客户/供应商主数据批导

客户/供应商主数据维护中，SAP提供了两个强大的类cmd_ei_api=>maintain_bapi和VMD_EI_API

BUS4001　　公司和组织的地址

下面是客户主数据维护的例子，供应商的类似。当客户编码为数据类型时，一定要将客户编码补0，否则系统会出现莫名其妙的错误。

```
*&---------------------------------------------------------------------*
*& 程序名称:ZSDB0002
*& 作者   :

*& 开发日期:
*& 请求号 :
*& 描述   :
*& 开发申请 :
*& 变更记录
*&
** 修改日期 开发人员  请求号 描述
*&---------------------------------------------------------------------*
REPORT  zsdb0002 MESSAGE-ID zl_mess.
INCLUDE ZSDB0002_VAR_INCLUDE.
INCLUDE ZSDB0002_SCREEN_INCLUDE.
INCLUDE ZSDB0002_FILE_INCLUDE.



 *&---------------------------------------------------------------------*
*&  Include           ZSDB0002_VAR_INCLUDE
*&---------------------------------------------------------------------*

TYPES:BEGIN OF ty_basis,
      ktokd   TYPE  kna1-ktokd  ,
      kunnr   TYPE  kna1-kunnr  ,
      chhu    type  c LENGTH 4,
      name1   TYPE  adrc-name1  ,
      name2   TYPE  adrc-name2  ,
      sortl   TYPE  kna1-sortl  ,
      sort2   TYPE  adrc-sort2  ,
      street  TYPE  adrc-street ,
      land1   TYPE  kna1-land1  ,
      regio   TYPE  kna1-regio  ,
      tel     TYPE  adrc-tel_number ,
      fax     TYPE  adrc-fax_number ,
      smtp    TYPE  adr6-smtp_addr  ,
      remark  TYPE  adrct-remark  ,
      stceg   TYPE  kna1-stceg  ,
      kukla   TYPE  kna1-kukla  ,
      brsch   TYPE  kna1-brsch  ,
      umsa1   TYPE  kna1-umsa1  ,
      uwaer   TYPE  kna1-uwaer  ,
      umjah   TYPE  kna1-umjah  ,
      katr1   TYPE  kna1-katr1  ,
      katr2   TYPE  kna1-katr2  ,
      katr3   TYPE  kna1-katr3  ,
      katr6   TYPE  kna1-katr6  ,
      zbz     TYPE  tdline  ,
      zywgg   TYPE  tdline  ,
       name11 TYPE  knvk-name1  ,
      telf11  TYPE  knvk-telf1  ,
      name12  TYPE  knvk-name1  ,
      telf12  TYPE  knvk-telf1  ,
      name13  TYPE  knvk-name1  ,
      telf13  TYPE  knvk-telf1  ,
  END OF ty_basis.

  TYPES:BEGIN OF ty_com,
     ktokd   TYPE  kna1-ktokd  ,
     kunnr   TYPE  kna1-kunnr  ,
     bukrs   TYPE  knb1-bukrs  ,
     akont   TYPE  knb1-akont  ,
  END OF ty_com.
    TYPES:BEGIN OF ty_org,
      ktokd   TYPE  kna1-ktokd  ,
      kunnr   TYPE  kna1-kunnr  ,
      vkorg   TYPE  knvv-vkorg  ,
      vtweg   TYPE  knvv-vtweg  ,
      spart   TYPE  knvv-spart  ,
      bzirk   TYPE  knvv-bzirk  ,
      vkbur   TYPE  knvv-vkbur  ,
      vkgrp   TYPE  knvv-vkgrp  ,
      kdgrp   TYPE  knvv-kdgrp  ,
      klabc   TYPE  knvv-klabc  ,
      waers   TYPE  knvv-waers  ,
      konda   TYPE  knvv-konda  ,
      pltyp   TYPE  knvv-pltyp  ,
      versg   TYPE  knvv-versg  ,
      vsbed   TYPE  knvv-vsbed  ,
      vwerk   TYPE  knvv-vwerk  ,
      inco1   TYPE  knvv-inco1  ,
      inco2   TYPE  knvv-inco2  ,
      zterm   TYPE  knvv-zterm  ,
      ktgrd   TYPE  knvv-ktgrd  ,
       taxkd   TYPE  knvi-taxkd  ,

      kvgr1   TYPE  knvv-kvgr1  ,
      kvgr2   TYPE  knvv-kvgr2  ,
      zczyq   TYPE  tdline  ,
      zqyg    TYPE  tdline  ,
      zmdg    TYPE  tdline  ,
      zdttj   TYPE  tdline  ,
       zbz     TYPE  tdline  ,
      END OF ty_org.
  TYPES:BEGIN OF ty_data,
      ktokd   TYPE  kna1-ktokd  ,
      kunnr   TYPE  kna1-kunnr  ,
      bukrs   TYPE  knb1-bukrs  ,
      vkorg   TYPE  knvv-vkorg  ,
      vtweg   TYPE  knvv-vtweg  ,
      spart   TYPE  knvv-spart  ,
      chhu    type  c LENGTH 4,
      name1   TYPE  adrc-name1  ,
      name2   TYPE  adrc-name2  ,
      sortl   TYPE  kna1-sortl  ,
      sort2   TYPE  adrc-sort2  ,
      street  TYPE  adrc-street ,
      land1   TYPE  kna1-land1  ,
      regio   TYPE  kna1-regio  ,
      tel     TYPE  adrc-tel_number ,
      fax     TYPE  adrc-fax_number ,
      smtp    TYPE  adr6-smtp_addr  ,
      remark  TYPE  adrct-remark  ,
      stceg   TYPE  kna1-stceg  ,
      kukla   TYPE  kna1-kukla  ,
      brsch   TYPE  kna1-brsch  ,
      umsa1   TYPE  kna1-umsa1  ,
      uwaer   TYPE  kna1-uwaer  ,
      umjah   TYPE  kna1-umjah  ,

      name11  TYPE  knvk-name1  ,
      telf11  TYPE  knvk-telf1  ,
      name12  TYPE  knvk-name1  ,
      telf12  TYPE  knvk-telf1  ,
      name13  TYPE  knvk-name1  ,
      telf13  TYPE  knvk-telf1  ,
      katr1   TYPE  kna1-katr1  ,
      katr2   TYPE  kna1-katr2  ,
      katr3   TYPE  kna1-katr3  ,
      katr6   TYPE  kna1-katr6  ,
      zbz     TYPE  tdline  ,
      zywgg   TYPE  tdline  ,
      akont   TYPE  knb1-akont  ,
      bzirk   TYPE  knvv-bzirk  ,
      vkbur   TYPE  knvv-vkbur  ,
      vkgrp   TYPE  knvv-vkgrp  ,
      kdgrp   TYPE  knvv-kdgrp  ,
      klabc   TYPE  knvv-klabc  ,
      waers   TYPE  knvv-waers  ,
      konda   TYPE  knvv-konda  ,
      pltyp   TYPE  knvv-pltyp  ,
      versg   TYPE  knvv-versg  ,
      vsbed   TYPE  knvv-vsbed  ,
      vwerk   TYPE  knvv-vwerk  ,
      inco1   TYPE  knvv-inco1  ,
      inco2   TYPE  knvv-inco2  ,
      zterm   TYPE  knvv-zterm  ,
      ktgrd   TYPE  knvv-ktgrd  ,
      taxkd   TYPE  knvi-taxkd  ,
      kvgr1   TYPE  knvv-kvgr1  ,
      kvgr2   TYPE  knvv-kvgr2  ,
      zczyq   TYPE  tdline  ,
      zqyg    TYPE  tdline  ,
      zmdg    TYPE  tdline  ,
      zdttj   TYPE  tdline  ,
    END OF ty_data.
   TYPES:BEGIN OF ty_partner,
      kunnr     TYPE kna1-kunnr,
      vkorg     TYPE  knvv-vkorg  ,
      vtweg     TYPE  knvv-vtweg  ,
      spart     TYPE  knvv-spart  ,
      parvw_we  TYPE knvp-parvw,
      kunnr_we  TYPE kna1-kunnr,
      parvw_zp  TYPE knvp-parvw,
      kunnr_zp  TYPE kna1-kunnr,
      parvw_sm  TYPE knvp-parvw,
      kunnr_sm  TYPE kna1-kunnr,
    END OF ty_partner.
    TYPES:BEGIN OF ty_knvp,
       kunnr type knvp-kunnr,
       vkorg type knvp-vkorg,
       vtweg type knvp-vtweg,
       spart type knvp-spart,
     END OF ty_knvp.
     data:w_knvp type ty_knvp.
     TYPES:BEGIN OF ty_log,
         row   type i,
         vkorg type knvv-vkorg,
         vtweg type knvv-vtweg,
         spart type knvv-spart,
         kunnr type kna1-kunnr,
         type  type bapiret2-type,
         mesg  type bapiret2-message,
         color type c LENGTH 6,
      END OF ty_log.
   DATA:
      t_log type STANDARD TABLE OF ty_log,
      w_log type ty_log.
   DATA:
      t_data TYPE STANDARD TABLE OF ty_data,
      w_data TYPE ty_data.
   data:
      t_partner TYPE STANDARD TABLE OF ty_partner,
      w_partner type ty_partner.
   DATA:lv_flg.
   DATA:g_parnr TYPE parnr..
   DATA:ls_main TYPE cmds_ei_main,
      ls_main1 TYPE cmds_ei_main,
      ls_main2 TYPE cmds_ei_main,
     ls_mesg TYPE cvis_message,
     ls_mesg1 TYPE cvis_message,
     lt_msg TYPE bapiret2_t,
     lwa_msg TYPE bapiret2,
     lt_cust TYPE cmds_ei_extern_t,
     lwa_cust TYPE cmds_ei_extern,
     lwa_sale TYPE cmds_ei_sales,
     lt_sale TYPE cmds_ei_sales_t,
     lt_functions TYPE cmds_ei_functions_t,
     lwa_functions TYPE cmds_ei_functions,
     lt_company TYPE cmds_ei_company_t,
     lt_text TYPE cvis_ei_text_t,
     lwa_text TYPE cvis_ei_text,
     lt_content TYPE tline_tab,
     lwa_content TYPE tline,
     lwa_tax TYPE cmds_ei_tax_ind,
     lt_tax TYPE cmds_ei_tax_ind_t,
     lt_remark TYPE cvis_ei_rem_t,
     lwa_remark TYPE cvis_ei_rem,
     lt_tel TYPE cvis_ei_tlx_t,
     lwa_tel TYPE cvis_ei_tlx_str,
     lt_fax TYPE cvis_ei_fax_t,
     lwa_fax TYPE cvis_ei_fax_str,
     lt_mail TYPE cvis_ei_smtp_t,
     lwa_mail TYPE cvis_ei_smtp_str,
     lt_tel01 TYPE cvis_ei_ttx_t,
     lwa_tel01 TYPE cvis_ei_ttx_str,
     lt_contact TYPE cmds_ei_contacts_t,
     lwa_contact TYPE cmds_ei_contacts,
     lt_version TYPE cvi_ei_version_type1_t,
     lwa_version TYPE cvi_ei_version_type1,
     lt_phone TYPE cvis_ei_phone_t,
     lwa_phone TYPE cvis_ei_phone_str,
     lt_phone1 TYPE cvis_ei_phone_t,
     lwa_phone1 TYPE cvis_ei_phone_str,
     lwa_company TYPE cmds_ei_company.
   CONSTANTS:con_insert TYPE c VALUE 'I',
           con_update TYPE c VALUE 'M'.
   DATA:t_tpakd TYPE STANDARD TABLE OF tpakd,
      w_tpakd TYPE tpakd.
   DATA: it_fieldcat TYPE slis_t_fieldcat_alv,
      g_layout    TYPE slis_layout_alv,
      g_repid     TYPE sy-repid,
      g_title     TYPE lvc_title.
   DATA: wa_fieldcat LIKE LINE OF it_fieldcat.
   data:
      t_basis type STANDARD TABLE OF ty_basis,
      w_basis type ty_basis,
      w_org   type ty_org,
      t_org   type STANDARD TABLE OF ty_org,
      t_com type STANDARD TABLE OF ty_com,
      w_com TYPE ty_com.



  *&---------------------------------------------------------------------*
   *&  Include           ZSDB0002_SCREEN_INCLUDE
   *&---------------------------------------------------------------------*

   SELECTION-SCREEN BEGIN OF BLOCK b0 WITH FRAME TITLE title3.

   PARAMETERS:p_test TYPE c AS CHECKBOX DEFAULT 'X'.
   *PARAMETERS:p_mod type c AS LISTBOX VISIBLE LENGTH 10.
   PARAMETERS: p_master TYPE c RADIOBUTTON GROUP g2 DEFAULT 'X', "不下载日志
             p_ptner TYPE c RADIOBUTTON GROUP g2.

   SELECTION-SCREEN END OF BLOCK b0.
   SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME TITLE title1.
   PARAMETERS: p_fn LIKE rlgrap-filename MEMORY ID zrf. "主数据文件路径
   *PARAMETERS:p_test TYPE c AS CHECKBOX DEFAULT 'X'.
   SELECTION-SCREEN END OF BLOCK b1.
   *SELECTION-SCREEN BEGIN OF BLOCK b2 WITH FRAME TITLE title2.
   *PARAMETERS: p_not TYPE c RADIOBUTTON GROUP g1 DEFAULT 'X', "不下载日志
   *           p_all TYPE c RADIOBUTTON GROUP g1, "下载全部日志
   *           p_eonly TYPE c RADIOBUTTON GROUP g1, "仅下载出错日志
   *           p_logfn LIKE rlgrap-filename. "日志文件路径
   *SELECTION-SCREEN END OF BLOCK b2.

   INITIALIZATION.
     title1 = '数据文件'.
   *  title2 = '处理日志'.
     title3 = '导入模式'.
   * CONCATENATE 'C:\LOG_SD_CUSTOMS_MASTER' sy-datum '-' sy-uzeit '.txt' INTO p_logfn.

   AT SELECTION-SCREEN ON VALUE-REQUEST FOR p_fn.
     PERFORM frm_get_fn. "取主数据文件路径


   *AT SELECTION-SCREEN ON VALUE-REQUEST FOR p_logfn.
   *  PERFORM frm_get_logfn.  "取日志文件路径

   AT SELECTION-SCREEN.
     PERFORM frm_chech_filename."检测文件名

   START-OF-SELECTION.
     IF p_ptner = 'X'.
     PERFORM frm_upload_data TABLES t_partner."上传文件到内表
     PERFORM frm_fill_partner_data.
   *    PERFORM frm_write_log.
     PERFORM frm_shor_message.
     ELSEIF p_master = 'X'.
     PERFORM frm_upload_data TABLES t_data.."上传文件到内表
     PERFORM frm_fill_data."将数据填充内表
   * PERFORM frm_write_log.
     PERFORM frm_shor_message.
     ENDIF.



  *&---------------------------------------------------------------------*
   *&  Include           ZSDB0002_FILE_INCLUDE
   *&---------------------------------------------------------------------*
   *&---------------------------------------------------------------------*
   *&      Form  frm_get_fn
   *&---------------------------------------------------------------------*
   *       text
   *----------------------------------------------------------------------*
```

```
FORM frm_get_fn .

  CALL FUNCTION 'WS_FILENAME_GET'
    EXPORTING
*     DEF_FILENAME     = ' '
*     DEF_PATH         = ' '
      mask             = ',*.txt.'
      mode             = 'O'
      title            = '客户主数据'
    IMPORTING
      filename         = p_fn
*     RC               =
    EXCEPTIONS
      inv_winsys       = 1
      no_batch         = 2
      selection_cancel = 3
      selection_error  = 4
      OTHERS           = 5.
  IF sy-subrc <> 0.
*   MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
*           WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
  ENDIF.

ENDFORM.                    "frm_get_fn
*&---------------------------------------------------------------------*
*&      Form  frm_get_logfn
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*FORM frm_get_logfn .
*
*  CALL FUNCTION 'WS_FILENAME_GET'
*    EXPORTING
**     DEF_FILENAME     = ' '
**     DEF_PATH         = ' '
*      mask             = ',*.txt.'
*      mode             = 'O'
*      title            = '日志文件'
*    IMPORTING
*      filename         = p_logfn
**     RC               =
*    EXCEPTIONS
*      inv_winsys       = 1
*      no_batch         = 2
*      selection_cancel = 3
*      selection_error  = 4
*      OTHERS           = 5.
*  IF sy-subrc <> 0.
**   MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
**           WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
*  ENDIF.
*
*ENDFORM.                    "frm_get_logfn
*&---------------------------------------------------------------------*
*&      Form  frm_chech_filename
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
FORM frm_chech_filename .
  IF p_fn IS INITIAL.
    MESSAGE i010."主数据文件，路径和文件名，不能为空！
    STOP.
  ENDIF.
ENDFORM.                    "frm_chech_filename
*&---------------------------------------------------------------------*
*&      Form  frm_upload_data
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
FORM frm_upload_data TABLES fu_data.

  REFRESH  fu_data[].
  CALL FUNCTION 'WS_UPLOAD'
    EXPORTING
*     CODEPAGE                      = ' '
      filename                      = p_fn
      filetype                      = 'DAT'
*     HEADLEN                       = ' '
*     LINE_EXIT                     = ' '
*     TRUNCLEN                      = ' '
*     USER_FORM                     = ' '
*     USER_PROG                   = ' '
*     DAT_D_FORMAT                = ' '
*   IMPORTING
*     FILELENGTH                  =
    TABLES
      data_tab                    = fu_data[]
    EXCEPTIONS
      conversion_error            = 1
      file_open_error             = 2
      file_read_error             = 3
      invalid_type                = 4
      no_batch                    = 5
      unknown_error               = 6
      invalid_table_width         = 7
      gui_refuse_filetransfer     = 8
      customer_error              = 9
      no_authority                = 10
      OTHERS                      = 11
            .
  IF sy-subrc <> 0.
    MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
            WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
  ENDIF.


  IF fu_data[] IS INITIAL.
    MESSAGE '文件为空！' TYPE 'I'.
    STOP.
  ENDIF.

ENDFORM.                    "frm_upload_data

*&---------------------------------------------------------------------*
*&      Form  frm_fill_partner
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
FORM frm_fill_partner_data.
 REFRESH lt_cust.
  DATA:lt_data TYPE STANDARD TABLE OF ty_knvp,
       lw_data TYPE ty_knvp.
  REFRESH lt_data.
  LOOP AT t_partner INTO w_partner.
    lwa_cust-header-object_task = 'C'.
    CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
      EXPORTING
        input  = w_partner-kunnr
      IMPORTING
        output = lwa_cust-header-object_instance-kunnr.
*   lwa_cust-header-object_instance-kunnr = w_partner-kunnr.
    CLEAR lwa_sale.
    REFRESH lt_sale.
    lwa_sale-task = 'U'.
    lwa_sale-data_key-vkorg = w_partner-vkorg.
    lwa_sale-data_key-vtweg = w_partner-vtweg.
    lwa_sale-data_key-spart = w_partner-spart.
*   lwa_cust-central_data-central-data-ktokd = 'Z002'."账户组
*       lwa_cust-central_data-central-datax-ktokd = 'X'.
    REFRESH lt_functions.
    CLEAR w_knvp.
    MOVE-CORRESPONDING w_partner TO w_knvp.
    PERFORM frm_fill_partner01 USING w_partner-parvw_zp w_partner-kunnr_zp w_knvp.
    PERFORM frm_fill_partner01 USING w_partner-parvw_sm w_partner-kunnr_sm w_knvp.
    PERFORM frm_fill_partner01 USING w_partner-parvw_we w_partner-kunnr_we w_knvp.

    lwa_sale-functions-functions = lt_functions.

    APPEND lwa_sale TO lt_sale.
    CLEAR lwa_sale.
    lwa_cust-sales_data-sales = lt_sale.
    REFRESH lt_sale.
    APPEND lwa_cust TO lt_cust.
    MOVE-CORRESPONDING w_partner TO lw_data.
    APPEND lw_data TO lt_data.
    CLEAR lw_data.

    CLEAR lwa_cust.

  ENDLOOP.
  ls_main-customers = lt_cust.
  PERFORM  frm_write_data_sap TABLES lt_data.
 FREE ls_main.
 REFRESH lt_cust.
ENDFORM.                    "frm_fill_partner
*&---------------------------------------------------------------------*
*&      Form  FRM_INPUT_DATA
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
FORM frm_fill_data .
* 主数据
 DATA:lt_data TYPE STANDARD TABLE OF ty_knvp,
      lw_data TYPE ty_knvp.
 REFRESH lt_cust.
 data:l_kunnr1 type knvp-kunnr.
 DATA:l_kunnr TYPE kna1-kunnr.
 DATA:lt_kna1 TYPE STANDARD TABLE OF kna1.
 DATA:lw_kna1 TYPE kna1.

*切割数据
 "客户+公司信息
 LOOP AT t_data INTO w_data.
   MOVE-CORRESPONDING w_data TO w_basis.
   MOVE-CORRESPONDING w_data TO w_com.
   MOVE-CORRESPONDING w_data TO w_org.
   APPEND w_basis TO t_basis.
   APPEND w_com TO t_com.
   APPEND w_org TO t_org.
   CLEAR:w_com,w_org,w_basis,w_data.
 ENDLOOP.
 SORT t_basis BY kunnr.
 SORT t_org   BY kunnr vkorg vtweg spart.
 SORT t_com BY kunnr bukrs.
 "客户+基本信息
 DELETE ADJACENT DUPLICATES FROM t_basis COMPARING kunnr.
 DELETE ADJACENT DUPLICATES FROM t_org COMPARING kunnr vkorg vtweg spart  .
 DELETE ADJACENT DUPLICATES FROM t_com COMPARING kunnr bukrs.
 "客户+销售组织信息

 LOOP AT t_basis INTO w_basis.
   "判定该客户是否存在



   "获取默认合作伙伴
   REFRESH t_tpakd.
   SELECT *
     FROM tpakd
     INTO CORRESPONDING FIELDS OF TABLE t_tpakd
     WHERE ktokd = w_basis-ktokd.

   CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
     EXPORTING
       input  = w_basis-kunnr
     IMPORTING
       output = lwa_cust-header-object_instance-kunnr.
   CLEAR l_kunnr.
   SELECT SINGLE kunnr
     FROM kna1
     INTO l_kunnr
     WHERE kunnr = lwa_cust-header-object_instance-kunnr.
*    lwa_cust-header-object_instance-kunnr = w_data-kunnr."客户编码

   lwa_cust-central_data-central-data-ktokd = w_basis-ktokd."账户组
   IF l_kunnr IS INITIAL.
     lwa_cust-header-object_task = 'I'.
   ELSE.
     lwa_cust-header-object_task = 'M'.

   ENDIF.


     lwa_cust-central_data-central-data-brsch = w_basis-brsch."行业
     lwa_cust-central_data-central-data-umsa1 = w_basis-umsa1.
     lwa_cust-central_data-central-data-kukla = w_basis-kukla.
     lwa_cust-central_data-central-data-katr1 = w_basis-katr1.
     lwa_cust-central_data-central-data-katr2 = w_basis-katr2.
     lwa_cust-central_data-central-data-katr3 = w_basis-katr3.
     lwa_cust-central_data-central-data-katr6 = w_basis-katr6.
```

```
      lwa_cust-central_data-central-data-uwaer = w_basis-uwaer.
      lwa_cust-central_data-central-data-stceg = w_basis-stceg."增值税号
      lwa_cust-central_data-central-data-umjah = w_basis-umjah."货币

      lwa_cust-central_data-central-datax-katr1 = 'X'.
      lwa_cust-central_data-central-datax-katr2 = 'X'.
      lwa_cust-central_data-central-datax-katr3 = 'X'.
      lwa_cust-central_data-central-datax-katr6 = 'X'.
      lwa_cust-central_data-central-datax-brsch = 'X'.
      lwa_cust-central_data-central-datax-umsa1 = 'X'.
      lwa_cust-central_data-central-datax-uwaer = 'X'."货币
      lwa_cust-central_data-central-datax-umjah = 'X'."货币
      lwa_cust-central_data-central-data-civve = 'X'.
      lwa_cust-central_data-central-datax-ktokd = 'X'.
      lwa_cust-central_data-central-datax-civve = 'X'.
* 地址数据
      lwa_cust-central_data-address-task = con_insert.
     lwa_cust-central_data-address-postal-data-title = w_basis-chhu.
      lwa_cust-central_data-address-postal-data-name = w_basis-name1.
      lwa_cust-central_data-address-postal-data-name_2 = w_basis-name2.
      lwa_cust-central_data-address-postal-data-street = w_basis-street.
      lwa_cust-central_data-address-postal-data-region = w_basis-regio.
      lwa_cust-central_data-address-postal-data-langu = sy-langu.
      lwa_cust-central_data-address-postal-data-country = w_basis-land1.
      lwa_cust-central_data-address-postal-data-sort1 = w_basis-sortl.
      lwa_cust-central_data-address-postal-datax-sort1 = 'X'.
      lwa_cust-central_data-address-postal-data-sort2 = w_basis-sort2.
      lwa_cust-central_data-address-postal-datax-sort2 = 'X'.
     lwa_cust-central_data-address-postal-datax-title = 'X'.
      lwa_cust-central_data-address-postal-datax-name = 'X'.
      lwa_cust-central_data-address-postal-datax-name_2 = 'X'.
      lwa_cust-central_data-address-postal-datax-langu = 'X'.
      lwa_cust-central_data-address-postal-datax-country = 'X'.
      lwa_cust-central_data-address-postal-datax-street = 'X'.
      lwa_cust-central_data-address-postal-datax-region = 'X'.
*地址注释
      CLEAR lwa_remark.
      REFRESH lt_remark.
      lwa_remark-task = con_insert.
      lwa_remark-data-langu = sy-langu.
      lwa_remark-data-adr_notes = w_basis-remark.
      lwa_remark-datax-langu = 'X'.
      lwa_remark-datax-adr_notes = 'X'.
      APPEND lwa_remark TO lt_remark.
      lwa_cust-central_data-address-remark-remarks = lt_remark.

*电话
      CLEAR:lwa_phone.
      REFRESH lt_phone.
      lwa_phone-contact-task = con_insert.
      lwa_phone-contact-data-telephone = w_basis-tel.
      lwa_phone-contact-datax-telephone = 'X'.
      APPEND lwa_phone TO lt_phone.
      lwa_cust-central_data-address-communication-phone-phone = lt_phone.
*传真
      CLEAR:lwa_fax.
      REFRESH lt_fax.
      lwa_fax-contact-task = con_insert.
      lwa_fax-contact-data-fax = w_basis-fax.
      lwa_fax-contact-datax-fax = 'X'.
      APPEND lwa_fax TO lt_fax.
      lwa_cust-central_data-address-communication-fax-fax = lt_fax.

*电子邮件
      CLEAR:lwa_mail.
      REFRESH lt_mail.
      lwa_mail-contact-task = con_insert.
      lwa_mail-contact-data-e_mail = w_basis-smtp.
      lwa_mail-contact-datax-e_mail = 'X'.
      APPEND lwa_mail TO lt_mail.
      lwa_cust-central_data-address-communication-smtp-smtp = lt_mail.

*联系人1
*CLEAR:g_parnr.
*CALL FUNCTION 'NUMBER_GET_NEXT'
*  EXPORTING
*    nr_range_nr            = 'AP'
*    object                 = 'PARTNER'
*    quantity               = '1'
*  IMPORTING
*    number                 = g_parnr
*  EXCEPTIONS
*    interval_not_found      = 1
*    number_range_not_intern = 2
*    object_not_found        = 3
*    quantity_is_0           = 4
*    quantity_is_not_1       = 5
*    interval_overflow       = 6
*    buffer_overflow         = 7
*    OTHERS                  = 8.
*
*lwa_contact-task = 'I'.
*lwa_contact-address_type_3-task = 'I'.
*lwa_contact-data_key-parnr = g_parnr.
*lwa_contact-address_type_3-postal-data-firstname = 'aa'.
*lwa_contact-address_type_3-postal-data-lastname = 'bb'.
*lwa_contact-address_type_3-postal-datax-firstname = 'X'.
*lwa_contact-address_type_3-postal-datax-lastname = 'X'.
*
*"联系人电话
*
*lwa_phone1-contact-task = 'I'.
*lwa_phone1-contact-data-telephone = '89999888'.
*lwa_phone1-contact-data-extension = '3434343'.
*lwa_phone1-contact-datax-extension = 'X'.
*lwa_phone1-contact-datax-telephone = 'X'.
*APPEND lwa_phone1 TO lt_phone1.
*
*lwa_contact-address_type_3-communication-phone-phone = lt_phone1.
*APPEND lwa_contact TO lt_contact.

    REFRESH lt_contact.
    PERFORM frm_fill_contact USING w_basis-name11 w_basis-telf11.
    PERFORM frm_fill_contact USING w_basis-name12 w_basis-telf12.
    PERFORM frm_fill_contact USING w_basis-name13 w_basis-telf13.
    lwa_cust-central_data-contact-contacts = lt_contact.

*会计注释
*    CLEAR:lwa_text.
   REFRESH lt_text.
*    CLEAR:lwa_content.
*    REFRESH lt_content.
*    lwa_text-task = 'M'.
*    lwa_text-data_key-text_id = '0002'.
*    lwa_text-data_key-langu = sy-langu.
*    lwa_content-tdformat = '*'.
*    lwa_content-tdline = 'TEST001'.
*    APPEND lwa_content TO lt_content.
*    CLEAR lwa_content.
*    lwa_text-data = lt_content.
*
*    APPEND lwa_text TO lt_text.

*备注
*    REFRESH lt_content.
*    CLEAR lwa_content.
*    CLEAR lwa_text.
*    lwa_text-task = 'M'.
*    lwa_text-data_key-text_id = 'Z301'.
*    lwa_text-data_key-langu = sy-langu.
*    lwa_content-tdformat = '*'.
*    lwa_content-tdline = 'TEST002'.
*    APPEND lwa_content TO lt_content.
*    CLEAR lwa_content.
*    lwa_text-data = lt_content.
*
*    APPEND lwa_text TO lt_text.
   PERFORM frm_fill_text USING 'Z301' w_basis-zbz.
*有无店招广告
*    REFRESH lt_content.
*    REFRESH lt_content.
*    CLEAR lwa_content.
*    CLEAR lwa_text.
*    lwa_text-task = 'M'.
*    lwa_text-data_key-text_id = 'Z302'.
*    lwa_text-data_key-langu = sy-langu.
*    lwa_content-tdformat = '*'.
*    lwa_content-tdline = 'TEST003'.
*    APPEND lwa_content TO lt_content.
*    CLEAR lwa_content.
*    lwa_text-data = lt_content.
*    APPEND lwa_text TO lt_text.
```

```
   PERFORM frm_fill_text USING 'Z302' w_basis-zywgg.
   lwa_cust-central_data-text-texts = lt_text.

 REFRESH lt_company.
* 公司代码数据
LOOP AT t_com INTO w_com WHERE kunnr = w_basis-kunnr.
 CLEAR lwa_company.

   lwa_company-task = 'M'.
   lwa_company-data_key-bukrs = w_com-bukrs.
*    lwa_company-data-zuawa = '001'.
   lwa_company-data-akont = w_com-akont.
*    lwa_company-datax-zuawa = 'X'.
   lwa_company-datax-akont = 'X'.
   APPEND lwa_company TO lt_company.
   CLEAR:w_com.
ENDLOOP.
    lwa_cust-company_data-company = lt_company.
*销售数据
   "销售数据文本
*称重要求
*tax

   REFRESH lt_sale.

    LOOP AT t_org INTO w_org WHERE kunnr = w_basis-kunnr.
 CLEAR lwa_sale.

   REFRESH lt_text.
   PERFORM frm_fill_text USING 'Z201' w_org-zbz.
*    REFRESH lt_content.
*    CLEAR:lwa_text,lwa_content.
*    lwa_text-task = 'M'.
*    lwa_text-data_key-text_id = 'Z201'.
*    lwa_text-data_key-langu = sy-langu.
*    lwa_content-tdformat = '*'.
*    lwa_content-tdline = w_data-zczyq.
*    APPEND lwa_content TO lt_content.
*    CLEAR lwa_content.
*    lwa_text-data = lt_content.
*    APPEND lwa_text TO lt_text.
   PERFORM frm_fill_text USING 'Z202' w_org-zbz.
*    REFRESH lt_content.
*    CLEAR:lwa_text,lwa_content.
*    lwa_text-task = 'M'.
*    lwa_text-data_key-text_id = 'Z202'.
*    lwa_text-data_key-langu = sy-langu.
*    lwa_content-tdformat = '*'.
*    lwa_content-tdline = w_data-zqyg.
*    APPEND lwa_content TO lt_content.
*    CLEAR lwa_content.
*    lwa_text-data = lt_content.
*    APPEND lwa_text TO lt_text.
   PERFORM frm_fill_text USING 'Z203' w_org-zmdg.
*    REFRESH lt_content.
*    CLEAR:lwa_text,lwa_content.
*    lwa_text-task = 'M'.
*    lwa_text-data_key-text_id = 'Z203'.
*    lwa_text-data_key-langu = sy-langu.
*    lwa_content-tdformat = '*'.
*    lwa_content-tdline = w_data-zmdg.
*    APPEND lwa_content TO lt_content.
*    CLEAR lwa_content.
*    lwa_text-data = lt_content.
*    APPEND lwa_text TO lt_text.

*    REFRESH lt_content.
*
*    lwa_text-task = 'M'.
*    lwa_text-data_key-text_id = 'Z204'.
*    lwa_text-data_key-langu = sy-langu.
*    lwa_content-tdformat = '*'.
*    lwa_content-tdline = w_data-zmdd.
*    APPEND lwa_content TO lt_content.
*    CLEAR lwa_content.
*    lwa_text-data = lt_content.
*
*    APPEND lwa_text TO lt_text.

*    REFRESH lt_content.
*    CLEAR:lwa_text,lwa_content.
*  lwa_text-task = 'M'.
*  lwa_text-data_key-text_id = 'Z205'.
*  lwa_text-data_key-langu = sy-langu.
*  lwa_content-tdformat = '*'.
*  lwa_content-tdline = w_data-zdttj.
*  APPEND lwa_content TO lt_content.
*  CLEAR lwa_content.
*  lwa_text-data = lt_content.
*  APPEND lwa_text TO lt_text.
  PERFORM frm_fill_text USING 'Z205' w_org-zdttj.
  lwa_sale-texts-texts = lt_text.

  lwa_sale-task = 'M'.
  lwa_sale-data_key-vkorg = w_org-vkorg.
  lwa_sale-data_key-vtweg = w_org-vtweg.
  lwa_sale-data_key-spart = w_org-spart.
  lwa_sale-data-zterm = w_org-zterm.
  lwa_sale-data-vkgrp = w_org-vkgrp.
  lwa_sale-data-vkbur = w_org-vkbur.
  lwa_sale-data-waers = w_org-waers.
  lwa_sale-data-kvgr1 = w_org-kvgr1.
  lwa_sale-data-kvgr2 = w_org-kvgr2.
  lwa_sale-data-inco1 = w_org-inco1.
  lwa_sale-data-inco2 = w_org-inco2.
  lwa_sale-data-bzirk = w_org-bzirk.
  lwa_sale-data-kdgrp = w_org-kdgrp."客户组
  lwa_sale-data-versg = w_org-versg."客户统计组
  lwa_sale-data-vsbed = w_org-vsbed."装运条件
  lwa_sale-data-vwerk = w_org-vwerk.
  lwa_sale-data-ktgrd = w_org-ktgrd."账户分配
  lwa_sale-data-klabc = w_org-klabc."ABC等级
  lwa_sale-data-konda = w_org-konda."价格租
  lwa_sale-data-pltyp = w_org-pltyp."价格清单
  IF w_org-konda IS NOT INITIAL.
   lwa_sale-datax-konda = 'X'."价格租
  ENDIF.
  IF w_org-pltyp IS NOT INITIAL.
   lwa_sale-datax-pltyp = 'X'."价格清单
  ENDIF.
  IF w_org-klabc IS NOT INITIAL.
   lwa_sale-datax-klabc = 'X'."ABC等级
  ENDIF.
  IF w_org-zterm IS NOT INITIAL.
   lwa_sale-datax-zterm = 'X'.
  ENDIF.
  IF w_org-vkgrp IS NOT INITIAL.
   lwa_sale-datax-vkgrp = 'X'.
  ENDIF.
  IF w_org-vkbur IS NOT INITIAL.
   lwa_sale-datax-vkbur = 'X'.
  ENDIF.
  IF w_org-waers IS NOT INITIAL.
   lwa_sale-datax-waers = 'X'.
  ENDIF.
  IF w_org-kvgr1 IS NOT INITIAL.
   lwa_sale-datax-kvgr1 = 'X'.
  ENDIF.
  IF w_org-kvgr2 IS NOT INITIAL.
   lwa_sale-datax-kvgr2 = 'X'.
  ENDIF.
  IF w_org-inco1 IS NOT INITIAL.
   lwa_sale-datax-inco1 = 'X'.
  ENDIF.
  IF w_org-inco2 IS NOT INITIAL.
   lwa_sale-datax-inco2 = 'X'.
  ENDIF.
  IF w_org-bzirk IS NOT INITIAL.
   lwa_sale-datax-bzirk = 'X'.
  ENDIF.
  IF w_org-kdgrp IS NOT INITIAL.
   lwa_sale-datax-kdgrp = 'X'."客户组
  ENDIF.
  IF w_org-versg IS NOT INITIAL.
   lwa_sale-datax-versg = 'X'."客户统计组
  ENDIF.
  IF w_org-vsbed IS NOT INITIAL.
   lwa_sale-datax-vsbed = 'X'."装运条件
  ENDIF.
  IF w_org-vwerk IS NOT INITIAL.
   lwa_sale-datax-vwerk = 'X'.
  ENDIF.
```

```
    IF w_org-ktgrd IS NOT INITIAL.
      lwa_sale-datax-ktgrd = 'X'."账户分配
    ENDIF.

    REFRESH lt_functions.
    CLEAR w_knvp.
    MOVE-CORRESPONDING w_org TO w_knvp.

    LOOP AT t_tpakd INTO w_tpakd.
      select SINGLE kunnr FROM knvp INTO l_kunnr1 WHERE kunnr = w_basis-kunnr and vkorg = w_org-vkorg and vtweg = w_org-vtweg and spart =
w_org-spart.
        if sy-subrc ne 0.
      PERFORM frm_fill_partner USING w_tpakd-parvw lwa_cust-header-object_instance-kunnr w_knvp.
      ENDIF.

    ENDLOOP.
*    CLEAR lwa_functions.
*    lwa_functions-task = 'I'.
*    lwa_functions-data_key-parvw = 'I'.
*    lwa_functions-data_key-parvw = 'WE'.
**    lwa_functions-data-partner = 'AY9'.
*    lwa_functions-data-defpa = 'X'.
*    lwa_functions-datax-defpa = 'X'.
*    lwa_functions-datax-partner = 'X'.
*    APPEND lwa_functions TO lt_functions.
*
*    CLEAR lwa_functions.
*    lwa_functions-task = 'I'.
*    lwa_functions-data_key-parvw = 'I'.
*    lwa_functions-data_key-parvw = 'RG'.
**    lwa_functions-data-partner = 'AY9'.
*    lwa_functions-data-defpa = 'X'.
*    lwa_functions-datax-defpa = 'X'.
*    lwa_functions-datax-partner = 'X'.
*    APPEND lwa_functions TO lt_functions.
*
*
*    CLEAR lwa_functions.
*    lwa_functions-task = 'I'.
*    lwa_functions-data_key-parvw = 'I'.
*    lwa_functions-data_key-parvw = 'SM'.
**    lwa_functions-data-partner = 'GX0001'.
*    lwa_functions-data-defpa = 'X'.
*    lwa_functions-datax-defpa = 'X'.
*    lwa_functions-datax-partner = 'X'.
*    APPEND lwa_functions TO lt_functions.
*
*
*    CLEAR lwa_functions.
*    lwa_functions-task = 'I'.
*    lwa_functions-data_key-parvw = 'I'.
*    lwa_functions-data_key-parvw = 'AG'.
*    lwa_functions-data-defpa = 'X'.
*    lwa_functions-datax-defpa = 'X'.
*    APPEND lwa_functions TO lt_functions.
*
*
*    CLEAR lwa_functions.
*    lwa_functions-task = 'I'.
*    lwa_functions-data_key-parvw = 'I'.
*    lwa_functions-data_key-parvw = 'RE'.
*    lwa_functions-data-defpa = 'X'.
*    lwa_functions-datax-defpa = 'X'.
*    APPEND lwa_functions TO lt_functions.

     CLEAR lwa_tax.
    REFRESH lt_tax.
    lwa_tax-task = 'M'.
    lwa_tax-data_key-aland = 'CN'.
    lwa_tax-data_key-tatyp = 'MWST' .
    lwa_tax-data-taxkd = w_org-taxkd.
    lwa_tax-datax-taxkd = 'X'.
    APPEND lwa_tax TO lt_tax.


    lwa_sale-functions-functions = lt_functions.

    APPEND lwa_sale TO lt_sale.
    CLEAR lwa_sale.
    CLEAR:w_org.
endloop.
```

```
lwa_cust-central_data-tax_ind-tax_ind = lt_tax.
    lwa_cust-sales_data-sales = lt_sale.
*    lwa_cust-sales_data-CURRENT_STATE = 'X'.
    REFRESH lt_sale.
    APPEND lwa_cust TO lt_cust.
    MOVE-CORRESPONDING w_basis TO lw_data.
    APPEND lw_data TO lt_data.
    CLEAR lw_data.
    CLEAR lwa_cust.
  ENDLOOP.
  ls_main-customers = lt_cust.
  PERFORM frm_write_data_sap TABLES lt_data."创建数据到SAP中
  FREE ls_main.
  REFRESH lt_cust.

ENDFORM.                    " FRM_INPUT_DATA

*&---------------------------------------------------------------------*
*&      Form  frm_fill_contact
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*      -->FU_NAME    text
*      -->FU_PHONE   text
*----------------------------------------------------------------------*
FORM frm_fill_contact USING fu_name fu_phone.
  IF fu_name IS NOT INITIAL OR fu_phone IS NOT INITIAL.
    CLEAR lwa_contact.
    CLEAR:g_parnr.
    CALL FUNCTION 'NUMBER_GET_NEXT'
      EXPORTING
        nr_range_nr             = 'AP'
        object                  = 'PARTNER'
        quantity                = '1'
      IMPORTING
        number                  = g_parnr
      EXCEPTIONS
        interval_not_found      = 1
        number_range_not_intern = 2
        object_not_found        = 3
        quantity_is_0           = 4
        quantity_is_not_1       = 5
        interval_overflow       = 6
        buffer_overflow         = 7
        OTHERS                  = 8.
    lwa_contact-task = 'I'.
    lwa_contact-address_type_3-task = 'I'.
    lwa_contact-data_key-parnr = g_parnr.
*lwa_contact-address_type_3-postal-data-firstname = 'aa'.
    lwa_contact-address_type_3-postal-data-lastname = fu_name.
*lwa_contact-address_type_3-postal-datax-firstname = 'X'.
    lwa_contact-address_type_3-postal-datax-lastname = 'X'.
    "联系人电话
    CLEAR:lwa_phone1.
    REFRESH lt_phone1.
    lwa_phone1-contact-task = 'I'.
    lwa_phone1-contact-data-telephone = fu_phone.
*lwa_phone1-contact-data-extension = '3434343'.
*lwa_phone1-contact-datax-extension = 'X'.
    lwa_phone1-contact-datax-telephone = 'X'.
    APPEND lwa_phone1 TO lt_phone1.
    lwa_contact-address_type_3-communication-phone-phone = lt_phone1.
    APPEND lwa_contact TO lt_contact.
  ENDIF.
ENDFORM.                    "frm_fill_contact

*&---------------------------------------------------------------------*
*&      Form  frm_fill_text
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*      -->FU_ID      text
*      -->FU_CONTENT text
*----------------------------------------------------------------------*
FORM frm_fill_text USING fu_id fu_content.
  REFRESH lt_content.
  REFRESH lt_content.
  CLEAR lwa_content.
  CLEAR lwa_text.
  lwa_text-task = 'M'.
  lwa_text-data_key-text_id = fu_id.
  lwa_text-data_key-langu = sy-langu.
  lwa_content-tdformat = '*'.
  lwa_content-tdline = fu_content.
  APPEND lwa_content TO lt_content.
  CLEAR lwa_content.
  lwa_text-data = lt_content.
  APPEND lwa_text TO lt_text.
ENDFORM.                    "frm_fill_text
*&---------------------------------------------------------------------*
*&      Form  FRM_WRITE_DATA_SAP
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
FORM frm_write_data_sap TABLES fu_data TYPE STANDARD TABLE.
  DATA:l_ilines TYPE i .
  DATA:l_message TYPE string.
  DATA:l_iindex TYPE i.
  FREE:ls_main1,ls_main2,ls_mesg,ls_mesg1.
  DATA:lw_data TYPE ty_knvp.
*  FREE:cmd_ei_api=>gt_global_kna1_old.
  DESCRIBE TABLE fu_data[] LINES l_ilines.
  CALL METHOD cmd_ei_api=>maintain_bapi
    EXPORTING
      iv_test_run              = p_test
      iv_collect_messages      = 'X'
      is_master_data           = ls_main
    IMPORTING
      es_master_data_correct   = ls_main1
      es_message_correct       = ls_mesg1
      es_master_data_defective = ls_main2
      es_message_defective     = ls_mesg.
  REFRESH lt_msg.
  CLEAR lv_flg.
  lt_msg = ls_mesg-messages.
  l_iindex = 0.
  DO l_ilines TIMES.
    l_iindex = l_iindex + 1.
    READ TABLE fu_data INTO lw_data INDEX l_iindex.
    READ TABLE lt_msg INTO lwa_msg WITH KEY row = l_iindex.
    IF sy-subrc NE 0."没有任何改行的信息，表明成功
      CLEAR w_log.
      CLEAR l_message.
      w_log-row = l_iindex.
      w_log-kunnr = lw_data-kunnr.
      w_log-type = 'S'.
      CONCATENATE '客户' w_log-kunnr '创建/修改成功' INTO l_message.
      w_log-mesg = l_message.
      APPEND w_log TO t_log.
      COMMIT WORK.
    ELSE.
      CLEAR lv_flg.
      LOOP AT lt_msg INTO lwa_msg WHERE row = l_iindex.
        CLEAR l_message.
        l_message = lwa_msg-message.

*    WRITE: / lwa_cust-header-OBJECT_INSTANCE-kunnr,lwa_msg-type,l_message.
        IF lwa_msg-type = 'E' OR lwa_msg-type = 'A'.
          CLEAR w_log.
          w_log-row = l_iindex.
          w_log-kunnr = lw_data-kunnr.
          w_log-type = lwa_msg-type.
          w_log-mesg = l_message.
          APPEND w_log TO t_log.
          lv_flg = 'X'.
        ENDIF.
        CLEAR:lwa_msg.
      ENDLOOP.

      IF lv_flg IS INITIAL.
        CLEAR w_log.
        CLEAR l_message.
        w_log-row = l_iindex.
        w_log-kunnr = lw_data-kunnr.
        w_log-type = 'S'.
        CONCATENATE '客户' w_log-kunnr '创建/修改成功' INTO l_message.
        w_log-mesg = l_message.
        APPEND w_log TO t_log.
        COMMIT WORK.
      ELSE.
        ROLLBACK WORK.
      ENDIF.
    ENDIF.
  ENDDO.
*  LOOP AT lt_msg INTO lwa_msg.
*    CLEAR l_message.
*    l_message = lwa_msg-message.
*
**    WRITE: / lwa_cust-header-OBJECT_INSTANCE-kunnr,lwa_msg-type,l_message.
*    IF lwa_msg-type = 'E' OR lwa_msg-type = 'A'.
*      CLEAR w_log.
*      w_log-kunnr = lwa_cust-header-object_instance-kunnr.
*      w_log-type = lwa_msg-type.
*      w_log-mesg = l_message.
*      APPEND w_log TO t_log.
*      lv_flg = 'X'.
*    ENDIF.
*    CLEAR:lwa_msg.
*  ENDLOOP.
*  IF lv_flg IS INITIAL.
*    CLEAR w_log.
*    CLEAR l_message.
*    w_log-kunnr = lwa_cust-header-object_instance-kunnr.
*    w_log-type = 'S'.
*    CONCATENATE '客户' w_log-kunnr '创建/修改成功' INTO l_message.
*    w_log-mesg = l_message.
*    APPEND w_log TO t_log.
*    COMMIT WORK.
*  ELSE.
*    ROLLBACK WORK.
*  ENDIF.
*  WAIT UP TO '0.5' SECONDS.
ENDFORM.                    " FRM_WRITE_DATA_SAP
*&---------------------------------------------------------------------*
*&      Form  frm_fill_partner
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*      -->FU_PARVW   text
*----------------------------------------------------------------------*
FORM frm_fill_partner USING fu_parvw fu_kunnr fu_data TYPE ty_knvp.

  DATA:l_parza TYPE knvp-parza.
  CLEAR l_parza.
  CLEAR lwa_functions.
  lwa_functions-task = 'M'.
  lwa_functions-data_key-parvw = 'M'.
  lwa_functions-data_key-parvw = fu_parvw.


*  IF NOT fu_kunnr IS INITIAL.
*    SELECT SINGLE MAX( parza )
*  FROM knvp
*  INTO l_parza
*  WHERE kunnr = fu_data-kunnr
*  AND   vkorg = fu_data-vkorg
*  AND   vtweg = fu_data-vtweg
*  AND   spart = fu_data-spart
*  AND   parvw = fu_parvw.
*    l_parza = l_parza + 1.
*    lwa_functions-data_key-parza = '000'.
    lwa_functions-data-partner = fu_kunnr.
    lwa_functions-datax-partner = 'X'.
*  ELSE.
*    lwa_functions-data-partner = 'AY9'.
*    lwa_functions-data-defpa = 'X'.
*    lwa_functions-datax-defpa = 'X'.
*    lwa_functions-datax-partner = 'X'.
*  ENDIF.
  APPEND lwa_functions TO lt_functions.


ENDFORM.                    "frm_fill_partner
*&---------------------------------------------------------------------*
*&      Form  FRM_INSERT_PARTNER
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
```

```
*----------------------------------------------------------------------*
FORM frm_insert_partner .
  CALL METHOD cmd_ei_api=>maintain_bapi
    EXPORTING
      iv_test_run              = p_test
      iv_collect_messages      = 'X'
      is_master_data           = ls_main
    IMPORTING
      es_master_data_correct   = ls_main1
      es_message_correct       = ls_mesg1
      es_master_data_defective = ls_main2
      es_message_defective     = ls_mesg.
  lt_msg = ls_mesg-messages.
  LOOP AT lt_msg INTO lwa_msg.
    WRITE: / lwa_msg-type,lwa_msg-message.
    IF lwa_msg-type = 'E' OR lwa_msg-type = 'A'.
      lv_flg = 'X'.
    ENDIF.
  ENDLOOP.
  IF lv_flg IS INITIAL.

    COMMIT WORK.
  ELSE.
    ROLLBACK WORK.
  ENDIF.

ENDFORM.                    " FRM_INSERT_PARTNER
*&---------------------------------------------------------------------*
*&      Form  FRM_WRITE_LOG
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
FORM frm_write_log .
  LOOP AT t_log INTO w_log.
    IF w_log-type = 'E'.
      WRITE:/ w_log-kunnr UNDER w_log-kunnr COLOR COL_NEGATIVE,
              w_log-type UNDER w_log-type COLOR COL_NEGATIVE,
             w_log-mesg UNDER w_log-mesg COLOR COL_NEGATIVE.
    ELSEIF w_log-type = 'A'.
      WRITE:/ w_log-kunnr UNDER w_log-kunnr COLOR COL_NEGATIVE,
            w_log-type UNDER w_log-type COLOR COL_NEGATIVE,
           w_log-mesg UNDER w_log-mesg COLOR COL_NEGATIVE.
    ELSEIF w_log-type = 'W'.
      WRITE:/ w_log-kunnr UNDER w_log-kunnr COLOR COL_TOTAL,
             w_log-type UNDER w_log-type COLOR COL_TOTAL,
            w_log-mesg UNDER w_log-mesg COLOR COL_TOTAL.
    ELSEIF w_log-type = 'S'.
      WRITE:/ w_log-kunnr UNDER w_log-kunnr COLOR COL_POSITIVE ,
            w_log-type UNDER w_log-type COLOR COL_POSITIVE ,
           w_log-mesg UNDER w_log-mesg COLOR COL_POSITIVE .
    ELSE.
*      WRITE:/ w_log-kunnr UNDER w_log-kunnr COLOR COL_HEADING ,
*             w_log-type UNDER w_log-type COLOR COL_HEADING ,
*            w_log-mesg UNDER w_log-mesg COLOR COL_HEADING .
    ENDIF.
    CLEAR w_log.
  ENDLOOP.
ENDFORM.                    " FRM_WRITE_LOG
*&---------------------------------------------------------------------*
*&      Form  FRM_SHOR_MESSAGE
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
FORM frm_shor_message .
  PERFORM frm_layout_build."设置 layout
  PERFORM frm_fieldcat_build."填充 fieldcat
  PERFORM frm_display_data."显示 ALV
ENDFORM.                    " FRM_SHOR_MESSAGE
*&---------------------------------------------------------------------*
*&      Form  FRM_LAYOUT_BUILD
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
```

```
*----------------------------------------------------------------------*
FORM frm_layout_build .
  CLEAR: g_title,g_repid,g_layout.

  g_title = '日志'.
  g_repid = sy-repid.

  g_layout-info_fieldname = 'COLOR'.

  g_layout-colwidth_optimize = 'X'."最优化列宽度
  g_layout-zebra = 'X'."可选行颜色,如果设置了，出现了间隔色带
*  g_layout-no_vline = 'X'."不显示列间竖线

  g_layout-f2code = '&ETA'."双击弹出显示详细信息的窗口
  g_layout-detail_popup = 'X'."是否弹出详细信息窗口
  g_layout-detail_initial_lines = 'X'."设置弹出窗口的？
  g_layout-detail_titlebar = '详细显示'."设置弹出窗口的标题栏
ENDFORM.                    " FRM_LAYOUT_BUILD
*&---------------------------------------------------------------------*
*&      Form  FRM_FIELDCAT_BUILD
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
FORM frm_fieldcat_build .
  CLEAR it_fieldcat[].

  DEFINE mc_fieldcatt_build.
    clear wa_fieldcat.
    wa_fieldcat-col_pos = &1.
    wa_fieldcat-fieldname = &2.
*    wa_fieldcat-seltext_s = &3.
    wa_fieldcat-seltext_m  = &3.
    wa_fieldcat-seltext_l  = &3.
    append wa_fieldcat to it_fieldcat.
  END-OF-DEFINITION.

  mc_fieldcatt_build:
    '1' 'ROW' '序号',
    '2' 'KUNNR' '客户',
    '3' 'TYPE' '信息类型',
    '4' 'MESG' '信息文本'.
ENDFORM.                    " FRM_FIELDCAT_BUILD
*&---------------------------------------------------------------------*
*&      Form  FRM_DISPLAY_DATA
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
FORM frm_display_data .
  CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'
    EXPORTING
*     I_INTERFACE_CHECK                 = ' '
*     I_BYPASSING_BUFFER                = ' '
*     I_BUFFER_ACTIVE                   = ' '
      i_callback_program                = g_repid
*     I_CALLBACK_PF_STATUS_SET          = 'SET_PF_STATUS'
*     I_CALLBACK_USER_COMMAND           = 'USER_COMMEND'
*     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                      = g_title
*     I_GRID_SETTINGS                   =
      is_layout                         = g_layout
      it_fieldcat                       = it_fieldcat[]
*     IT_EXCLUDING                      =
*     IT_SPECIAL_GROUPS                 =
*     IT_SORT                           =
*     IT_FILTER                         =
*     IS_SEL_HIDE                       =
*     I_DEFAULT                         = 'X'
      i_save                            = 'A'"可以保存所有变式
*     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        =
*  IMPORTING
*    E_EXIT_CAUSED_BY_CALLER           =
*    ES_EXIT_CAUSED_BY_USER            =
    TABLES
      t_outtab                          = t_log[]
    EXCEPTIONS
     program_error                     = 1
     OTHERS                            = 2
            .
  IF sy-subrc <> 0.
    MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
        WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
  ENDIF.
ENDFORM.                    " FRM_DISPLAY_DATA


*&---------------------------------------------------------------------*
*&      Form  frm_fill_partner01
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*      -->FU_PARVW   text
*      -->FU_KUNNR   text
*      -->FU_DATA    text
*----------------------------------------------------------------------*
FORM frm_fill_partner01 USING fu_parvw fu_kunnr fu_data TYPE ty_knvp.
  IF fu_kunnr IS NOT INITIAL.
    DATA:l_parza TYPE knvp-parza.
    CLEAR l_parza.
    CLEAR lwa_functions.
    lwa_functions-task = 'M'.
    lwa_functions-data_key-parvw = 'M'.
    lwa_functions-data_key-parvw = fu_parvw.


    IF NOT fu_kunnr IS INITIAL.
      SELECT SINGLE MAX( parza )
    FROM knvp
    INTO l_parza
    WHERE kunnr = fu_data-kunnr
    AND   vkorg = fu_data-vkorg
    AND   vtweg = fu_data-vtweg
    AND   spart = fu_data-spart
    AND   parvw = fu_parvw.
      l_parza = l_parza + 1.
      lwa_functions-data_key-parza = l_parza.
          CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
      EXPORTING
        input  = fu_kunnr
      IMPORTING
        output = lwa_functions-data-partner.
*      lwa_functions-data-partner = fu_kunnr.
      lwa_functions-datax-partner = 'X'.
    ELSE.
*    lwa_functions-data-partner = 'AY9'.
      lwa_functions-data-defpa = 'X'.
      lwa_functions-datax-defpa = 'X'.
      lwa_functions-datax-partner = 'X'.
    ENDIF.
    APPEND lwa_functions TO lt_functions.
  ENDIF.

ENDFORM.                    "frm_fill_partner01
```