

```
*&-----*
*& Report ZERCAN_ABAP_0008
*&-----*
*&
*&-----*
REPORT ZERCAN_ABAP_0008.

INCLUDE ZERCAN_ABAP_0008_TOP.
INCLUDE ZERCAN_ABAP_0008_FRM.
INCLUDE ZERCAN_ABAP_0008_PBO.
INCLUDE ZERCAN_ABAP_0008_PAI.
INCLUDE ZERCAN_ABAP_0008_ALV.

START-OF-SELECTION.

CALL SCREEN '100'.

CONTROLS: TBCONTROL TYPE TABLEVIEW USING SCREEN 0102.
```

```
*&-----*
*& Include          ZERCAN_ABAP_0008_TOP
*&-----*
```

CONTROLS TB_ID TYPE TABSTRIP.

TB_ID-ACTIVETAB = 'TAB1'.

DATA: GV_FATUNO TYPE ZEO_SIPAMAS-FATURA_NO,
GV_CARI TYPE ZEO_SIPAMAS-CARI,
GV_TARIH TYPE ZEO_SIPAMAS-TARIH,
GV_TESTARIH TYPE ZEO_SIPAMAS-TESLIM_TARIHI,
GV_ISTTARIH TYPE ZEO_SIPAMAS-ISTENILEN_TESLIM_TARIHI,
GV_DOVTARIH TYPE ZEO_SIPAMAS-DOVIZ_BAZ_TARIH,
GV_TIPI TYPE ZEO_SIPAMAS-TIPI,
GV_IHIR TYPE ZEO_SIPAMAS-ITHALAT_IHRACAT,
GV_EXP TYPE ZEO_SIPAMAS-EXPREFNO,
GV_OZEL TYPE ZEO_SIPAMAS-OZEL_KOD,
GV_ACIKLAMA TYPE ZEO_SIPAMAS-ACIKLAMA,
GV_KDV TYPE ZEO_SIPAMAS-KDV_DAHILMI.

DATA : LV_CAD TYPE CHAR200,
LV_CADRES TYPE CHAR200,
LV_ILCE TYPE CHAR15,
LV_IL TYPE CHAR20,
LV_VERDA TYPE CHAR20,
LV_VERNO TYPE CHAR20.

DATA: GV_FISNO TYPE ZEO_SIPATRA-FISNO,
GV_STOKKODU TYPE ZEO_SIPATRA-STOK_KODU,
GV_MIKTAR TYPE ZEO_SIPATRA-MIKTAR,
GV_DVZTIPI TYPE ZEO_SIPATRA-DOVIZ_TIPI,
GV_FIYAT TYPE ZEO_SIPATRA-FIYAT.

DATA:GV_ID TYPE VRM_ID,
GT_VALUES TYPE VRM_VALUES,
GS_VALUE TYPE VRM_VALUE.

DATA:GV_IDI TYPE VRM_ID,
GT_VALUESI TYPE VRM_VALUES,
GS_VALUEI TYPE VRM_VALUE.

DATA:GV_IDIHR TYPE VRM_ID,
GT_VALUESIHR TYPE VRM_VALUES,
GS_VALUEIHR TYPE VRM_VALUE.

DATA:GS_SIPAMAS TYPE ZEO_SIPAMAS.

DATA:GS_SIPATRA TYPE ZEO_SIPATRA.

DATA:GO_SLV TYPE REF TO CL_SALV_TABLE.

GV_TARIH = SY-DATUM.

GV_TESTARIH = SY-DATUM.

GV_ISTTARIH = SY-DATUM.

GV_DOVTARIH = SY-DATUM.

TYPES:

BEGIN OF TY_SIPATRA,
FISNO TYPE ZEO_SIPATRA-FISNO,
STOKKODU TYPE ZEO_SIPATRA-STOK_KODU,
MIKTAR TYPE ZEO_SIPATRA-MIKTAR,
DVZTIPI TYPE ZEO_SIPATRA-DOVIZ_TIPI,
FIYAT TYPE ZEO_SIPATRA-FIYAT,

END OF TY_SIPATRA.

DATA: WA_SIPATRA TYPE TY_SIPATRA,
IT_SIPATRA TYPE TABLE OF TY_SIPATRA.

DATA:GO_ALV TYPE REF TO CL_GUI_ALV_GRID.

TYPES: BEGIN OF GTY_SCARR,

FATURA_NO TYPE ZEO_SIPAMAS-FATURA_NO,
CARI TYPE ZEO_SIPAMAS-CARI,
TARIH TYPE ZEO_SIPAMAS-TARIH,
TIPI TYPE ZEO_SIPAMAS-TIPI,
FISNO TYPE ZEO_SIPATRA-FISNO,
STOK_KODU TYPE ZEO_SIPATRA-STOK_KODU,
MIKTAR TYPE ZEO_SIPATRA-MIKTAR,
DOVIZ_TIPI TYPE ZEO_SIPATRA-DOVIZ_TIPI,
FIYAT TYPE ZEO_SIPATRA-FIYAT,
LINE_COLOR TYPE CHAR4,

CELL_COLOR TYPE LVC_T_SCOL,
COST TYPE INT4,
LOCATION TYPE CHAR20,

SEATP TYPE CHAR10,
DD_HANDLE TYPE INT4,
CELLSTYLE TYPE LVC_T_STYL,
END OF GTY_SCARR.

DATA: GS_CELLSTYLE TYPE LVC_S_STYL.

DATA: GS_CELL_COLOR TYPE LVC_S_SCOL.

DATA: GT_SCARR TYPE TABLE OF GTY_SCARR,
GS_SCARR TYPE GTY_SCARR,
GT_FCAT TYPE LVC_T_FCAT,
GS_GCAT TYPE LVC_S_FCAT,
GS_LAYO TYPE LVC_S_LAYO.

FIELD-SYMBOLS: <GFS_FC> TYPE LVC_S_FCAT,
<GFS_SCARR> TYPE GTY_SCARR.

```
*&-----*
*& Include          ZERCAN_ABAP_0008_FRM
*&-----*
```

```
FORM SIPAMAS_TEMIZLE .
  GV_KDV          = ABAP_FALSE.
  CLEAR:
```

```
    GV_FATUNO    ,
    GV_CARI      ,
    GV_TARIH     ,
    GV_TESTARIH  ,
    GV_ISTTARIH  ,
    GV_DOVTARIH  ,
    GV_TIPI      ,
    GV_IHIR      ,
    GV_EXP       ,
    GV_OZEL      ,
    GV_ACIKLAMA  ,
    LV_CAD,
    LV_ILCE,
    LV_IL,
    LV_VERDA,
    LV_VERNO.
```

```
ENDFORM.
```

```
FORM SIPAMAS_KAYDET .
```

```
  GS_SIPAMAS-FATURA_NO      = GV_FATUNO.
  GS_SIPAMAS-CARI           = GV_CARI.
  GS_SIPAMAS-TARIH          = GV_TARIH.
  GS_SIPAMAS-TIPI           = GV_TIPI.
  GS_SIPAMAS-TESLIM_TARIHI  = GV_TESTARIH .
  GS_SIPAMAS-ISTENILEN_TESLIM_TARIHI = GV_ISTTARIH .
  GS_SIPAMAS-DOVIZ_BAZ_TARIH = GV_DOVTARIH .
  GS_SIPAMAS-ITHALAT_IHRACAT = GV_IHIR .
  GS_SIPAMAS-EXPREFNO       = GV_EXP .
  GS_SIPAMAS-OZEL_KOD       = GV_OZEL .
  GS_SIPAMAS-ACIKLAMA       = GV_ACIKLAMA .
  GS_SIPAMAS-KDV_DAHILMI    = GV_KDV.
```

```
  INSERT ZEO_SIPAMAS FROM GS_SIPAMAS.
  COMMIT WORK AND WAIT.
```

```
  MESSAGE 'Kayıt İşlemi Tamamlanmıştır' TYPE 'I'.
```

```
ENDFORM.
```

```
FORM SIPATRA_TEMIZLE .
```

```
  CLEAR:
```

```
    GV_FISNO    ,
    GV_STOKKODU ,
    GV_MIKTAR   ,
    GV_DVZTIPI  ,
    GV_FIYAT    .
```

```
ENDFORM.
```

```
FORM SIPATRA_KAYDET .
```

```
  GS_SIPATRA-FISNO      = GV_FISNO.
  GS_SIPATRA-DOVIZ_TIPI = GV_DVZTIPI.
  GS_SIPATRA-FIYAT      = GV_FIYAT.
  GS_SIPATRA-MIKTAR     = GV_MIKTAR.
  GS_SIPATRA-STOK_KODU  = GV_STOKKODU.
  INSERT ZEO_SIPATRA FROM GS_SIPATRA.
```

```
  COMMIT WORK AND WAIT.
```

```
  MESSAGE 'Kayıt İşlemi Tamamlanmıştır' TYPE 'I'.
```

```
ENDFORM.
```

```

*&-----*
*& Form CARI_BUL
*&-----*
*& text
*&-----*
*& --> p1      text
*& <-- p2      text
*&-----*
FORM CARI_BUL .

```

```

DATA: S_SRCH TYPE DSELC,
      T_SRCH TYPE STANDARD TABLE OF DSELC.

```

```

DATA : LV_CKODBUL TYPE ZEO_CASABIT-CARI_KODU,
       LV_CADBUL  TYPE ZEO_CASABIT-CARI_ADI,
       LV_CADRBUL TYPE ZEO_CASABIT-CARI_ADRES,
       LV_CLCEBUL TYPE ZEO_CASABIT-CARI_ILCE,
       LV_CLBUL   TYPE ZEO_CASABIT-CARI_IL,
       LV_CVDBUL  TYPE ZEO_CASABIT-CARI_VERGI_DAIRESI,
       LV_CVNBUL  TYPE ZEO_CASABIT-CARI_VERGI_NO.

```

```

DATA : LT_DEMO_TABLE TYPE TABLE OF ZEO_CASABIT,
      LT_RETURN      TYPE TABLE OF DDSHRETVAL.

```

```

" Eğer Üretim Yeri bilgisi girilmişse, girilen üretim yeri
" numarasına göre tablodaki veriler çekilir.

```

```

IF GV_CARI IS NOT INITIAL.
  SELECT * INTO CORRESPONDING FIELDS OF TABLE LT_DEMO_TABLE
  FROM ZEO_CASABIT WHERE CARI_KODU EQ GV_CARI.

```

```

" Eğer Üretim Yeri bilgisi girilmemişse, tablodaki tüm
" veriler çekilir.

```

```

ELSE.
  SELECT * INTO CORRESPONDING FIELDS OF TABLE LT_DEMO_TABLE
  FROM ZEO_CASABIT.
ENDIF.

```

```

S_SRCH-FLDNAME      = 'F0002'.
S_SRCH-DYFLDNAME    = 'LV_CKODBUL'.
APPEND S_SRCH TO T_SRCH.
CLEAR S_SRCH.

```

```

S_SRCH-FLDNAME      = 'F0003'.
S_SRCH-DYFLDNAME    = 'LV_CADBUL'.
APPEND S_SRCH TO T_SRCH.
CLEAR S_SRCH.

```

```

S_SRCH-FLDNAME      = 'F0004'.
S_SRCH-DYFLDNAME    = 'LV_CADRBUL'.
APPEND S_SRCH TO T_SRCH.
CLEAR S_SRCH.

```

```

S_SRCH-FLDNAME      = 'F0005'.
S_SRCH-DYFLDNAME    = 'LV_CLCEBUL'.
APPEND S_SRCH TO T_SRCH.
CLEAR S_SRCH.

```

```

S_SRCH-FLDNAME      = 'F0006'.
S_SRCH-DYFLDNAME    = 'LV_CLBUL'.
APPEND S_SRCH TO T_SRCH.
CLEAR S_SRCH.

```

```

S_SRCH-FLDNAME      = 'F0007'.
S_SRCH-DYFLDNAME    = 'LV_CVDBUL'.
APPEND S_SRCH TO T_SRCH.
CLEAR S_SRCH.

```

```
S_SRCH-FLDNAME      = 'F0008'.
S_SRCH-DYFLDNAME   = 'LV_CVNBUL'.
APPEND S_SRCH TO T_SRCH.
CLEAR S_SRCH.
```

```
CALL FUNCTION 'F4IF_INT_TABLE_VALUE_REQUEST'
```

```
EXPORTING
```

```
  RETFIELD          = 'MATNR'
  DYNPROFIELD       = 'LV_CKODBUL'
  WINDOW_TITLE      = 'Cari Seçim Ekranı'
  VALUE_ORG         = 'S'
```

```
TABLES
```

```
  VALUE_TAB        = LT_DEMO_TABLE
  RETURN_TAB        = LT_RETURN
  DYNPFELD_MAPPING = T_SRCH
```

```
EXCEPTIONS
```

```
  PARAMETER_ERROR = 1
  NO_VALUES_FOUND = 2
  OTHERS           = 3.
```

```
IF SY-SUBRC EQ 0.
```

```
  IF LT_RETURN IS NOT INITIAL.
```

```
    FIELD-SYMBOLS : <WA> TYPE DDSHRETVAL.
```

```
    READ TABLE LT_RETURN ASSIGNING <WA> WITH KEY RETFIELD = 'LV_CKODBUL'.
    GV_CARI = <WA>-FIELDVAL.
```

```
    READ TABLE LT_RETURN ASSIGNING <WA> WITH KEY RETFIELD = 'LV_CADBUL'.
    LV_CAD = <WA>-FIELDVAL.
```

```
    READ TABLE LT_RETURN ASSIGNING <WA> WITH KEY RETFIELD = 'LV_CADRBUL'.
    LV_CADRES = <WA>-FIELDVAL.
```

```
    READ TABLE LT_RETURN ASSIGNING <WA> WITH KEY RETFIELD = 'LV_CLCEBUL'.
    LV_ILCE = <WA>-FIELDVAL.
```

```
    READ TABLE LT_RETURN ASSIGNING <WA> WITH KEY RETFIELD = 'LV_CILBUL'.
    LV_IL = <WA>-FIELDVAL.
```

```
    READ TABLE LT_RETURN ASSIGNING <WA> WITH KEY RETFIELD = 'LV_CVDBUL'.
    LV_VERDA = <WA>-FIELDVAL.
```

```
    READ TABLE LT_RETURN ASSIGNING <WA> WITH KEY RETFIELD = 'LV_CVNBUL'.
    LV_VERNO = <WA>-FIELDVAL.
```

```
  ENDIF.
```

```
ENDIF.
```

```
ENDFORM.
```

```

*-----*
*   INCLUDE TABLECONTROL_FORMS                                     *
*-----*

*&-----*
*&   Form   USER_OK_TC                                           *
*&-----*

FORM USER_OK_TC USING      P_TC_NAME TYPE DYNFNAM
                          P_TABLE_NAME
                          P_MARK_NAME
                          CHANGING P_OK      LIKE SY-UCOMM.

DATA: L_OK      TYPE SY-UCOMM,
      L_OFFSET TYPE I.

SEARCH P_OK FOR P_TC_NAME.
IF SY-SUBRC <> 0.
  EXIT.
ENDIF.
L_OFFSET = STRLEN( P_TC_NAME ) + 1.
L_OK = P_OK+L_OFFSET.
*
CASE L_OK.
  WHEN 'INSR'.
    PERFORM FCODE_INSERT_ROW USING      P_TC_NAME
                                        P_TABLE_NAME.

    CLEAR P_OK.

  WHEN 'DELE'.
    PERFORM FCODE_DELETE_ROW USING      P_TC_NAME
                                        P_TABLE_NAME
                                        P_MARK_NAME.

    CLEAR P_OK.

  WHEN 'P--' OR
      'P-'  OR
      'P+'  OR
      'P++'.
    PERFORM COMPUTE_SCROLLING_IN_TC USING P_TC_NAME
                                        L_OK.

    CLEAR P_OK.

  WHEN 'MARK'.
    PERFORM FCODE_TC_MARK_LINES USING P_TC_NAME
                                        P_TABLE_NAME
                                        P_MARK_NAME .

    CLEAR P_OK.

  WHEN 'DMRK'.
    PERFORM FCODE_TC_DEMARK_LINES USING P_TC_NAME
                                        P_TABLE_NAME
                                        P_MARK_NAME .

    CLEAR P_OK.

ENDCASE.

ENDFORM.

```

```

*&-----*
*&      Form  FCODE_INSERT_ROW                                     *
*&-----*
FORM FCODE_INSERT_ROW
      USING      P_TC_NAME          TYPE DYNFNAM
                P_TABLE_NAME          .

DATA L_LINES_NAME      LIKE FELD-NAME.
DATA L_SELLINE         LIKE SY-STEPL.
DATA L_LASTLINE       TYPE I.
DATA L_LINE           TYPE I.
DATA L_TABLE_NAME     LIKE FELD-NAME.
FIELD-SYMBOLS <TC>          TYPE CXTAB_CONTROL.
FIELD-SYMBOLS <TABLE>      TYPE STANDARD TABLE.
FIELD-SYMBOLS <LINES>     TYPE I.

ASSIGN (P_TC_NAME) TO <TC>.

CONCATENATE P_TABLE_NAME '[' INTO L_TABLE_NAME.

ASSIGN (L_TABLE_NAME) TO <TABLE>.

CONCATENATE 'G_' P_TC_NAME '_LINES' INTO L_LINES_NAME.
ASSIGN (L_LINES_NAME) TO <LINES>.

GET CURSOR LINE L_SELLINE.
IF SY-SUBRC <> 0.
  L_SELLINE = <TC>-LINES + 1.

  IF L_SELLINE > <LINES>.
    <TC>-TOP_LINE = L_SELLINE - <LINES> + 1 .
  ELSE.
    <TC>-TOP_LINE = 1.
  ENDIF.
ELSE.
  L_SELLINE = <TC>-TOP_LINE + L_SELLINE - 1.
  L_LASTLINE = <TC>-TOP_LINE + <LINES> - 1.
ENDIF.

L_LINE = L_SELLINE - <TC>-TOP_LINE + 1.

INSERT INITIAL LINE INTO <TABLE> INDEX L_SELLINE.
<TC>-LINES = <TC>-LINES + 1.

SET CURSOR LINE L_LINE.

ENDFORM.

```



```

*&-----*
*&      Form  FCODE_DELETE_ROW                               *
*&-----*
FORM FCODE_DELETE_ROW
      USING      P_TC_NAME          TYPE DYNFNAM
                P_TABLE_NAME
                P_MARK_NAME      .

DATA L_TABLE_NAME          LIKE FELD-NAME.

FIELD-SYMBOLS <TC>          TYPE CXTAB_CONTROL.
FIELD-SYMBOLS <TABLE>       TYPE STANDARD TABLE.
FIELD-SYMBOLS <WA> .
FIELD-SYMBOLS <MARK_FIELD>.

ASSIGN (P_TC_NAME) TO <TC>.

CONCATENATE P_TABLE_NAME '[' INTO L_TABLE_NAME.
ASSIGN (L_TABLE_NAME) TO <TABLE>.

DESCRIBE TABLE <TABLE> LINES <TC>-LINES.

LOOP AT <TABLE> ASSIGNING <WA>.

      ASSIGN COMPONENT P_MARK_NAME OF STRUCTURE <WA> TO <MARK_FIELD>.

      IF <MARK_FIELD> = 'X'.
        DELETE <TABLE> INDEX SYST-TABIX.
        IF SY-SUBRC = 0.
          <TC>-LINES = <TC>-LINES - 1.
        ENDIF.
      ENDIF.
    ENDLOOP.

ENDFORM.

```

```

*&-----*
*&      Form  COMPUTE_SCROLLING_IN_TC
*&-----*
*      text
*-----*
*      -->P_TC_NAME  name of tablecontrol
*      -->P_OK      ok code
*-----*
FORM COMPUTE_SCROLLING_IN_TC USING      P_TC_NAME
                                       P_OK.

DATA L_TC_NEW_TOP_LINE      TYPE I.
DATA L_TC_NAME              LIKE FELD-NAME.
DATA L_TC_LINES_NAME       LIKE FELD-NAME.
DATA L_TC_FIELD_NAME       LIKE FELD-NAME.

FIELD-SYMBOLS <TC>        TYPE CXTAB_CONTROL.
FIELD-SYMBOLS <LINES>    TYPE I.

ASSIGN (P_TC_NAME) TO <TC>.

CONCATENATE 'G_' P_TC_NAME '_LINES' INTO L_TC_LINES_NAME.
ASSIGN (L_TC_LINES_NAME) TO <LINES>.

IF <TC>-LINES = 0.

    L_TC_NEW_TOP_LINE = 1.
ELSE.

    CALL FUNCTION 'SCROLLING_IN_TABLE'
    EXPORTING
        ENTRY_ACT      = <TC>-TOP_LINE
        ENTRY_FROM     = 1
        ENTRY_TO       = <TC>-LINES
        LAST_PAGE_FULL = 'X'
        LOOPS          = <LINES>
        OK_CODE        = P_OK
        OVERLAPPING    = 'X'
    IMPORTING
        ENTRY_NEW      = L_TC_NEW_TOP_LINE
    EXCEPTIONS
        OTHERS         = 0.
ENDIF.

GET CURSOR FIELD L_TC_FIELD_NAME
                AREA L_TC_NAME.

IF SYST-SUBRC = 0.
    IF L_TC_NAME = P_TC_NAME.

        SET CURSOR FIELD L_TC_FIELD_NAME LINE 1.
    ENDIF.
ENDIF.

<TC>-TOP_LINE = L_TC_NEW_TOP_LINE.

ENDFORM.

```

```

*&-----*
*&      Form  FCODE_TC_MARK_LINES
*&-----*
*      marks all TableControl lines
*-----*
*      -->P_TC_NAME  name of tablecontrol
*-----*
FORM FCODE_TC_MARK_LINES USING P_TC_NAME
                                P_TABLE_NAME
                                P_MARK_NAME.

DATA L_TABLE_NAME              LIKE FELD-NAME.

FIELD-SYMBOLS <TC>             TYPE CXTAB_CONTROL.
FIELD-SYMBOLS <TABLE>         TYPE STANDARD TABLE.
FIELD-SYMBOLS <WA>.
FIELD-SYMBOLS <MARK_FIELD>.

ASSIGN (P_TC_NAME) TO <TC>.

CONCATENATE P_TABLE_NAME '[' INTO L_TABLE_NAME.
ASSIGN (L_TABLE_NAME) TO <TABLE>.

LOOP AT <TABLE> ASSIGNING <WA>.

    ASSIGN COMPONENT P_MARK_NAME OF STRUCTURE <WA> TO <MARK_FIELD>.

    <MARK_FIELD> = 'X'.
ENDLOOP.
ENDFORM.

```

```

*&-----*
*&      Form  FCODE_TC_DEMARK_LINES
*&-----*
*      demarks all TableControl lines
*-----*
*      -->P_TC_NAME  name of tablecontrol
*-----*
FORM FCODE_TC_DEMARK_LINES USING P_TC_NAME
                                P_TABLE_NAME
                                P_MARK_NAME .

DATA L_TABLE_NAME              LIKE FELD-NAME.

FIELD-SYMBOLS <TC>             TYPE CXTAB_CONTROL.
FIELD-SYMBOLS <TABLE>          TYPE STANDARD TABLE.
FIELD-SYMBOLS <WA>.
FIELD-SYMBOLS <MARK_FIELD>.

ASSIGN (P_TC_NAME) TO <TC>.

CONCATENATE P_TABLE_NAME '[' INTO L_TABLE_NAME.
ASSIGN (L_TABLE_NAME) TO <TABLE>.

LOOP AT <TABLE> ASSIGNING <WA>.

    ASSIGN COMPONENT P_MARK_NAME OF STRUCTURE <WA> TO <MARK_FIELD>.

    <MARK_FIELD> = SPACE.
ENDLOOP.
ENDFORM.

```

```
*&-----*
*& Include          ZERCAN_ABAP_0008_PBO
*&-----*

*&-----*
*& Module STATUS_0100 OUTPUT
*&-----*
*&
*&-----*
```

```
MODULE STATUS_0100 OUTPUT.
SET PF-STATUS '200'.
SET TITLEBAR '0300'.
```

```
CLEAR:
```

```
GS_VALUE,
GV_TIPI,
GT_VALUES,
GS_VALUEI,
GV_DVZTIPI,
GT_VALUESI,
GS_VALUEIHR,
GV_IHIR,
GT_VALUESIHR.
```

```
GV_ID = 'GV_TIPI'.
GS_VALUE-KEY = '1'.
GS_VALUE-TEXT = 'YURTICI'.
APPEND GS_VALUE TO GT_VALUES.
GV_ID = 'GV_TIPI'.
GS_VALUE-KEY = '2'.
GS_VALUE-TEXT = 'YURTDISI'.
APPEND GS_VALUE TO GT_VALUES.
```

```
GV_IDI = 'GV_DVZTIPI'.
GS_VALUEI-KEY = '1'.
GS_VALUEI-TEXT = 'EUR'.
APPEND GS_VALUEI TO GT_VALUESI.
GV_IDI = 'GV_DVZTIPI'.
GS_VALUEI-KEY = '2'.
GS_VALUEI-TEXT = 'USD'.
APPEND GS_VALUEI TO GT_VALUESI.
GV_IDI = 'GV_DVZTIPI'.
GS_VALUEI-KEY = '3'.
GS_VALUEI-TEXT = 'TL'.
APPEND GS_VALUEI TO GT_VALUESI.
```

```
GV_IDIHR = 'GV_IHIR'.
GS_VALUEIHR-KEY = '1'.
GS_VALUEIHR-TEXT = 'FOB'.
APPEND GS_VALUEIHR TO GT_VALUESIHR.
```

```
GV_IDIHR = 'GV_IHIR'.
GS_VALUEIHR-KEY = '2'.
GS_VALUEIHR-TEXT = 'CIF'.
APPEND GS_VALUEIHR TO GT_VALUESIHR.
```

```
GV_IDIHR = 'GV_IHIR'.
GS_VALUEIHR-KEY = '3'.
GS_VALUEIHR-TEXT = 'CF'.
APPEND GS_VALUEIHR TO GT_VALUESIHR.
```

```
GV_IDIHR = 'GV_IHIR'.
GS_VALUEIHR-KEY = '4'.
GS_VALUEIHR-TEXT = 'FOT'.
APPEND GS_VALUEIHR TO GT_VALUESIHR.
```

```
GV_IDIHR = 'GV_IHR'.
GS_VALUEIHR-KEY = '5'.
GS_VALUEIHR-TEXT = 'DAF'.
APPEND GS_VALUEIHR TO GT_VALUESIHR.
```

```
GV_IDIHR = 'GV_IHR'.
GS_VALUEIHR-KEY = '6'.
GS_VALUEIHR-TEXT = 'EXW'.
APPEND GS_VALUEIHR TO GT_VALUESIHR.
```

```
CALL FUNCTION 'VRM_SET_VALUES'
  EXPORTING
    ID      = GV_ID
    VALUES = GT_VALUES.
CALL FUNCTION 'VRM_SET_VALUES'
  EXPORTING
    ID      = GV_IDI
    VALUES = GT_VALUESI.
CALL FUNCTION 'VRM_SET_VALUES'
  EXPORTING
    ID      = GV_IDIHR
    VALUES = GT_VALUESIHR.
ENDMODULE.
```

```
MODULE TBCONTROL_CHANGE_TC_ATTR OUTPUT.
```

```
ENDMODULE.
```

```
*&-----*
*& Include          ZERCAN_ABAP_0008_PAI
*&-----*
*&-----*
*&      Module  USER_COMMAND_0100  INPUT
*&-----*
*      text
*-----*
```

```
MODULE USER_COMMAND_0100 INPUT.
  CASE SY-UCOMM.
    WHEN '&BACK'.
      LEAVE TO SCREEN 0.
    WHEN 'TAB1'.
      TB_ID-ACTIVETAB = 'TAB1'.
    WHEN 'TAB2'.
      TB_ID-ACTIVETAB = 'TAB2'.
    WHEN '&TEMIZLE'.
      PERFORM SIPAMAS_TEMIZLE.
    WHEN '&KAYDET'.
      PERFORM SIPAMAS_KAYDET.
    WHEN '&CLEAR'.
      PERFORM SIPATRA_TEMIZLE.
    WHEN '&SAVE'.
      PERFORM SIPATRA_KAYDET.
    WHEN '&ALV'.
      PERFORM ALV.
    WHEN '&CBUL'.
      PERFORM CARI_BUL.
  ENDCASE.
ENDMODULE.
```