```
*&---------------------------------------------------------------------*
*& Report  ZFI_REP_FB60
*&
REPORT  zfi_rep_fb60.
*----------------------------------------------------------------------*
*       Type pools
*----------------------------------------------------------------------*

TYPE-POOLS : slis.
*----------------------------------------------------------------------*
*       Types
*----------------------------------------------------------------------*
TYPES : BEGIN OF ty_final,
        sno   TYPE i,
        bukrs TYPE bkpf-bukrs,
        hkont TYPE bseg-hkont,
        bldat TYPE char10,
        ref_doc_no TYPE bkpf-xblnr,
        budat TYPE char10,
        blart TYPE bkpf-blart,
        shkzg TYPE bseg-shkzg,
        dmbtr TYPE bseg-dmbtr,
        waers TYPE bkpf-waers,
        bupla TYPE bseg-bupla,
        secco TYPE bseg-secco,
        sgtxt TYPE bseg-sgtxt,
        gst_part TYPE bseg-gst_part,
        plc_sup TYPE bseg-plc_sup,
        zuonr TYPE bseg-zuonr,
        bktxt TYPE bkpf-bktxt,
        gsber TYPE bseg-gsber,
        vatdate TYPE char10,
        xmwst TYPE char2,
        zfbdt TYPE char10,
        zterm TYPE char4,
        mwskz TYPE bseg-mwskz,
        kostl TYPE bseg-kostl,
*        prctr TYPE bseg-prctr,
        aufnr TYPE bseg-aufnr,
        hsn_sac TYPE bseg-hsn_sac,
        wtax TYPE wt_withcd,
*************************************  Added by Ankit
        fis_period TYPE monat, "Period
        kursf TYPE kursk,      "Exchange Rate
        umskz TYPE umskz,      "Special G/L Indicator
        cs_trans_t  TYPE  rmvct,       "Transaction Type
        value_date  TYPE  valut,       "Value Date
        pmnt_block  TYPE acpi_zlspr,   "Payment Block
        pymt_meth   TYPE acpi_zlsch,   "Payment Method
        wbs_element  TYPE ps_posid,     "WBS Element
        profit_ctr  TYPE prctr,        "Profit Center
        ref_key_1 TYPE xref1,          "Reference Key 1
        ref_key_2 TYPE xref2,          "Reference Key 2
        ref_key_3  TYPE xref3,          "Reference Key 3
        END OF ty_final,

        BEGIN OF ty_log,
         sno TYPE i,
         msgtype TYPE char10,
         belnr TYPE belnr_d,
         message TYPE char200,
        END OF ty_log.

*---------------------------------------------------------------------*
*       work area and internal table
*---------------------------------------------------------------------*
DATA :  gs_log   TYPE ty_log,
        gs_final TYPE ty_final,
        gs_final1 TYPE ty_final,
        gt_log   TYPE TABLE OF ty_log,
        gt_final TYPE TABLE OF ty_final.

*---------------------------------------------------------------------*
*       alv fieldcat
*---------------------------------------------------------------------*
DATA : gs_fieldcat TYPE slis_fieldcat_alv,
       gt_fieldcat TYPE slis_t_fieldcat_alv.
*---------------------------------------------------------------------*
*       bapi related declartion
*---------------------------------------------------------------------*

DATA :  gt_accountgl          TYPE STANDARD TABLE OF bapiacgl09,
        gt_accountreceivable  TYPE STANDARD TABLE OF bapiacar09,
        gt_accountpayable  TYPE STANDARD TABLE OF bapiacap09,
        gt_currencyamount     TYPE STANDARD TABLE OF bapiaccr09,
        gt_currencyamount1    TYPE STANDARD TABLE OF bapiaccr09,
        gt_accounttax         TYPE STANDARD TABLE OF bapiactx09,
        gt_accountwt          TYPE STANDARD TABLE OF bapiacwt09,
        gt_return             TYPE STANDARD TABLE OF bapiret2,
        gt_extension1         TYPE STANDARD TABLE OF bapiacextc,
        gt_extension2         TYPE STANDARD TABLE OF bapiparex.

DATA : gs_documentheader     TYPE bapiache09,
       gs_accountgl          TYPE bapiacgl09,
       gs_accountreceivable  TYPE bapiacar09,
       gs_accountpayable     TYPE bapiacap09,
       gs_currencyamount     TYPE bapiaccr09,
       gs_return             TYPE bapiret2,
       gs_accounttax         TYPE bapiactx09,
       gs_accountwt          TYPE bapiacwt09,
       gs_extension1         TYPE bapiacextc,
       gs_extension2         TYPE bapiparex,
       gs_customercpd        TYPE bapiacpa09.

DATA:  gv_type               TYPE bapiache09-obj_type,
       gv_key                TYPE bapiache09-obj_key,
       gv_sys                TYPE bapiache09-obj_sys.
DATA : lv_amount    TYPE bseg-wrbtr,
         lv_baseamt TYPE bseg-wrbtr,
         lv_index   TYPE sy-tabix,
         wa_tax     TYPE rtax1u15,
         lt_tax     TYPE TABLE OF rtax1u15,
         lwa_tax    TYPE j_1it030k,
         lt_j1tax   TYPE TABLE OF j_1it030k.
*---------------------------------------------------------------------*
*       Global variables
*---------------------------------------------------------------------*

DATA : gv_posnr TYPE bapiacgl09-itemno_acc.
DATA : gv_net_amount TYPE bseg-dmbtr,
       gv_flag TYPE c.
```

```
*----------------------------------------------------------------------*
*
*----------------------------------------------------------------------*
*Selection screen
*----------------------------------------------------------------------*

SELECTION-SCREEN : BEGIN OF BLOCK blk2 WITH FRAME TITLE text-002.
PARAMETERS   p_file TYPE rlgrap-filename OBLIGATORY.
PARAMETERS : pr_test RADIOBUTTON GROUP grp1 DEFAULT 'X',
             pr_post RADIOBUTTON GROUP grp1.
SELECTION-SCREEN : END OF BLOCK blk2.

*----------------------------------------------------------------------*
*       Initialisation
*----------------------------------------------------------------------*
INITIALIZATION.

*----------------------------------------------------------------------**
*       At selection-screen
*----------------------------------------------------------------------*
*AT SELECTION-SCREEN.
AT SELECTION-SCREEN ON VALUE-REQUEST FOR p_file.
  " this perform is used to get the select file location
  PERFORM p_file.
***********************************************************************
*       S T A R T   O F   S E L E C T I O N
***********************************************************************
START-OF-SELECTION.
  " this perform is used to upload data from excel into internal table
  PERFORM upload_excel_to_int.
  " this perform is used to upload data through bapi
  PERFORM process_data.

END-OF-SELECTION.
  " this perform is used to display log of uploaded data
  PERFORM display_data.


*&---------------------------------------------------------------------*
*&      Form  FILL_AMOUNT_TO_GL
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*      -->P_0728   text
*      -->P_GS_ADPROINV_DTL_PEAMOUNT  text
*      -->P_GS_ADPRO_GL_MAP_HKONT  text
*----------------------------------------------------------------------*
FORM fill_amount_to_gl.

  ADD 1 TO gv_posnr.
  CLEAR : gs_accountgl, gs_extension2.

  CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
    EXPORTING
      input  = gs_final-hkont
    IMPORTING
      output = gs_final-hkont.

  lv_baseamt               = gs_final-dmbtr.
  gs_accountgl-itemno_acc = gv_posnr.
  gs_accountgl-gl_account = gs_final-hkont.
  gs_accountgl-comp_code  = gs_final-bukrs.

  gs_accountgl-orderid = gs_final-aufnr.

  CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
    EXPORTING
      input  = gs_final-kostl
    IMPORTING
      output = gs_final-kostl.

  gs_accountgl-costcenter = gs_final-kostl.
  gs_accountgl-item_text  = gs_final-sgtxt.
  gs_accountgl-alloc_nmbr = gs_final-zuonr.

  IF gs_final-mwskz IS NOT INITIAL.
    gs_accountgl-tax_code   = gs_final-mwskz.
    gs_accountgl-itemno_tax = gv_posnr.
  ENDIF.

************Period***********  Added by Ankit Kumar on 22.09.2017
  IF gs_final-fis_period IS NOT INITIAL.
    CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
      EXPORTING
        input  = gs_final-fis_period
      IMPORTING
        output = gs_final-fis_period.

    MOVE gs_final-fis_period TO gs_accountgl-fis_period.
  ENDIF.
****************************************************
************Transaction Type***********  Added by Ankit Kumar on 22.09.2017
  IF gs_final-cs_trans_t IS NOT INITIAL.
    CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
      EXPORTING
        input  = gs_final-cs_trans_t
      IMPORTING
        output = gs_final-cs_trans_t.

    MOVE gs_final-cs_trans_t TO gs_accountgl-cs_trans_t.
  ENDIF.
****************************************************
************Value Date***********  Added by Ankit Kumar on 22.09.2017
  IF gs_final-value_date IS NOT INITIAL.
    PERFORM date_change USING gs_final-value_date CHANGING gs_accountgl-value_date.
  ENDIF.
****************************************************
************WBS Element***********  Added by Ankit Kumar on 22.09.2017
  IF gs_final-wbs_element IS NOT INITIAL.
    CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
      EXPORTING
        input  = gs_final-wbs_element
      IMPORTING
        output = gs_final-wbs_element.

    MOVE gs_final-wbs_element TO gs_accountgl-wbs_element.
  ENDIF.
****************************************************
************Profit Center***********  Added by Ankit Kumar on 22.09.2017
  IF gs_final-profit_ctr IS NOT INITIAL.
    CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
      EXPORTING
```

```
        input  = gs_final-profit_ctr
      IMPORTING
        output = gs_final-profit_ctr.

    MOVE gs_final-profit_ctr TO gs_accountgl-profit_ctr.
  ENDIF.
****************************************************
************Profit Center***********  Added by Ankit Kumar on 22.09.2017
  IF gs_final-ref_key_1 IS NOT INITIAL.
    CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
      EXPORTING
        input  = gs_final-ref_key_1
      IMPORTING
        output = gs_final-ref_key_1.

    MOVE gs_final-ref_key_1 TO gs_accountgl-ref_key_1.
  ENDIF.
****************************************************
************Profit Center***********  Added by Ankit Kumar on 22.09.2017
  IF gs_final-ref_key_2 IS NOT INITIAL.
    CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
      EXPORTING
        input  = gs_final-ref_key_2
      IMPORTING
        output = gs_final-ref_key_2.

    MOVE gs_final-ref_key_2 TO gs_accountgl-ref_key_2.
  ENDIF.
****************************************************
************Profit Center***********  Added by Ankit Kumar on 22.09.2017
  IF gs_final-ref_key_3 IS NOT INITIAL.
    CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
      EXPORTING
        input  = gs_final-ref_key_3
      IMPORTING
        output = gs_final-ref_key_3.

    MOVE gs_final-ref_key_3 TO gs_accountgl-ref_key_3.
  ENDIF.
****************************************************
  APPEND gs_accountgl TO gt_accountgl.

  gs_currencyamount-itemno_acc = gv_posnr.
  gs_currencyamount-curr_type  = '00'.
  gs_currencyamount-currency   = gs_final-waers.

  IF gs_final-shkzg = 'S'.
    gs_currencyamount-amt_doccur = lv_baseamt.
  ELSE.
    gs_currencyamount-amt_doccur = lv_baseamt * -1.
  ENDIF.
  gs_currencyamount-amt_base = gs_currencyamount-amt_doccur.

************Exchange Rate***********  Added by Ankit Kumar on 22.09.2017
  IF gs_final-kursf IS NOT INITIAL.
    CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
      EXPORTING
        input  = gs_final-kursf
      IMPORTING
        output = gs_final-kursf.
    MOVE gs_final-kursf TO gs_currencyamount-exch_rate.
  ENDIF.
****************************************************
  APPEND gs_currencyamount TO gt_currencyamount.

  " custom field be added in bapi extension2
  " these field will be handled in class ZBADI_ACC_DOCUEMNT_CLASS
  " badi BADI_ACC_DOCUMENT
  gs_extension2-structure  = gv_posnr.
  gs_extension2-valuepart1 = gs_final-bupla.
  gs_extension2-valuepart2 = gs_final-gst_part.
  gs_extension2-valuepart3 = gs_final-plc_sup.
  gs_extension2-valuepart4 = gs_final-hsn_sac.

  APPEND gs_extension2 TO gt_extension2.

  " if tax code is exist then calcualte tax
  IF gs_final-mwskz IS NOT INITIAL.


    lv_amount = gs_currencyamount-amt_doccur.
    CALL FUNCTION 'CALCULATE_TAX_FROM_NET_AMOUNT'
      EXPORTING
        i_bukrs                    = gs_final-bukrs
        i_mwskz                    = gs_final-mwskz
*       I_TXJCD                    = ' '
        i_waers                    = 'INR'
        i_wrbtr                    = lv_amount
"gs_currencyamount-amt_doccur
*       I_ZBD1P                    = 0
*       I_PRSDT                    =
*       I_PROTOKOLL                =
*       I_TAXPS                    =
*       I_ACCNT_EXT                =
*       I_ACCDATA                  =
*       I_PRICING_REFRESH_TX       = ' '
*     IMPORTING
*       E_FWNAV                    =
*       E_FWNVV                    =
*       E_FWSTE                    =
*       E_FWAST                    =
      TABLES
        t_mwdat                    = lt_tax
     EXCEPTIONS
       bukrs_not_found            = 1
       country_not_found          = 2
       mwskz_not_defined          = 3
       mwskz_not_valid            = 4
       ktosl_not_found            = 5
       kalsm_not_found            = 6
       parameter_error            = 7
       knumh_not_found            = 8
       kschl_not_found            = 9
       unknown_error              = 10
       account_not_found          = 11
       txjcd_not_valid            = 12
       OTHERS                     = 13
              .
    IF sy-subrc <> 0.
* Implement suitable error handling here
    ENDIF.
    LOOP AT lt_tax INTO wa_tax.

      gs_accounttax-gl_account = wa_tax-hkont.
      gs_accounttax-tax_code = gs_final-mwskz.
      gs_accounttax-itemno_tax = gv_posnr.
      gs_accounttax-cond_key = wa_tax-kschl.
      gs_accounttax-acct_key = wa_tax-ktosl.
      APPEND gs_accounttax TO gt_accounttax.
      CLEAR : gs_accounttax, lwa_tax.

      gs_currencyamount-curr_type  = '00'.
      gs_currencyamount-currency   = 'INR'.
      gs_currencyamount-amt_base = lv_amount.
      gs_currencyamount-amt_doccur = wa_tax-wmwst.
      gs_currencyamount-tax_amt = wa_tax-wmwst.
      APPEND gs_currencyamount TO gt_currencyamount1.
      CLEAR gs_currencyamount.
    ENDLOOP.

  ENDIF.

  " tax line item should be added in main amount line item
  IF gv_flag EQ 'X'.
    CLEAR : gs_currencyamount.
    LOOP AT gt_currencyamount1 INTO gs_currencyamount.
      lv_index = sy-tabix.
      ADD 1 TO gv_posnr.
      gs_currencyamount-itemno_acc = gv_posnr.
      APPEND gs_currencyamount TO gt_currencyamount.
      CLEAR gs_currencyamount.

      READ TABLE gt_accounttax INTO gs_accounttax INDEX lv_index.
      IF sy-subrc = 0.
        gs_accounttax-itemno_acc = gv_posnr.
        MODIFY gt_accounttax FROM gs_accounttax INDEX lv_index
        TRANSPORTING itemno_acc.
      ENDIF.
    ENDLOOP.

    REFRESH : gt_currencyamount1.
    CLEAR: gv_flag, gs_accounttax.

  ENDIF.

ENDFORM.                    " FILL_AMOUNT_TO_GL
*&----------------------------------------------------------------------*
*&      Form  PROCESS_DATA1
*&----------------------------------------------------------------------*
*       text
*-----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*-----------------------------------------------------------------------*
FORM process_data .

* SORT gt_final BY sno.
  DATA lv_continue.
  DATA :lv_kunnr TYPE kna1-kunnr,
        lv_onecust1 TYPE kna1-kunnr,
       lv_lifnr TYPE lfa1-lifnr.
  DATA : ls_knb1 LIKE knb1,
         lv_term TYPE knb1-zterm,
         ls_kna1 TYPE kna1.
  DATA lv_sgtxt TYPE bseg-sgtxt.
  DATA : lv_cdate(10),
         lv_header TYPE c.

  DATA : lt_lfbw TYPE TABLE OF lfbw,
         lwa_lfbw TYPE lfbw.
  DATA: lt_t059z TYPE TABLE OF t059z,
        lwa_t059z TYPE t059z.
  LOOP AT gt_final INTO gs_final.
    AT NEW sno.
      lv_header = 'X'.
    ENDAT.

    AT END OF sno.
      gv_flag = 'X'.
    ENDAT.

    IF lv_header EQ 'X'.
      CLEAR : gv_net_amount, lv_header.
      CLEAR gv_posnr.
      CLEAR lv_sgtxt.
      REFRESH : gt_accountgl,gt_accountreceivable,gt_accountpayable,
                gt_currencyamount,gt_extension1,gt_accounttax,
                gt_return, gt_extension2, gt_accountwt.
      CLEAR : gs_documentheader.
      CLEAR lv_continue.
      lv_lifnr = gs_final-hkont.

      CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
        EXPORTING
          input  = lv_lifnr
        IMPORTING
          output = lv_lifnr.

      gv_posnr = '0000000001'.
      gs_documentheader-username = sy-uname .
      gs_documentheader-comp_code  = gs_final-bukrs.

      " change date format into system format
      PERFORM date_change USING gs_final-bldat CHANGING
gs_documentheader-doc_date.
      PERFORM date_change USING gs_final-budat CHANGING
gs_documentheader-pstng_date.
      gs_documentheader-ref_doc_no = gs_final-ref_doc_no.
      gs_documentheader-header_txt = gs_final-bktxt.
      gs_documentheader-doc_type   = gs_final-blart.
      PERFORM date_change USING gs_final-vatdate CHANGING
gs_documentheader-vatdate.
      gs_accountpayable-itemno_acc = gv_posnr.
      gs_accountpayable-vendor_no   = lv_lifnr.

      gs_accountpayable-pmnttrms = gs_final-zterm.
      CLEAR lv_term.
      gs_accountpayable-item_text = gs_final-sgtxt.
      gs_accountpayable-alloc_nmbr = gs_final-zuonr.
      gs_accountpayable-businessplace = gs_final-bupla.
      gs_accountpayable-bus_area = gs_final-gsber.
      gs_accountpayable-sectioncode = gs_final-secco.
```

```
************Special G/L Indicator***********  Added by Ankit Kumar on 22.09.2017
      CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
        EXPORTING
          input  = gs_final-umskz
        IMPORTING
          output = gs_final-umskz.

      MOVE gs_final-umskz TO gs_accountpayable-sp_gl_ind.
****************************************************
************Payment Block***********  Added by Ankit Kumar on 22.09.2017
      IF  gs_final-pmnt_block IS NOT INITIAL.
        CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
          EXPORTING
            input  = gs_final-pmnt_block
          IMPORTING
            output = gs_final-pmnt_block.

        MOVE gs_final-pmnt_block TO gs_accountpayable-pmnt_block.
      ENDIF.
****************************************************
************Payment Method***********  Added by Ankit Kumar on 22.09.2017
      IF  gs_final-pymt_meth IS NOT INITIAL.
        CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
          EXPORTING
            input  = gs_final-pymt_meth
          IMPORTING
            output = gs_final-pymt_meth.

        MOVE gs_final-pymt_meth TO gs_accountpayable-pymt_meth.
      ENDIF.
****************************************************
      PERFORM date_change USING gs_final-zfbdt CHANGING
gs_accountpayable-bline_date.
      APPEND  gs_accountpayable TO gt_accountpayable.

      gs_currencyamount-itemno_acc = gv_posnr.
      gs_currencyamount-curr_type  = '00'.
      gs_currencyamount-currency   = gs_final-waers.
      IF gs_final-shkzg = 'S'.
        gs_currencyamount-amt_doccur = gs_final-dmbtr.
      ELSE.
        " if tax code is exist then calcualte tax
*        IF gs_final-mwskz IS NOT INITIAL.

        READ TABLE gt_final INTO gs_final1 INDEX 2.
        lv_amount = gs_final-dmbtr.
        IF gs_final1-mwskz NE 'RC'.
          CALL FUNCTION 'CALCULATE_TAX_FROM_NET_AMOUNT'
            EXPORTING
              i_bukrs                    = gs_final-bukrs
              i_mwskz                    = gs_final1-mwskz
*       I_TXJCD                    = ' '
              i_waers                    = 'INR'
              i_wrbtr                    = lv_amount
      "gs_currencyamount-amt_doccur
*       I_ZBD1P                    = 0
*       I_PRSDT                    =
*       I_PROTOKOLL                =
*       I_TAXPS                    =
*       I_ACCNT_EXT                =
*       I_ACCDATA                  =
*       I_PRICING_REFRESH_TX       = ' '
*     IMPORTING
*       E_FWNAV                    =
*       E_FWNVV                    =
*       E_FWSTE                    =
*       E_FWAST                    =
            TABLES
              t_mwdat                    = lt_tax
           EXCEPTIONS
             bukrs_not_found            = 1
             country_not_found          = 2
             mwskz_not_defined          = 3
             mwskz_not_valid            = 4
             ktosl_not_found            = 5
             kalsm_not_found            = 6
             parameter_error            = 7
             knumh_not_found            = 8
             kschl_not_found            = 9
             unknown_error              = 10
             account_not_found          = 11
             txjcd_not_valid            = 12
             OTHERS                     = 13
                    .
          IF sy-subrc <> 0.
* Implement suitable error handling here
          ENDIF.

          gs_currencyamount-amt_doccur = lv_amount. "MOD-001

          LOOP AT lt_tax INTO wa_tax.

*            gs_currencyamount-amt_doccur = lv_amount + wa_tax-wmwst. "MOD-001
            gs_currencyamount-amt_doccur = gs_currencyamount-amt_doccur + wa_tax-wmwst. "MOD-001

          ENDLOOP.

        ENDIF.
        gs_currencyamount-amt_doccur = gs_currencyamount-amt_doccur * -1.
*        gs_final-dmbtr * -1.
        CLEAR:lv_amount,lt_tax ,wa_tax.
      ENDIF.
      APPEND gs_currencyamount TO gt_currencyamount.
      SORT gt_currencyamount BY itemno_acc.

      " code for withholding tax
      REFRESH lt_lfbw.

      SELECT * FROM lfbw
      INTO TABLE lt_lfbw
      WHERE lifnr = lv_lifnr
      AND bukrs = gs_final-bukrs
      AND wt_withcd NE space.

      SELECT * FROM t059z
               INTO TABLE lt_t059z
               FOR ALL ENTRIES IN lt_lfbw
                      WHERE land1 = 'IN'
                        AND witht = lt_lfbw-witht.

      LOOP AT lt_lfbw INTO lwa_lfbw.
```

```
        gs_accountwt-itemno_acc = gv_posnr.
        gs_accountwt-wt_type = lwa_lfbw-witht.
        gs_accountwt-wt_code = lwa_lfbw-wt_withcd.
        gs_accountwt-bas_amt_lc = gs_final-dmbtr.

        READ TABLE lt_t059z INTO lwa_t059z WITH KEY witht = lwa_lfbw-witht.
        gs_accountwt-man_amt_tc = lwa_t059z-qsatz * 100 .
        gs_accountwt-man_amt_tc = gs_final-dmbtr /
gs_accountwt-man_amt_tc.

        IF gs_accountwt-man_amt_tc IS NOT INITIAL.
          DATA  x TYPE p .

          x = gs_accountwt-man_amt_tc.
          gs_accountwt-man_amt_tc = x.
        ENDIF.


        IF  gs_accountwt-man_amt_tc < 0.
          gs_accountwt-man_amt_tc = gs_accountwt-man_amt_tc * -1.

        ENDIF.
        gs_accountwt-man_amt_lc = gs_accountwt-man_amt_tc.
        gs_accountwt-bas_amt_ind = 'X'.
        gs_accountwt-man_amt_ind = 'X'.
        APPEND gs_accountwt TO gt_accountwt.
        CLEAR: lwa_lfbw,lwa_t059z.
      ENDLOOP.

      gs_extension2-structure  = gv_posnr.
      gs_extension2-valuepart1 = gs_final-bupla.
      gs_extension2-valuepart2 = gs_final-gst_part.
      gs_extension2-valuepart3 = gs_final-plc_sup.
      gs_extension2-valuepart4 = gs_final-hsn_sac.

      APPEND gs_extension2 TO gt_extension2.
    ELSE.

      PERFORM fill_amount_to_gl.
    ENDIF.

    AT END OF sno.

      IF pr_test = 'X'.
        CALL FUNCTION 'BAPI_ACC_DOCUMENT_CHECK'
          EXPORTING
            documentheader    = gs_documentheader
*           customercpd       = gs_customercpd
          TABLES
            accountgl         = gt_accountgl
*           accountreceivable = gt_accountreceivable
            accounttax        = gt_accounttax
            accountpayable    = gt_accountpayable
            currencyamount    = gt_currencyamount
            extension1        = gt_extension1
            return            = gt_return
            extension2        = gt_extension2
            accountwt         = gt_accountwt.
        READ TABLE gt_return INTO gs_return WITH KEY type = 'E'.
        IF sy-subrc <> 0.
          gs_log-sno = gs_final-sno.
          gs_log-message = 'Testing Successfull'.
          gs_log-msgtype = 'Success'.
          APPEND gs_log TO gt_log.
          CLEAR gs_log.
        ELSE.
          WRITE : /01 gs_final-sno, 21 'Error while testing'.
          LOOP AT gt_return INTO gs_return.
            gs_log-sno = gs_final-sno.
            gs_log-message = gs_return-message.
            gs_log-msgtype = 'Error'.
            APPEND gs_log TO gt_log.
            CLEAR gs_log.
          ENDLOOP.
        ENDIF.
      ELSE.

        CLEAR gt_return[].
        CALL FUNCTION 'BAPI_ACC_DOCUMENT_POST'
          EXPORTING
            documentheader    = gs_documentheader
            customercpd       = gs_customercpd
          IMPORTING                      ";Commentfor>;=ECC5.0
            obj_type          = gv_type  ";Commentfor>;=ECC5.0
            obj_key           = gv_key   ";Commentfor>;=ECC5.0
            obj_sys           = gv_sys   ";Commentfor>;=ECC5.0
          TABLES
            accountgl         = gt_accountgl
*           accountreceivable = gt_accountreceivable
            accounttax        = gt_accounttax
            accountpayable    = gt_accountpayable
            currencyamount    = gt_currencyamount
            extension1        = gt_extension1
            return            = gt_return
            extension2        = gt_extension2
            accountwt         = gt_accountwt.

        READ TABLE gt_return INTO gs_return WITH KEY type = 'E'.
        IF sy-subrc <> 0.
          CALL FUNCTION 'BAPI_TRANSACTION_COMMIT'
            EXPORTING
              wait = 'X'.
          READ TABLE gt_return INTO gs_return
                   WITH KEY type = 'S' id = 'RW' number = '605'.

*          WRITE : /01 gs_final-sno, 21 gs_return-message.
          gs_log-sno = gs_final-sno.
          gs_log-belnr = gs_return-message_v2+(10).
          gs_log-message = gs_return-message.
          gs_log-msgtype = 'Success'.
          APPEND gs_log TO gt_log.
          CLEAR gs_log.

        ELSE.
          CALL FUNCTION 'BAPI_TRANSACTION_ROLLBACK'.
*          WRITE : /01 gs_final-sno, 21 'Error while Posting'.
          LOOP AT gt_return INTO gs_return.
            gs_log-sno = gs_final-sno.
            gs_log-message = gs_return-message.
            gs_log-msgtype = 'Error'.
            APPEND gs_log TO gt_log.
            CLEAR gs_log.
          ENDLOOP.
        ENDIF.

      ENDIF.
      CLEAR : gs_return,gs_accountreceivable,gs_accountpayable,
gs_currencyamount.
    ENDAT.
  ENDLOOP.

ENDFORM.                    " PROCESS_DATA1
*&---------------------------------------------------------------------*
*&      Form  FILL_AMOUNT_TO_GL1
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*      -->P_0818   text
*      -->P_GS_ADPROCM_DTL_PEAMOUNT  text
*      -->P_GS_ADPRO_GL_MAP_HKONT  text
*      -->P_LV_KUNNR  text
*      -->P_GS_ADPROCM_HDR_CLIENTID  text
*----------------------------------------------------------------------*
FORM fill_amount_to_gl1  USING    uv_fieldname
                                 uv_amount
                                 uv_hkont
                                 uv_kunnr
                                 uv_clientid.
  ADD 1 TO gv_posnr.
  gs_accountreceivable-itemno_acc = gv_posnr.
  gs_accountreceivable-customer  = uv_kunnr.
  gs_accountreceivable-item_text = gs_final-sgtxt.
*  uv_kunnr = gs_adprocm_hdr-clientid.
  CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
    EXPORTING
      input  = uv_kunnr
    IMPORTING
      output = uv_kunnr.
  gs_accountreceivable-ref_key_1 = uv_kunnr.
  gs_accountreceivable-comp_code  = gs_final-bukrs.
  gs_accountreceivable-item_text = gs_final-sgtxt.

  CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
    EXPORTING
      input  = uv_clientid
    IMPORTING
      output = uv_clientid.
  gs_accountreceivable-ref_key_1 = uv_clientid.
*  gs_accountreceivable-alloc_nmbr = gs_adprocm_hdr-ronumber.
  APPEND gs_accountreceivable TO gt_accountreceivable.


  gs_currencyamount-itemno_acc = gv_posnr.
  gs_currencyamount-curr_type  = '00'.
  gs_currencyamount-currency   = 'INR'.
  gs_currencyamount-amt_doccur = uv_amount * -1.


  APPEND gs_currencyamount TO gt_currencyamount.



  " custom field be added in bapi extension2
  " these field will be handled in class ZZBADI_ACC_DOCUEMNT_CLASS
  " badi BADI_ACC_DOCUMENT
  gs_extension2-structure  = gv_posnr.
  gs_extension2-valuepart1 = gs_final-bupla.
  gs_extension2-valuepart2 = gs_final-gst_part.
  gs_extension2-valuepart3 = gs_final-plc_sup.
  gs_extension2-valuepart4 = gs_final-hsn_sac.

  APPEND gs_extension2 TO gt_extension2.
  CLEAR gs_extension2.


ENDFORM.                    " FILL_AMOUNT_TO_GL1



*&---------------------------------------------------------------------*
*&      Form  P_FILE
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
FORM p_file .
  CALL FUNCTION 'F4_FILENAME'
    EXPORTING
      program_name  = syst-cprog
      dynpro_number = syst-dynnr
      field_name    = ' '
    IMPORTING
      file_name     = p_file.
ENDFORM.                    " P_FILE
*&---------------------------------------------------------------------*
*&      Form  UPLOAD_EXCEL_TO_INT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
FORM upload_excel_to_int .
  DATA : li_itab TYPE TABLE OF alsmex_tabline,
         lwa_excel TYPE alsmex_tabline,
         lv_index TYPE sy-tabix.
  DATA: count TYPE i.
  FIELD-SYMBOLS : <fs>.

  CALL FUNCTION 'ALSM_EXCEL_TO_INTERNAL_TABLE'
    EXPORTING
      filename                = p_file
      i_begin_col             = '1'
      i_begin_row             = '3'
      i_end_col               = '30'
      i_end_row               = '5000'
    TABLES
      intern                  = li_itab
    EXCEPTIONS
      inconsistent_parameters = 1
      upload_ole              = 2
      OTHERS                  = 3.
  IF sy-subrc NE 0.
    FORMAT COLOR COL_BACKGROUND INTENSIFIED.
    WRITE : / 'ERROR IN EXCEL FILE'.
    EXIT.
  ELSE.
    SORT li_itab BY row col.
    LOOP AT li_itab INTO lwa_excel.
      MOVE lwa_excel-col TO lv_index.
      ASSIGN COMPONENT lv_index OF STRUCTURE  gs_final TO <fs>.
      MOVE lwa_excel-value TO <fs>.
      AT END OF row.

        APPEND gs_final TO gt_final.
        CLEAR gs_final.
      ENDAT.
    ENDLOOP.
  ENDIF.

  IF <fs> IS ASSIGNED.
    UNASSIGN <fs>.
  ENDIF.

ENDFORM.                    " UPLOAD_EXCEL_TO_INT
*&---------------------------------------------------------------------*
*&      Form  DATE_CHANGE
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*      -->P_GS_FINAL_BLDAT  text
*      <--P_GS_DOCUMENTHEADER_DOC_DATE  text
*----------------------------------------------------------------------*
FORM date_change  USING    p_f_date
                  CHANGING p_f_cdate.
  p_f_cdate+0(4) = p_f_date+6(4).
  p_f_cdate+4(2) = p_f_date+3(2).
  p_f_cdate+6(2) = p_f_date+0(2).
ENDFORM.                    " DATE_CHANGE
*&---------------------------------------------------------------------*
*&      Form  DISPLAY_DATA
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
FORM display_data .
  DATA : lv_layout TYPE slis_layout_alv.
  lv_layout-colwidth_optimize = 'X'.

  gs_fieldcat-fieldname = 'SNO'.
  gs_fieldcat-seltext_l = 'SNO'.
  gs_fieldcat-seltext_m = 'SNO'.
  APPEND gs_fieldcat TO gt_fieldcat.

  gs_fieldcat-fieldname = 'BELNR'.
  gs_fieldcat-seltext_l = 'Document Number'.
  gs_fieldcat-seltext_m = 'Document Number'.
  APPEND gs_fieldcat TO gt_fieldcat.

  gs_fieldcat-fieldname = 'MSGTYPE'.
  gs_fieldcat-seltext_l = 'Message Type'.
  gs_fieldcat-seltext_m = 'Message Type'.
  APPEND gs_fieldcat TO gt_fieldcat.

  gs_fieldcat-fieldname = 'MESSAGE'.
  gs_fieldcat-seltext_l = 'Message'.
  gs_fieldcat-seltext_m = 'Message'.
  APPEND gs_fieldcat TO gt_fieldcat.

  CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'
    EXPORTING
*     I_INTERFACE_CHECK        = ' '
*     I_BYPASSING_BUFFER       = ' '
*     I_BUFFER_ACTIVE          = ' '
      i_callback_program       = sy-repid
*     I_CALLBACK_PF_STATUS_SET = ' '
      is_layout                = lv_layout
      it_fieldcat              = gt_fieldcat
*     IT_EXCLUDING             =
*     IT_SPECIAL_GROUPS        =
*     IT_SORT                  =
*     IT_FILTER                =
*     IS_SEL_HIDE              =
*     I_DEFAULT                = 'X'
      i_save                   = 'X'
    TABLES
      t_outtab                 = gt_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.                            " DISPLAY_DATA
```