\*&---------------------------------------------------------------------\*  
\*& Include          ZFIR\_CASH\_JOURNAL\_SIMPLE\_F01  
\*&---------------------------------------------------------------------\*  
  
\*----------------------------------------------------------------------\*  
FORM park\_data.  
  DATA: l\_valid            TYPE c,  
        lv\_parking\_allowed TYPE flag.  
\* §7.Check if any errors exist in protocol by using method  
\*    CHECK\_CHANGED\_DATA of your ALV Grid instance.  
  
\* The method CHECK\_CHANGED\_DATA checks all new cells syntactically,  
\* raises event DATA\_CHANGED and looks then for any entries  
\* in the error protocol. If any exist the parameter e\_valid  
\* is initial ('X' in the other case).  
\*  
  CALL METHOD go\_grid->check\_changed\_data  
    IMPORTING  
      e\_valid = l\_valid.  
  
  IF l\_valid IS INITIAL.  
    CALL FUNCTION 'POPUP\_TO\_INFORM'  
      EXPORTING  
        titel = TEXT-i06  
        txt1  = TEXT-i07  
        txt2  = TEXT-i08  
        txt3  = TEXT-i09.  
  
  ELSE.  
    CLEAR: gt\_disp\_msgs.  
    PERFORM park\_document USING abap\_false  
                       CHANGING lv\_parking\_allowed. "Check  
  
    IF lv\_parking\_allowed = abap\_true.  
      PERFORM park\_document USING abap\_true  
                         CHANGING lv\_parking\_allowed.  "Park  
    ENDIF.  
  
    go\_grid->refresh\_table\_display( ).  
  ENDIF.  
ENDFORM.                    "save\_data  
\*------------------------------------------  
FORM park\_document USING iv\_park  
                CHANGING ev\_parking\_allowed.  
  DATA: ls\_outtab         LIKE LINE OF gt\_outtab,  
        ls\_outtab\_current LIKE LINE OF gt\_outtab,  
        ls\_outtab\_next    LIKE LINE OF gt\_outtab.  
  
  DATA: lv\_bldat         TYPE char10,  
        lv\_budat         TYPE char10,  
        lv\_index         TYPE sy-tabix,  
        lv\_index\_next    TYPE sy-tabix,  
        lv\_ok\_code\_final TYPE bdc\_fval.  
  
  REFRESH bdcdata.  
\* Convert Dates to proper format for BDC processing  
  CALL FUNCTION 'CONVERSION\_EXIT\_PDATE\_OUTPUT'  
    EXPORTING  
      input  = 0200\_i\_doc\_date  
    IMPORTING  
      output = lv\_bldat.  
  
  CALL FUNCTION 'CONVERSION\_EXIT\_PDATE\_OUTPUT'  
    EXPORTING  
      input  = 0200\_i\_posting\_date  
    IMPORTING  
      output = lv\_budat.  
  
\* Fill Header Data  
  PERFORM bdc\_dynpro      USING 'SAPLF040' '0100'.  
  PERFORM bdc\_field       USING 'BDC\_OKCODE'  
                                '/00'.  
  
  PERFORM bdc\_field       USING 'BKPF-BLDAT'  
                                 lv\_bldat.  
  PERFORM bdc\_field       USING 'BKPF-BLART'  
                                0200\_i\_document\_type.  
  PERFORM bdc\_field       USING 'BKPF-BUKRS'  
                                0200\_i\_company\_code .  
  PERFORM bdc\_field       USING 'BKPF-BUDAT'  
                                lv\_budat.  
  PERFORM bdc\_field       USING 'BKPF-WAERS'  
                                0200\_i\_currency .  
  PERFORM bdc\_field       USING 'BKPF-BKTXT'  
                                0200\_i\_header\_text.  
  
  "NEW REFERENCE Field  
  PERFORM bdc\_field       USING 'BKPF-XBLNR'  
                              0200\_i\_reference.  
  
\* Delete lines that do not have a receipt key  
  DELETE gt\_outtab WHERE receipt\_key IS INITIAL.  
  
\* Fill Line Item Data  
  LOOP AT gt\_outtab INTO ls\_outtab.  
    ls\_outtab\_current = ls\_outtab.  
    lv\_index = sy-tabix.  
    lv\_index\_next = lv\_index + 1.  
  
    AT FIRST.  
      PERFORM bdc\_field       USING 'RF05V-NEWBS'  
                                    ls\_outtab\_current-posting\_key.  
      PERFORM bdc\_field       USING 'RF05V-NEWUM'  
                                    ls\_outtab\_current-special\_gl\_ind.  
      IF ls\_outtab\_current-gl\_account IS NOT INITIAL.  
        PERFORM bdc\_field       USING 'RF05V-NEWKO'  
                                      ls\_outtab\_current-gl\_account.  
      ELSEIF ls\_outtab\_current-customer IS NOT INITIAL.  
        PERFORM bdc\_field       USING 'RF05V-NEWKO'  
                                      ls\_outtab\_current-customer.  
      ELSEIF ls\_outtab\_current-vendor IS NOT INITIAL.  
        PERFORM bdc\_field       USING 'RF05V-NEWKO'  
                                      ls\_outtab\_current-vendor.  
      ENDIF.  
  
    ENDAT.  
    READ TABLE gt\_outtab INTO ls\_outtab\_next INDEX lv\_index\_next.  
    IF sy-subrc = 0.  
      IF ls\_outtab\_current-gl\_account IS NOT INITIAL.  
        IF ls\_outtab\_next-gl\_account IS NOT INITIAL.  
          PERFORM bdc\_gl\_account USING '/00'  
                                       ls\_outtab\_current  
                                       ls\_outtab\_next-posting\_key  
                                       ls\_outtab\_next-gl\_account  
                                       ls\_outtab\_next-special\_gl\_ind.  
        ELSEIF ls\_outtab\_next-customer IS NOT INITIAL.  
          PERFORM bdc\_gl\_account USING '/00'  
                                       ls\_outtab\_current  
                                       ls\_outtab\_next-posting\_key  
                                       ls\_outtab\_next-customer  
                                       ls\_outtab\_next-special\_gl\_ind.  
        ELSEIF ls\_outtab\_next-vendor IS NOT INITIAL.  
          PERFORM bdc\_gl\_account USING '/00'  
                                       ls\_outtab\_current  
                                       ls\_outtab\_next-posting\_key  
                                       ls\_outtab\_next-vendor  
                                       ls\_outtab\_next-special\_gl\_ind.  
        ENDIF.  
      ELSEIF ls\_outtab\_current-customer IS NOT INITIAL.  
        IF ls\_outtab\_next-gl\_account IS NOT INITIAL.  
          PERFORM bdc\_customer USING '/00'  
                                     ls\_outtab\_current  
                                     ls\_outtab\_next-posting\_key  
                                     ls\_outtab\_next-gl\_account  
                                     ls\_outtab\_next-special\_gl\_ind.  
        ELSEIF ls\_outtab\_next-customer IS NOT INITIAL.  
          PERFORM bdc\_customer USING '/00'  
                                     ls\_outtab\_current  
                                     ls\_outtab\_next-posting\_key  
                                     ls\_outtab\_next-customer  
                                     ls\_outtab\_next-special\_gl\_ind..  
        ELSEIF ls\_outtab\_next-vendor IS NOT INITIAL.  
          PERFORM bdc\_customer USING '/00'  
                                     ls\_outtab\_current  
                                     ls\_outtab\_next-posting\_key  
                                     ls\_outtab\_next-vendor  
                                     ls\_outtab\_next-special\_gl\_ind..  
        ENDIF.  
      ELSEIF ls\_outtab-vendor IS NOT INITIAL.  
        IF ls\_outtab\_next-gl\_account IS NOT INITIAL.  
          PERFORM bdc\_vendor USING '/00'  
                                     ls\_outtab\_current  
                                     ls\_outtab\_next-posting\_key  
                                     ls\_outtab\_next-gl\_account.  
        ELSEIF ls\_outtab\_next-customer IS NOT INITIAL.  
          PERFORM bdc\_vendor USING '/00'  
                                     ls\_outtab\_current  
                                     ls\_outtab\_next-posting\_key  
                                     ls\_outtab\_next-customer.  
        ELSEIF ls\_outtab\_next-vendor IS NOT INITIAL.  
          PERFORM bdc\_vendor USING '/00'  
                                     ls\_outtab\_current  
                                     ls\_outtab\_next-posting\_key  
                                     ls\_outtab\_next-vendor.  
        ENDIF.  
      ENDIF.  
    ELSE.  
\* This is the last record in the table.  
\* Determine if this is a check or a park  
      IF iv\_park = abap\_true.  
        lv\_ok\_code\_final = 'PBBP'.  " old call to park changed to park-completed '=BP'.  
      ELSE.  
        lv\_ok\_code\_final = '=PB'.   " preliminary check in FBV1  
      ENDIF.  
  
\* Fill last line of data  
      IF ls\_outtab\_current-gl\_account IS NOT INITIAL.  
        PERFORM bdc\_gl\_account USING lv\_ok\_code\_final  
                                     ls\_outtab\_current  
                                     ' '  
                                     ' '  
                                     ' '.  
      ELSEIF ls\_outtab\_current-customer IS NOT INITIAL.  
        PERFORM bdc\_customer USING lv\_ok\_code\_final  
                                   ls\_outtab\_current  
                                   ' '  
                                   ' '  
                                   ' '.  
      ELSEIF ls\_outtab\_current-vendor IS NOT INITIAL.  
        PERFORM bdc\_vendor USING lv\_ok\_code\_final  
                                 ls\_outtab\_current  
                                 ' '  
                                 ' '.  
      ENDIF.  
  
      IF iv\_park = abap\_false.  
        PERFORM bdc\_dynpro      USING 'SAPLF040' '0700'.  
        PERFORM bdc\_field       USING 'BDC\_OKCODE'  
                                      '=RW'.  
  
        PERFORM bdc\_dynpro      USING 'SAPLSPO1' '0200'.  
        PERFORM bdc\_field       USING 'BDC\_OKCODE'  
                                      '=YES'.  
      ENDIF.  
    ENDIF.  
  ENDLOOP.  
  
\* Call Transaction  
  PERFORM bdc\_transaction USING 'FBV1'  
                       CHANGING ev\_parking\_allowed.  
ENDFORM.                    "update\_database  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  GET\_INITIAL\_DATA  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*  -->  p1        text  
\*  <--  p2        text  
\*----------------------------------------------------------------------\*  
FORM get\_initial\_data .  
  SELECT \*  
    FROM zfit\_cr\_def\_val  
    INTO TABLE gt\_zfi\_cr\_def\_val.  
ENDFORM.                    " GET\_INITIAL\_DATA  
  
\*----------------------------------------------------------------------\*  
\*        Start new screen                                              \*  
\*----------------------------------------------------------------------\*  
FORM bdc\_dynpro USING program dynpro.  
  CLEAR bdcdata.  
  bdcdata-program  = program.  
  bdcdata-dynpro   = dynpro.  
  bdcdata-dynbegin = 'X'.  
  APPEND bdcdata.  
ENDFORM.                    "BDC\_DYNPRO  
\*----------------------------------------------------------------------\*  
\*        Insert field                                                  \*  
\*----------------------------------------------------------------------\*  
FORM bdc\_field USING fnam fval.  
  IF fval IS NOT INITIAL.  
    CLEAR bdcdata.  
    bdcdata-fnam = fnam.  
    bdcdata-fval = fval.  
    APPEND bdcdata.  
  ENDIF.  
ENDFORM.                    "BDC\_FIELD  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  BDC\_TRANSACTION  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*      -->TCODE      text  
\*----------------------------------------------------------------------\*  
FORM bdc\_transaction USING tcode  
                  CHANGING ev\_parking\_allowed.  
  
  DATA: ctumode LIKE ctu\_params-dismode VALUE 'N'.  
  DATA: cupdate LIKE ctu\_params-updmode VALUE 'L'.  
  DATA: lt\_messtab TYPE STANDARD TABLE OF  bdcmsgcoll.  
  
  DATA: l\_mstring TYPE string.  
  
  FIELD-SYMBOLS: <fs\_messtab> LIKE LINE OF lt\_messtab.  
  
  REFRESH lt\_messtab.  
  CALL TRANSACTION tcode USING bdcdata  
                   MODE   ctumode  
                   UPDATE cupdate  
                   MESSAGES INTO lt\_messtab.  
  
  LOOP AT lt\_messtab ASSIGNING <fs\_messtab>.  
    READ TABLE gt\_disp\_msgs WITH TABLE KEY table\_line = <fs\_messtab> TRANSPORTING NO FIELDS.  
    CHECK sy-subrc <> 0.  
  
    INSERT <fs\_messtab> INTO TABLE gt\_disp\_msgs.  
  
    MESSAGE ID <fs\_messtab>-msgid TYPE 'I' NUMBER <fs\_messtab>-msgnr  
      WITH <fs\_messtab>-msgv1 <fs\_messtab>-msgv2 <fs\_messtab>-msgv3 <fs\_messtab>-msgv4  
      INTO l\_mstring.  
  
    MESSAGE l\_mstring TYPE 'I'.  
  ENDLOOP.  
  
\* If check is successful set parking allowed indicator to true  
  READ TABLE lt\_messtab WITH KEY msgtyp = 'S'  
                                 msgid  = 'FP'  
                                 msgnr  = '093' TRANSPORTING NO FIELDS.  
  IF sy-subrc = 0.  
    ev\_parking\_allowed = abap\_true.  
  ELSE.  
    ev\_parking\_allowed = abap\_false.  
  ENDIF.  
  
\* If posting is successful return to home screen.  
  READ TABLE lt\_messtab WITH KEY msgtyp = 'S'  
                                 msgid  = 'FP'  
                                 msgnr  = '001' TRANSPORTING NO FIELDS.  
  IF sy-subrc = 0.  
    PERFORM unlock\_record.  
    PERFORM delete\_saved\_record.  
    PERFORM refresh\_global\_data.  
    CALL SCREEN 100.  
  ENDIF.  
ENDFORM.                    "BDC\_TRANSACTION  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  bdc\_gl\_account  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*      -->I\_GL\_ACCOUNT  text  
\*      -->I\_AMOUNT      text  
\*----------------------------------------------------------------------\*  
FORM bdc\_gl\_account USING i\_okcode  
                          is\_outtab\_current TYPE lty\_outtab  
                          i\_posting\_key\_next  
                          i\_gl\_account\_next  
                          i\_spec\_gl\_ind\_next.  
  
  DATA: lv\_wrbtr(16) TYPE c,  
        lv\_sgtxt     TYPE text50,  
        xusr01       TYPE usr01.  
  
  CONSTANTS: lv\_dot(1) TYPE c VALUE '.',  
             lv\_com(1) TYPE c VALUE ','.  
  
  SELECT SINGLE \* FROM usr01 INTO xusr01  
  WHERE bname = sy-uname.  
  
  MOVE is\_outtab\_current-amount TO lv\_wrbtr.  
  
  CASE xusr01-dcpfm.  
    WHEN  ' ' OR 'Y'.  
      REPLACE ALL OCCURRENCES OF lv\_dot IN lv\_wrbtr WITH lv\_com.  
  ENDCASE.  
  
  CONDENSE lv\_wrbtr .  
  
\*  IF is\_outtab\_current-special\_gl\_ind IS INITIAL.  
\*    PERFORM bdc\_dynpro      USING 'SAPLF040' '0301'.  
\*    PERFORM bdc\_field       USING 'BDC\_OKCODE'  
\*                                  i\_okcode.  
\*  ELSE.  
\*    PERFORM bdc\_dynpro      USING 'SAPLF040' '0303'.  
\*    PERFORM bdc\_field       USING 'BDC\_OKCODE'  
\*                                  i\_okcode.  
\*  ENDIF.  
  
  PERFORM bdc\_dynpro      USING 'SAPLF040' '0300'.  
  PERFORM bdc\_field       USING 'BDC\_OKCODE'  
                                i\_okcode.  
  
  PERFORM bdc\_field       USING 'BSEG-WRBTR'  
                                lv\_wrbtr.  
  
\*  CONCATENATE is\_outtab\_current-receipt\_key  
\*              is\_outtab\_current-receipt\_key\_desc  
\*         INTO lv\_sgtxt.  
  
  lv\_sgtxt = is\_outtab\_current-sgtxt.  
  
  PERFORM bdc\_field       USING 'BSEG-SGTXT'  
                                lv\_sgtxt.  
  
  PERFORM bdc\_field       USING 'DKACB-FMORE'  
                                'X'.  
  PERFORM bdc\_field       USING 'RF05V-NEWBS'  
                                i\_posting\_key\_next.  
  PERFORM bdc\_field       USING 'RF05V-NEWKO'  
                                i\_gl\_account\_next.  
  PERFORM bdc\_field       USING 'RF05V-NEWUM'  
                                i\_spec\_gl\_ind\_next.  
  
\* Populate Coding Block Data  
  PERFORM bdc\_dynpro      USING 'SAPLKACB' '0002'.  
  PERFORM bdc\_field       USING 'BDC\_OKCODE'  
                                '=ENTE'.  
  PERFORM bdc\_field       USING 'COBL-GSBER'  
                                is\_outtab\_current-business\_area.  
  PERFORM bdc\_field       USING 'COBL-KOSTL'  
                                is\_outtab\_current-cost\_center.  
  PERFORM bdc\_field       USING 'COBL-GEBER'  
                                is\_outtab\_current-fund.  
  PERFORM bdc\_field       USING 'COBL-GRANT\_NBR'  
                                is\_outtab\_current-grant\_nbr.  
  PERFORM bdc\_field       USING 'COBL-AUFNR'  
                                is\_outtab\_current-internal\_order.  
  PERFORM bdc\_field       USING 'COBL-PS\_POSID'  
                                is\_outtab\_current-wbs.  
  "New Item Fields  
  PERFORM bdc\_field       USING 'COBL-FKBER'  
                              is\_outtab\_current-fkber.  
  PERFORM bdc\_field       USING 'SKAT-TXT20'  
                            is\_outtab\_current-txt20.  
  PERFORM bdc\_field       USING 'COBL-FISTL'  
                        is\_outtab\_current-fistl.  
  PERFORM bdc\_field       USING 'COBL-MEASURE'  
                    is\_outtab\_current-measure.  
  PERFORM bdc\_field       USING 'COBL-BUDGET\_PD'  
                   is\_outtab\_current-budget\_pd.  
  PERFORM bdc\_field       USING 'ACGL\_ITEM-ZUONR'  
                                 is\_outtab\_current-zuonr.  
\*  PERFORM bdc\_field       USING 'BSEG-SGTXT'  
\*                                 is\_outtab\_current-sgtxt.  
  
ENDFORM.                    "bdc\_gl\_account  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  bdc\_customer  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
FORM bdc\_customer USING i\_okcode  
                        is\_outtab\_current TYPE lty\_outtab  
                        i\_posting\_key\_next  
                        i\_account\_next  
                        i\_spec\_gl\_ind\_next.  
  
  DATA: lv\_wrbtr(16)        TYPE c,  
        lv\_one\_time\_account TYPE xcpdk,  
        lv\_baseline\_date    TYPE char10,  
        lv\_sgtxt            TYPE text50,  
        xusr01              TYPE usr01.  
  
  CONSTANTS: lv\_dot(1) TYPE c VALUE '.',  
             lv\_com(1) TYPE c VALUE ','.  
  
  SELECT SINGLE \* FROM usr01 INTO xusr01  
               WHERE bname = sy-uname.  
  
  MOVE is\_outtab\_current-amount TO lv\_wrbtr.  
  
  
  CASE xusr01-dcpfm.  
    WHEN  ' ' OR 'Y'.  
      REPLACE ALL OCCURRENCES OF lv\_dot IN lv\_wrbtr WITH lv\_com.  
  ENDCASE.  
  CONDENSE lv\_wrbtr.  
  
\* Convert Dates to proper format for BDC processing  
  IF is\_outtab\_current-baseline\_date IS NOT INITIAL.  
    CALL FUNCTION 'CONVERSION\_EXIT\_PDATE\_OUTPUT'  
      EXPORTING  
        input  = is\_outtab\_current-baseline\_date  
      IMPORTING  
        output = lv\_baseline\_date.  
  ENDIF.  
  
  SELECT SINGLE xcpdk  
    FROM kna1  
    INTO lv\_one\_time\_account  
   WHERE kunnr = is\_outtab\_current-customer.  
  IF lv\_one\_time\_account IS NOT INITIAL.  
\*   Fill OTA Data  
    PERFORM bdc\_dynpro      USING 'SAPLFCPD' '0100'.  
    PERFORM bdc\_field       USING 'BDC\_OKCODE'  
                                  '/00'.  
  
    PERFORM bdc\_field       USING 'BSEC-SPRAS'  
                                  'EN'.  
    PERFORM bdc\_field       USING 'BSEC-NAME1'  
                                  is\_outtab\_current-name1.  
    PERFORM bdc\_field       USING 'BSEC-NAME2'  
                                  is\_outtab\_current-name2.  
    PERFORM bdc\_field       USING 'BSEC-STRAS'  
                                  is\_outtab\_current-street.  
    PERFORM bdc\_field       USING 'BSEC-ORT01'  
                                  is\_outtab\_current-city.  
    PERFORM bdc\_field       USING 'BSEC-PSTLZ'  
                                  is\_outtab\_current-postal\_code.  
    PERFORM bdc\_field       USING 'BSEC-LAND1'  
                                  is\_outtab\_current-country.  
    PERFORM bdc\_field       USING 'BSEC-REGIO'  
                                  is\_outtab\_current-region.  
  ENDIF.  
  
  IF is\_outtab\_current-special\_gl\_ind IS INITIAL.  
    PERFORM bdc\_dynpro      USING 'SAPLF040' '0301'.  
    PERFORM bdc\_field       USING 'BDC\_OKCODE'  
                                  i\_okcode.  
  ELSE.  
    PERFORM bdc\_dynpro      USING 'SAPLF040' '0303'.  
    PERFORM bdc\_field       USING 'BDC\_OKCODE'  
                                  i\_okcode.  
  ENDIF.  
  
  PERFORM bdc\_field       USING 'BSEG-WRBTR'  
                                lv\_wrbtr.  
  
\*  CONCATENATE is\_outtab\_current-receipt\_key  
\*              is\_outtab\_current-receipt\_key\_desc  
\*         INTO lv\_sgtxt.  
\*  
\*  PERFORM bdc\_field       USING 'BSEG-SGTXT'  
\*                                lv\_sgtxt.  
  
  PERFORM bdc\_field       USING 'BSEG-REBZG '  
                                is\_outtab\_current-ref\_invoice.  
  PERFORM bdc\_field       USING 'BSEG-GSBER'  
                                is\_outtab\_current-business\_area.  
  PERFORM bdc\_field       USING 'BSEG-ZFBDT'  
                                lv\_baseline\_date.  
  
  IF is\_outtab\_current-fund IS NOT INITIAL OR  
     is\_outtab\_current-grant\_nbr IS NOT INITIAL.  
  
    PERFORM bdc\_field       USING 'BDC\_OKCODE'  
                                  '=ZK'.  
  
    PERFORM bdc\_dynpro      USING 'SAPLF040' '0331'.  
    PERFORM bdc\_field       USING 'BSEG-GRANT\_NBR'  
                                  is\_outtab\_current-grant\_nbr.  
  
    PERFORM bdc\_field       USING 'BSEG-GEBER'  
                                  is\_outtab\_current-fund.  
  
\*    PERFORM bdc\_field       USING 'BDC\_OKCODE'  
\*                                  i\_okcode.  
\*  
\*    IF lv\_one\_time\_account IS NOT INITIAL.  
\*      PERFORM bdc\_dynpro      USING 'SAPLFCPD' '0100'.  
\*      PERFORM bdc\_field       USING 'BDC\_OKCODE'  
\*                                    i\_okcode.  
\*    ENDIF.  
  ENDIF.  
  
  PERFORM bdc\_field       USING 'RF05V-NEWBS'  
                                i\_posting\_key\_next.  
  PERFORM bdc\_field       USING 'RF05V-NEWKO'  
                                i\_account\_next.  
  PERFORM bdc\_field       USING 'RF05V-NEWUM'  
                                i\_spec\_gl\_ind\_next.  
  
ENDFORM.                    "bdc\_customer  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  SET\_DEFAULT\_VALUES  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*  -->  p1        text  
\*  <--  p2        text  
\*----------------------------------------------------------------------\*  
FORM set\_default\_values .  
  0200\_i\_doc\_date = sy-datum.  
  0200\_i\_document\_type = 'ZC'."0100\_I\_DOCUMENT\_TYPE.  "'Z1'.  
  0200\_i\_company\_code  = '1000'.  
  0200\_i\_currency = 'USD'.  
ENDFORM.                    " SET\_DEFAULT\_VALUES  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  bdc\_vendor  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
FORM bdc\_vendor   USING i\_okcode  
                        is\_outtab\_current TYPE lty\_outtab  
                        i\_posting\_key\_next  
                        i\_account\_next.  
  
  DATA: lv\_wrbtr(16)        TYPE c,  
        lv\_baseline\_date    TYPE char10,  
        lv\_one\_time\_account TYPE lfa1-xcpdk,  
        xusr01              TYPE usr01.  
  
  CONSTANTS: lv\_dot(1) TYPE c VALUE '.',  
             lv\_com(1) TYPE c VALUE ','.  
  
  SELECT SINGLE \* FROM usr01 INTO xusr01  
               WHERE bname = sy-uname.  
  
  MOVE is\_outtab\_current-amount TO lv\_wrbtr.  
  
  CASE xusr01-dcpfm.  
    WHEN  ' ' OR 'Y'.  
      REPLACE ALL OCCURRENCES OF lv\_dot IN lv\_wrbtr WITH lv\_com.  
  ENDCASE.  
  CONDENSE lv\_wrbtr.  
  
\* Convert Dates to proper format for BDC processing  
  CALL FUNCTION 'CONVERSION\_EXIT\_PDATE\_OUTPUT'  
    EXPORTING  
      input  = is\_outtab\_current-baseline\_date  
    IMPORTING  
      output = lv\_baseline\_date.  
  
  
  SELECT SINGLE xcpdk  
    FROM lfa1  
    INTO lv\_one\_time\_account  
   WHERE lifnr = is\_outtab\_current-vendor.  
  
  IF lv\_one\_time\_account IS NOT INITIAL.  
\* Fill OTA Data  
  
    PERFORM bdc\_dynpro      USING 'SAPLFCPD' '0100'.  
    PERFORM bdc\_field       USING 'BDC\_OKCODE'  
                                  '/00'.  
  
    PERFORM bdc\_field       USING 'BSEC-SPRAS'  
                                  'EN'.  
    PERFORM bdc\_field       USING 'BSEC-NAME1'  
                                  is\_outtab\_current-name1.  
    PERFORM bdc\_field       USING 'BSEC-NAME2'  
                                  is\_outtab\_current-name2.  
    PERFORM bdc\_field       USING 'BSEC-STRAS'  
                                  is\_outtab\_current-street.  
    PERFORM bdc\_field       USING 'BSEC-ORT01'  
                                  is\_outtab\_current-city.  
    PERFORM bdc\_field       USING 'BSEC-PSTLZ'  
                                  is\_outtab\_current-postal\_code.  
    PERFORM bdc\_field       USING 'BSEC-LAND1'  
                                  is\_outtab\_current-country.  
    PERFORM bdc\_field       USING 'BSEC-REGIO'  
                                  is\_outtab\_current-region.  
  ENDIF.  
  
  
  PERFORM bdc\_dynpro      USING 'SAPLF040' '0302'.  
  PERFORM bdc\_field       USING 'BDC\_OKCODE'  
                                i\_okcode.  
  
  PERFORM bdc\_field       USING 'BSEG-WRBTR'  
                                lv\_wrbtr.  
  PERFORM bdc\_field       USING 'BSEG-REBZG '  
                                is\_outtab\_current-ref\_invoice.  
  PERFORM bdc\_field       USING 'BSEG-GSBER'  
                                is\_outtab\_current-business\_area.  
  PERFORM bdc\_field       USING 'BSEG-ZFBDT'  
                                lv\_baseline\_date.  
  
  PERFORM bdc\_field       USING 'RF05V-NEWBS'  
                                i\_posting\_key\_next.  
  PERFORM bdc\_field       USING 'RF05V-NEWKO'  
                                i\_account\_next.  
ENDFORM.                    "bdc\_vendor  
\*&---------------------------------------------------------------------\*  
\*&      Form  ATTACH\_DOCUMENT  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*  -->  p1        text  
\*  <--  p2        text  
\*----------------------------------------------------------------------\*  
\*FORM attach\_document .  
\*data: ls\_object type borident.  
\*  
\*create object go\_gos\_manager.  
\*  
\*ls\_object-OBJKEY = '100041000000732012'.  
\*ls\_object-OBJTYPE = 'FIPP'.  
\*  
\*CALL METHOD MANAGER->START\_SERVICE\_DIRECT  
\*      EXPORTING  
\*        IP\_SERVICE       = 'CREATE\_ATTA'  
\*        IS\_OBJECT        = OBJECT  
\*EXCEPTIONS  
\*        NO\_OBJECT        = 1  
\*        OBJECT\_INVALID   = 2  
\*        EXECUTION\_FAILED = 3  
\*        OTHERS           = 4.  
\*  
\*go\_gos\_manager->display\_toolbox( exporting is\_object = ls\_object ).  
\*ENDFORM.                    " ATTACH\_DOCUMENT  
\*&---------------------------------------------------------------------\*  
\*&      Form  DISABLE\_SCREEN\_FIELDS  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*  -->  p1        text  
\*  <--  p2        text  
\*----------------------------------------------------------------------\*  
FORM disable\_screen\_fields .  
  LOOP AT SCREEN.  
    CASE screen-name.  
      WHEN '0200\_I\_DOC\_DATE'  
        OR '0200\_I\_POSTING\_DATE'  
        OR '0200\_I\_DOCUMENT\_TYPE'  
        OR '0200\_I\_COMPANY\_CODE'  
        OR '0200\_I\_CURRENCY'  
        OR '0200\_I\_HEADER\_TEXT'  
        OR '0200\_I\_REFERENCE'.  
        screen-input = 0.  
        screen-required = 0.  
    ENDCASE.  
    MODIFY SCREEN.  
  ENDLOOP.  
ENDFORM.                    " DISABLE\_SCREEN\_FIELDS  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  GET\_PARKED\_DOCUMENT\_DATA  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*  -->  p1        text  
\*  <--  p2        text  
\*----------------------------------------------------------------------\*  
FORM get\_parked\_document\_data.  
  TYPES: BEGIN OF lty\_vbkpf,  
           bukrs TYPE vbkpf-bukrs,  
           blart TYPE vbkpf-blart,  
           bldat TYPE vbkpf-bldat,  
           budat TYPE vbkpf-budat,  
           waers TYPE vbkpf-waers,  
           bktxt TYPE vbkpf-bktxt,  
           xblnr TYPE vbkpf-xblnr,  
         END OF lty\_vbkpf.  
  
  TYPES: BEGIN OF lty\_vbsegs,  
           bzkey      TYPE vbsegs-bzkey,  
           bschl      TYPE vbsegs-bschl,  
           gsber      TYPE vbsegs-gsber,  
           wrbtr      TYPE vbsegs-wrbtr,  
           kostl      TYPE vbsegs-kostl,  
           aufnr      TYPE vbsegs-aufnr,  
           saknr      TYPE vbsegs-saknr,  
           grant\_nbr  TYPE vbseg-grant\_nbr,  
           geber      TYPE vbsegs-geber,  
           ps\_psp\_pnr TYPE vbsegs-ps\_psp\_pnr,  
           sgtxt      TYPE vbsegs-sgtxt,  
           "Additional Fields  
           fkber      TYPE vbsegs-fkber,  
           fistl      TYPE vbsegs-fistl,  
           measure    TYPE vbsegs-measure,  
           budget\_pd  TYPE vbsegs-budget\_pd,  
           zuonr      TYPE vbsegs-zuonr,  
         END OF lty\_vbsegs.  
\*  
  TYPES: BEGIN OF lty\_vbsegd,  
           bzkey     TYPE vbsegd-bzkey,  
           bschl     TYPE vbsegd-bschl,  
           umskz     TYPE vbsegd-umskz,  
           gsber     TYPE vbsegd-gsber,  
           wrbtr     TYPE vbsegd-wrbtr,  
           kunnr     TYPE vbsegd-kunnr,  
           rebzg     TYPE vbsegd-rebzg,  
           grant\_nbr TYPE vbseg-grant\_nbr,  
           xcpdd     TYPE vbsegd-xcpdd,  
           ausbk     TYPE vbsegd-ausbk,  
           buzei     TYPE vbsegd-buzei,  
           geber     TYPE vbsegd-geber,  
           sgtxt     TYPE vbsegd-sgtxt,  
         END OF lty\_vbsegd.  
  
  TYPES: BEGIN OF lty\_vbsegk,  
           bzkey     TYPE vbsegk-bzkey,  
           bschl     TYPE vbsegk-bschl,  
           umskz     TYPE vbsegk-umskz,  
           gsber     TYPE vbsegk-gsber,  
           wrbtr     TYPE vbsegk-wrbtr,  
           lifnr     TYPE vbsegk-lifnr,  
           rebzg     TYPE vbsegk-rebzg,  
           grant\_nbr TYPE vbseg-grant\_nbr,  
           xcpdd     TYPE vbsegk-xcpdd,  
           ausbk     TYPE vbsegk-ausbk,  
           buzei     TYPE vbsegk-buzei,  
         END OF lty\_vbsegk.  
  
  
  DATA: ls\_vbkpf       TYPE lty\_vbkpf,  
        lt\_vbsegs      TYPE STANDARD TABLE OF lty\_vbsegs,  
        ls\_vbsegs      LIKE LINE OF lt\_vbsegs,  
        lt\_vbsegd      TYPE STANDARD TABLE OF lty\_vbsegd,  
        ls\_vbsegd      LIKE LINE OF lt\_vbsegd,  
        lt\_vbsegk      TYPE STANDARD TABLE OF lty\_vbsegk,  
        ls\_vbsegk      LIKE LINE OF lt\_vbsegk,  
        lt\_outtab\_line TYPE STANDARD TABLE OF lty\_outtab\_line,  
        ls\_outtab\_line LIKE LINE OF lt\_outtab\_line,  
        ls\_outtab      LIKE LINE OF gt\_outtab.  
  
  SELECT SINGLE bukrs  
                blart  
                bldat  
                budat  
                waers  
                bktxt  
                xblnr  
    FROM vbkpf  
    INTO ls\_vbkpf  
   WHERE belnr = 0100\_i\_document\_number  
     AND bukrs = 0100\_i\_company\_code  
     AND gjahr = 0100\_i\_fiscal\_year  
     AND blart = 0100\_i\_document\_type.  "'Z1'.  
  IF sy-subrc = 0.  
    0200\_i\_doc\_date      = ls\_vbkpf-bldat.  
    0200\_i\_posting\_date  = ls\_vbkpf-budat.  
    0200\_i\_document\_type = ls\_vbkpf-blart.  
    0200\_i\_company\_code  = ls\_vbkpf-bukrs.  
    0200\_i\_currency      = ls\_vbkpf-waers.  
    0200\_i\_header\_text   = ls\_vbkpf-bktxt.  
    0200\_i\_reference   = ls\_vbkpf-xblnr.  
  
    SELECT bzkey  
           bschl  
           gsber  
           wrbtr  
           kostl  
           aufnr  
           saknr  
           grant\_nbr  
           geber  
           ps\_psp\_pnr  
           sgtxt  
           fkber  
           fistl  
           measure  
           budget\_pd  
           zuonr  
      FROM vbsegs  
      INTO TABLE lt\_vbsegs  
     WHERE belnr = 0100\_i\_document\_number  
       AND bukrs = 0100\_i\_company\_code  
       AND gjahr = 0100\_i\_fiscal\_year.  
  
    IF sy-subrc = 0.  
      LOOP AT lt\_vbsegs INTO ls\_vbsegs.  
        CLEAR ls\_outtab\_line.  
        ls\_outtab\_line-bzkey = ls\_vbsegs-bzkey.  
        ls\_outtab\_line-posting\_key = ls\_vbsegs-bschl.  
        ls\_outtab\_line-business\_area = ls\_vbsegs-gsber.  
        ls\_outtab\_line-amount        = ls\_vbsegs-wrbtr.  
        ls\_outtab\_line-cost\_center   = ls\_vbsegs-kostl.  
        ls\_outtab\_line-internal\_order = ls\_vbsegs-aufnr.  
        ls\_outtab\_line-gl\_account     = ls\_vbsegs-saknr.  
        ls\_outtab\_line-grant\_nbr      = ls\_vbsegs-grant\_nbr.  
        ls\_outtab\_line-fund           = ls\_vbsegs-geber.  
        "Additional Fields  
        ls\_outtab\_line-fkber          = ls\_vbsegs-fkber.  
        ls\_outtab\_line-fistl           = ls\_vbsegs-fistl.  
        ls\_outtab\_line-measure           = ls\_vbsegs-measure.  
        ls\_outtab\_line-budget\_pd           = ls\_vbsegs-budget\_pd.  
        ls\_outtab\_line-zuonr           = ls\_vbsegs-zuonr.  
        ls\_outtab\_line-sgtxt   = ls\_vbsegs-sgtxt.  
        IF ls\_vbsegs-ps\_psp\_pnr NE '00000000'.  
          ls\_outtab\_line-wbs            = ls\_vbsegs-ps\_psp\_pnr.  
        ENDIF.  
  
  
\*        ls\_outtab\_line-receipt\_key = ls\_vbsegs-sgtxt(4).  
\*        ls\_outtab\_line-receipt\_key\_desc = ls\_vbsegs-sgtxt+4.  
  
        APPEND ls\_outtab\_line TO lt\_outtab\_line.  
      ENDLOOP.  
    ENDIF.  
  
    SELECT bzkey  
           bschl  
           umskz  
           gsber  
           wrbtr  
           kunnr  
           rebzg  
           grant\_nbr  
           xcpdd  
           ausbk  
           buzei  
           geber  
           sgtxt  
      FROM vbsegd  
      APPENDING TABLE lt\_vbsegd  
     WHERE belnr = 0100\_i\_document\_number  
       AND bukrs = 0100\_i\_company\_code  
       AND gjahr = 0100\_i\_fiscal\_year.  
    IF sy-subrc = 0.  
      LOOP AT lt\_vbsegd INTO ls\_vbsegd.  
        CLEAR ls\_outtab\_line.  
        ls\_outtab\_line-bzkey = ls\_vbsegd-bzkey.  
        ls\_outtab\_line-posting\_key = ls\_vbsegd-bschl.  
        ls\_outtab\_line-special\_gl\_ind = ls\_vbsegd-umskz.  
        ls\_outtab\_line-business\_area  = ls\_vbsegd-gsber.  
        ls\_outtab\_line-amount         = ls\_vbsegd-wrbtr.  
        ls\_outtab\_line-customer       = ls\_vbsegd-kunnr.  
        ls\_outtab\_line-ref\_invoice    = ls\_vbsegd-rebzg.  
        ls\_outtab\_line-grant\_nbr      = ls\_vbsegd-grant\_nbr.  
        ls\_outtab\_line-fund           = ls\_vbsegd-geber.  
        ls\_outtab\_line-sgtxt           = ls\_vbsegd-sgtxt.  
  
        IF ls\_vbsegd-xcpdd IS NOT INITIAL.  
          PERFORM read\_one\_time\_info USING ls\_vbsegd-ausbk  
                                           0100\_i\_document\_number  
                                           0100\_i\_fiscal\_year  
                                           ls\_vbsegd-buzei  
                                     CHANGING ls\_outtab\_line.  
        ENDIF.  
  
        APPEND ls\_outtab\_line TO lt\_outtab\_line.  
      ENDLOOP.  
    ENDIF.  
  
    SELECT bzkey  
           bschl  
           umskz  
           gsber  
           wrbtr  
           lifnr  
           rebzg  
           grant\_nbr  
           xcpdd  
           ausbk  
           buzei  
      FROM vbsegk  
      INTO TABLE lt\_vbsegk  
     WHERE belnr = 0100\_i\_document\_number  
       AND bukrs = 0100\_i\_company\_code  
       AND gjahr = 0100\_i\_fiscal\_year.  
  
    IF sy-subrc = 0.  
      LOOP AT lt\_vbsegk INTO ls\_vbsegk.  
        CLEAR ls\_outtab\_line.  
        ls\_outtab\_line-bzkey = ls\_vbsegk-bzkey.  
        ls\_outtab\_line-posting\_key = ls\_vbsegk-bschl.  
        ls\_outtab\_line-special\_gl\_ind = ls\_vbsegk-umskz.  
        ls\_outtab\_line-business\_area  = ls\_vbsegk-gsber.  
        ls\_outtab\_line-amount         = ls\_vbsegk-wrbtr.  
        ls\_outtab\_line-vendor       = ls\_vbsegk-lifnr.  
        ls\_outtab\_line-ref\_invoice    = ls\_vbsegk-rebzg.  
        ls\_outtab\_line-grant\_nbr      = ls\_vbsegk-grant\_nbr.  
  
        IF ls\_vbsegk-xcpdd IS NOT INITIAL.  
          PERFORM read\_one\_time\_info USING ls\_vbsegk-ausbk  
                                           0100\_i\_document\_number  
                                           0100\_i\_fiscal\_year  
                                           ls\_vbsegk-buzei  
                                     CHANGING ls\_outtab\_line.  
        ENDIF.  
  
        APPEND ls\_outtab\_line TO lt\_outtab\_line.  
      ENDLOOP.  
    ENDIF.  
  
\*Build data table for output  
    SORT lt\_outtab\_line BY bzkey.  
  
    LOOP AT lt\_outtab\_line INTO ls\_outtab\_line.  
      CLEAR ls\_outtab.  
      MOVE-CORRESPONDING ls\_outtab\_line TO ls\_outtab.  
      APPEND ls\_outtab TO gt\_outtab.  
    ENDLOOP.  
  
  ELSE.  
\* Document does not exist  
    MESSAGE s006(zfi\_cj) WITH 0100\_i\_document\_number DISPLAY LIKE 'E'.  
    CALL SCREEN 100.  
  ENDIF.  
ENDFORM.                    " GET\_PARKED\_DOCUMENT\_DATA  
\*&---------------------------------------------------------------------\*  
\*&      Form  SELECT\_DATA  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*  -->  p1        text  
\*  <--  p2        text  
\*----------------------------------------------------------------------\*  
FORM select\_data .  
  DATA: lt\_zfi\_cr\_data      TYPE STANDARD TABLE OF zfit\_cr\_data,  
        ls\_zfi\_cr\_data      LIKE LINE OF lt\_zfi\_cr\_data,  
        ls\_outtab           LIKE LINE OF gt\_outtab,  
        lt\_zfi\_cr\_line\_stat TYPE STANDARD TABLE OF zfit\_crline\_stat,  
        ls\_zfi\_cr\_line\_stat LIKE LINE OF lt\_zfi\_cr\_line\_stat,  
        ls\_celltab          TYPE lvc\_s\_styl.  
  
  REFRESH: lt\_zfi\_cr\_data,  
           lt\_zfi\_cr\_line\_stat.  
  
  CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'  
    EXPORTING  
      input  = 0100\_i\_document\_number  
    IMPORTING  
      output = 0100\_i\_document\_number.  
  
  SELECT \*  
    FROM zfit\_cr\_data  
    INTO TABLE lt\_zfi\_cr\_data  
   WHERE document\_number = 0100\_i\_document\_number  
     AND company\_code    = 0100\_i\_company\_code  
     AND fiscal\_year     = 0100\_i\_fiscal\_year.  
  IF sy-subrc = 0.  
    SELECT \*  
      FROM zfit\_crline\_stat  
      INTO TABLE lt\_zfi\_cr\_line\_stat  
     WHERE document\_number = 0100\_i\_document\_number  
       AND company\_code    = 0100\_i\_company\_code  
       AND fiscal\_year     = 0100\_i\_fiscal\_year.  
    LOOP AT lt\_zfi\_cr\_data INTO ls\_zfi\_cr\_data.  
      CLEAR ls\_outtab.  
      IF sy-tabix = 1.  
        0200\_i\_doc\_date = ls\_zfi\_cr\_data-document\_date.  
        0200\_i\_posting\_date  = ls\_zfi\_cr\_data-posting\_date.  
        0200\_i\_document\_type = ls\_zfi\_cr\_data-document\_type.  
        0200\_i\_company\_code  = ls\_zfi\_cr\_data-company\_code.  
        0200\_i\_currency      = ls\_zfi\_cr\_data-currency.  
        0200\_i\_header\_text     = ls\_zfi\_cr\_data-header\_text.  
        0200\_i\_reference     = ls\_zfi\_cr\_data-reference.  
      ENDIF.  
      MOVE-CORRESPONDING ls\_zfi\_cr\_data TO ls\_outtab.  
      LOOP AT lt\_zfi\_cr\_line\_stat INTO ls\_zfi\_cr\_line\_stat WHERE item = ls\_zfi\_cr\_data-item.  
        MOVE-CORRESPONDING ls\_zfi\_cr\_line\_stat TO ls\_celltab.  
        APPEND ls\_celltab TO ls\_outtab-celltab.  
      ENDLOOP.  
      APPEND ls\_outtab TO gt\_outtab.  
    ENDLOOP.  
  ELSE.  
\* Document does not exist  
    MESSAGE s006(zfi\_cj) WITH 0100\_i\_document\_number DISPLAY LIKE 'E'.  
    CALL SCREEN 100.  
  ENDIF.  
ENDFORM.                    " SELECT\_DATA  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  SAVE\_DATA  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*  -->  p1        text  
\*  <--  p2        text  
\*----------------------------------------------------------------------\*  
FORM save\_data .  
  DATA: l\_valid             TYPE c,  
        ls\_outtab           LIKE LINE OF gt\_outtab,  
        ls\_celltab          TYPE lvc\_s\_styl,  
        lt\_zfi\_cr\_data      TYPE STANDARD TABLE OF zfit\_cr\_data,  
        ls\_zfi\_cr\_data      LIKE LINE OF lt\_zfi\_cr\_data,  
        lt\_zfi\_cr\_line\_stat TYPE STANDARD TABLE OF zfit\_crline\_stat,  
        ls\_zfi\_cr\_line\_stat LIKE LINE OF lt\_zfi\_cr\_line\_stat.  
  
  DATA: lv\_document\_number TYPE belnr\_d,  
        lv\_item            TYPE buzei,  
        lv\_fiscal\_year     TYPE bapi0002\_4-fiscal\_year.  
  
  CALL METHOD go\_grid->check\_changed\_data  
    IMPORTING  
      e\_valid = l\_valid.  
  
  IF l\_valid IS INITIAL.  
    CALL FUNCTION 'POPUP\_TO\_INFORM'  
      EXPORTING  
        titel = TEXT-i06  
        txt1  = TEXT-i07  
        txt2  = TEXT-i08  
        txt3  = TEXT-i09.  
  ELSE.  
    IF 0100\_i\_document\_number IS INITIAL.  
      CALL FUNCTION 'NUMBER\_RANGE\_ENQUEUE'  
        EXPORTING  
          object           = 'ZFI\_CR'  
        EXCEPTIONS  
          foreign\_lock     = 1  
          object\_not\_found = 2  
          system\_failure   = 3  
          OTHERS           = 4.  
      IF sy-subrc = 0.  
        CALL FUNCTION 'NUMBER\_GET\_NEXT'  
          EXPORTING  
            nr\_range\_nr             = '01'  
            object                  = 'ZFI\_CR'  
\*           QUANTITY                = '1'  
\*           SUBOBJECT               = ' '  
\*           TOYEAR                  = '0000'  
\*           IGNORE\_BUFFER           = ' '  
          IMPORTING  
            number                  = lv\_document\_number  
\*           QUANTITY                =  
\*           RETURNCODE              =  
          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.  
        IF sy-subrc <> 0.  
\* Implement suitable error handling here  
        ENDIF.  
  
        CALL FUNCTION 'NUMBER\_RANGE\_DEQUEUE'  
          EXPORTING  
            object           = 'ZFI\_CR'  
          EXCEPTIONS  
            object\_not\_found = 1  
            OTHERS           = 2.  
  
        IF sy-subrc <> 0.  
  
        ENDIF.  
  
      ENDIF.  
    ELSE.  
      lv\_document\_number = 0100\_i\_document\_number.  
    ENDIF.  
  
\* Get Fiscal Year.  
    CALL FUNCTION 'BAPI\_COMPANYCODE\_GET\_PERIOD'  
      EXPORTING  
        companycodeid = 0200\_i\_company\_code  
        posting\_date  = 0200\_i\_posting\_date  
      IMPORTING  
        fiscal\_year   = lv\_fiscal\_year  
\*       FISCAL\_PERIOD =  
\*       RETURN        =  
      .  
  
\* Delete lines that do not have a receipt key  
    DELETE gt\_outtab WHERE receipt\_key IS INITIAL.  
  
    LOOP AT gt\_outtab INTO ls\_outtab.  
      CLEAR ls\_zfi\_cr\_data.  
      ADD 1 TO lv\_item.  
\* Set header data  
      ls\_zfi\_cr\_data-document\_number = lv\_document\_number.  
      ls\_zfi\_cr\_data-company\_code    = 0200\_i\_company\_code.  
      ls\_zfi\_cr\_data-fiscal\_year     = lv\_fiscal\_year.  
      ls\_zfi\_cr\_data-item            = lv\_item.  
      ls\_zfi\_cr\_data-posting\_date    = 0200\_i\_posting\_date.  
      ls\_zfi\_cr\_data-document\_date   = 0200\_i\_doc\_date.  
      ls\_zfi\_cr\_data-document\_type   = 0200\_i\_document\_type.  
      ls\_zfi\_cr\_data-currency        = 0200\_i\_currency.  
      ls\_zfi\_cr\_data-header\_text     = 0200\_i\_header\_text.  
      ls\_zfi\_cr\_data-reference     = 0200\_i\_reference.  
  
\* Set Item Data  
      MOVE-CORRESPONDING ls\_outtab TO ls\_zfi\_cr\_data.  
  
      LOOP AT ls\_outtab-celltab INTO ls\_celltab.  
        ls\_zfi\_cr\_line\_stat-document\_number = lv\_document\_number.  
        ls\_zfi\_cr\_line\_stat-company\_code    = 0200\_i\_company\_code.  
        ls\_zfi\_cr\_line\_stat-fiscal\_year     = lv\_fiscal\_year.  
        ls\_zfi\_cr\_line\_stat-item            = lv\_item.  
  
        MOVE-CORRESPONDING ls\_celltab TO ls\_zfi\_cr\_line\_stat.  
        APPEND ls\_zfi\_cr\_line\_stat TO lt\_zfi\_cr\_line\_stat.  
      ENDLOOP.  
  
      APPEND ls\_zfi\_cr\_data TO lt\_zfi\_cr\_data.  
    ENDLOOP.  
    IF sy-subrc = 0.  
      PERFORM delete\_saved\_record.  
  
      INSERT zfit\_cr\_data FROM TABLE lt\_zfi\_cr\_data.  
      IF sy-subrc = 0.  
        INSERT zfit\_crline\_stat FROM TABLE lt\_zfi\_cr\_line\_stat.  
        IF sy-subrc = 0.  
          MESSAGE i001(zfi\_cj) WITH lv\_document\_number DISPLAY LIKE 'S'.  
  
\* Store saved document key information  
          0100\_i\_document\_number = lv\_document\_number.  
          0100\_i\_company\_code    = 0200\_i\_company\_code.  
          0100\_i\_fiscal\_year     = lv\_fiscal\_year.  
  
\*        PERFORM unlock\_record.  
\*        PERFORM refresh\_global\_data.  
\*        CALL SCREEN 100.  
        ENDIF.  
      ENDIF.  
    ELSE.  
      MESSAGE i007(zfi\_cj) DISPLAY LIKE 'E'.  
    ENDIF.  
\* BEGIN OF ADDITION ED1K903487  
\* Refresh the table to take into account empty lines that have been deleted  
    go\_grid->refresh\_table\_display( ).  
\* END OF ADDITION ED1K903487  
  ENDIF.  
ENDFORM.                    " SAVE\_DATA  
\*&---------------------------------------------------------------------\*  
\*&      Form  LOCK\_RECORD  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*  -->  p1        text  
\*  <--  p2        text  
\*----------------------------------------------------------------------\*  
FORM lock\_record .  
  
  CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'  
    EXPORTING  
      input  = 0100\_i\_document\_number  
    IMPORTING  
      output = 0100\_i\_document\_number.  
  
  CALL FUNCTION 'ENQUEUE\_EZ\_CR'  
    EXPORTING  
\*     mode\_zfi\_cr\_data  = 'X'  
      mode\_zfit\_cr\_data = 'X'  
\*     MANDT             = SY-MANDT  
      document\_number   = 0100\_i\_document\_number  
      company\_code      = 0100\_i\_company\_code  
      fiscal\_year       = 0100\_i\_fiscal\_year  
\*     ITEM              =  
\*     X\_DOCUMENT\_NUMBER = ' '  
\*     X\_COMPANY\_CODE    = ' '  
\*     X\_FISCAL\_YEAR     = ' '  
\*     X\_ITEM            = ' '  
\*     \_SCOPE            = '2'  
\*     \_WAIT             = ' '  
\*     \_COLLECT          = ' '  
    EXCEPTIONS  
      foreign\_lock      = 1  
      system\_failure    = 2  
      OTHERS            = 3.  
  IF sy-subrc <> 0.  
    MESSAGE i002(zfi\_cj) DISPLAY LIKE 'E'.  
    CALL SCREEN 100.  
  ENDIF.  
  
ENDFORM.                    " LOCK\_RECORD  
\*&---------------------------------------------------------------------\*  
\*&      Form  UNLOCK\_RECORD  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*  -->  p1        text  
\*  <--  p2        text  
\*----------------------------------------------------------------------\*  
FORM unlock\_record .  
  CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'  
    EXPORTING  
      input  = 0100\_i\_document\_number  
    IMPORTING  
      output = 0100\_i\_document\_number.  
  
  CALL FUNCTION 'DEQUEUE\_EZ\_CR'  
    EXPORTING  
      mode\_zfit\_cr\_data = 'X'  
\*     MANDT             = SY-MANDT  
      document\_number   = 0100\_i\_document\_number  
      company\_code      = 0100\_i\_company\_code  
      fiscal\_year       = 0100\_i\_fiscal\_year  
\*     ITEM              =  
\*     X\_DOCUMENT\_NUMBER = ' '  
\*     X\_COMPANY\_CODE    = ' '  
\*     X\_FISCAL\_YEAR     = ' '  
\*     X\_ITEM            = ' '  
\*     \_SCOPE            = '3'  
\*     \_SYNCHRON         = ' '  
\*     \_COLLECT          = ' '  
    .  
  
ENDFORM.                    " UNLOCK\_RECORD  
\*&---------------------------------------------------------------------\*  
\*&      Form  INITIALIZE\_ALL\_DATA  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*  -->  p1        text  
\*  <--  p2        text  
\*----------------------------------------------------------------------\*  
FORM initialize\_all\_data .  
  CLEAR: gv\_edit,  
         gv\_create,  
         gv\_display.  
ENDFORM.                    " INITIALIZE\_ALL\_DATA  
\*&---------------------------------------------------------------------\*  
\*&      Form  REFRESH\_GLOBAL\_DATA  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*  -->  p1        text  
\*  <--  p2        text  
\*----------------------------------------------------------------------\*  
FORM refresh\_global\_data .  
\* Tables  
  FREE gt\_outtab.  
  
\* Objects  
  go\_grid->free( ).  
  go\_custom\_container->free( ).  
  cl\_gui\_cfw=>flush( ).  
  
  FREE: go\_verifier,  
        go\_onf4,  
        go\_main,  
        go\_gos\_manager,  
        go\_grid,  
        go\_custom\_container.  
  
\* Screen Variables  
  CLEAR: 0100\_i\_document\_number,  
         0100\_i\_company\_code,  
         0100\_i\_fiscal\_year,  
         0200\_i\_doc\_date,  
         0200\_i\_posting\_date,  
         0200\_i\_document\_type,  
         0200\_i\_company\_code,  
         0200\_i\_currency,  
         0200\_i\_header\_text,  
         0200\_i\_reference.  
  
  lcl\_event\_receiver=>clear\_rec\_buffer( ).  
  
ENDFORM.                    " REFRESH\_GLOBAL\_DATA  
\*&---------------------------------------------------------------------\*  
\*&      Form  DELETE\_SAVED\_RECORD  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*  -->  p1        text  
\*  <--  p2        text  
\*----------------------------------------------------------------------\*  
FORM delete\_saved\_record .  
  IF gv\_edit = abap\_true OR gv\_create = abap\_true.  
\* Delete saved line item data  
    DELETE FROM zfit\_cr\_data  
          WHERE document\_number = 0100\_i\_document\_number  
            AND company\_code    = 0100\_i\_company\_code  
            AND fiscal\_year     = 0100\_i\_fiscal\_year.  
  
\* Delete saved line line item status  
    DELETE FROM zfit\_crline\_stat  
          WHERE document\_number = 0100\_i\_document\_number  
            AND company\_code    = 0100\_i\_company\_code  
            AND fiscal\_year     = 0100\_i\_fiscal\_year.  
  ENDIF.  
ENDFORM.                    " DELETE\_SAVED\_RECORD  
\*&---------------------------------------------------------------------\*  
\*&      Form  START\_WF  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*      -->P\_LS\_MESSTAB\_MSGV1  text  
\*----------------------------------------------------------------------\*  
FORM start\_wf  USING iv\_company\_code  
                     iv\_document\_number  
                     iv\_fiscal\_year.  
  INCLUDE <cntn01>.  
  
  DATA: lo\_fipp TYPE swc\_object.  
  DATA: lv\_key(18) TYPE c..  
  
  
  CONCATENATE iv\_company\_code  
              iv\_document\_number  
              iv\_fiscal\_year  
         INTO lv\_key.  
  
  swc\_container l\_container.  
  swc\_create\_object lo\_fipp 'FIPP' lv\_key.  
  swc\_raise\_event lo\_fipp 'ZTriggerParkedDocWF' l\_container.  
  COMMIT WORK.  
  MESSAGE i008(zfi\_cj) DISPLAY LIKE 'S'.  
ENDFORM.                    " START\_WF  
\*&---------------------------------------------------------------------\*  
\*&      Form  DETERMINE\_WF\_STATUS  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*      <--P\_LV\_WF\_STARTED  text  
\*----------------------------------------------------------------------\*  
FORM determine\_wf\_status  CHANGING iv\_wf\_started.  
  DATA: lv\_instid TYPE sww\_wi2obj-instid,  
        lv\_objkey TYPE sww\_wi2obj-instid.  
  
  CONSTANTS: lc\_type TYPE swwwihead-wi\_type VALUE 'F',  
             lc\_task TYPE swwwihead-wi\_rh\_task VALUE 'WS90000011'.  
  
  CLEAR iv\_wf\_started.  
  
  CONCATENATE 0100\_i\_company\_code  
              0100\_i\_document\_number  
              0100\_i\_fiscal\_year  
         INTO lv\_objkey.  
  
  SELECT SINGLE sww\_wi2obj~instid  
    FROM swwwihead  
   INNER JOIN sww\_wi2obj  
      ON swwwihead~wi\_id = sww\_wi2obj~wi\_id  
    INTO lv\_instid  
   WHERE swwwihead~wi\_type = lc\_type  
     AND swwwihead~wi\_rh\_task = lc\_task  
     AND sww\_wi2obj~instid = lv\_objkey.  
  IF sy-subrc = 0.  
    iv\_wf\_started = abap\_true.  
  ENDIF.  
ENDFORM.                    " DETERMINE\_WF\_STATUS  
\*&---------------------------------------------------------------------\*  
\*&      Form  READ\_ONE\_TIME\_INFO  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*      -->P\_AUSBK  text  
\*      -->P\_BELNR  text  
\*      -->P\_GJAHR  text  
\*      -->P\_BUZEI  text  
\*      <--P\_LS\_OUTTAB  text  
\*----------------------------------------------------------------------\*  
FORM read\_one\_time\_info  USING    iv\_ausbk TYPE ausbk  
                                  iv\_belnr TYPE belnr\_d  
                                  iv\_gjahr TYPE gjahr  
                                  iv\_buzei TYPE buzei  
                         CHANGING is\_outtab TYPE lty\_outtab\_line.  
  
  TYPES: BEGIN OF lty\_vbsec,  
           name1 TYPE vbsec-name1,  
           name2 TYPE vbsec-name2,  
           ort01 TYPE vbsec-ort01,  
           pstlz TYPE vbsec-pstlz,  
           stras TYPE vbsec-stras,  
           land1 TYPE vbsec-land1,  
           regio TYPE vbsec-regio,  
         END OF lty\_vbsec.  
  
  DATA: ls\_vbsec TYPE lty\_vbsec.  
  
  SELECT SINGLE name1 name2 ort01 pstlz stras land1 regio  
   INTO ls\_vbsec  
   FROM vbsec  
   WHERE ausbk = iv\_ausbk  
     AND belnr = iv\_belnr  
     AND gjahr = iv\_gjahr  
     AND buzei = iv\_buzei.  
  
  IF sy-subrc = 0.  
    is\_outtab-name1 = ls\_vbsec-name1.  
    is\_outtab-name2 = ls\_vbsec-name2.  
    is\_outtab-street = ls\_vbsec-stras.  
    is\_outtab-city = ls\_vbsec-ort01.  
    is\_outtab-postal\_code = ls\_vbsec-pstlz.  
    is\_outtab-region = ls\_vbsec-regio.  
    is\_outtab-country = ls\_vbsec-land1.  
  ENDIF.  
  
ENDFORM.                    " READ\_ONE\_TIME\_INFO