\*&---------------------------------------------------------------------\*  
\*& Report ZMMR004\_VENDOR\_MASTER INTERFACE WITH LOGS  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT zmmr004\_vendor\_master.  
  
\*----------------------------------------------------------------------\*  
\*---------------------------------------------------------------------\*  
\* Types  
\*---------------------------------------------------------------------\*  
TYPES : BEGIN OF ty\_data,  
          filler1          TYPE c LENGTH 2,  
          vendor\_no        TYPE c LENGTH 10,  
          vendor\_no\_sfx    TYPE c LENGTH 2,  
          name             TYPE c LENGTH 30,  
          addr1            TYPE c LENGTH 30,  
          addr2            TYPE c LENGTH 30,  
          addr3            TYPE c LENGTH 30,  
          city             TYPE c LENGTH 19,  
          state            TYPE c LENGTH 2,  
          zip              TYPE c LENGTH 9,  
          phone            TYPE c LENGTH 10,  
          county           TYPE c LENGTH 2,  
          district         TYPE c LENGTH 2,  
          country          TYPE c LENGTH 2,  
          vendor\_type      TYPE c LENGTH 1,  
          vendor\_conv      TYPE c LENGTH 12,  
          minority\_bus\_ind TYPE c LENGTH 1,  
          small\_bus\_ind    TYPE c LENGTH 1,  
          contact\_name     TYPE c LENGTH 30,  
          vendor\_status    TYPE c LENGTH 1,  
          ss\_fid\_no        TYPE c LENGTH 14,  
          dept\_ind         TYPE c LENGTH 50,  
          create\_date      TYPE c LENGTH 7,  
          last\_update\_date TYPE c LENGTH 7,  
          sort\_seq         TYPE c LENGTH 15,  
          filler2          TYPE c LENGTH 21,  
        END OF ty\_data.  
  
TYPES : BEGIN OF ty\_log,  
          vendor\_no     TYPE zmd\_vendorno,  
          vendor\_no\_sfx TYPE zmd\_vendorno\_sfx,  
          er\_fg         TYPE char1,  
          chind         TYPE char1,  
          msg           TYPE text255,  
        END OF   ty\_log.  
\*---------------------------------------------------------------------\*  
\* Global Data  
\*---------------------------------------------------------------------\*  
DATA : gt\_data        TYPE TABLE OF ty\_data,  
       gs\_data        TYPE ty\_data,  
       gt\_vendorfamis TYPE TABLE OF zmmd\_vendorfamis,  
       gs\_vendorfamis TYPE zmmd\_vendorfamis,  
       gs\_log         TYPE ty\_log,  
       gt\_log         TYPE TABLE OF ty\_log,  
       gv\_no\_file     TYPE flag,  
       gv\_no\_data     TYPE flag,  
       gv\_db\_upd      TYPE flag,  
       gv\_total       TYPE numc10,  
       gv\_success     TYPE numc10,  
       gv\_error       TYPE numc10,  
       gv\_delete      TYPE numc10,  
       gv\_stamp       TYPE ertsp,  
       gv\_arch\_file   TYPE text256,  
       gv\_log\_file    TYPE text256,  
       gv\_file        TYPE text256,  
       gv\_email       TYPE char80.  
  
DATA : gt\_text TYPE bcsy\_text. "Table for body  
DATA : gs\_text LIKE LINE OF gt\_text. "work area for message body  
\*---------------------------------------------------------------------\*  
\* Constants  
\*---------------------------------------------------------------------\*  
CONSTANTS : gc\_def\_path   TYPE string VALUE '/usr/sap/interfaces/',  
            gc\_archive    TYPE string VALUE '/FAMIS/archive/',  
            gc\_log        TYPE string VALUE '/FAMIS/log/',  
            gc\_load\_files TYPE string VALUE '/FAMIS/inbound/'.  
\*---------------------------------------------------------------------\*  
\*Selection screen  
\*---------------------------------------------------------------------\*  
  
PARAMETERS : pr\_load RADIOBUTTON GROUP gr1 DEFAULT 'X' USER-COMMAND rad.  
PARAMETERS : pr\_log RADIOBUTTON GROUP gr1.  
PARAMETERS : p\_file  TYPE text256 MODIF ID a.  
SELECT-OPTIONS : s\_email FOR gv\_email  MODIF ID a NO INTERVALS.  
PARAMETERS: p\_lfile  TYPE text256 MODIF ID b.  
  
AT SELECTION-SCREEN OUTPUT.  
  PERFORM screen\_visiblity.  
  
AT SELECTION-SCREEN ON VALUE-REQUEST FOR p\_file.  
  PERFORM f4\_app\_file CHANGING p\_file.  
  
AT SELECTION-SCREEN ON VALUE-REQUEST FOR p\_lfile.  
  PERFORM f4\_app\_file CHANGING p\_lfile.  
  
\*---------------------------------------------------------------------\*  
\* Start of Selection  
\*---------------------------------------------------------------------\*  
  
START-OF-SELECTION.  
  IF pr\_load = abap\_true  
    AND p\_file IS INITIAL.  
    MESSAGE TEXT-029 TYPE 'I'.  
    EXIT.  
  ELSEIF pr\_log = abap\_true  
    AND p\_lfile IS INITIAL.  
    MESSAGE TEXT-030 TYPE 'I'.  
    EXIT.  
  ENDIF.  
  
  IF pr\_load = abap\_true  
    AND s\_email[] IS INITIAL.  
    MESSAGE TEXT-034 TYPE 'I'.  
    EXIT.  
  ENDIF.  
  
  IF pr\_load = abap\_true.  
  
    GET TIME STAMP FIELD gv\_stamp.  
    PERFORM read\_file.  
  
    IF gv\_no\_data = abap\_false  
      AND gv\_no\_file = abap\_false.  
      CLEAR : gv\_db\_upd, gv\_total, gv\_success,  
              gv\_error, gv\_delete.  
  
      PERFORM update\_table.  
  
      IF gv\_db\_upd = abap\_true.  
        PERFORM archive.  
        PERFORM write\_logs.  
        PERFORM send\_email.  
      ENDIF.  
    ELSE.  
      PERFORM send\_email.  
    ENDIF.  
  
  ELSEIF pr\_log = abap\_true.  
    CLEAR : gv\_db\_upd, gv\_total, gv\_success,  
              gv\_error, gv\_delete.  
    PERFORM read\_log\_file.  
    PERFORM display\_log.  
  ENDIF.  
  
\*&---------------------------------------------------------------------\*  
\*& Form f4\_app\_file  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*&      <-- P\_AFILE1  
\*&---------------------------------------------------------------------\*  
FORM f4\_app\_file  CHANGING cv\_afile.  
  DATA: lv\_path TYPE string.  
  DATA: lv\_def\_path  TYPE c LENGTH 500.  
  
  IF pr\_load = abap\_true.  
    lv\_def\_path = gc\_def\_path && sy-mandt && gc\_load\_files.  
  ELSE.  
    lv\_def\_path = gc\_def\_path && sy-mandt && gc\_log.  
  ENDIF.  
  
  CALL FUNCTION '/SAPDMC/LSM\_F4\_SERVER\_FILE'  
    EXPORTING  
      directory        = lv\_def\_path  
      filemask         = '\*.txt'  
    IMPORTING  
      serverfile       = lv\_path  
    EXCEPTIONS  
      canceled\_by\_user = 1  
      OTHERS           = 2.  
  IF sy-subrc = 0 AND lv\_path IS NOT INITIAL.  
    cv\_afile = lv\_path.  
  ENDIF.  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form read\_file  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*& -->  p1        text  
\*& <--  p2        text  
\*&---------------------------------------------------------------------\*  
FORM read\_file .  
  DATA : lv\_line TYPE string.  
  DATA : lt\_string TYPE TABLE OF string.  
  
\* Get file names for Logs.  
  CLEAR : gv\_file, lt\_string[].  
  SPLIT p\_file AT '/' INTO TABLE lt\_string.  
  gv\_file = lt\_string[ lines( lt\_string ) ].  
  TRANSLATE gv\_file TO UPPER CASE.  
  
  gv\_file = replace( val = gv\_file  
                   regex = '.TXT'  
                   with  = space  
                   occ   = 0 ).  
  CONDENSE gv\_file.  
  
\* -----Read file Data.  
  CLEAR : gv\_no\_file, gv\_no\_data.  
  OPEN DATASET p\_file FOR INPUT IN TEXT MODE ENCODING DEFAULT.  
  IF sy-subrc <> 0.  
    gv\_no\_file = abap\_true.  
    WRITE : /1 TEXT-015.  
  ELSE.  
    DO.  
      CLEAR lv\_line.  
      READ DATASET p\_file INTO lv\_line.  
      IF sy-subrc EQ 0.  
        CLEAR :gs\_data, gs\_vendorfamis.  
        IF lv\_line IS NOT INITIAL.  
          gs\_data = lv\_line.  
          IF gs\_data-vendor\_no IS NOT INITIAL.  
            gs\_vendorfamis-mandt          = sy-mandt.  
            gs\_vendorfamis-vendor\_no      = gs\_data-vendor\_no.  
            gs\_vendorfamis-vendor\_no\_sfx  = gs\_data-vendor\_no\_sfx.  
            gs\_vendorfamis-name           = gs\_data-name.  
            gs\_vendorfamis-addr1          = gs\_data-addr1.  
            gs\_vendorfamis-addr2          = gs\_data-addr2.  
            gs\_vendorfamis-addr3          = gs\_data-addr3.  
            gs\_vendorfamis-city           = gs\_data-city.  
            gs\_vendorfamis-state          = gs\_data-state.  
            gs\_vendorfamis-zip            = gs\_data-zip.  
            gs\_vendorfamis-phone          = gs\_data-phone.  
            gs\_vendorfamis-county         = gs\_data-county.  
            gs\_vendorfamis-district       = gs\_data-district.  
            gs\_vendorfamis-country        = gs\_data-country.  
            gs\_vendorfamis-vendor\_type    = gs\_data-vendor\_type.  
            gs\_vendorfamis-vendor\_conv    = gs\_data-vendor\_conv.  
            gs\_vendorfamis-minority\_busind = gs\_data-minority\_bus\_ind.  
            gs\_vendorfamis-small\_busind   = gs\_data-small\_bus\_ind.  
            gs\_vendorfamis-contact\_name   = gs\_data-contact\_name..  
            gs\_vendorfamis-vendor\_status  = gs\_data-vendor\_status.  
            gs\_vendorfamis-ss\_fid\_no      = gs\_data-ss\_fid\_no.  
            gs\_vendorfamis-dept\_ind       = gs\_data-dept\_ind+4(1).  
  
\*        gs\_vendorfamis-create\_date  
            IF gs\_data-create\_date IS NOT INITIAL.  
              PERFORM conv\_date USING gs\_data-create\_date  
                    CHANGING gs\_vendorfamis-create\_date.  
            ENDIF.  
  
\*        gs\_vendorfamis-last\_update  
            IF gs\_data-last\_update\_date IS NOT INITIAL.  
              PERFORM conv\_date USING gs\_data-last\_update\_date  
                    CHANGING gs\_vendorfamis-last\_update.  
            ENDIF.  
  
            gs\_vendorfamis-date\_of\_load   = sy-datum.  
            APPEND gs\_vendorfamis TO gt\_vendorfamis.  
          ENDIF.  
        ENDIF.  
      ELSE.  
        EXIT.  
      ENDIF.  
    ENDDO.  
    CLOSE DATASET p\_file.  
    IF gt\_vendorfamis[] IS INITIAL.  
      gv\_no\_data = abap\_true.  
      WRITE : /1 TEXT-016.  
    ENDIF.  
  
  ENDIF.  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form conv\_date  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*&      --> JDATE  
\*&      <-- DATE  
\*&---------------------------------------------------------------------\*  
FORM conv\_date  USING    p\_jdate  
                CHANGING p\_date.  
  
  DATA: lv\_date  TYPE char10,  
        lv\_rdate TYPE char10,  
        lv\_rcode TYPE c LENGTH 2.  
  
  lv\_date = p\_jdate.  
  CALL FUNCTION '/PRA/GET\_JULIAN\_DATE'  
    CHANGING  
      date    = lv\_date  
      rdate   = lv\_rdate  
      rcode   = lv\_rcode  
    EXCEPTIONS  
      goback  = 1  
      failed  = 2  
      warning = 3  
      OTHERS  = 4.  
  
  IF sy-subrc = 1.  
    p\_date = lv\_rdate+0(4) && lv\_rdate+5(2) && lv\_rdate+8(2).  
  ENDIF.  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form update\_table  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*& -->  p1        text  
\*& <--  p2        text  
\*&---------------------------------------------------------------------\*  
FORM update\_table .  
  
  TYPES : BEGIN OF lty\_del,  
            vendor\_no     TYPE zmd\_vendorno,  
            vendor\_no\_sfx TYPE zmd\_vendorno\_sfx,  
            del\_fg        TYPE flag,  
          END OF lty\_del.  
  
  DATA : lt\_del TYPE TABLE OF lty\_del,  
         ls\_del TYPE lty\_del.  
  
  DATA: lv\_index TYPE sy-tabix.  
  DATA : lv\_val\_str TYPE string VALUE ' 1234567890'.  
  SORT gt\_vendorfamis BY vendor\_no vendor\_no\_sfx.  
  IF gt\_vendorfamis[] IS NOT INITIAL.  
    SELECT  
       a~vendor\_no,  
       a~vendor\_no\_sfx,  
       a~del\_fg,  
       b~partner,  
       b~bu\_group,  
       a~CRTBY,  
       a~CRTTS,  
       a~LCHBY,  
       a~LCHTS  
      FROM zmmd\_vendorfamis AS a  
      LEFT OUTER JOIN but000 AS b  
      ON concat( a~vendor\_no, a~vendor\_no\_sfx ) = b~bpext  
      ORDER BY a~vendor\_no, a~vendor\_no\_sfx  
      INTO TABLE @DATA(lt\_db\_vend).  
    IF sy-subrc = 0.  
  
    ENDIF.  
  
    DATA(lt\_tmp) = gt\_vendorfamis[].  
  
\*    GET TIME STAMP FIELD DATA(lv\_stamp).  
  
    LOOP AT gt\_vendorfamis ASSIGNING FIELD-SYMBOL(<lfs\_vendorfamis>). "INTO gs\_vendorfamis.  
      lv\_index = sy-tabix.  
      CLEAR gs\_log.  
      gs\_log-vendor\_no     = <lfs\_vendorfamis>-vendor\_no.  
      gs\_log-vendor\_no\_sfx = <lfs\_vendorfamis>-vendor\_no\_sfx.  
  
      if <lfs\_vendorfamis>-zip is NOT INITIAL.  
        CALL FUNCTION 'PREPARE\_STRING'  
          EXPORTING  
            i\_valid\_chars        = lv\_val\_str  
\*           I\_XVALID\_CHECK       = 'X'  
\*           I\_XCHAR\_REPL         = 'X'  
\*           I\_XTOUPPER           = 'X'  
          changing  
            c\_string             = <lfs\_vendorfamis>-zip.  
      endif.  
  
  
      READ TABLE lt\_db\_vend INTO DATA(ls\_db\_vend)  
      WITH KEY vendor\_no = <lfs\_vendorfamis>-vendor\_no  
               vendor\_no\_sfx = <lfs\_vendorfamis>-vendor\_no\_sfx BINARY SEARCH.  
      IF sy-subrc = 0.  
\*   Update.  
        gs\_log-chind = 'U'.  
        gs\_log-er\_fg = 'S'.  
        gs\_log-msg   = TEXT-018.  
        <lfs\_vendorfamis>-lchby = sy-uname.  
        <lfs\_vendorfamis>-lchts = gv\_stamp.  
        <lfs\_vendorfamis>-crtby = ls\_db\_vend-crtby.  
        <lfs\_vendorfamis>-crtts = ls\_db\_vend-crtts.  
        IF ls\_db\_vend-bu\_group = 'HR'.  
           <lfs\_vendorfamis>-emp\_fg  = abap\_true.  
        ENDIF.  
  
      ELSE.  
\*   Insert  
        gs\_log-chind = 'I'.  
        gs\_log-er\_fg = 'S'.  
        gs\_log-msg   = TEXT-017.  
        <lfs\_vendorfamis>-crtby = sy-uname.  
        <lfs\_vendorfamis>-crtts = gv\_stamp.  
      ENDIF.  
  
\* Validations--  
\* Check duplicate record.  
      READ TABLE lt\_tmp TRANSPORTING NO FIELDS  
      WITH KEY vendor\_no = <lfs\_vendorfamis>-vendor\_no  
            vendor\_no\_sfx = <lfs\_vendorfamis>-vendor\_no\_sfx BINARY SEARCH.  
      IF sy-subrc = 0.  
        IF lv\_index <> sy-tabix.  
          gs\_log-chind = '-'.  
          gs\_log-er\_fg = 'E'.  
          gs\_log-msg   = TEXT-019.  
        ENDIF.  
      ENDIF.  
  
      gv\_total    = gv\_total + 1.  
      IF gs\_log-er\_fg = 'S'.  
        gv\_success  = gv\_success + 1.  
      ELSEIF gs\_log-er\_fg = 'E'.  
        gv\_error    = gv\_error + 1.  
      ENDIF.  
      APPEND gs\_log TO gt\_log.  
    ENDLOOP.  
  
    LOOP AT lt\_db\_vend ASSIGNING FIELD-SYMBOL(<lfs\_db>).  
  
      READ TABLE gt\_vendorfamis TRANSPORTING NO FIELDS  
      WITH KEY vendor\_no = <lfs\_db>-vendor\_no  
               vendor\_no\_sfx = <lfs\_db>-vendor\_no\_sfx BINARY SEARCH.  
      IF sy-subrc <> 0.  
        CLEAR ls\_del.  
        ls\_del = CORRESPONDING #( <lfs\_db> ).  
        IF <lfs\_db>-partner IS NOT INITIAL.  
          ls\_del-del\_fg = abap\_true.  
        ENDIF.  
        APPEND ls\_del TO lt\_del.  
      ENDIF.  
    ENDLOOP.  
  
    IF lt\_del[] IS NOT INITIAL.  
      LOOP AT lt\_del INTO ls\_del.  
        IF ls\_del-del\_fg = abap\_true.  
          UPDATE zmmd\_vendorfamis SET del\_fg = abap\_true  
                                  WHERE vendor\_no = ls\_del-vendor\_no  
                                  AND   vendor\_no\_sfx = ls\_del-vendor\_no\_sfx.  
          IF sy-subrc = 0.  
            CLEAR gs\_log.  
            gs\_log-vendor\_no     = ls\_del-vendor\_no.  
            gs\_log-vendor\_no\_sfx = ls\_del-vendor\_no\_sfx.  
            gs\_log-chind = 'D'.  
            gs\_log-er\_fg = 'S'.  
            gs\_log-msg   = TEXT-032.  
            APPEND gs\_log TO gt\_log.  
            gv\_delete = gv\_delete + 1.  
          ENDIF.  
        ELSE.  
          DELETE FROM zmmd\_vendorfamis WHERE vendor\_no = ls\_del-vendor\_no  
                                       AND   vendor\_no\_sfx = ls\_del-vendor\_no\_sfx.  
          IF sy-subrc = 0.  
            CLEAR gs\_log.  
            gs\_log-vendor\_no     = ls\_del-vendor\_no.  
            gs\_log-vendor\_no\_sfx = ls\_del-vendor\_no\_sfx.  
            gs\_log-chind = 'D'.  
            gs\_log-er\_fg = 'S'.  
            gs\_log-msg   = TEXT-032.  
            APPEND gs\_log TO gt\_log.  
            gv\_delete = gv\_delete + 1.  
          ENDIF.  
        ENDIF.  
  
      ENDLOOP.  
  
      COMMIT WORK.  
    ENDIF.  
  
    DELETE ADJACENT DUPLICATES FROM gt\_vendorfamis COMPARING vendor\_no vendor\_no\_sfx.  
    MODIFY zmmd\_vendorfamis FROM TABLE gt\_vendorfamis.  
    IF sy-subrc = 0.  
      COMMIT WORK.  
      gv\_db\_upd = abap\_true.  
      WRITE /1 TEXT-021.  
    ELSE.  
      ROLLBACK WORK.  
      gv\_db\_upd = abap\_false.  
      WRITE /1 TEXT-022.  
    ENDIF.  
  ENDIF.  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form send\_email  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*& -->  p1        text  
\*& <--  p2        text  
\*&---------------------------------------------------------------------\*  
FORM send\_email .  
  
\*For Email.  
  DATA lo\_send\_request   TYPE REF TO cl\_bcs.  
  DATA lo\_document       TYPE REF TO cl\_document\_bcs.  
  DATA lo\_recipient      TYPE REF TO if\_recipient\_bcs.  
  DATA lo\_bcs\_exception  TYPE REF TO cx\_bcs.  
  DATA lv\_sent\_to\_all    TYPE os\_boolean.  
  DATA lv\_mailto         TYPE ad\_smtpadr.  
  DATA lv\_sub            TYPE so\_obj\_des.  
  
  DATA: lo\_ex\_ref   TYPE REF TO cx\_root,  
        lv\_msg\_text TYPE string.  
  
  CONSTANTS: lc\_htm  TYPE so\_obj\_tp VALUE 'HTM',  
             lc\_tab  TYPE c VALUE cl\_bcs\_convert=>gc\_tab,  
             lc\_crlf TYPE c VALUE cl\_bcs\_convert=>gc\_crlf.  
  
  TRY.  
\*     Create persistent send request ------------------------  
      lo\_send\_request = cl\_bcs=>create\_persistent( ).  
\*     Create Email body using HTML.  
      PERFORM create\_email\_body.  
  
\*     Create document object from internal table with text  
      lv\_sub = TEXT-028.  
      lo\_document = cl\_document\_bcs=>create\_document(  
        i\_type    = lc\_htm  
        i\_text    = gt\_text  
        i\_subject = lv\_sub ).                               "#EC NOTEXT  
  
\*     Add document object to send request  
      lo\_send\_request->set\_document( lo\_document ).  
  
\*--------- add recipient (e-mail address) -----------------------  
      LOOP AT s\_email.  
\*     Create recipient object  
        lv\_mailto = s\_email-low.  
        TRY."Here Recipient is email input p\_email  
            lo\_recipient = cl\_cam\_address\_bcs=>create\_internet\_address( lv\_mailto ). ""#EC CI\_SEL\_NESTED  
          CATCH  cx\_address\_bcs INTO lo\_ex\_ref.  
            lv\_msg\_text = lo\_ex\_ref->get\_text( ).  
\*      Catch exception here  
        ENDTRY.  
        TRY.  
\*      Add recipient object to send request  
            lo\_send\_request->add\_recipient( lo\_recipient ).  
          CATCH  cx\_send\_req\_bcs INTO lo\_ex\_ref.  
            lv\_msg\_text = lo\_ex\_ref->get\_text( ).  
\*      Catch exception here  
        ENDTRY.  
      ENDLOOP.  
\*     ---------- send document ---------------------------------------  
      lv\_sent\_to\_all = lo\_send\_request->send( i\_with\_error\_screen = 'X' ).  
      COMMIT WORK.  
      IF lv\_sent\_to\_all IS INITIAL.  
        MESSAGE i500(sbcoms) WITH lv\_mailto INTO DATA(lv\_msg).  
        WRITE : /1 lv\_msg.  
      ELSE.  
        MESSAGE s022(so) INTO lv\_msg.  
        WRITE: /1 TEXT-033, 15 lv\_mailto.  
      ENDIF.  
  
\*   ------------ exception handling ----------------------------------  
    CATCH cx\_bcs INTO lo\_bcs\_exception.  
      MESSAGE i865(so) WITH lo\_bcs\_exception->error\_type.  
  ENDTRY.  
  
  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form archive  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*& -->  p1        text  
\*& <--  p2        text  
\*&---------------------------------------------------------------------\*  
FORM archive .  
  
  DATA : lv\_source TYPE saepathful,  
         lv\_target TYPE saepathful.  
  
  DATA : lv\_file  TYPE string.  
  
  lv\_source = p\_file.  
  lv\_target = gc\_def\_path && sy-mandt && gc\_archive.  
  lv\_file = gv\_stamp.  
  CONDENSE lv\_file NO-GAPS.  
\*  CONCATENATE 'ZMMR004ARCH\_' lv\_file '.TXT' INTO lv\_file.  
  CONCATENATE gv\_file '\_ARCH\_' lv\_file '.TXT' INTO lv\_file.  
  lv\_target = lv\_target && lv\_file.  
  
  gv\_arch\_file = lv\_target.  
  
  CALL FUNCTION 'ZFM\_ARCHIVE\_SERVER\_TO\_SERVER'  
    EXPORTING  
      sourcepath        = lv\_source  
      targetpath        = lv\_target  
      remove\_sourcefile = 'X'  
\* IMPORTING  
\*     LENGTH            =  
    EXCEPTIONS  
      error             = 1  
      OTHERS            = 2.  
  IF sy-subrc = 0.  
    WRITE : /1 TEXT-020, 16 gv\_arch\_file.  
  ENDIF.  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form write\_logs  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*& -->  p1        text  
\*& <--  p2        text  
\*&---------------------------------------------------------------------\*  
FORM write\_logs .  
  DATA : lv\_file  TYPE string.  
  DATA: lv\_string TYPE string.  
  CONSTANTS  lc\_pipe TYPE c LENGTH 1 VALUE '|'.  
  
  gv\_log\_file = gc\_def\_path && sy-mandt && gc\_log.  
  lv\_file = gv\_stamp.  
  CONDENSE lv\_file NO-GAPS.  
\*  CONCATENATE 'ZMMR004LOG\_' lv\_file '.TXT' INTO lv\_file.  
  CONCATENATE gv\_file '\_LOG\_' lv\_file '.TXT' INTO lv\_file.  
  gv\_log\_file = gv\_log\_file && lv\_file.  
  
  OPEN DATASET gv\_log\_file FOR OUTPUT IN TEXT MODE ENCODING DEFAULT.  
  IF sy-subrc NE 0.  
    WRITE : /1 TEXT-024.  
  ELSE.  
  
    CLEAR lv\_string.  
    lv\_string = 'TOTAL:' && gv\_total.  
    TRANSFER lv\_string TO gv\_log\_file.  
  
    CLEAR lv\_string.  
    lv\_string = 'ERROR:' && gv\_error.  
    TRANSFER lv\_string TO gv\_log\_file.  
  
    CLEAR lv\_string.  
    lv\_string = 'SUCCESS:' && gv\_success.  
    TRANSFER lv\_string TO gv\_log\_file.  
  
    CLEAR lv\_string.  
    lv\_string = 'DELETE:' && gv\_delete.  
    TRANSFER lv\_string TO gv\_log\_file.  
  
    LOOP AT gt\_log INTO gs\_log.  
      CLEAR lv\_string.  
      CONCATENATE gs\_log-vendor\_no gs\_log-vendor\_no\_sfx gs\_log-chind gs\_log-er\_fg gs\_log-msg INTO lv\_string SEPARATED BY lc\_pipe.  
      TRANSFER lv\_string TO gv\_log\_file.  
    ENDLOOP.  
    CLOSE DATASET gv\_log\_file.  
    WRITE : /1 TEXT-023, 20 gv\_log\_file.  
  ENDIF.  
  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form create\_email\_body  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*& -->  p1        text  
\*& <--  p2        text  
\*&---------------------------------------------------------------------\*  
FORM create\_email\_body .  
  
  gs\_text-line = '<html>'.  
  APPEND gs\_text TO gt\_text.  
  
  gs\_text-line = '<body>'.  
  APPEND gs\_text TO gt\_text.  
  
  gs\_text-line = '<p>'.  
  APPEND gs\_text TO gt\_text.  
  gs\_text-line = 'Dear Sir/ Madam,'.  
  APPEND gs\_text TO gt\_text.  
  gs\_text-line = '</p>'.  
  APPEND gs\_text TO gt\_text.  
  
\*Table update  summary.  
  IF gv\_no\_file = abap\_false  
    AND gv\_no\_data =  abap\_false  
    AND gv\_db\_upd = abap\_true.  
  
    gs\_text-line = '<p>'.  
    APPEND gs\_text TO gt\_text.  
    gs\_text-line = 'The vendor Master data interface run is executed and here is the summary of the results'.  
    APPEND gs\_text TO gt\_text.  
    gs\_text-line = '</p>'.  
    APPEND gs\_text TO gt\_text.  
  
    gs\_text-line = '<p></p>'.  
    APPEND gs\_text TO gt\_text.  
  
    gs\_text-line = '<p><u>SUMMARY:</u></p>'.  
    APPEND gs\_text TO gt\_text.  
  
    gs\_text-line = '<p></p>'.  
    APPEND gs\_text TO gt\_text.  
  
  
    gs\_text-line = '<table border = "1" cellpadding = "2" style="color:#000">'.  
    APPEND gs\_text TO gt\_text.  
  
    gs\_text-line = '<tr bgcolor="#f3f9ff" >'  
      && '<th>' && 'Total Records'    && '</th>'  
      && '<th>' && 'Successfully updated' && '</th>'  
      && '<th>' && 'Error'  && '</th></tr>'.  
    APPEND gs\_text TO gt\_text.  
  
    gs\_text-line = '<tr bgcolor="#f3f9ff" >'  
      && '<th>' && |{ gv\_total   ALPHA = OUT }|  && '</th>'  
      && '<th>' && |{ gv\_success ALPHA = OUT }| && '</th>'  
      && '<th>' && |{ gv\_error   ALPHA = OUT }| && '</th></tr>'.  
    APPEND gs\_text TO gt\_text.  
  
    gs\_text-line = '</table>'.  
    APPEND gs\_text TO gt\_text.  
  
\* Blank line  
    gs\_text-line = '<p>'.  
    APPEND gs\_text TO gt\_text.  
    gs\_text-line = '</p>'.  
    APPEND gs\_text TO gt\_text.  
  
\* Log File  
    gs\_text-line = '<p>'.  
    APPEND gs\_text TO gt\_text.  
    gs\_text-line = TEXT-025. " For more details, please check below  
    APPEND gs\_text TO gt\_text.  
    gs\_text-line = '</p>'.  
    APPEND gs\_text TO gt\_text.  
  
    gs\_text-line = '<p><b>'.  
    APPEND gs\_text TO gt\_text.  
    gs\_text-line = TEXT-031. " Log file in the AL11  
    APPEND gs\_text TO gt\_text.  
    gs\_text-line = '</b></p>'.  
    APPEND gs\_text TO gt\_text.  
  
    gs\_text-line = '<p><b>'.  
    APPEND gs\_text TO gt\_text.  
    gs\_text-line = gv\_log\_file.  
    APPEND gs\_text TO gt\_text.  
    gs\_text-line = '</b></p>'.  
    APPEND gs\_text TO gt\_text.  
  
\* Archive File  
    gs\_text-line = '<p><b>'.  
    APPEND gs\_text TO gt\_text.  
    gs\_text-line = TEXT-026.  
    APPEND gs\_text TO gt\_text.  
    gs\_text-line = '</b></p>'.  
    APPEND gs\_text TO gt\_text.  
  
    gs\_text-line = '<p><b>'.  
    APPEND gs\_text TO gt\_text.  
    gs\_text-line = gv\_arch\_file.  
    APPEND gs\_text TO gt\_text.  
    gs\_text-line = '</b></p>'.  
    APPEND gs\_text TO gt\_text.  
  
  
  ELSEIF gv\_no\_file = abap\_true.  
  
    gs\_text-line = '<p>'.  
    APPEND gs\_text TO gt\_text.  
    gs\_text-line = 'The vendor Master data interface run is executed and endded with below error'.  
    APPEND gs\_text TO gt\_text.  
    gs\_text-line = '</p>'.  
    APPEND gs\_text TO gt\_text.  
  
    gs\_text-line = '<p>'.  
    APPEND gs\_text TO gt\_text.  
    gs\_text-line = '</p>'.  
    APPEND gs\_text TO gt\_text.  
  
    gs\_text-line = '<p>'.  
    APPEND gs\_text TO gt\_text.  
    gs\_text-line = TEXT-015.  
    APPEND gs\_text TO gt\_text.  
    gs\_text-line = '</p>'.  
    APPEND gs\_text TO gt\_text.  
  
    gs\_text-line = '<p>'.  
    APPEND gs\_text TO gt\_text.  
    gs\_text-line = p\_file.  
    APPEND gs\_text TO gt\_text.  
    gs\_text-line = '</p>'.  
    APPEND gs\_text TO gt\_text.  
  ELSEIF gv\_no\_data = abap\_true.  
  
    gs\_text-line = '<p>'.  
    APPEND gs\_text TO gt\_text.  
    gs\_text-line = 'The vendor Master data interface run is executed and endded with below error'.  
    APPEND gs\_text TO gt\_text.  
    gs\_text-line = '</p>'.  
    APPEND gs\_text TO gt\_text.  
  
    gs\_text-line = '<p>'.  
    APPEND gs\_text TO gt\_text.  
    gs\_text-line = '</p>'.  
    APPEND gs\_text TO gt\_text.  
  
    gs\_text-line = '<p>'.  
    APPEND gs\_text TO gt\_text.  
    gs\_text-line = TEXT-016.  
    APPEND gs\_text TO gt\_text.  
    gs\_text-line = '</p>'.  
    APPEND gs\_text TO gt\_text.  
  
    gs\_text-line = '<p>'.  
    APPEND gs\_text TO gt\_text.  
    gs\_text-line = p\_file.  
    APPEND gs\_text TO gt\_text.  
    gs\_text-line = '</p>'.  
    APPEND gs\_text TO gt\_text.  
  
  ENDIF.  
  
  
  gs\_text-line = '<p><br></br></p>'.  
  APPEND gs\_text TO gt\_text.  
  
\* Thank you!  
  gs\_text-line = '<p>'.  
  APPEND gs\_text TO gt\_text.  
  gs\_text-line = TEXT-027.  
  APPEND gs\_text TO gt\_text.  
  gs\_text-line = '</p>'.  
  
  APPEND gs\_text TO gt\_text.  
  gs\_text-line = '</body>'.  
  APPEND gs\_text TO gt\_text.  
  gs\_text-line = '</html>'.  
  APPEND gs\_text TO gt\_text.  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form screen\_visiblity  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*& -->  p1        text  
\*& <--  p2        text  
\*&---------------------------------------------------------------------\*  
FORM screen\_visiblity .  
  LOOP AT SCREEN.  
    IF screen-group1 = 'A'.  
      IF pr\_load = abap\_true.  
        screen-input = 1.  
        screen-invisible = 0.  
        MODIFY SCREEN.  
      ELSE.  
        screen-input = 0.  
        screen-invisible = 1.  
        MODIFY SCREEN.  
      ENDIF.  
    ENDIF.  
  
    IF screen-group1 = 'B'.  
      IF pr\_load = abap\_true.  
        screen-input = 0.  
        screen-invisible = 1.  
        MODIFY SCREEN.  
      ELSE.  
        screen-input = 1.  
        screen-invisible = 0.  
        MODIFY SCREEN.  
      ENDIF.  
    ENDIF.  
  
  ENDLOOP.  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form read\_log\_file  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*& -->  p1        text  
\*& <--  p2        text  
\*&---------------------------------------------------------------------\*  
FORM read\_log\_file .  
  DATA : lv\_string TYPE string.  
  
  DATA : lv\_total   TYPE c LENGTH 12,  
         lv\_success TYPE c LENGTH 10,  
         lv\_error   TYPE c LENGTH 10,  
         lv\_delete  TYPE c LENGTH 10.  
  
  CLEAR gt\_log[].  
  OPEN DATASET p\_lfile FOR INPUT IN TEXT MODE ENCODING DEFAULT.  
  IF sy-subrc <> 0.  
    MESSAGE TEXT-001 TYPE 'I'.  
  ELSE.  
  
    DO.  
      CLEAR : lv\_string.  
      READ DATASET p\_lfile INTO lv\_string.  
      IF sy-subrc = 0.  
        IF lv\_string+0(6) = 'TOTAL:'.  
          lv\_total = lv\_string+6(10).  
        ELSEIF lv\_string+0(6) = 'ERROR:'.  
          lv\_error = lv\_string+6(10).  
        ELSEIF lv\_string+0(8) = 'SUCCESS:'.  
          lv\_success = lv\_string+8(10).  
        ELSEIF lv\_string+0(7) = 'DELETE:'.  
          lv\_delete = lv\_string+7(10).  
        ELSE.  
          CLEAR gs\_log.  
          SPLIT lv\_string AT '|' INTO  
          gs\_log-vendor\_no  
          gs\_log-vendor\_no\_sfx  
          gs\_log-chind  
          gs\_log-er\_fg  
          gs\_log-msg.  
          APPEND gs\_log TO gt\_log.  
        ENDIF.  
      ELSE.  
        EXIT.  
      ENDIF.  
  
    ENDDO.  
    CLOSE DATASET p\_lfile.  
  
    gv\_total   = lv\_total.  
    gv\_success = lv\_success.  
    gv\_error   = lv\_error.  
    gv\_delete  = lv\_delete.  
  
  ENDIF.  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form display\_log  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*& -->  p1        text  
\*& <--  p2        text  
\*&---------------------------------------------------------------------\*  
FORM display\_log .  
  DATA: lo\_gr\_alv       TYPE REF TO cl\_salv\_table,  
        lo\_gr\_functions TYPE REF TO cl\_salv\_functions\_list.  
  
  DATA: "lo\_selections TYPE REF TO cl\_salv\_selections,  
    lo\_columns TYPE REF TO cl\_salv\_columns,  
    lo\_column  TYPE REF TO cl\_salv\_column\_table.  
  
  DATA: lo\_header  TYPE REF TO cl\_salv\_form\_layout\_grid,  
        lo\_h\_label TYPE REF TO cl\_salv\_form\_label.  
  DATA: lv\_text TYPE char100.  
  
\*Create ALV object.  
  TRY.  
      CALL METHOD cl\_salv\_table=>factory  
        IMPORTING  
          r\_salv\_table = lo\_gr\_alv  
        CHANGING  
          t\_table      = gt\_log.  
  
    CATCH cx\_salv\_msg.  
  
  ENDTRY.  
  
\* Show all ALV functions.  
  lo\_gr\_functions = lo\_gr\_alv->get\_functions( ).  
  lo\_gr\_functions->set\_all( abap\_true ).  
  
\* Fit the columns  
  lo\_columns = lo\_gr\_alv->get\_columns( ).  
  lo\_columns->set\_optimize( abap\_true ).  
  
\*\* Code to remove coloumn or change text for reference  
\*   TRY.  
\*       lo\_column ?= lo\_columns->get\_column( 'MANDT' ).  
\*       lo\_column->set\_visible( if\_salv\_c\_bool\_sap=>false ).  
\*\*      lo\_column->set\_long\_text( 'MyTitle' ).  
\*\*      lo\_column->set\_medium\_text( 'MyTitle' ).  
\*\*      lo\_column->set\_short\_text( 'MyTitle' ).  
\*     CATCH cx\_salv\_not\_found.  
\*     CATCH cx\_salv\_existing.  
\*     CATCH cx\_salv\_data\_error.  
\*   ENDTRY.  
  
  
  TRY.  
      lo\_column ?= lo\_columns->get\_column( 'ER\_FG' ).  
      lo\_column->set\_visible( if\_salv\_c\_bool\_sap=>true ).  
      lo\_column->set\_long\_text( 'Status' ).  
      lo\_column->set\_medium\_text( 'Status' ).  
      lo\_column->set\_short\_text( 'Status' ).  
    CATCH cx\_salv\_not\_found.  
    CATCH cx\_salv\_existing.  
    CATCH cx\_salv\_data\_error.  
  ENDTRY.  
  
  TRY.  
      lo\_column ?= lo\_columns->get\_column( 'CHIND' ).  
      lo\_column->set\_visible( if\_salv\_c\_bool\_sap=>true ).  
      lo\_column->set\_long\_text( 'Change indicator' ).  
      lo\_column->set\_medium\_text( 'Change indicator' ).  
      lo\_column->set\_short\_text( 'Indicator' ).  
    CATCH cx\_salv\_not\_found.  
    CATCH cx\_salv\_existing.  
    CATCH cx\_salv\_data\_error.  
  ENDTRY.  
  
  TRY.  
      lo\_column ?= lo\_columns->get\_column( 'MSG' ).  
      lo\_column->set\_visible( if\_salv\_c\_bool\_sap=>true ).  
      lo\_column->set\_long\_text( 'Message' ).  
      lo\_column->set\_medium\_text( 'Message' ).  
      lo\_column->set\_short\_text( 'Message' ).  
    CATCH cx\_salv\_not\_found.  
    CATCH cx\_salv\_existing.  
    CATCH cx\_salv\_data\_error.  
  ENDTRY.  
  
  
  CREATE OBJECT lo\_header.  
\* information in Bold  
  
  lo\_h\_label = lo\_header->create\_label( row = 1 column = 1 ).  
  lv\_text = |{ gv\_total ALPHA = OUT }|.  
  CONDENSE lv\_text.  
  lv\_text = TEXT-002 && lv\_text.  
  lo\_h\_label->set\_text( lv\_text ).  
  
  lo\_h\_label = lo\_header->create\_label( row = 2 column = 1 ).  
  lv\_text = |{ gv\_success ALPHA = OUT }|.  
  CONDENSE lv\_text.  
  lv\_text = TEXT-003 && lv\_text.  
  lo\_h\_label->set\_text( lv\_text ).  
  
  lo\_h\_label = lo\_header->create\_label( row = 3 column = 1 ).  
  lv\_text = |{ gv\_error ALPHA = OUT }|.  
  CONDENSE lv\_text.  
  lv\_text = TEXT-004 && lv\_text.  
  lo\_h\_label->set\_text( lv\_text ).  
  
  lo\_gr\_alv->set\_top\_of\_list( lo\_header ).  
  
\* Display ALV  
  lo\_gr\_alv->display( ).  
  
ENDFORM.