
  
  
      ENDIF.

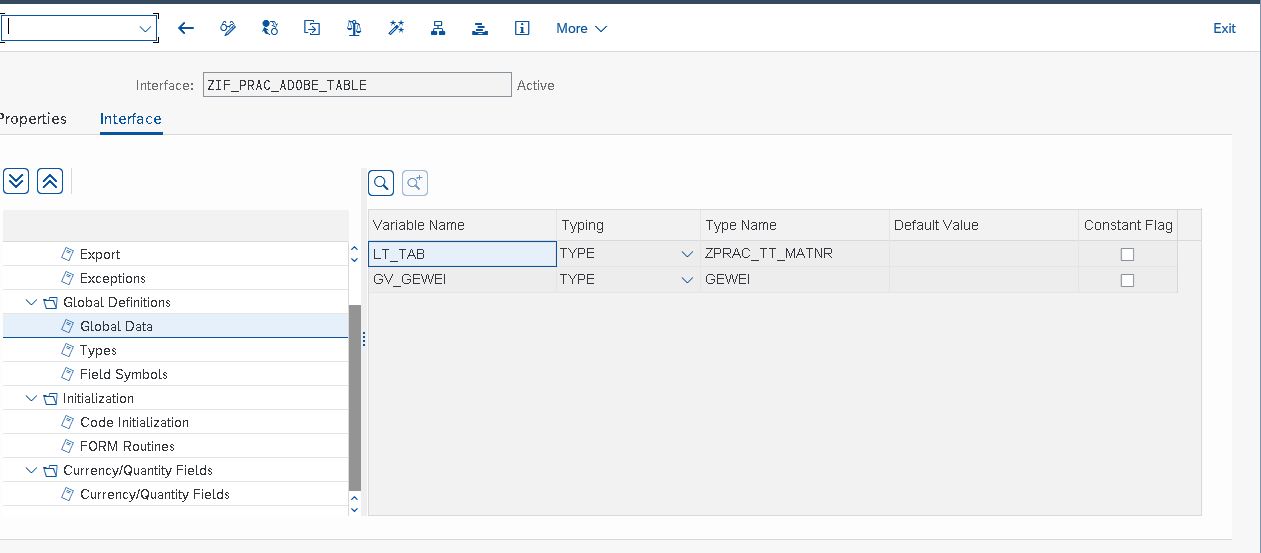
OUTPUT:

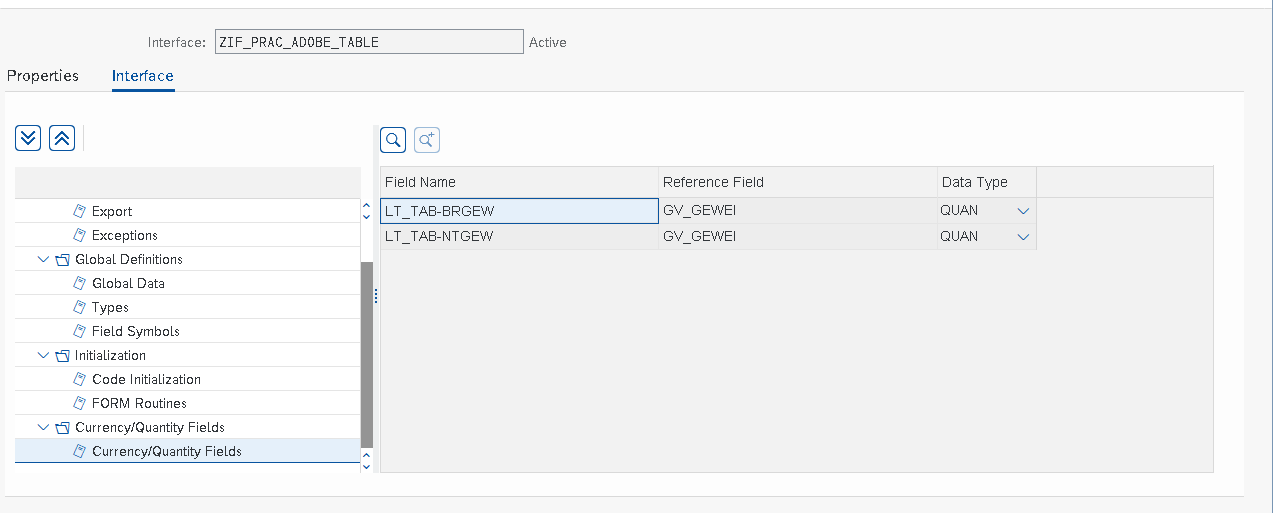


ADOBE FORM TABLE PROG:

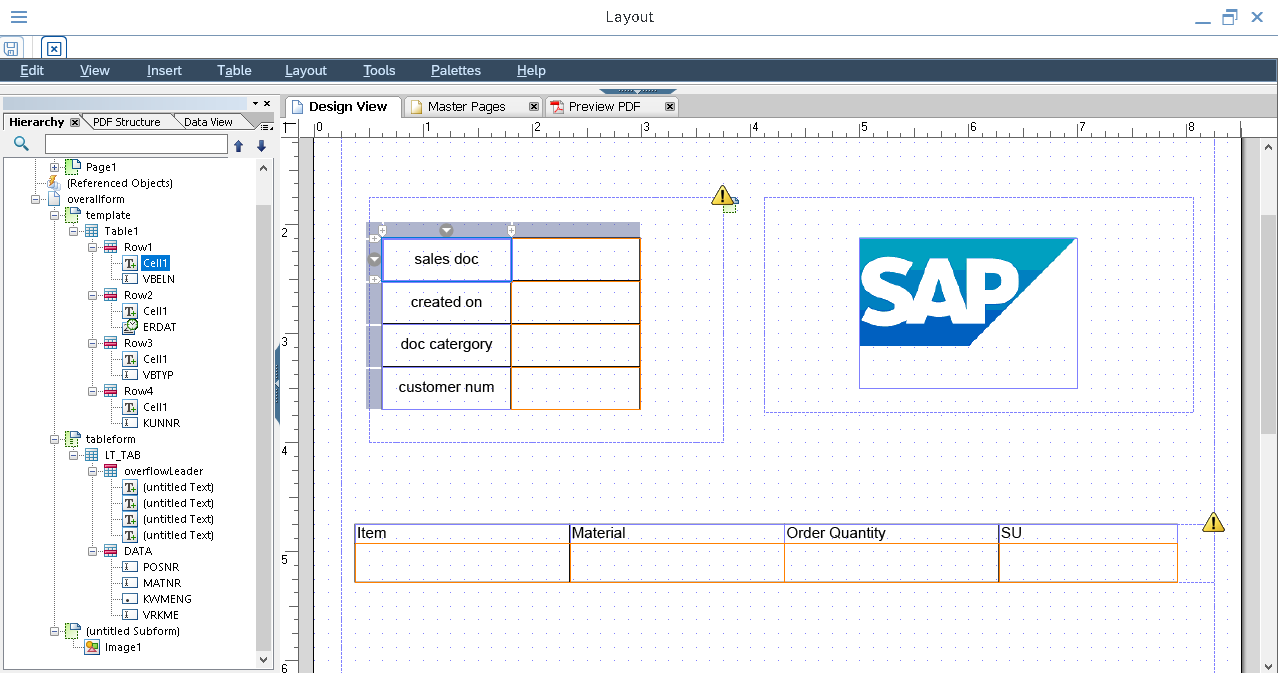


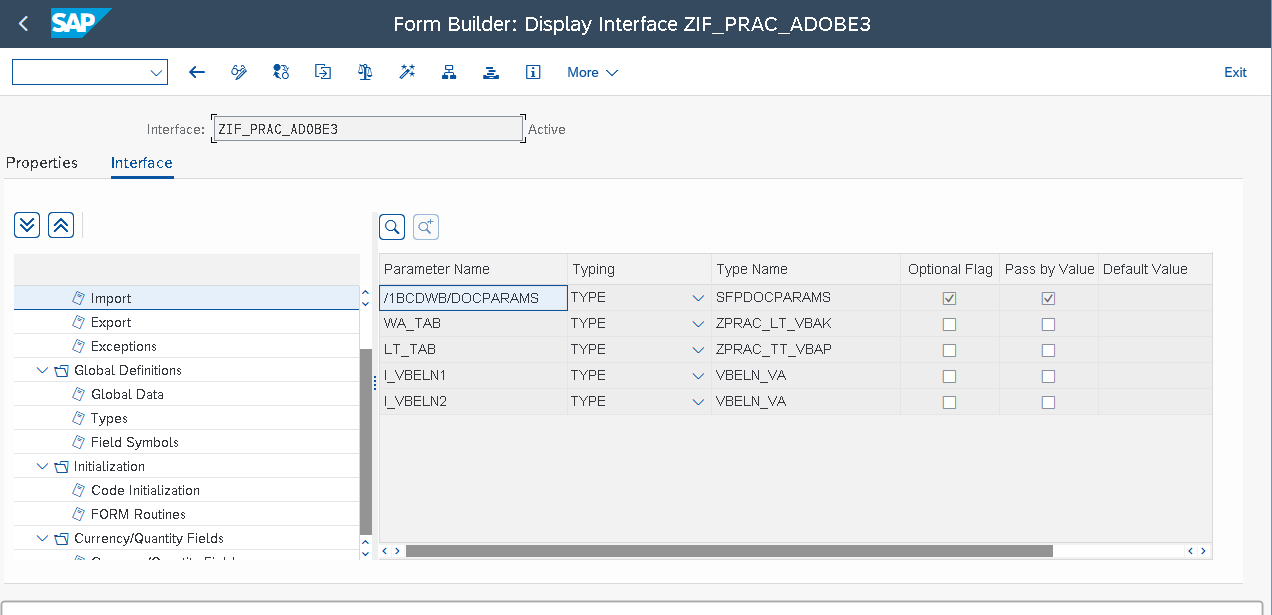


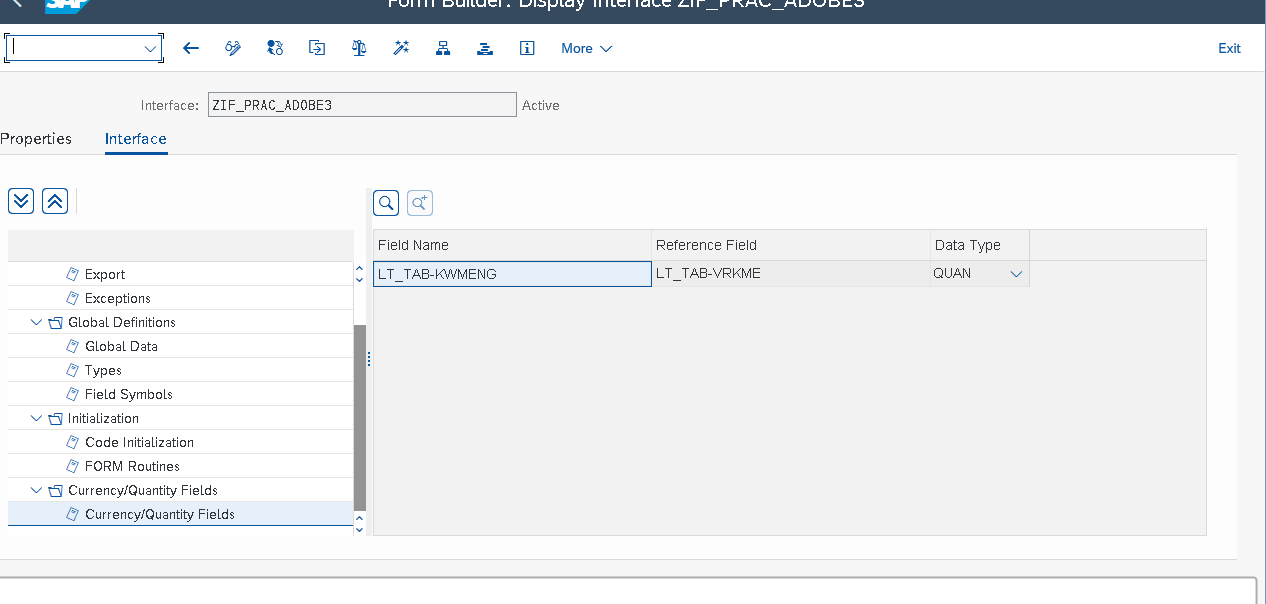


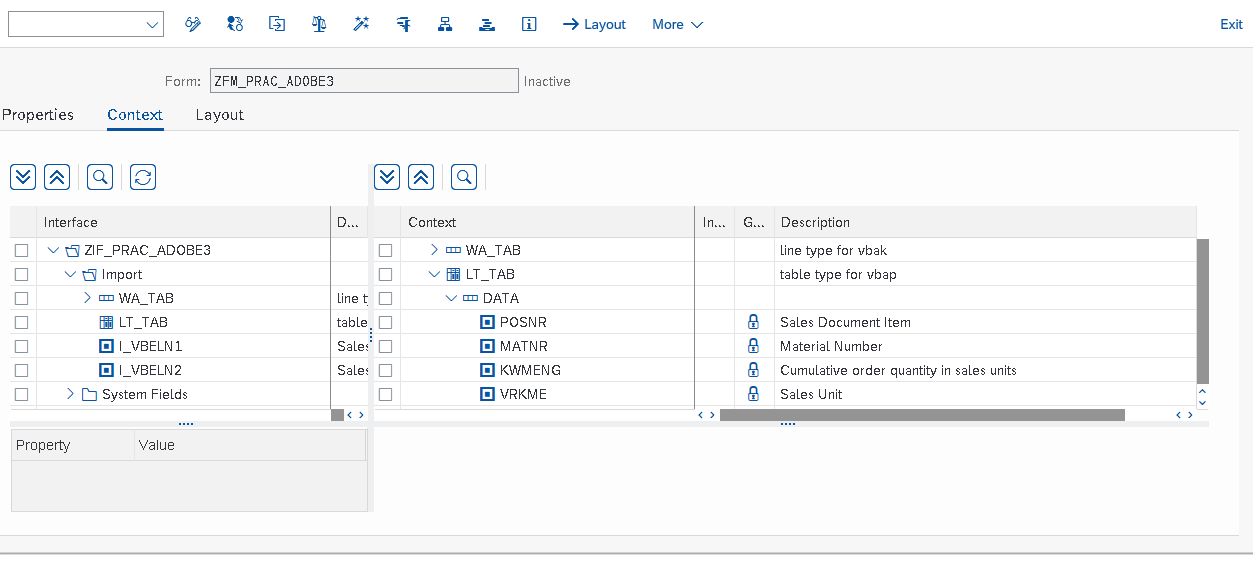


*\*&---------------------------------------------------------------------\**  
*\*& Report ZFM\_PRAC\_ADOBE\_TABLE\_PROG*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZFM\_PRAC\_ADOBE\_TABLE\_PROG.  
  
INCLUDE ZFM\_PRAC\_ADOBE\_TABLE\_TOP.  
  
DATA: lv\_matnr type matnr.  
  
SELECT-OPTIONS s\_matnr for lv\_matnr.  
  
START-OF-SELECTION.  
  
wa\_outputparams-device =  'PRINTER'.  
wa\_outputparams-dest  =   'LP01' .  
wa\_outputparams-nodialog =  'X'.  
wa\_outputparams-preview =  'X'.  
  
CALL FUNCTION 'FP\_JOB\_OPEN'  
  CHANGING  
    ie\_outputparams       = wa\_outputparams  
*\* EXCEPTIONS*  
*\*   CANCEL                = 1*  
*\*   USAGE\_ERROR           = 2*  
*\*   SYSTEM\_ERROR          = 3*  
*\*   INTERNAL\_ERROR        = 4*  
*\*   OTHERS                = 5*  
          .  
IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
ENDIF.  
  
CALL FUNCTION 'FP\_FUNCTION\_MODULE\_NAME'  
  EXPORTING  
    i\_name                     =  'ZFM\_PRAC\_ADOBE\_TABLE'  
 IMPORTING  
   E\_FUNCNAME                 = gv\_fm\_nam  
*\*   E\_INTERFACE\_TYPE           =*  
*\*   EV\_FUNCNAME\_INBOUND        =*  
          .  
  
wa\_docparams-langu  ='E'.  
wa\_docparams-country  ='IN'.  
  
select matnr mbrsh mtart meins brgew ntgew into table lt\_tab  
  from mara  
 where matnr eq s\_matnr.  
  
  CALL FUNCTION gv\_fm\_nam  *"'/1BCDWB/SM00000339'*  
    EXPORTING  
     /1BCDWB/DOCPARAMS        = wa\_docparams  
      i\_matnr1                 = s\_matnr-low  
      i\_matnr2                 = s\_matnr-high  
*\*   IMPORTING*  
*\*     /1BCDWB/FORMOUTPUT       =*  
*\*   EXCEPTIONS*  
*\*     USAGE\_ERROR              = 1*  
*\*     SYSTEM\_ERROR             = 2*  
*\*     INTERNAL\_ERROR           = 3*  
*\*     OTHERS                   = 4*  
            .  
  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.

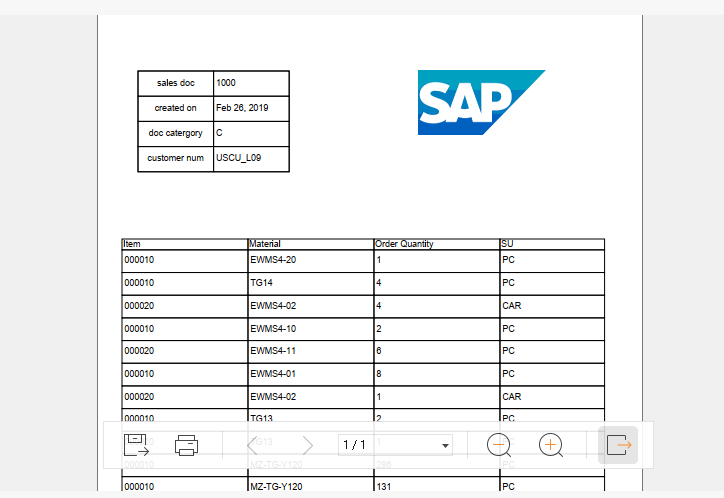


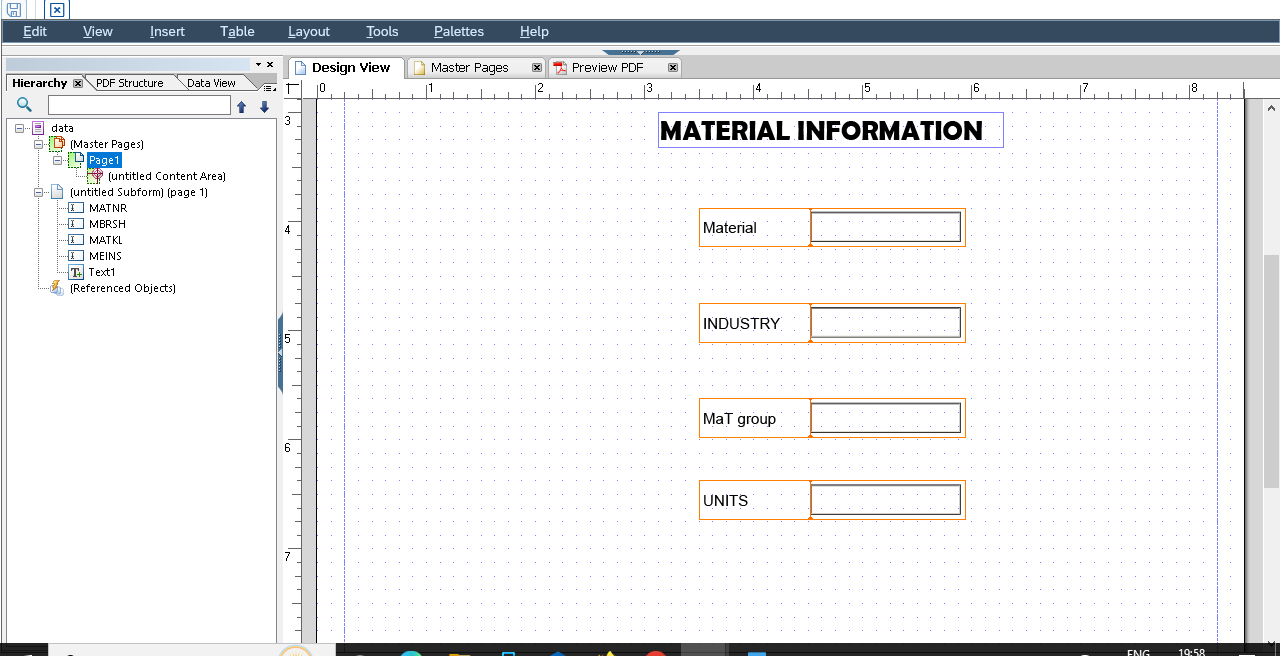


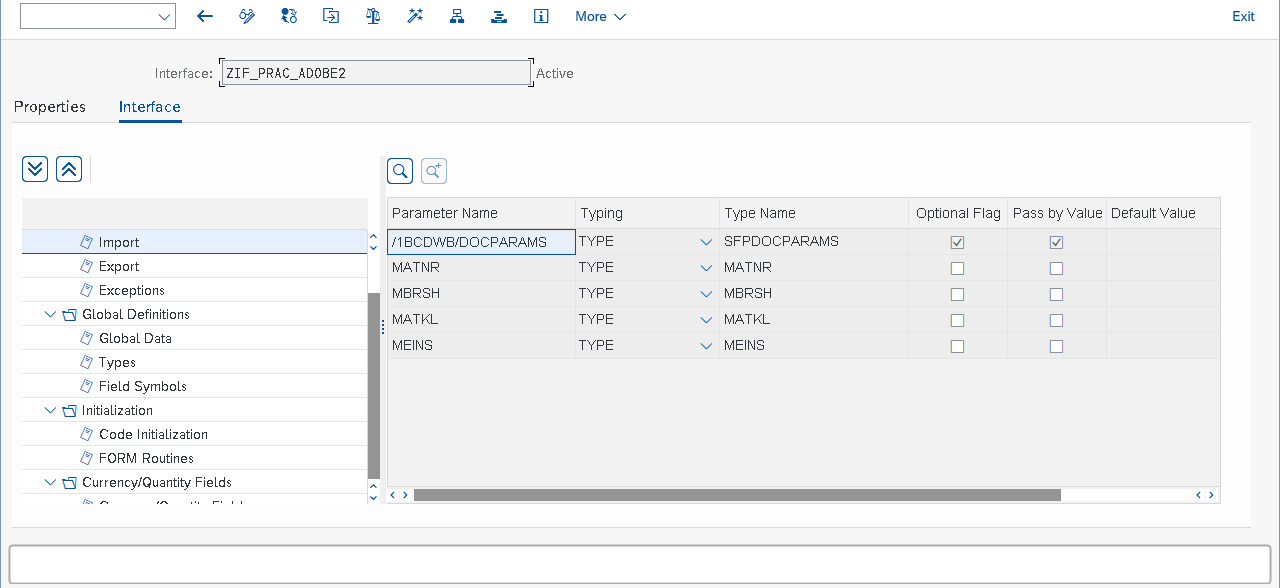




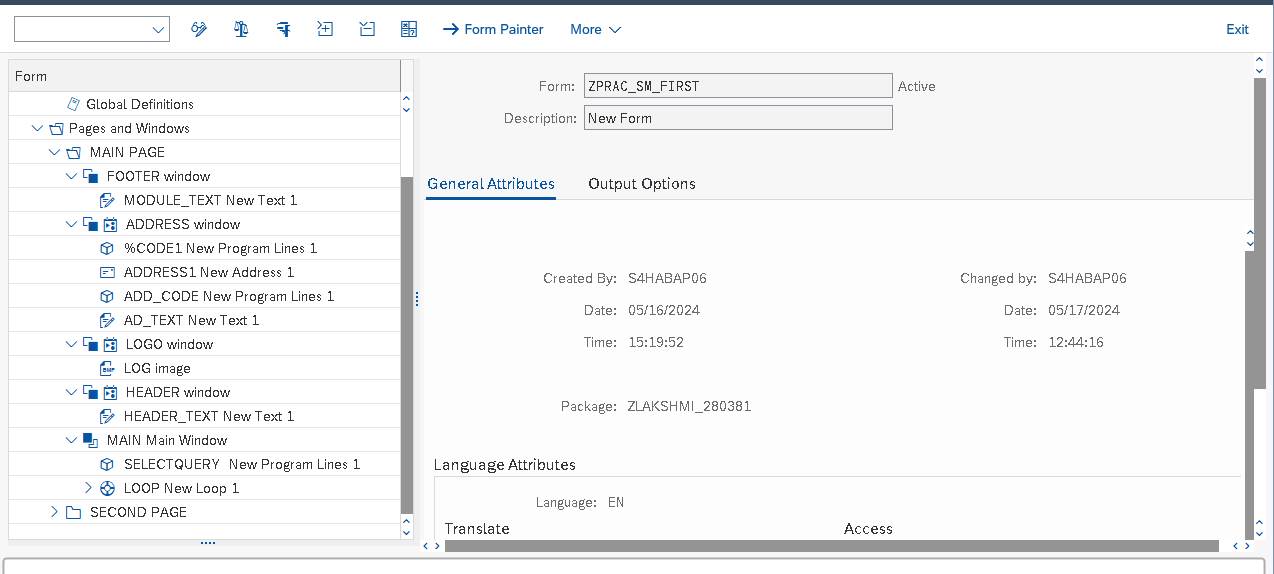
*\*&---------------------------------------------------------------------\**  
*\*& Report ZFM\_PRAC\_ADOBE3\_PROG*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZFM\_PRAC\_ADOBE3\_PROG.  
  
TABLES: vbak, vbap.  
  
INCLUDE ZFM\_PRAC\_ADOBE3\_TOP.  
  
DATA: lv\_vbeln type vbeln.  
  
PARAMETERS p\_vbeln type vbak-vbeln.  
  
SELECT-OPTIONS s\_vbeln for lv\_vbeln.  
  
  
START-OF-SELECTION.  
  
wa\_outputparams-device =  'PRINTER'.  
wa\_outputparams-dest  =   'LP01' .  
wa\_outputparams-nodialog =  'X'.  
wa\_outputparams-preview =  'X'.  
  
CALL FUNCTION 'FP\_JOB\_OPEN'  
  CHANGING  
    ie\_outputparams       = wa\_outputparams  
*\* EXCEPTIONS*  
*\*   CANCEL                = 1*  
*\*   USAGE\_ERROR           = 2*  
*\*   SYSTEM\_ERROR          = 3*  
*\*   INTERNAL\_ERROR        = 4*  
*\*   OTHERS                = 5*  
          .  
IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
ENDIF.  
  
CALL FUNCTION 'FP\_FUNCTION\_MODULE\_NAME'  
  EXPORTING  
    i\_name                     =  'ZFM\_PRAC\_ADOBE3'  
 IMPORTING  
   E\_FUNCNAME                 = gv\_fm\_nam  
*\*   E\_INTERFACE\_TYPE           =*  
*\*   EV\_FUNCNAME\_INBOUND        =*  
          .  
  
select single vbeln erdat vbtyp kunnr into wa\_tab from vbak                    *"""line items*  
  where vbeln eq p\_vbeln.  
  
select posnr matnr kwmeng vrkme from vbap  
  into table lt\_tab            *""header*  
  where vbeln in s\_vbeln.  
  
CALL FUNCTION gv\_fm\_nam *"'/1BCDWB/SM00000610'*  
  EXPORTING  
   /1BCDWB/DOCPARAMS        = wa\_docparams  
    wa\_tab                   =  wa\_tab  
    lt\_tab                   =  lt\_tab  
    i\_vbeln1                 =  s\_vbeln-low  
    i\_vbeln2                 =  s\_vbeln-high  
*\* IMPORTING*  
*\*   /1BCDWB/FORMOUTPUT       =*  
*\* EXCEPTIONS*  
*\*   USAGE\_ERROR              = 1*  
*\*   SYSTEM\_ERROR             = 2*  
*\*   INTERNAL\_ERROR           = 3*  
*\*   OTHERS                   = 4*  
          .  
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.

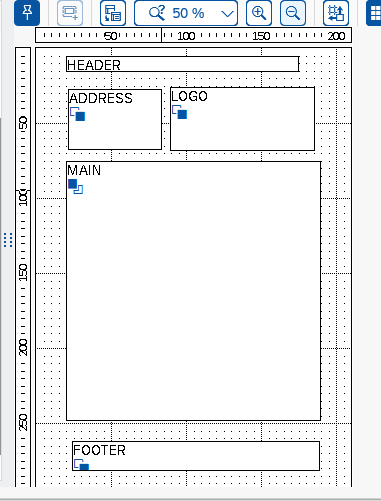


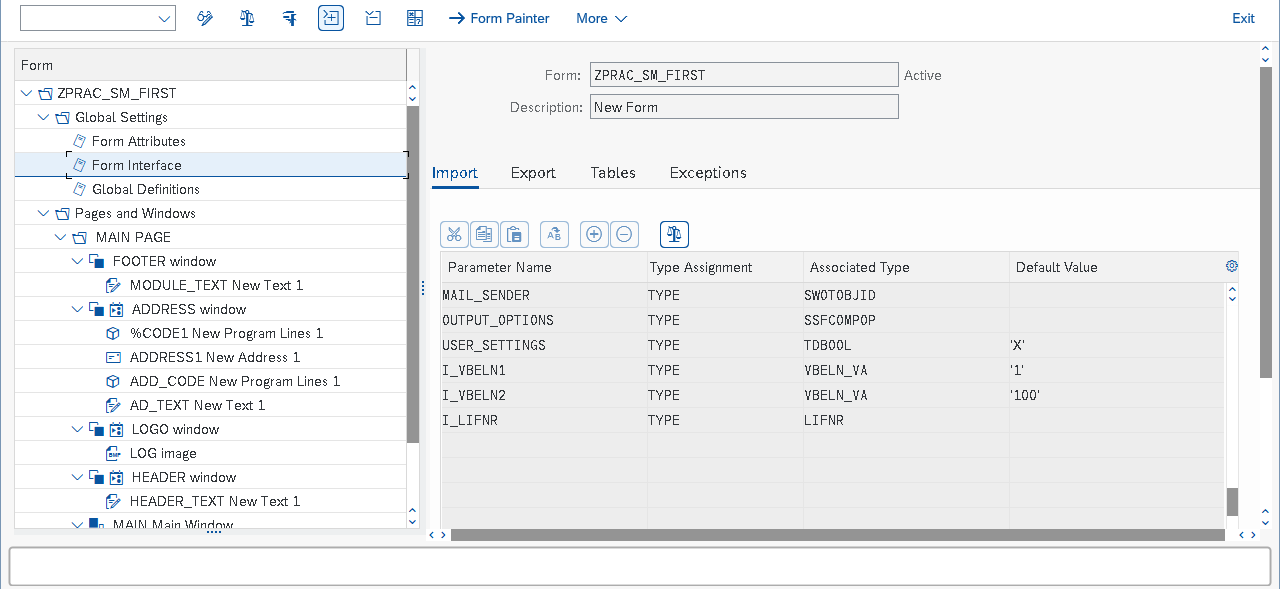


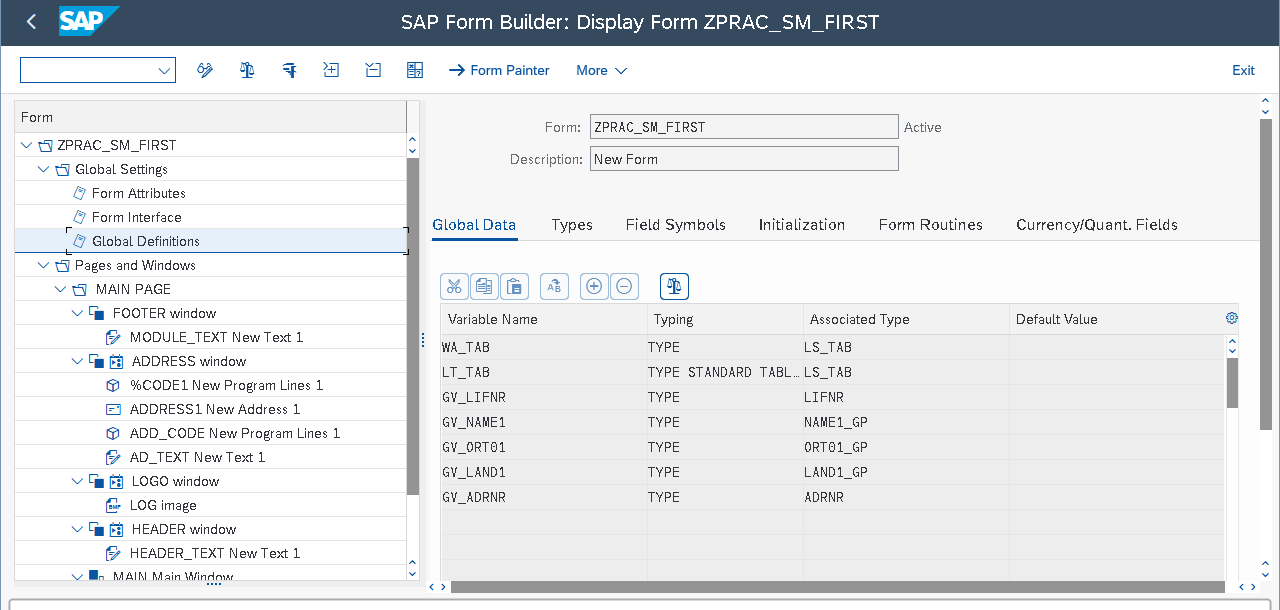


*\*&---------------------------------------------------------------------\**  
*\*& Report ZFM\_PRAC\_ADOBE2\_PROG*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZFM\_PRAC\_ADOBE2\_PROG.  
  
INCLUDE ZFM\_PRAC\_ADOBE2\_PROG\_TOP.  
  
PARAMETERS: p\_matnr type matnr.  
  
START-OF-SELECTION.  
  
wa\_outputparams-device =  'PRINTER'.  
wa\_outputparams-dest  =   'LP01' .  
wa\_outputparams-nodialog =  'X'.  
wa\_outputparams-preview =  'X'.  
  
CALL FUNCTION 'FP\_JOB\_OPEN'  
  CHANGING  
    ie\_outputparams       = wa\_outputparams  
*\* EXCEPTIONS*  
*\*   CANCEL                = 1*  
*\*   USAGE\_ERROR           = 2*  
*\*   SYSTEM\_ERROR          = 3*  
*\*   INTERNAL\_ERROR        = 4*  
*\*   OTHERS                = 5*  
          .  
IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
ENDIF.  
  
CALL FUNCTION 'FP\_FUNCTION\_MODULE\_NAME'  
  EXPORTING  
    i\_name                     =   'ZFM\_PRAC\_ADOBE2'  
 IMPORTING  
   E\_FUNCNAME                 =   gv\_fm\_nam  
*\*   E\_INTERFACE\_TYPE           =*  
*\*   EV\_FUNCNAME\_INBOUND        =*  
          .  
  
wa\_docparams-langu  ='E'.  
wa\_docparams-country  ='IN'.  
  
select single matnr mbrsh matkl meins from mara  
  into (gv\_matnr, gv\_mbrsh, gv\_matkl, gv\_meins)  
  where matnr eq p\_matnr.  
  
  CALL FUNCTION '/1BCDWB/SM00000095'  
    EXPORTING  
   /1BCDWB/DOCPARAMS        =   wa\_docparams  
      matnr                    =  gv\_matnr  
      mbrsh                    =  gv\_mbrsh  
      matkl                    =  gv\_matkl  
      meins                    =  gv\_meins  
*\*   IMPORTING*  
*\*     /1BCDWB/FORMOUTPUT       =*  
*\*   EXCEPTIONS*  
*\*     USAGE\_ERROR              = 1*  
*\*     SYSTEM\_ERROR             = 2*  
*\*     INTERNAL\_ERROR           = 3*  
*\*     OTHERS                   = 4*  
            .  
  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.

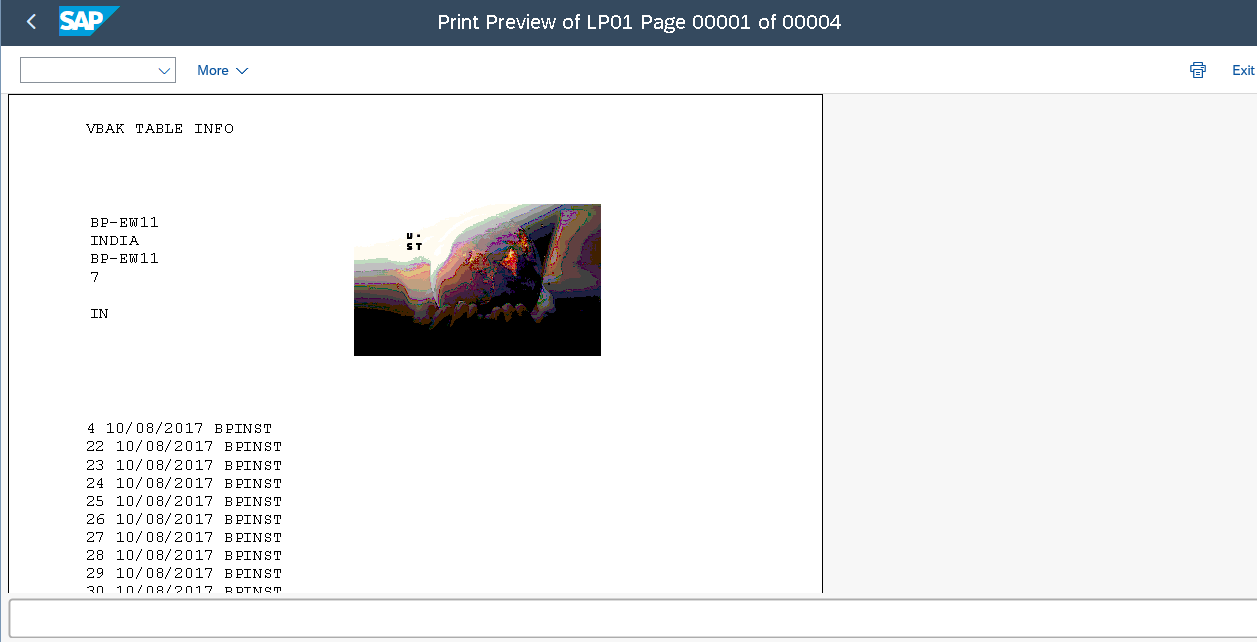
SMARTFORMS  


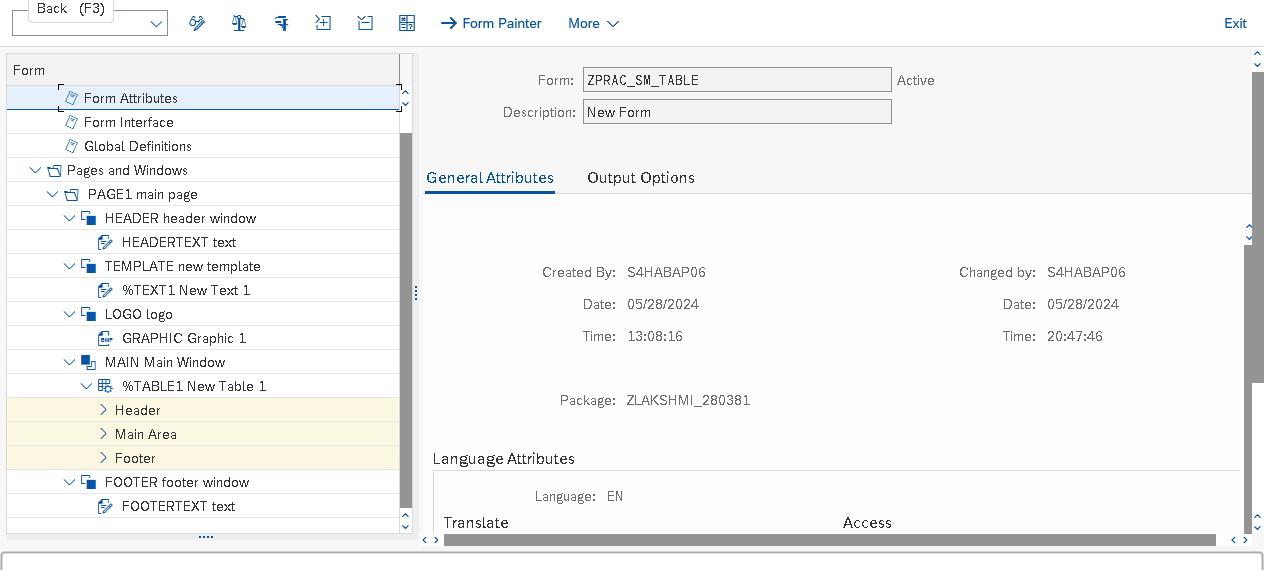


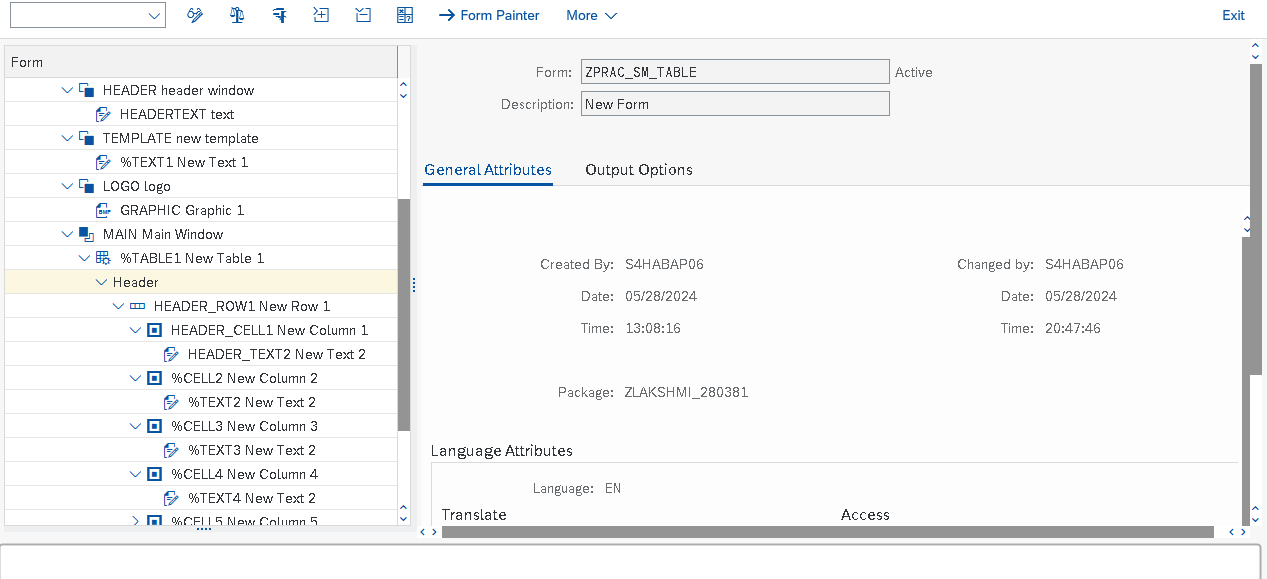


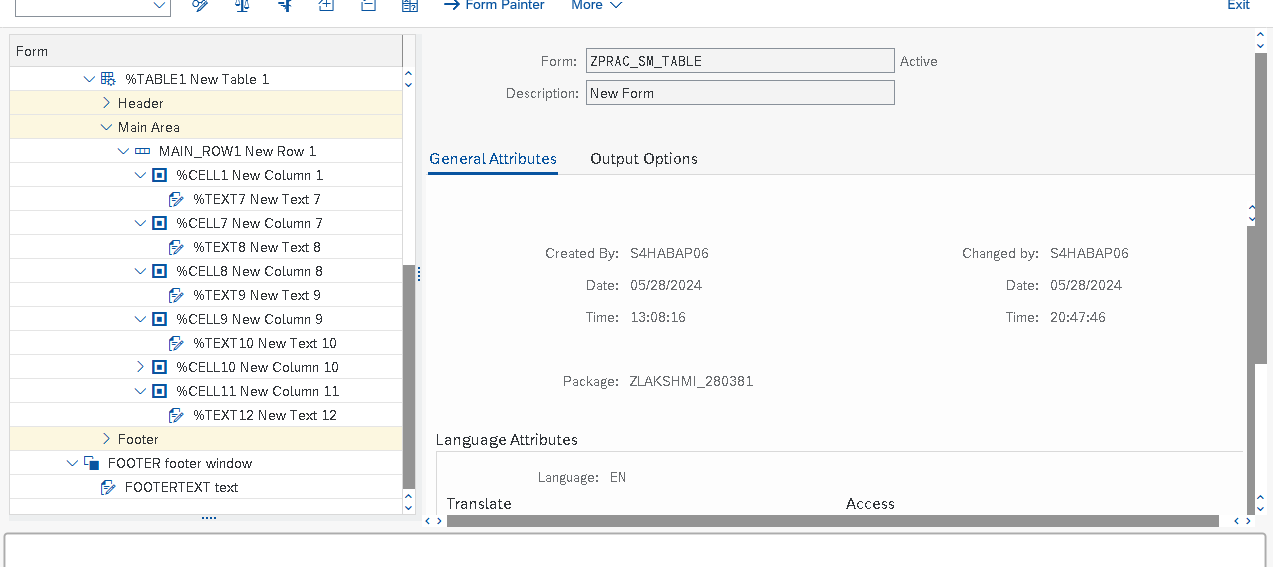


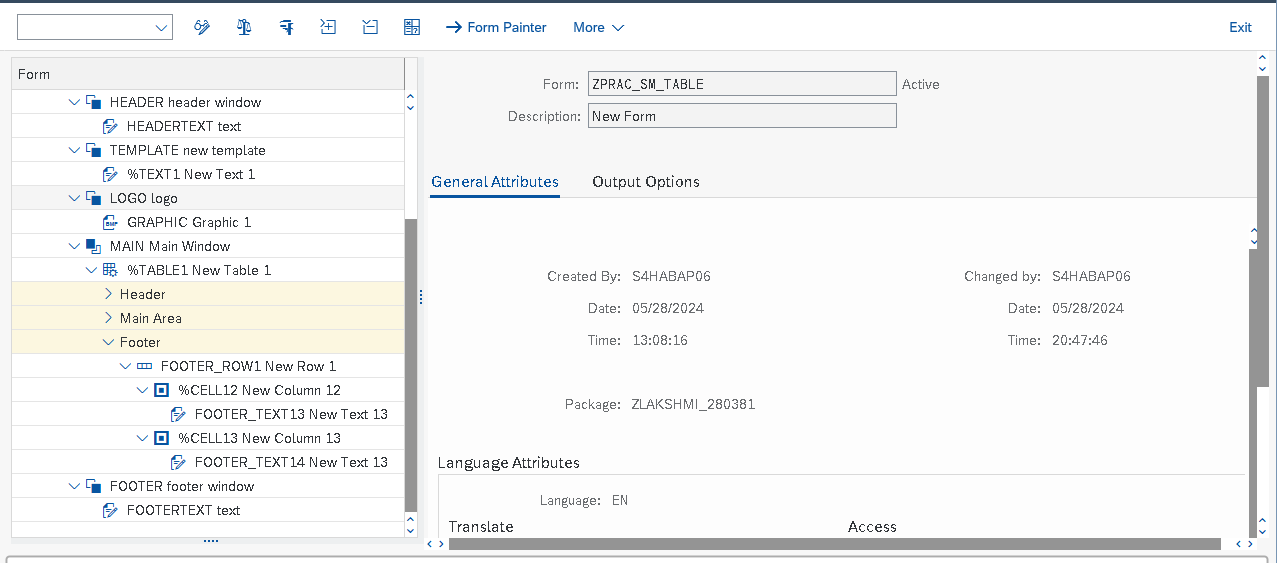
*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_SM\_FIRST\_PROG*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_SM\_FIRST\_PROG.  
  
DATA: LV\_FM\_NAME type RS38L\_FNAM,  
      lv\_vbeln type vbeln\_va.                       *""local variable for function module*  
  
select-OPTIONS: s\_vbeln for lv\_vbeln.  
PARAMETERS p\_lifnr type lifnr.  
  
START-OF-SELECTION.  
  
CALL FUNCTION 'SSF\_FUNCTION\_MODULE\_NAME'  
  EXPORTING  
    formname                 = 'ZPRAC\_SM\_FIRST'  
*\*   VARIANT                  = ' '*  
*\*   DIRECT\_CALL              = ' '*  
 IMPORTING  
   FM\_NAME                  =  LV\_FM\_NAME               *"/1BCDWB/SF00000060*  
*\* EXCEPTIONS*  
*\*   NO\_FORM                  = 1*  
*\*   NO\_FUNCTION\_MODULE       = 2*  
*\*   OTHERS                   = 3*  
          .  
IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
ENDIF.  
  
*\*\*\*\*\*\*\*\*CALL FUNCTION LV\_FM\_NAME                                 "'/1BCDWB/SF00000060'*  
*\*\*\*\*\*\*\*\* EXPORTING*  
*\*\*\*\*\*\*\*\*\*   ARCHIVE\_INDEX              =*  
*\*\*\*\*\*\*\*\*\*   ARCHIVE\_INDEX\_TAB          =*  
*\*\*\*\*\*\*\*\*\*   ARCHIVE\_PARAMETERS         =*  
*\*\*\*\*\*\*\*\*\*   CONTROL\_PARAMETERS         =*  
*\*\*\*\*\*\*\*\*\*   MAIL\_APPL\_OBJ              =*  
*\*\*\*\*\*\*\*\*\*   MAIL\_RECIPIENT             =*  
*\*\*\*\*\*\*\*\*\*   MAIL\_SENDER                =*  
*\*\*\*\*\*\*\*\*\*   OUTPUT\_OPTIONS             =*  
*\*\*\*\*\*\*\*\*\*   USER\_SETTINGS              = 'X'*  
*\*\*\*\*\*\*\*\*   I\_VBELN1                   = S\_VBELN-low*  
*\*\*\*\*\*\*\*\*   I\_VBELN2                   = S\_VBELN-high*  
*\*\*\*\*\*\*\*\*\* IMPORTING*  
*\*\*\*\*\*\*\*\*\*   DOCUMENT\_OUTPUT\_INFO       =*  
*\*\*\*\*\*\*\*\*\*   JOB\_OUTPUT\_INFO            =*  
*\*\*\*\*\*\*\*\*\*   JOB\_OUTPUT\_OPTIONS         =*  
*\*\*\*\*\*\*\*\*\* EXCEPTIONS*  
*\*\*\*\*\*\*\*\*\*   FORMATTING\_ERROR           = 1*  
*\*\*\*\*\*\*\*\*\*   INTERNAL\_ERROR             = 2*  
*\*\*\*\*\*\*\*\*\*   SEND\_ERROR                 = 3*  
*\*\*\*\*\*\*\*\*\*   USER\_CANCELED              = 4*  
*\*\*\*\*\*\*\*\*\*   OTHERS                     = 5*  
*\*\*\*\*\*\*\*\*          .*  
*\*\*\*\*\*\*\*\*IF sy-subrc <> 0.*  
*\*\*\*\*\*\*\*\*\* Implement suitable error handling here*  
*\*\*\*\*\*\*\*\*ENDIF.*  
  
CALL FUNCTION LV\_FM\_NAME                                *"'/1BCDWB/SF00000060'*  
  EXPORTING  
*\*   ARCHIVE\_INDEX              =*  
*\*   ARCHIVE\_INDEX\_TAB          =*  
*\*   ARCHIVE\_PARAMETERS         =*  
*\*   CONTROL\_PARAMETERS         =*  
*\*   MAIL\_APPL\_OBJ              =*  
*\*   MAIL\_RECIPIENT             =*  
*\*   MAIL\_SENDER                =*  
*\*   OUTPUT\_OPTIONS             =*  
*\*   USER\_SETTINGS              = 'X'*  
   I\_VBELN1                   = S\_VBELN-low  
   I\_VBELN2                   = S\_VBELN-high  
    i\_lifnr                    = P\_LIFNR  
*\* IMPORTING*  
*\*   DOCUMENT\_OUTPUT\_INFO       =*  
*\*   JOB\_OUTPUT\_INFO            =*  
*\*   JOB\_OUTPUT\_OPTIONS         =*  
*\* EXCEPTIONS*  
*\*   FORMATTING\_ERROR           = 1*  
*\*   INTERNAL\_ERROR             = 2*  
*\*   SEND\_ERROR                 = 3*  
*\*   USER\_CANCELED              = 4*  
*\*   OTHERS                     = 5*  
          .  
IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
ENDIF.

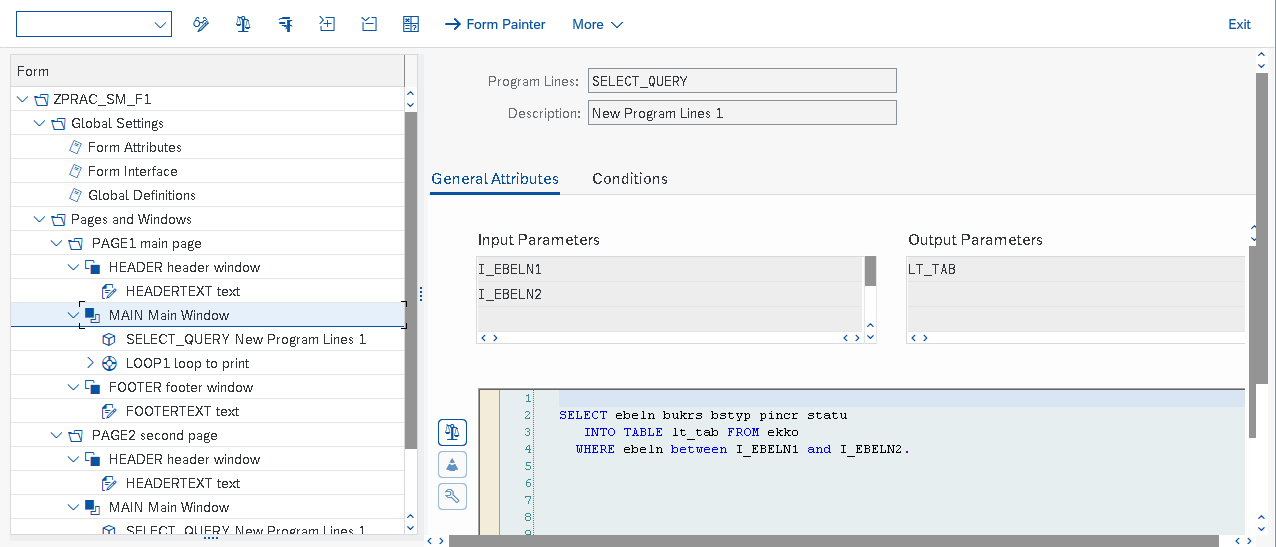


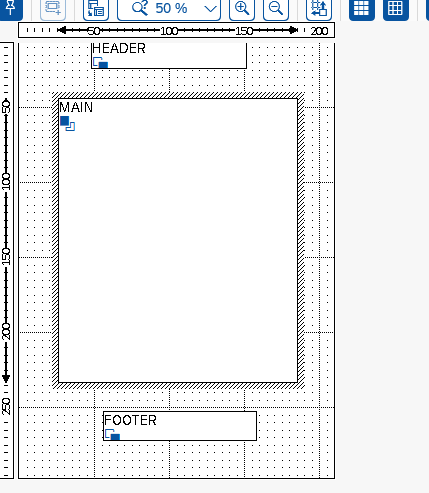


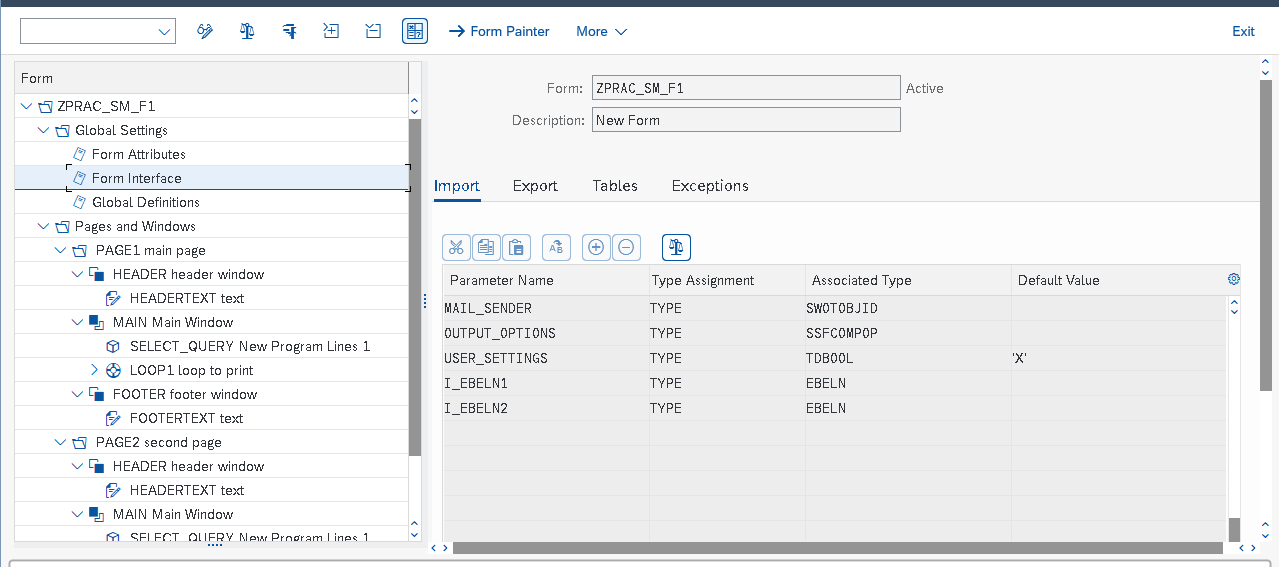


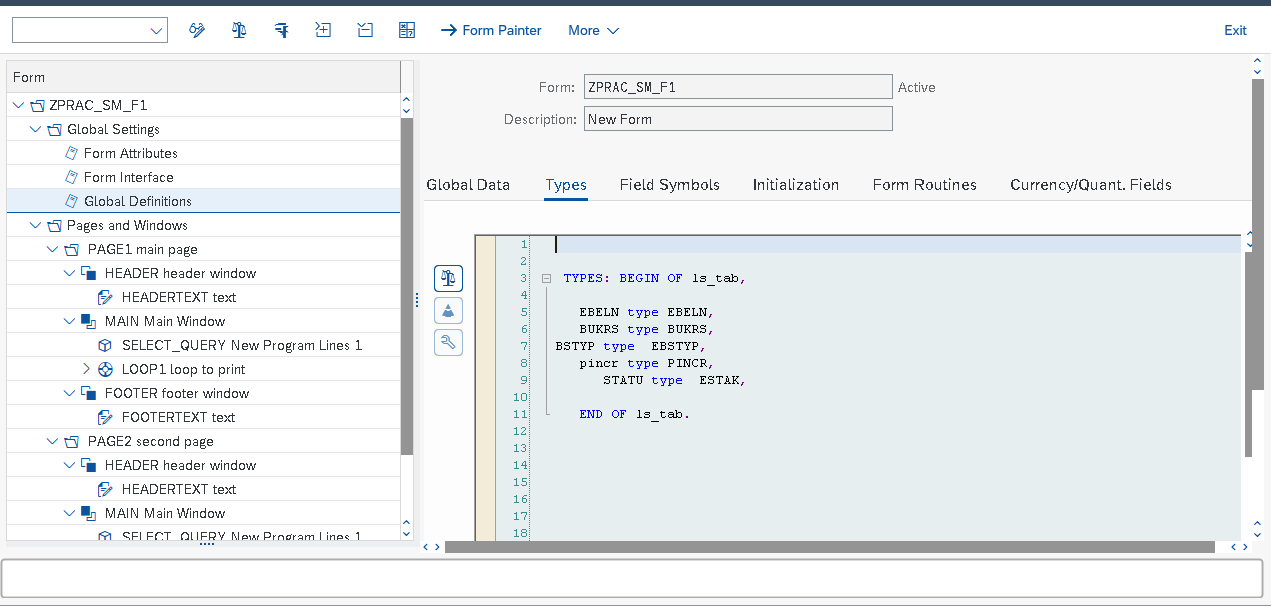




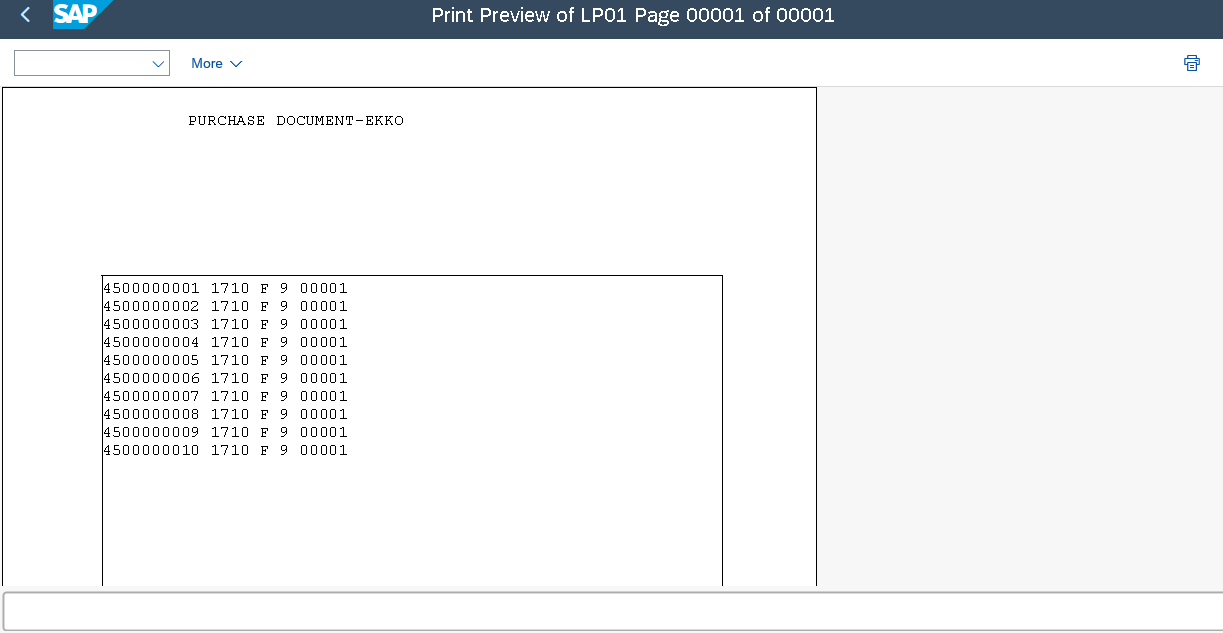




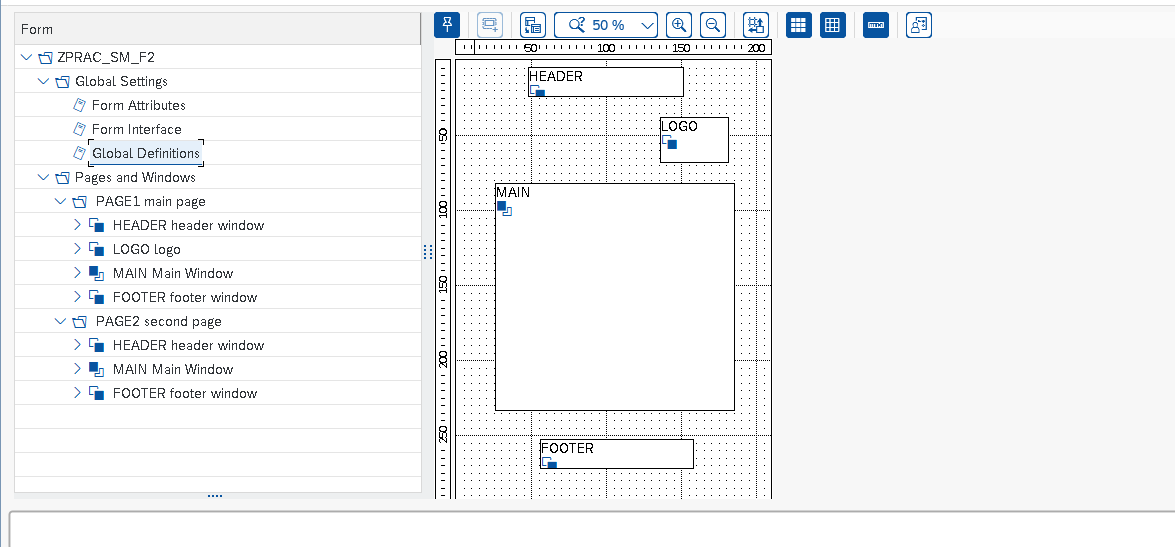


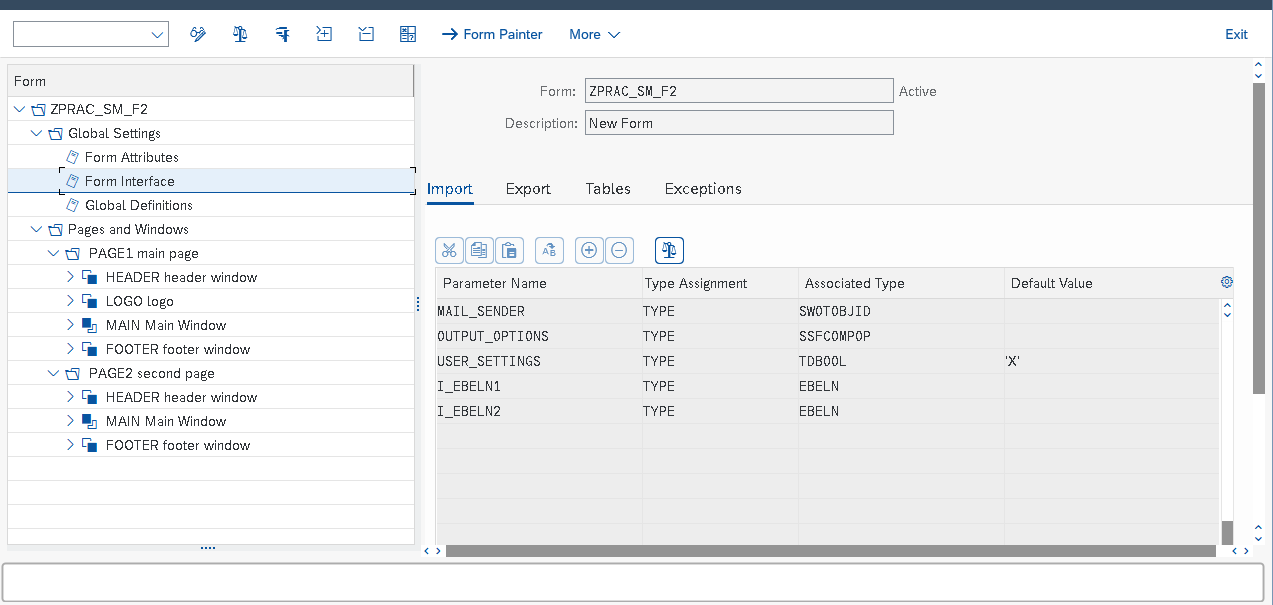


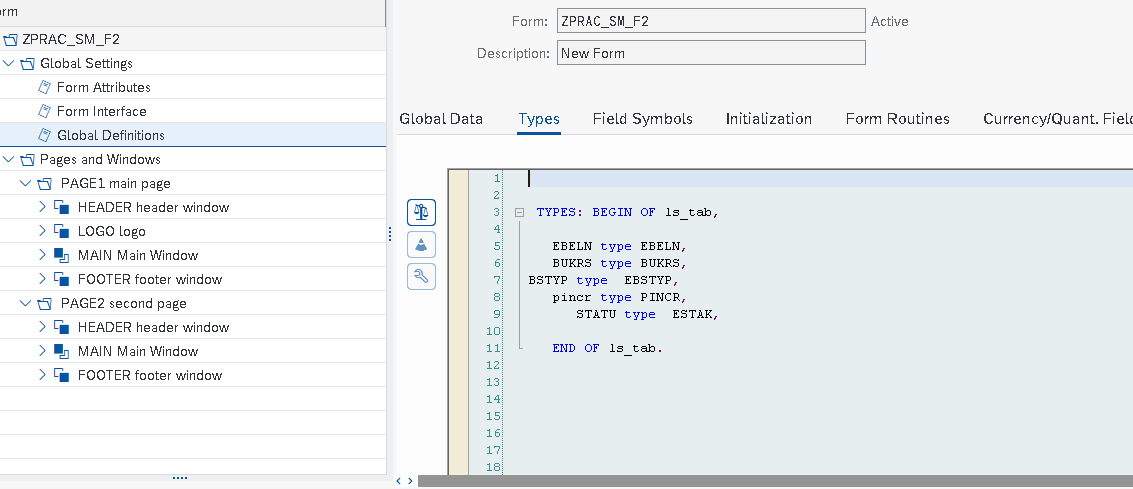
*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_SM\_F2\_PROG*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_SM\_F1\_PROG.  
  
DATA: LV\_FM\_NAME type RS38L\_FNAM,  
      lv\_ebeln type ebeln.                       *""local variable for function module*  
  
select-OPTIONS: s\_ebeln for lv\_ebeln.  
  
START-OF-SELECTION.  
  
CALL FUNCTION 'SSF\_FUNCTION\_MODULE\_NAME'  
  EXPORTING  
    formname                 =  'ZPRAC\_SM\_F1'  
*\*   VARIANT                  = ' '*  
*\*   DIRECT\_CALL              = ' '*  
 IMPORTING  
   FM\_NAME                  =   LV\_FM\_NAME  
*\* EXCEPTIONS*  
*\*   NO\_FORM                  = 1*  
*\*   NO\_FUNCTION\_MODULE       = 2*  
*\*   OTHERS                   = 3*  
          .  
IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
ENDIF.  
  
CALL FUNCTION LV\_FM\_NAME         *""'/1BCDWB/SF00000168'*  
  EXPORTING  
*\*   ARCHIVE\_INDEX              =*  
*\*   ARCHIVE\_INDEX\_TAB          =*  
*\*   ARCHIVE\_PARAMETERS         =*  
*\*   CONTROL\_PARAMETERS         =*  
*\*   MAIL\_APPL\_OBJ              =*  
*\*   MAIL\_RECIPIENT             =*  
*\*   MAIL\_SENDER                =*  
*\*   OUTPUT\_OPTIONS             =*  
*\*   USER\_SETTINGS              = 'X'*  
    i\_ebeln1                   =  S\_EBELN-LOW  
    i\_ebeln2                   =  S\_EBELN-HIGH  
*\* IMPORTING*  
*\*   DOCUMENT\_OUTPUT\_INFO       =*  
*\*   JOB\_OUTPUT\_INFO            =*  
*\*   JOB\_OUTPUT\_OPTIONS         =*  
*\* EXCEPTIONS*  
*\*   FORMATTING\_ERROR           = 1*  
*\*   INTERNAL\_ERROR             = 2*  
*\*   SEND\_ERROR                 = 3*  
*\*   USER\_CANCELED              = 4*  
*\*   OTHERS                     = 5*  
  
          .  
IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
ENDIF.



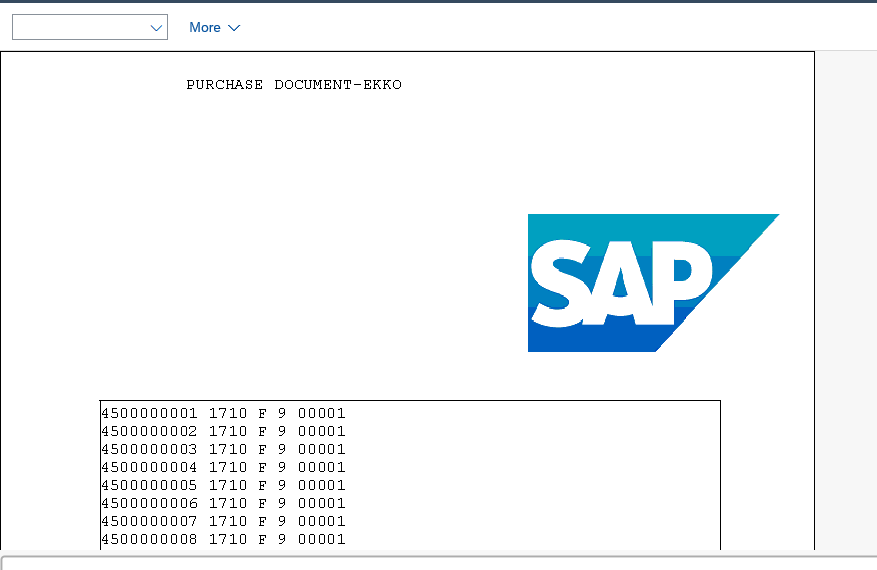
Smart form 2







*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_SM\_F2\_PROG*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_SM\_F2\_PROG.  
  
DATA: LV\_FM\_NAME type RS38L\_FNAM,  
      lv\_ebeln type ebeln.                       *""local variable for function module*  
  
select-OPTIONS: s\_ebeln for lv\_ebeln.  
  
START-OF-SELECTION.  
  
CALL FUNCTION 'SSF\_FUNCTION\_MODULE\_NAME'  
  EXPORTING  
    formname                 =  'ZPRAC\_SM\_F2'  
*\*   VARIANT                  = ' '*  
*\*   DIRECT\_CALL              = ' '*  
 IMPORTING  
   FM\_NAME                  =   LV\_FM\_NAME  
*\* EXCEPTIONS*  
*\*   NO\_FORM                  = 1*  
*\*   NO\_FUNCTION\_MODULE       = 2*  
*\*   OTHERS                   = 3*  
          .  
IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
ENDIF.  
  
CALL FUNCTION LV\_FM\_NAME         *""'/1BCDWB/SF00000168'*  
  EXPORTING  
*\*   ARCHIVE\_INDEX              =*  
*\*   ARCHIVE\_INDEX\_TAB          =*  
*\*   ARCHIVE\_PARAMETERS         =*  
*\*   CONTROL\_PARAMETERS         =*  
*\*   MAIL\_APPL\_OBJ              =*  
*\*   MAIL\_RECIPIENT             =*  
*\*   MAIL\_SENDER                =*  
*\*   OUTPUT\_OPTIONS             =*  
*\*   USER\_SETTINGS              = 'X'*  
    i\_ebeln1                   =  S\_EBELN-LOW  
    i\_ebeln2                   =  S\_EBELN-HIGH  
*\* IMPORTING*  
*\*   DOCUMENT\_OUTPUT\_INFO       =*  
*\*   JOB\_OUTPUT\_INFO            =*  
*\*   JOB\_OUTPUT\_OPTIONS         =*  
*\* EXCEPTIONS*  
*\*   FORMATTING\_ERROR           = 1*  
*\*   INTERNAL\_ERROR             = 2*  
*\*   SEND\_ERROR                 = 3*  
*\*   USER\_CANCELED              = 4*  
*\*   OTHERS                     = 5*  
  
          .  
IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
ENDIF.

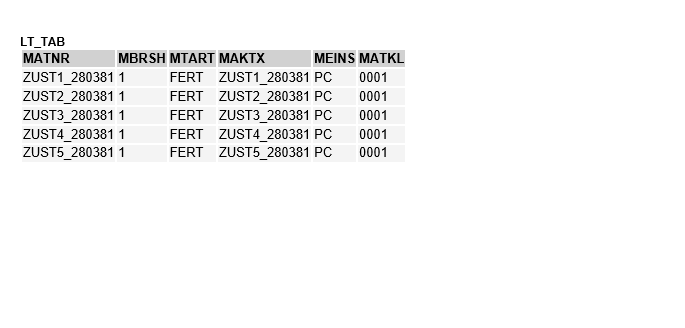


FILE HANDLING

Upload material from local server to SAP 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.  
DATA wa\_tab type ls\_tab.  
  
PARAMETERS p\_file type rlgrap-filename.  
  
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 p\_file1 TYPE string.  
 p\_file1 = p\_file.  
  
 CALL FUNCTION 'GUI\_UPLOAD'  
   EXPORTING  
     filename                      = p\_file1              "Parameter for file 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                      = lt\_tab                   "internal table  
\*  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.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   Writing or creating file  
  
 OPEN DATASET 'sample' for output in text mode encoding default.  
 loop at lt\_tab into wa\_tab.  
   transfer wa\_tab to 'sample'.  
   clear wa\_tab.  
  
  ENDLOOP.  
  
  close dataset 'sample'.  
  
cl\_demo\_output=>display\_data( lt\_tab ).

OUTPUT:



Uploading xls file local to SAP 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.  
DATA wa\_tab TYPE ls\_tab.  
  
PARAMETERS p\_file TYPE rlgrap-filename.  
  
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 lt\_raw TYPE truxs\_t\_text\_data.  
  
  CALL FUNCTION 'TEXT\_CONVERT\_XLS\_TO\_SAP'  
    EXPORTING  
      i\_field\_seperator    = 'X'  
      i\_line\_header        = 'X'  
      i\_tab\_raw\_data       = LT\_RAW  
     I\_FILENAME           = p\_file  
\*     I\_STEP               = 1  
\*     I\_FILENAME\_LONG      =  
    TABLES  
      i\_tab\_converted\_data = LT\_TAB  
\* EXCEPTIONS  
\*     CONVERSION\_FAILED    = 1  
\*     OTHERS               = 2  
    .  
  IF sy-subrc <> 0.  
\* Implement suitable error handling here  
  ENDIF.  
  
  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   Writing or creating file  
  
  OPEN DATASET 'MAT\_280381' FOR OUTPUT IN TEXT MODE ENCODING DEFAULT.  
  LOOP AT lt\_tab INTO wa\_tab.  
    TRANSFER wa\_tab TO 'MAT\_280381'.  
    CLEAR wa\_tab.  
  
  ENDLOOP.  
  
  CLOSE DATASET 'MAT\_280381'.  
  
  cl\_demo\_output=>display\_data( LT\_TAB ).

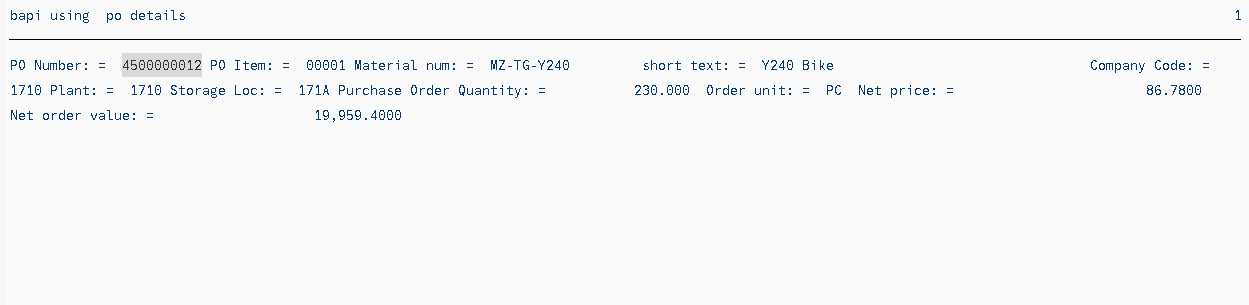
CUSTOM BAPIS RFC prog:

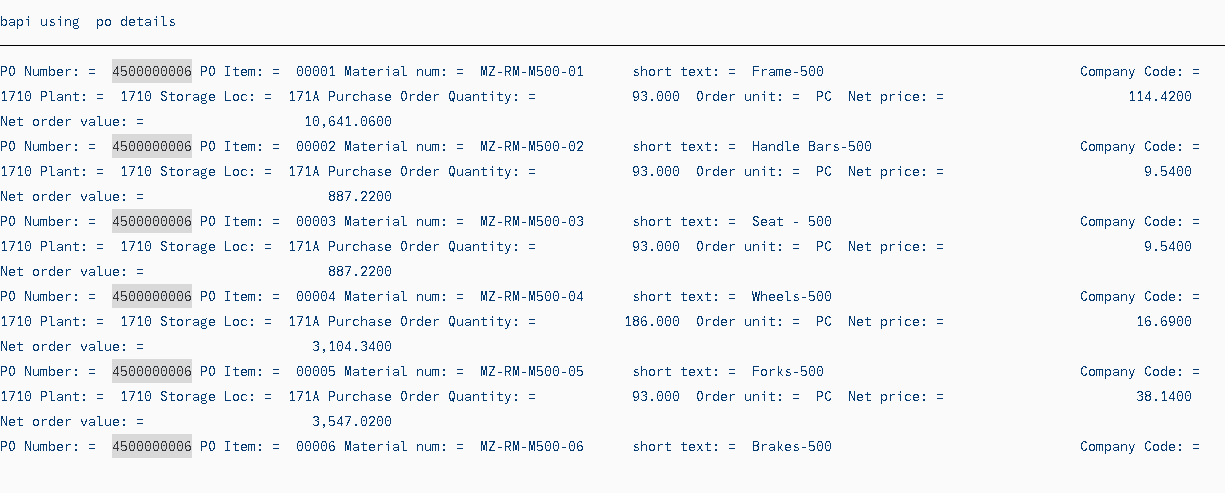
*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_FUNCMOD\_RFC\_PROG*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZPRAC\_FUNCMOD\_RFC\_PROG.  
  
DATA lv\_matnr type matnr.  
DATA: lt\_tab type table of mara,  
      wa\_tab type mara.  
select-OPTIONS s\_matnr for lv\_matnr.  
  
CALL FUNCTION 'ZPRAC\_FUNCMOD\_RFC'  DESTINATION 'ZPRAC\_RFC\_CON'  
 EXPORTING  
   I\_MATNR1       = S\_MATNR-low  
   I\_MATNR2       = S\_MATNR-high  
  TABLES  
    lt\_tab         = LT\_TAB  
          .  
  
LOOP AT lt\_tab into wa\_tab.  
  
  write:/ wa\_tab-mandt, wa\_tab-matnr, wa\_tab-mtart.  
  
ENDLOOP.

BAPIs using table Purchase Order Details,

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_BAPI\_PO\_DETAILS*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT zprac\_bapi\_po\_details.  
  
DATA: BEGIN OF it\_tab OCCURS 0.  
        INCLUDE STRUCTURE bapiekpo.               *""EKPO structure for ref*  
DATA: END OF it\_tab.  
  
PARAMETERS p\_ebeln TYPE ekko-ebeln.  
  
START-OF-SELECTION.  
  
  CALL FUNCTION 'BAPI\_PO\_GETDETAIL'  
    EXPORTING  
      purchaseorder = p\_ebeln  
    TABLES  
      po\_items      = it\_tab.  
  
  LOOP AT it\_tab.  
    WRITE: / 'PO Number: = ', it\_tab-po\_number COLOR COL\_HEADING,  
            'PO Item: = ', it\_tab-PO\_Item,  
            'Material num: = ', it\_tab-material,  
            'short text: = ', it\_tab-short\_text,  
            'Company Code: = ', it\_tab-co\_code,  
            'Plant: = ', it\_tab-plant,  
            'Storage Loc: = ', it\_tab-store\_loc,  
            'Purchase Order Quantity: = ', it\_tab-quantity,  
            'Order unit: = ', it\_tab-unit,  
            'Net price: = ', it\_tab-net\_price,  
            'Net order value: = ', it\_tab-net\_value.  
  
  ENDLOOP.

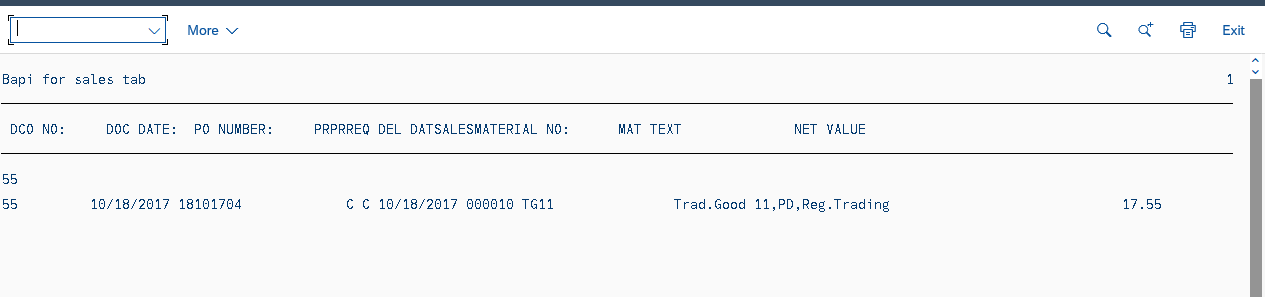
*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_BAPI\_PO\_DETAILS*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT zprac\_bapi\_po\_details.  
  
DATA: BEGIN OF it\_tab OCCURS 0.  
        INCLUDE STRUCTURE bapiekpo.               *""EKPO structure for ref*  
DATA: END OF it\_tab.  
  
PARAMETERS p\_ebeln TYPE ekko-ebeln.  
  
START-OF-SELECTION.  
  
  CALL FUNCTION 'BAPI\_PO\_GETDETAIL'  
    EXPORTING  
      purchaseorder = p\_ebeln  
    TABLES  
      po\_items      = it\_tab.  
  
  LOOP AT it\_tab.  
    WRITE: / 'PO Number: = ', it\_tab-po\_number COLOR COL\_HEADING,  
            'PO Item: = ', it\_tab-PO\_Item,  
            'Material num: = ', it\_tab-material,  
            'short text: = ', it\_tab-short\_text,  
            'Company Code: = ', it\_tab-co\_code,  
            'Plant: = ', it\_tab-plant,  
            'Storage Loc: = ', it\_tab-store\_loc,  
            'Purchase Order Quantity: = ', it\_tab-quantity,  
            'Order unit: = ', it\_tab-unit,  
            'Net price: = ', it\_tab-net\_price,  
            'Net order value: = ', it\_tab-net\_value.  
  
  ENDLOOP.





BAPI using table SALES TABLE,

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_BAPI\_SALES\_DEATILS*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT zprac\_bapi\_sales\_deatils.  
  
TABLES vbap.  
  
DATA: wa\_return TYPE bapireturn,  
      wa\_tab    TYPE bapisdstat.  
  
DATA it\_tab1 LIKE STANDARD TABLE OF wa\_tab.  
  
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.  
  
PARAMETERS p\_vbeln TYPE vbap-vbeln.  
  
START-OF-SELECTION.  
  
  SELECT matnr maktx FROM makt INTO TABLE it\_tab2.  
  
  CALL FUNCTION 'BAPI\_SALESORDER\_GETSTATUS'  
    EXPORTING  
      salesdocument = p\_vbeln  
    IMPORTING  
      return        = wa\_return  
    TABLES  
      statusinfo    = it\_tab1.  
  
  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 it\_tab1 INTO wa\_tab.  
      WRITE: wa\_TAB-doc\_number.  
  
      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 it\_tab1[].  
  ELSE.  
    WRITE wa\_return-message.  
  ENDIF.



BAPIS using table material master,

*\*&---------------------------------------------------------------------\**  
*\*& Report ZPRAC\_BAPI\_MATMASTER*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT zprac\_bapi\_matmaster.  
  
TABLES: bapimathead,      *"HEADER segment with control information*  
        bapi\_mara,        *"Material Data at client level*  
        bapi\_marax,       *"checkbox structure for mara bapi.*  
        bapi\_marc,        *"Material Data at plant level*  
        bapi\_marcx,       *"checkbox structure for marc bapi*  
        bapi\_makt,        *"Material descriptions*  
        bapiret2.         *"return parameter bapi to return the values.*  
  
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.  
  
TYPES: BEGIN OF ls\_tab1,  
         langu(1),  
         langu\_iso(2),  
         matl\_desc(40),  
         del\_flag(1),  
       END OF ls\_tab1.  
  
DATA: lt\_tab1 TYPE STANDARD TABLE OF ls\_tab1,  
      wa\_tab1 TYPE ls\_tab1.  
  
DATA: BEGIN OF it\_return OCCURS 0.  
  
        INCLUDE STRUCTURE bapiret2.  
  
DATA: END OF it\_return.  
  
PARAMETERS p\_file TYPE rlgrap-filename.  
  
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 p\_file1 TYPE string.  
  p\_file1 = p\_file.  
  
  CALL FUNCTION 'GUI\_UPLOAD'  
    EXPORTING  
      filename            = p\_file1  
      filetype            = 'ASC'  
      has\_field\_separator = 'X'  
    TABLES  
      data\_tab            = 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\_mara-matl\_group  = wa\_tab-matkl.  
    bapi\_mara-base\_uom  = wa\_tab-meins.  
  
    bapi\_marax-matl\_group  = 'X'.  
    bapi\_marax-base\_uom  = 'X'.  
  
    wa\_tab1-langu = 'E'.  
    wa\_tab1-matl\_desc = wa\_tab-maktx.  
    APPEND wa\_tab1 TO lt\_tab1.  
  
  
    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\_tab1.  
  
    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.

OUTPUT:

Create notepad file with mara to upload. And after entering input it shows status message or error messages and materials will insert in the mara.