\*&---------------------------------------------------------------------\*  
\*& Include          ZFIR\_CASH\_JOURNAL\_SIMPLE\_I01  
\*&---------------------------------------------------------------------\*  
  
  DATA: lt\_return\_tab      TYPE STANDARD TABLE OF ddshretval,  
        ls\_return\_tab      LIKE LINE OF lt\_return\_tab,  
        lt\_field\_tab       TYPE STANDARD TABLE OF dfies,  
        lt\_dynpfld\_mapping TYPE STANDARD TABLE OF dselc,  
        ls\_dynpfld\_mapping LIKE LINE OF lt\_dynpfld\_mapping.  
  
  DATA: lt\_dynpfields LIKE STANDARD TABLE OF dynpread,  
        ls\_dynpfields TYPE dynpread.  
  
  DATA: lv\_dynpprog TYPE sy-repid,  
        lv\_dynpnr   TYPE sy-dynnr,  
        lv\_answer.  
\*&---------------------------------------------------------------------\*  
\*&      Module  PAI\_0200  INPUT  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
  MODULE pai\_0200 INPUT.  
  
    ok\_code = sy-ucomm.  
  
    CASE ok\_code.  
      WHEN 'PARK'.  
        PERFORM park\_data.  
      WHEN 'SAVE'.  
        PERFORM save\_data.  
      WHEN 'ATTACH'.  
        CALL METHOD go\_gos\_manager->start\_service\_direct  
          EXPORTING  
            ip\_service       = 'PCATTA\_CREA'  
            is\_object        = ls\_object  
          EXCEPTIONS  
            no\_object        = 1  
            object\_invalid   = 2  
            execution\_failed = 3  
            OTHERS           = 4.  
  
      WHEN 'LIST'.  
        CALL METHOD go\_gos\_manager->start\_service\_direct  
          EXPORTING  
            ip\_service       = 'VIEW\_ATTA'  
            is\_object        = ls\_object  
          EXCEPTIONS  
            no\_object        = 1  
            object\_invalid   = 2  
            execution\_failed = 3  
            OTHERS           = 4.  
      WHEN 'START\_WF'.  
        PERFORM start\_wf USING 0100\_i\_company\_code  
                               0100\_i\_document\_number  
                               0100\_i\_fiscal\_year.  
        PERFORM unlock\_record.  
        PERFORM refresh\_global\_data.  
        SET SCREEN 0.  
        LEAVE SCREEN.  
      WHEN 'BACK'.  
        IF gv\_edit = abap\_true OR gv\_create = abap\_true.  
          CALL FUNCTION 'POPUP\_TO\_CONFIRM'  
            EXPORTING  
              text\_question = TEXT-001  
            IMPORTING  
              answer        = lv\_answer.  
          IF lv\_answer = '1'. " Yes  
            PERFORM save\_data.  
            PERFORM unlock\_record.  
            PERFORM refresh\_global\_data.  
            SET SCREEN 0.  
            LEAVE SCREEN.  
          ELSEIF lv\_answer = '2'. "No  
            PERFORM unlock\_record.  
            PERFORM refresh\_global\_data.  
            SET SCREEN 0.  
            LEAVE SCREEN.  
          ELSE. "Cancel  
  
          ENDIF.  
        ELSEIF gv\_display = abap\_true.  
          PERFORM unlock\_record.  
          PERFORM refresh\_global\_data.  
          SET SCREEN 0.  
          LEAVE SCREEN.  
        ENDIF.  
      WHEN 'EXIT'.  
        IF gv\_edit = abap\_true OR gv\_create = abap\_true.  
          CALL FUNCTION 'POPUP\_TO\_CONFIRM'  
            EXPORTING  
              text\_question = TEXT-001  
            IMPORTING  
              answer        = lv\_answer.  
          IF lv\_answer = '1'. " Yes  
            PERFORM save\_data.  
            PERFORM unlock\_record.  
            PERFORM refresh\_global\_data.  
            LEAVE PROGRAM.  
          ELSEIF lv\_answer = '2'. "No  
            PERFORM unlock\_record.  
            PERFORM refresh\_global\_data.  
            LEAVE PROGRAM.  
          ELSE. "Cancel  
  
          ENDIF.  
        ELSEIF gv\_display = abap\_true.  
          PERFORM unlock\_record.  
          PERFORM refresh\_global\_data.  
          LEAVE PROGRAM.  
        ENDIF.  
      WHEN OTHERS.  
  
    ENDCASE.  
  ENDMODULE.                 " PAI\_0200  INPUT  
  
\*&---------------------------------------------------------------------\*  
\*&      Module  USER\_COMMAND\_0200  INPUT  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
  MODULE user\_command\_0200 INPUT.  
    PERFORM unlock\_record.  
    PERFORM refresh\_global\_data.  
    SET SCREEN 0.  
    LEAVE SCREEN.  
  ENDMODULE.                 " USER\_COMMAND\_0200  INPUT  
\*&---------------------------------------------------------------------\*  
\*&      Module  USER\_COMMAND\_0100  INPUT  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
  MODULE user\_command\_0100 INPUT.  
    ok\_code = sy-ucomm.  
    CASE ok\_code.  
      WHEN 'DISPLAY'.  
        gv\_display = abap\_true.  
        IF 0100\_i\_document\_number IS INITIAL.  
          MESSAGE s003(zfi\_cj) DISPLAY LIKE 'E'.  
          LEAVE TO SCREEN 100.  
        ELSEIF 0100\_i\_company\_code IS INITIAL .  
          MESSAGE s004(zfi\_cj) DISPLAY LIKE 'E'.  
          LEAVE TO SCREEN 100.  
        ELSEIF 0100\_i\_fiscal\_year IS INITIAL.  
          MESSAGE s005(zfi\_cj) DISPLAY LIKE 'E'.  
          LEAVE TO SCREEN 100.  
        ELSE.  
          PERFORM get\_parked\_document\_data.  
          CALL SCREEN 200.  
        ENDIF.  
      WHEN 'CREATE'.  
        gv\_create = abap\_true.  
        CALL SCREEN 200.  
      WHEN 'CHANGE'.  
        gv\_edit = abap\_true.  
  
        IF 0100\_i\_document\_number IS INITIAL.  
          MESSAGE s003(zfi\_cj) DISPLAY LIKE 'E'.  
          LEAVE TO SCREEN 100.  
        ELSEIF 0100\_i\_company\_code IS INITIAL .  
          MESSAGE s004(zfi\_cj) DISPLAY LIKE 'E'.  
          LEAVE TO SCREEN 100.  
        ELSEIF 0100\_i\_fiscal\_year IS INITIAL.  
          MESSAGE s005(zfi\_cj) DISPLAY LIKE 'E'.  
          LEAVE TO SCREEN 100.  
        ELSE.  
          gv\_edit = abap\_true.  
          PERFORM lock\_record.  
          PERFORM select\_data.  
          CALL SCREEN 200.  
        ENDIF.  
      WHEN 'BACK'.  
        LEAVE PROGRAM.  
      WHEN 'EXIT'.  
        LEAVE PROGRAM.  
    ENDCASE.  
  ENDMODULE.                 " USER\_COMMAND\_0100  INPUT  
\*&---------------------------------------------------------------------\*  
\*&      Module  F4\_DOC\_NUM  INPUT  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
  MODULE f4\_doc\_num INPUT.  
    TYPES: BEGIN OF lty\_value\_saved,  
             document\_number TYPE zfit\_cr\_data-document\_number,  
             company\_code    TYPE zfit\_cr\_data-company\_code,  
             fiscal\_year     TYPE zfit\_cr\_data-fiscal\_year,  
             header\_text     TYPE zfit\_cr\_data-header\_text,  
             reference     TYPE zfit\_cr\_data-reference,  
           END OF lty\_value\_saved.  
  
    TYPES: BEGIN OF lty\_value\_parked,  
             belnr TYPE vbkpf-belnr,  
             bukrs TYPE vbkpf-bukrs,  
             gjahr TYPE vbkpf-gjahr,  
             bktxt TYPE vbkpf-bktxt,  
             xblnr TYPE vbkpf-xblnr,  
           END OF lty\_value\_parked.  
  
    DATA: lt\_value\_saved  TYPE STANDARD TABLE OF lty\_value\_saved,  
          lt\_value\_parked TYPE STANDARD TABLE OF lty\_value\_parked.  
  
    REFRESH: lt\_value\_saved,  
             lt\_value\_parked,  
             lt\_return\_tab,  
             lt\_field\_tab,  
             lt\_dynpfld\_mapping,  
             lt\_dynpfields.  
  
    lv\_dynpprog = sy-repid.  
    lv\_dynpnr  = sy-dynnr.  
  
    ls\_dynpfld\_mapping-fldname   = 'F0001'.  
    ls\_dynpfld\_mapping-dyfldname = '0100\_I\_DOCUMENT\_NUMBER'.  
    APPEND ls\_dynpfld\_mapping TO lt\_dynpfld\_mapping.  
  
    ls\_dynpfld\_mapping-fldname   = 'F0002'.  
    ls\_dynpfld\_mapping-dyfldname = '0100\_I\_COMPANY\_CODE'.  
    APPEND ls\_dynpfld\_mapping TO lt\_dynpfld\_mapping.  
  
    ls\_dynpfld\_mapping-fldname   = 'F0003'.  
    ls\_dynpfld\_mapping-dyfldname = '0100\_I\_FISCAL\_YEAR'.  
    APPEND ls\_dynpfld\_mapping TO lt\_dynpfld\_mapping.  
  
    IF 0100\_rb\_saved = abap\_true.  
      SELECT document\_number  
             company\_code  
             fiscal\_year  
             header\_text  
             reference  
        FROM zfit\_cr\_data  
        INTO TABLE lt\_value\_saved.  
      IF sy-subrc = 0.  
        SORT lt\_value\_saved BY document\_number.  
        DELETE ADJACENT DUPLICATES FROM lt\_value\_saved.  
  
        CALL FUNCTION 'F4IF\_INT\_TABLE\_VALUE\_REQUEST'  
          EXPORTING  
            retfield        = 'DOCUMENT\_NUMBER'  
            value\_org       = 'S'  
          TABLES  
            value\_tab       = lt\_value\_saved  
            field\_tab       = lt\_field\_tab  
            return\_tab      = lt\_return\_tab  
            dynpfld\_mapping = lt\_dynpfld\_mapping  
          EXCEPTIONS  
            parameter\_error = 1  
            no\_values\_found = 2  
            OTHERS          = 3.  
      ENDIF.  
    ELSEIF 0100\_rb\_parked = abap\_true.  
      SELECT belnr  
             bukrs  
             gjahr  
             bktxt  
             xblnr  
        FROM vbkpf  
        INTO TABLE lt\_value\_parked  
      WHERE blart = 0100\_i\_document\_type. "'Z1'.  
  
      CALL FUNCTION 'F4IF\_INT\_TABLE\_VALUE\_REQUEST'  
        EXPORTING  
          retfield        = 'BELNR'  
          value\_org       = 'S'  
        TABLES  
          value\_tab       = lt\_value\_parked  
          field\_tab       = lt\_field\_tab  
          return\_tab      = lt\_return\_tab  
          dynpfld\_mapping = lt\_dynpfld\_mapping  
        EXCEPTIONS  
          parameter\_error = 1  
          no\_values\_found = 2  
          OTHERS          = 3.  
    ENDIF.  
  
    LOOP AT lt\_return\_tab INTO ls\_return\_tab.  
      CASE ls\_return\_tab-fieldname.  
        WHEN 'F0001'.  
          ls\_dynpfields-fieldname  = '0100\_I\_DOCUMENT\_NUMBER'.  
          ls\_dynpfields-fieldvalue = ls\_return\_tab-fieldval.  
        WHEN 'F0002'.  
          ls\_dynpfields-fieldname  = '0100\_I\_COMPANY\_CODE'.  
          ls\_dynpfields-fieldvalue = ls\_return\_tab-fieldval.  
        WHEN 'F0003'.  
          ls\_dynpfields-fieldname  = '0100\_I\_FISCAL\_YEAR'.  
          ls\_dynpfields-fieldvalue = ls\_return\_tab-fieldval.  
      ENDCASE.  
      APPEND ls\_dynpfields TO lt\_dynpfields.  
    ENDLOOP.  
  
  
    CALL FUNCTION 'DYNP\_VALUES\_UPDATE'  
      EXPORTING  
        dyname     = sy-repid  
        dynumb     = sy-dynnr  
      TABLES  
        dynpfields = lt\_dynpfields.  
  ENDMODULE.                 " F4\_DOC\_NUM  INPUT  
\*&---------------------------------------------------------------------\*  
\*&      Module  F4\_HEADER\_TEXT  INPUT  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
  MODULE f4\_header\_text INPUT.  
    TYPES: BEGIN OF lty\_value\_header\_text,  
             abtei    TYPE zfit\_abtei-abtei,  
             zzabteit TYPE zfit\_abtei-zzabteit,  
           END OF lty\_value\_header\_text.  
  
    DATA: lt\_value\_header\_text TYPE STANDARD TABLE OF lty\_value\_header\_text,  
          ls\_value\_header\_text LIKE LINE OF lt\_value\_header\_text.  
  
    REFRESH: lt\_value\_header\_text,  
             lt\_return\_tab,  
             lt\_field\_tab,  
             lt\_dynpfld\_mapping,  
             lt\_dynpfields.  
  
    lv\_dynpprog = sy-repid.  
    lv\_dynpnr  = sy-dynnr.  
  
    SELECT abtei  
           zzabteit  
      FROM zfit\_abtei  
      INTO TABLE lt\_value\_header\_text.  
  
    CALL FUNCTION 'F4IF\_INT\_TABLE\_VALUE\_REQUEST'  
      EXPORTING  
        retfield        = 'ABTEI'  
        value\_org       = 'S'  
        dynpprog        = lv\_dynpprog  
        dynpnr          = lv\_dynpnr  
        dynprofield     = '0200\_I\_HEADER\_TEXT'  
      TABLES  
        value\_tab       = lt\_value\_header\_text  
\*       field\_tab       = lt\_field\_tab  
        return\_tab      = lt\_return\_tab  
\*       dynpfld\_mapping = lt\_dynpfld\_mapping  
      EXCEPTIONS  
        parameter\_error = 1  
        no\_values\_found = 2  
        OTHERS          = 3.  
  
    READ TABLE lt\_return\_tab INTO ls\_return\_tab INDEX 1.  
    IF sy-subrc = 0.  
      READ TABLE lt\_value\_header\_text INTO ls\_value\_header\_text WITH KEY abtei = ls\_return\_tab-fieldval.  
      IF sy-subrc = 0.  
  
        ls\_dynpfields-fieldname  = '0200\_I\_HEADER\_TEXT'.  
  
        CONCATENATE ls\_value\_header\_text-abtei  
                    ls\_value\_header\_text-zzabteit  
               INTO ls\_dynpfields-fieldvalue SEPARATED BY space.  
  
        APPEND ls\_dynpfields TO lt\_dynpfields.  
      ENDIF.  
  
    ENDIF.  
  
    CALL FUNCTION 'DYNP\_VALUES\_UPDATE'  
      EXPORTING  
        dyname     = sy-repid  
        dynumb     = sy-dynnr  
      TABLES  
        dynpfields = lt\_dynpfields.  
  
  
  ENDMODULE.                 " F4\_HEADER\_TEXT  INPUT