\*&---------------------------------------------------------------------\*  
\*& Include          ZDYS\_I\_0001\_CLS  
\*&---------------------------------------------------------------------\*  
CLASS lcl\_main DEFINITION FINAL.  
  PUBLIC SECTION.  
    TYPES: BEGIN OF ty\_fname,  
             fname TYPE fieldname,  
           END OF ty\_fname.  
    TYPES: tty\_fname TYPE TABLE OF ty\_fname.  
  
    CONSTANTS: mc\_msgid      TYPE sy-msgid VALUE 'ZDYS1',  
               mc\_fc\_add\_row TYPE ui\_func VALUE 'ADD\_ROW',  
               mc\_fc\_del\_row TYPE ui\_func VALUE 'DEL\_ROW'.  
  
    DATA: ms\_t001   TYPE t001,  
          ms\_item   TYPE zdys\_s\_003,  
          mt\_detail TYPE zdys\_tt\_002,  
          mt\_wf\_log TYPE TABLE OF zdys\_s\_009,  
          mt\_ltxt   TYPE TABLE OF zdys\_s\_019,  
          mr\_rol    TYPE zdys\_tt\_001.  
  
    DATA: mt\_dysid\_selected TYPE zdys\_tt\_003.  
  
    DATA: mt\_zdys\_t\_016 TYPE TABLE OF zdys\_t\_016,  
          ms\_zdys\_t\_010 TYPE zdys\_t\_010,  
          mt\_zdys\_t\_027 TYPE TABLE OF zdys\_t\_027.  
  
    DATA: ms\_variant TYPE disvariant.  
  
\*  ALV Event Handler Methods  
    METHODS:  
      handle\_button\_click FOR EVENT button\_click OF cl\_gui\_alv\_grid  
        IMPORTING es\_col\_id es\_row\_no sender,  
      handle\_hotspot\_click FOR EVENT hotspot\_click OF cl\_gui\_alv\_grid  
        IMPORTING e\_row\_id e\_column\_id es\_row\_no sender,  
      handle\_toolbar FOR EVENT toolbar OF cl\_gui\_alv\_grid  
        IMPORTING e\_object e\_interactive sender,  
      handle\_user\_command FOR EVENT user\_command OF cl\_gui\_alv\_grid  
        IMPORTING e\_ucomm sender,  
      handle\_double\_click FOR EVENT double\_click OF cl\_gui\_alv\_grid  
        IMPORTING e\_row e\_column es\_row\_no sender,  
      handle\_data\_changed\_finished FOR EVENT data\_changed\_finished OF cl\_gui\_alv\_grid  
        IMPORTING e\_modified et\_good\_cells sender.  
  
  
    METHODS:  
      constructor,  
      init\_selection\_screen,  
      pbo\_selection\_screen,  
      call\_maintain\_screen IMPORTING iv\_viewmaint TYPE vim\_name,  
      call\_transaction IMPORTING iv\_tcode TYPE tcode,  
      check\_init RETURNING VALUE(rv\_error) TYPE xfeld,  
      get\_data\_inb RETURNING VALUE(rt\_item) TYPE zdys\_tt\_015,  
      get\_data\_oub,  
      get\_domvalue\_text IMPORTING iv\_domname    TYPE domname  
                                  iv\_domvalue\_l TYPE any  
                        EXPORTING ev\_ddtext     TYPE any,  
      check\_input\_partner CHANGING cv\_partner  TYPE bu\_partner  
                                   cv\_partnert TYPE ad\_namtext,  
      check\_input\_marka CHANGING cv\_marka  TYPE zdys\_e\_marka  
                                 cv\_markat TYPE zdys\_e\_markat,  
      check\_input\_matkl CHANGING cv\_matkl TYPE matkl  
                                 cv\_wgbez TYPE wgbez,  
      check\_stcd2\_inconsistency IMPORTING iv\_dysid          TYPE zdys\_e\_dysid  
                                          iv\_partner        TYPE bu\_partner  
                                          iv\_stcd2          TYPE stcd2  
                                          iv\_popup\_msg      TYPE xfeld OPTIONAL  
                                EXPORTING et\_return         TYPE bapiret2\_t  
                                RETURNING VALUE(rv\_matched) TYPE xfeld,  
      check\_auth\_gdr CHANGING ct\_zdys\_t\_001 TYPE zdys\_tt\_011,  
      pbo\_0100\_screen,  
      pbo\_0200\_screen,  
      pbo\_0300\_screen,  
      pbo\_0400\_screen,  
      pbo\_0500\_screen,  
      pbo\_0600\_screen,  
      pbo\_0700\_screen,  
      pai\_0100\_screen,  
      pai\_0200\_screen,  
      pai\_0300\_screen,  
      pai\_0400\_screen,  
      pai\_0500\_screen,  
      pai\_0600\_screen,  
      pai\_0700\_screen,  
      refresh\_0100\_screen,  
      refresh\_0200\_screen,  
      refresh\_0300\_screen,  
      refresh\_0700\_screen,  
      get\_fieldcatalog IMPORTING iv\_alv\_name TYPE char20  
                       EXPORTING et\_fieldcat TYPE lvc\_t\_fcat,  
      change\_col\_pos IMPORTING iv\_fieldname      TYPE lvc\_fname  
                               it\_fieldcat       TYPE lvc\_t\_fcat  
                     RETURNING VALUE(rv\_col\_pos) TYPE lvc\_colpos,  
      get\_layout IMPORTING iv\_alv\_name TYPE char20  
                 EXPORTING es\_layout   TYPE lvc\_s\_layo,  
      set\_event\_handlers,  
      check\_selected\_rows IMPORTING iv\_sel\_method   TYPE c  " S => Single, M => Multi  
                          RETURNING VALUE(rv\_error) TYPE xfeld,  
      refresh\_data\_inb,  
      refresh\_list,  
      execute\_process,  
      save\_screen\_0400,  
      save\_screen\_0500,  
      list\_process\_details,  
      check\_before\_update\_invoice IMPORTING is\_item   TYPE zdys\_s\_003  
                                  EXPORTING et\_return TYPE bapiret2\_t,  
      get\_auth\_input\_for\_update,  
      update\_invoice\_data\_single,  
      update\_invoice\_data\_multi,  
      adjust\_updated\_inv\_values IMPORTING it\_fname\_updated TYPE tty\_fname  
                                          is\_item\_old      TYPE zdys\_t\_001  
                                EXPORTING et\_return        TYPE bapiret2\_t  
                                CHANGING  cs\_item\_new      TYPE zdys\_t\_001,  
      save\_screen\_0300,  
      save\_screen\_0600,  
      restart\_workflow,  
      complete\_workflow\_manually,  
      display\_workflow\_log,  
      display\_attachments,  
      workflow\_process,  
      run\_esignature\_process,  
      run\_expense\_invoice IMPORTING iv\_dispmode TYPE xfeld OPTIONAL,  
      add\_order RETURNING VALUE(rv\_modified) TYPE xfeld,  
      delete\_order RETURNING VALUE(rv\_modified) TYPE xfeld,  
      delete\_invoice,  
      run\_return\_reason RETURNING VALUE(rv\_modified) TYPE xfeld,  
      run\_auto\_return RETURNING VALUE(rv\_modified) TYPE xfeld,  
      run\_update\_md RETURNING VALUE(rv\_modified) TYPE xfeld,  
      call\_rebilling\_screen RETURNING VALUE(rv\_called) TYPE xfeld,  
      call\_ltxt\_screen IMPORTING is\_item TYPE zdys\_s\_003,  
      get\_ltxt\_data IMPORTING iv\_dysid TYPE zdys\_e\_dysid,  
      create\_new\_ltxt IMPORTING iv\_dysid           TYPE zdys\_e\_dysid  
                      RETURNING VALUE(rv\_modified) TYPE xfeld,  
      display\_conf\_hierarchy IMPORTING iv\_dysid TYPE zdys\_e\_dysid,  
      display\_conf\_history IMPORTING iv\_dysid TYPE zdys\_e\_dysid,  
      display\_gib\_xml IMPORTING is\_item TYPE zdys\_s\_003,  
      update\_idoc\_statu IMPORTING is\_item   TYPE zdys\_s\_003  
                                  iv\_status TYPE /kspack/ef\_de\_status  
                                  it\_text   TYPE tlinet  
                        EXPORTING et\_return TYPE bapiret2\_t,  
      download\_as\_zip,  
      match\_dys,  
      display\_edocu IMPORTING iv\_ettn  TYPE zdys\_e\_ettn  
                              iv\_fatur TYPE zdys\_e\_fatur,  
      create\_sales\_inv IMPORTING is\_item   TYPE zdys\_s\_003  
                       CHANGING  ct\_return TYPE bapiret2\_t,  
      f4\_variant RETURNING VALUE(rv\_variant) TYPE slis\_vari,  
      run\_bg\_process. " Artalanda çalışacak işlemler  
  
  PRIVATE SECTION.  
    DATA: mt\_zdys\_t\_004 TYPE HASHED TABLE OF zdys\_t\_004 WITH UNIQUE KEY marka,  
          mt\_dd07t      TYPE SORTED TABLE OF dd07t WITH NON-UNIQUE KEY domname domvalue\_l,  
          mt\_zdys\_t\_006 TYPE TABLE OF zdys\_t\_006,  
          mt\_zdys\_t\_007 TYPE HASHED TABLE OF zdys\_t\_007 WITH UNIQUE KEY gdrtp,  
          mt\_zdys\_t\_021 TYPE TABLE OF zdys\_t\_021,  
          mt\_zdys\_t\_022 TYPE TABLE OF zdys\_t\_022.  
  
ENDCLASS.  
  
CLASS lcl\_main IMPLEMENTATION.  
  
  METHOD handle\_button\_click.  
    DATA: lt\_lines      TYPE TABLE OF tline,  
          lt\_txwnote    TYPE TABLE OF txw\_note,  
          lv\_name       TYPE thead-tdname,  
          lv\_ar\_object  TYPE saeobjart,  
          lv\_file\_count TYPE int4.  
  
    CASE sender.  
      WHEN go\_alv\_grid\_0100.  
        CASE es\_col\_id.  
          WHEN 'BTN\_FAT' OR " Fatura görseli  
               'BTN\_FATEK'. " Fatura eki görseli  
            READ TABLE gt\_item INTO DATA(ls\_item) INDEX es\_row\_no-row\_id.  
            CHECK sy-subrc = 0.  
            " E-fatura görseli varsa  
            IF es\_col\_id = 'BTN\_FAT' AND ls\_item-eguid IS NOT INITIAL.  
              display\_edocu( iv\_ettn  = ls\_item-ettn  
                             iv\_fatur = ls\_item-fatur ).  
            ELSE.  
              IF es\_col\_id = 'BTN\_FAT'.  
                lv\_ar\_object = zdys\_cl\_attachment=>mc\_ar\_object\_fatura.  
              ELSEIF es\_col\_id = 'BTN\_FATEK'.  
                lv\_ar\_object = zdys\_cl\_attachment=>mc\_ar\_object\_faturaek.  
              ENDIF.  
  
              DATA(lo\_attachment) = NEW zdys\_cl\_attachment( iv\_dysid     = ls\_item-dysid  
                                                            iv\_ar\_object = lv\_ar\_object ).  
              CLEAR lv\_file\_count.  
              LOOP AT lo\_attachment->ms\_data-attachments INTO DATA(ls\_attachments)  
                                                         WHERE ar\_object = lv\_ar\_object.  
                ADD 1 TO lv\_file\_count.  
              ENDLOOP.  
              " Eğer seçilen doküman türünden 1 tane varsa direkt görüntülenir  
              " 1'den fazla var ise ek işlemleri ekranı çıkarılır  
              IF lv\_file\_count = 1.  
                lo\_attachment->display\_document( ls\_attachments ).  
              ELSE.  
                lo\_attachment->call\_attachment\_screen( ).  
              ENDIF.  
            ENDIF.  
          WHEN 'BTN\_TXT'.   " Açıklama  
            READ TABLE gt\_item INTO ls\_item INDEX es\_row\_no-row\_id.  
            IF sy-subrc = 0.  
              call\_ltxt\_screen( ls\_item ).  
            ENDIF.  
          WHEN 'BTN\_ONYH'.  " Onay hiyerarşisi  
            READ TABLE gt\_item INTO ls\_item INDEX es\_row\_no-row\_id.  
            IF sy-subrc = 0.  
              display\_conf\_hierarchy( ls\_item-dysid ).  
            ENDIF.  
          WHEN 'BTN\_ONYT'.  " Onay tarihçesi  
            READ TABLE gt\_item INTO ls\_item INDEX es\_row\_no-row\_id.  
            IF sy-subrc = 0.  
              display\_conf\_history( ls\_item-dysid ).  
            ENDIF.  
          WHEN 'BTN\_XML'.  " GIB XML  
            READ TABLE gt\_item INTO ls\_item INDEX es\_row\_no-row\_id.  
            IF sy-subrc = 0.  
              display\_gib\_xml( ls\_item ).  
            ENDIF.  
        ENDCASE.  
  
      WHEN go\_alv\_grid\_0200.  
        CASE es\_col\_id.  
          WHEN 'ACKLM'. " Açıklama  
            READ TABLE mt\_detail INTO DATA(ls\_detail) INDEX es\_row\_no-row\_id.  
            IF sy-subrc = 0.  
              lv\_name = ls\_detail-dysid && ls\_detail-islno.  
              CALL FUNCTION 'READ\_TEXT'  
                EXPORTING  
                  id                      = 'Z001'  
                  language                = sy-langu  
                  name                    = lv\_name  
                  object                  = zdys\_cl\_general=>mc\_tdobject  
                TABLES  
                  lines                   = lt\_lines  
                EXCEPTIONS  
                  id                      = 1  
                  language                = 2  
                  name                    = 3  
                  not\_found               = 4  
                  object                  = 5  
                  reference\_check         = 6  
                  wrong\_access\_to\_archive = 7  
                  OTHERS                  = 8.  
              IF sy-subrc = 0.  
                LOOP AT lt\_lines INTO DATA(ls\_lines).  
                  APPEND VALUE #( line = ls\_lines-tdline ) TO lt\_txwnote.  
                ENDLOOP.  
                CALL FUNCTION 'TXW\_TEXTNOTE\_EDIT'  
                  EXPORTING  
                    edit\_mode = ''  
                  TABLES  
                    t\_txwnote = lt\_txwnote.  
              ENDIF.  
            ENDIF.  
        ENDCASE.  
    ENDCASE.  
  ENDMETHOD.  
  
  METHOD handle\_hotspot\_click.  
    DATA: lt\_rsparams   TYPE TABLE OF rsparams,  
          lt\_lines      TYPE TABLE OF tline,  
          lt\_dynpfields TYPE TABLE OF dynpread,  
          lv\_dyname     TYPE d020s-prog.  
  
    CASE sender.  
      WHEN go\_alv\_grid\_0100.  
        IF e\_column\_id = 'BLGNO'.  
          READ TABLE gt\_item INTO DATA(ls\_item) INDEX e\_row\_id-index.  
          IF sy-subrc = 0 AND ls\_item-blgno IS NOT INITIAL.  
            zdys\_cl\_general=>get\_awkey\_values(  
              EXPORTING  
                iv\_awtyp     = ls\_item-awtyp  
                iv\_awkey     = ls\_item-awkey  
              IMPORTING  
                ev\_bukrs\_fi  = DATA(lv\_bukrs\_fi)  
                ev\_belnr\_fi  = DATA(lv\_belnr\_fi)  
                ev\_gjahr\_fi  = DATA(lv\_gjahr\_fi)  
                ev\_belnr\_mm  = DATA(lv\_belnr\_mm)  
                ev\_gjahr\_mm  = DATA(lv\_gjahr\_mm)  
                ev\_vbeln\_sd  = DATA(lv\_vbeln\_sd)  
                ev\_blgno     = DATA(lv\_blgno) ).  
            IF lv\_blgno IS NOT INITIAL.  
              CASE ls\_item-awtyp.  
                WHEN 'RMRP'.  
                  APPEND VALUE #( selname = 'S\_BELNR'  
                                  kind    = 'S'  
                                  sign    = 'I'  
                                  option  = 'EQ'  
                                  low     = lv\_belnr\_mm  
                                 ) TO lt\_rsparams.  
                  APPEND VALUE #( selname = 'S\_GJAHR'  
                                  kind    = 'S'  
                                  sign    = 'I'  
                                  option  = 'EQ'  
                                  low     = lv\_gjahr\_mm  
                                 ) TO lt\_rsparams.  
                  APPEND VALUE #( selname = 'S\_BUDAT'  
                                  kind    = 'S'  
                                  sign    = 'I'  
                                  option  = 'BT'  
                                  low     = '20000101'  
                                  high    = '99991231'  
                                 ) TO lt\_rsparams.  
                  SUBMIT zmmgen\_r\_0001 WITH SELECTION-TABLE lt\_rsparams AND RETURN.  
                WHEN 'VBRK'.  
                  APPEND VALUE #( selname = 'S\_VBELN'  
                                  kind    = 'S'  
                                  sign    = 'I'  
                                  option  = 'EQ'  
                                  low     = lv\_vbeln\_sd  
                                 ) TO lt\_rsparams.  
                  APPEND VALUE #( selname = 'S\_FKDAT'  
                                  kind    = 'S'  
                                  sign    = 'I'  
                                  option  = 'BT'  
                                  low     = '20000101'  
                                  high    = '99991231'  
                                 ) TO lt\_rsparams.  
                  SUBMIT zsdgen\_r\_0001 WITH SELECTION-TABLE lt\_rsparams AND RETURN.  
                WHEN 'BKPF' OR 'BKPFF'.  
                  SET PARAMETER ID 'BLN' FIELD lv\_belnr\_fi.  
                  SET PARAMETER ID 'BUK' FIELD lv\_bukrs\_fi.  
                  SET PARAMETER ID 'GJR' FIELD lv\_gjahr\_fi.  
                  CALL TRANSACTION 'FB03' WITH AUTHORITY-CHECK AND SKIP FIRST SCREEN.  
              ENDCASE.  
            ENDIF.  
          ENDIF.  
        ENDIF.  
  
      WHEN go\_alv\_grid\_0700.  
        IF e\_column\_id = 'ACKLM'.  
          READ TABLE mt\_ltxt INTO DATA(ls\_itxt) INDEX e\_row\_id-index.  
          CHECK sy-subrc = 0.  
          REFRESH gs\_0700-text.  
          gs\_0700-header\_txt = |{ ls\_itxt-islno ALPHA = OUT }|.  
          gs\_0700-header\_txt = gs\_0700-header\_txt && ` ` && TEXT-049.  
          CALL FUNCTION 'READ\_TEXT'  
            EXPORTING  
              id                      = ls\_itxt-tdid  
              language                = ls\_itxt-tdspras  
              name                    = ls\_itxt-tdname  
              object                  = ls\_itxt-tdobject  
            TABLES  
              lines                   = lt\_lines  
            EXCEPTIONS  
              id                      = 1  
              language                = 2  
              name                    = 3  
              not\_found               = 4  
              object                  = 5  
              reference\_check         = 6  
              wrong\_access\_to\_archive = 7  
              OTHERS                  = 8.  
          IF sy-subrc = 0.  
            CALL FUNCTION 'CONVERT\_ITF\_TO\_STREAM\_TEXT'  
              EXPORTING  
                language    = 'T'  
              TABLES  
                itf\_text    = lt\_lines  
                text\_stream = gs\_0700-text.  
          ENDIF.  
          CALL METHOD go\_textedit\_0700->set\_text\_as\_stream  
            EXPORTING  
              text = gs\_0700-text.  
          lv\_dyname = sy-repid.  
          APPEND VALUE #( fieldname = 'GS\_0700-HEADER\_TXT'  
                          fieldvalue = gs\_0700-header\_txt ) TO lt\_dynpfields.  
          CALL FUNCTION 'DYNP\_UPDATE\_FIELDS'  
            EXPORTING  
              dyname     = lv\_dyname  
              dynumb     = '0700'  
              request    = 'A'  
            TABLES  
              dynpfields = lt\_dynpfields.  
        ENDIF.  
    ENDCASE.  
  ENDMETHOD.  
  
  METHOD handle\_toolbar.  
    CASE sender.  
      WHEN go\_alv\_grid\_0300.  
        APPEND VALUE #( butn\_type = 3 ) TO e\_object->mt\_toolbar.  
        APPEND VALUE #( function  = mc\_fc\_add\_row  
                        butn\_type = 0  
                        icon      = icon\_insert\_row  
                        quickinfo = TEXT-029  
                        text      = TEXT-029 ) TO e\_object->mt\_toolbar.  
        APPEND VALUE #( function  = mc\_fc\_del\_row  
                        butn\_type = 0  
                        icon      = icon\_delete\_row  
                        quickinfo = TEXT-030  
                        text      = TEXT-030 ) TO e\_object->mt\_toolbar.  
        APPEND VALUE #( butn\_type = 3 ) TO e\_object->mt\_toolbar.  
    ENDCASE.  
  ENDMETHOD.  
  
  METHOD handle\_user\_command.  
    CASE sender.  
      WHEN go\_alv\_grid\_0300.  
        CASE e\_ucomm.  
          WHEN mc\_fc\_add\_row.  
            IF add\_order( ) = 'X'.  
              gs\_0300-order\_changed = 'X'.  
              refresh\_0300\_screen( ).  
            ENDIF.  
          WHEN mc\_fc\_del\_row.  
            IF delete\_order( ) = 'X'.  
              gs\_0300-order\_changed = 'X'.  
              refresh\_0300\_screen( ).  
            ENDIF.  
        ENDCASE.  
    ENDCASE.  
  ENDMETHOD.  
  
  METHOD handle\_double\_click.  
    CASE sender.  
      WHEN go\_alv\_grid\_0100.  
    ENDCASE.  
  ENDMETHOD.  
  
  METHOD handle\_data\_changed\_finished.  
    CASE sender.  
      WHEN go\_alv\_grid\_0100.  
    ENDCASE.  
  ENDMETHOD.  
  
  METHOD constructor.  
    IF sy-tcode = 'ZDYS0001'.  
      p\_report = ''.  
    ENDIF.  
  
    CLEAR ms\_variant.  
    ms\_variant-report = sy-repid.  
    ms\_variant-handle = 'H1'.  
  
    SELECT \*  
           FROM zdys\_t\_004  
           INTO TABLE mt\_zdys\_t\_004.                    "#EC CI\_NOWHERE  
    SELECT \*  
           FROM zdys\_t\_007  
           INTO TABLE mt\_zdys\_t\_007.                    "#EC CI\_NOWHERE  
  
    SELECT SINGLE \*  
           FROM zdys\_t\_010  
           INTO ms\_zdys\_t\_010  
           WHERE uname = sy-uname  
           AND aktif = 'X'.  
    SELECT \*  
           FROM zdys\_t\_027  
           INTO TABLE mt\_zdys\_t\_027  
           WHERE vkedc = sy-uname  
           AND begda <= sy-datum  
           AND endda >= sy-datum.  
  ENDMETHOD.  
  
  METHOD init\_selection\_screen.  
    DATA: lt\_variants TYPE TABLE OF ltvariant,  
          lr\_report   TYPE RANGE OF ltdx-report,  
          lr\_handle   TYPE RANGE OF ltdx-handle,  
          ls\_fattr    LIKE LINE OF s\_fattr.  
  
    " Seçim ekranı başlık adı(Tittle)  
    IF p\_report = 'X'.  
      sy-title = TEXT-032.  
    ELSE.  
      sy-title = TEXT-031.  
    ENDIF.  
  
    " Fatura tarihi  
    IF p\_report = 'X'.  
      CLEAR ls\_fattr.  
      REFRESH s\_fattr.  
      CALL FUNCTION 'LAST\_DAY\_OF\_MONTHS'  
        EXPORTING  
          day\_in            = sy-datum  
        IMPORTING  
          last\_day\_of\_month = ls\_fattr-high  
        EXCEPTIONS  
          day\_in\_no\_date    = 1  
          OTHERS            = 2.  
      IF sy-subrc = 0.  
        CALL FUNCTION 'BKK\_ADD\_MONTH\_TO\_DATE'  
          EXPORTING  
            months  = -1  
            olddate = ls\_fattr-high  
          IMPORTING  
            newdate = ls\_fattr-low.  
        ls\_fattr-low+6(2) = '01'.  
        APPEND ls\_fattr TO s\_fattr.  
      ENDIF.  
    ENDIF.  
  
    " Başka ekrandan DYS Id bazında görüntüleme için eklendi  
    GET PARAMETER ID 'ZDYSID' FIELD s\_dysid-low.  
    IF s\_dysid-low IS NOT INITIAL.  
      APPEND VALUE #( sign = 'I'  
                      option = 'EQ'  
                      low = s\_dysid-low  
                     ) TO s\_dysid.  
      CLEAR s\_dysid-low.  
      SET PARAMETER ID 'ZDYSID' FIELD s\_dysid-low.  
      REFRESH s\_fattr.  
      APPEND VALUE #( sign = 'I'  
                      option = 'BT'  
                      low = '20240101'  
                      high = '99991231'  
                     ) TO s\_fattr.  
    ENDIF.  
  
    " Varsayılan varyant seçimi  
    APPEND VALUE #( sign   = 'I'  
                    option = 'EQ'  
                    low    = ms\_variant-report  
                   ) TO lr\_report.  
    APPEND VALUE #( sign   = 'I'  
                    option = 'EQ'  
                    low    = ms\_variant-handle  
                   ) TO lr\_handle.  
    CALL FUNCTION 'LT\_VARIANTS\_READ\_FROM\_LTDX'  
      TABLES  
        et\_variants  = lt\_variants  
        it\_ra\_report = lr\_report  
        it\_ra\_handle = lr\_handle  
      EXCEPTIONS  
        not\_found    = 1  
        OTHERS       = 2.  
    DO 1 TIMES.  
      READ TABLE lt\_variants INTO DATA(ls\_variants) WITH KEY erfname    = sy-uname  
                                                             defaultvar = 'X'.  
      CHECK sy-subrc <> 0.  
      READ TABLE lt\_variants INTO ls\_variants WITH KEY erfname = sy-uname.  
      CHECK sy-subrc <> 0.  
      READ TABLE lt\_variants INTO ls\_variants WITH KEY defaultvar = 'X'.  
    ENDDO.  
    p\_varian = ls\_variants-variant.  
\*    CALL FUNCTION 'REUSE\_ALV\_VARIANT\_DEFAULT\_GET'  
\*      EXPORTING  
\*        i\_save        = 'A'  
\*      CHANGING  
\*        cs\_variant    = ms\_variant  
\*      EXCEPTIONS  
\*        wrong\_input   = 1  
\*        not\_found     = 2  
\*        program\_error = 3  
\*        OTHERS        = 4.  
\*    IF sy-subrc = 0 AND ms\_variant-variant IS NOT INITIAL.  
\*      p\_varian = ms\_variant-variant.  
\*    ENDIF.  
  
  ENDMETHOD.  
  
  METHOD pbo\_selection\_screen.  
    LOOP AT SCREEN.  
      CASE screen-group1.  
        WHEN 'RAP'.  
          IF p\_report = 'X' AND p\_inb = 'X'.  
            screen-active = 1.  
          ELSE.  
            screen-active = 0.  
          ENDIF.  
          MODIFY SCREEN.  
        WHEN 'INB'.  
          IF p\_inb = 'X'.  
            screen-active = 1.  
          ELSE.  
            screen-active = 0.  
          ENDIF.  
          MODIFY SCREEN.  
        WHEN 'YON'.  
          IF p\_report = 'X'.  
            screen-active = 1.  
          ELSE.  
            screen-active = 0.  
          ENDIF.  
          MODIFY SCREEN.  
      ENDCASE.  
  
      CASE screen-name.  
        WHEN 'S\_FATTR-LOW'.  
          IF p\_report = 'X' AND p\_updty IS INITIAL.  
            screen-required = 1.  
          ELSE.  
            screen-required = 0.  
          ENDIF.  
          MODIFY SCREEN.  
      ENDCASE.  
    ENDLOOP.  
  ENDMETHOD.  
  
  METHOD call\_maintain\_screen.  
    CALL FUNCTION 'AUTHORITY\_CHECK\_TCODE'  
      EXPORTING  
        tcode  = 'SM30'  
      EXCEPTIONS  
        ok     = 1  
        not\_ok = 2.  
    IF sy-subrc = 2.  
      MESSAGE i077(s#) WITH 'SM30' DISPLAY LIKE 'E'.  
    ELSE.  
      CALL FUNCTION 'VIEW\_MAINTENANCE\_CALL'  
        EXPORTING  
          action                       = 'U'  
          view\_name                    = iv\_viewmaint  
        EXCEPTIONS  
          client\_reference             = 1  
          foreign\_lock                 = 2  
          invalid\_action               = 3  
          no\_clientindependent\_auth    = 4  
          no\_database\_function         = 5  
          no\_editor\_function           = 6  
          no\_show\_auth                 = 7  
          no\_tvdir\_entry               = 8  
          no\_upd\_auth                  = 9  
          only\_show\_allowed            = 10  
          system\_failure               = 11  
          unknown\_field\_in\_dba\_sellist = 12  
          view\_not\_found               = 13  
          maintenance\_prohibited       = 14  
          OTHERS                       = 15.  
      IF sy-subrc <> 0.  
        ##FM\_SUBRC\_OK  
      ENDIF.  
    ENDIF.  
  ENDMETHOD.  
  
  METHOD call\_transaction.  
    CALL FUNCTION 'AUTHORITY\_CHECK\_TCODE'  
      EXPORTING  
        tcode  = iv\_tcode  
      EXCEPTIONS  
        ok     = 1  
        not\_ok = 2.  
    IF sy-subrc = 2.  
      MESSAGE i077(s#) WITH iv\_tcode DISPLAY LIKE 'E'.  
    ELSE.  
      CALL TRANSACTION iv\_tcode WITH AUTHORITY-CHECK.  
    ENDIF.  
  ENDMETHOD.  
  
  METHOD check\_init.  
    REFRESH: mr\_rol, mt\_dysid\_selected.  
    IF p\_report = ''.  " Fatura işleme iş listesi ekranı ise  
      " Supervisor kullanıcı ise rolden bağımsız bütün faturaları görebilir  
      IF ms\_zdys\_t\_010-uname IS NOT INITIAL.  
        APPEND VALUE #( sign   = 'I'  
                        option = 'CP'  
                        low    = '\*'  
                       ) TO mr\_rol.  
      ELSE.  
        " Kullanıcıya ait roller  
        zdys\_cl\_general=>get\_roles\_of\_user(  
          RECEIVING  
            rt\_rol        = mr\_rol  
          EXCEPTIONS  
            no\_role\_found = 1  
            OTHERS        = 2 ).  
        IF sy-subrc <> 0.  
          MESSAGE s012 WITH sy-uname DISPLAY LIKE 'E'.  
          rv\_error = 'X'.  
        ENDIF.  
      ENDIF.  
    ENDIF.  
  
    SELECT SINGLE \* FROM t001 INTO ms\_t001 WHERE bukrs = p\_bukrs.  
  
    " Düzen(ALV varyantı) kontrolü  
    IF p\_varian IS INITIAL.  
      CALL FUNCTION 'REUSE\_ALV\_VARIANT\_DEFAULT\_GET'  
        EXPORTING  
          i\_save        = 'A'  
        CHANGING  
          cs\_variant    = ms\_variant  
        EXCEPTIONS  
          wrong\_input   = 1  
          not\_found     = 2  
          program\_error = 3  
          OTHERS        = 4.  
      IF sy-subrc = 0 AND ms\_variant-variant IS NOT INITIAL.  
        p\_varian = ms\_variant-variant.  
      ENDIF.  
    ELSE.  
      ms\_variant-variant = p\_varian.  
      CALL FUNCTION 'REUSE\_ALV\_VARIANT\_EXISTENCE'  
        EXPORTING  
          i\_save        = 'A'  
        CHANGING  
          cs\_variant    = ms\_variant  
        EXCEPTIONS  
          wrong\_input   = 1  
          not\_found     = 2  
          program\_error = 3  
          OTHERS        = 4.  
      IF sy-subrc <> 0.  
        rv\_error = 'X'.  
        CLEAR: ms\_variant-variant, p\_varian.  
        MESSAGE ID sy-msgid TYPE 'I' NUMBER sy-msgno  
                WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4  
                DISPLAY LIKE 'E'.  
      ENDIF.  
    ENDIF.  
  ENDMETHOD.  
  
  METHOD get\_data\_inb.  
    TYPES: BEGIN OF ty\_but000,  
             partner   TYPE but000-partner,  
             name\_org1 TYPE but000-name\_org1,  
             name\_org2 TYPE but000-name\_org2,  
           END OF ty\_but000.  
    TYPES: BEGIN OF ty\_t023t,  
             matkl TYPE t023t-matkl,  
             wgbez TYPE t023t-wgbez,  
           END OF ty\_t023t.  
    TYPES: BEGIN OF ty\_rseg,  
             belnr  TYPE rseg-belnr,  
             gjahr  TYPE rseg-gjahr,  
             buzei  TYPE rseg-buzei,  
             ebeln  TYPE rseg-ebeln,  
             ebelp  TYPE rseg-ebelp,  
             rbstat TYPE rbkp-rbstat,  
           END OF ty\_rseg.  
    TYPES: BEGIN OF ty\_vbfa,  
             vbelv   TYPE vbfa-vbelv,  
             posnv   TYPE vbfa-posnv,  
             vbeln   TYPE vbfa-vbeln,  
             posnn   TYPE vbfa-posnn,  
             vbtyp\_n TYPE vbfa-vbtyp\_n,  
           END OF ty\_vbfa.  
    TYPES: BEGIN OF ty\_lips,  
             vbeln TYPE lips-vbeln,  
             posnr TYPE lips-posnr,  
             vgbel TYPE lips-vgbel,  
             vgpos TYPE lips-vgpos,  
             faksp TYPE lips-faksp,  
             fksta TYPE lips-fksta,  
             faksk TYPE likp-faksk,  
             bolnr TYPE likp-bolnr,  
           END OF ty\_lips.  
    TYPES: BEGIN OF ty\_vbrp,  
             vbeln TYPE vbrp-vbeln,  
             posnr TYPE vbrp-posnr,  
             vgbel TYPE vbrp-vgbel,  
             vgpos TYPE vbrp-vgpos,  
             objky TYPE nast-objky,  
           END OF ty\_vbrp.  
  
    DATA: lt\_but000 TYPE HASHED TABLE OF ty\_but000 WITH UNIQUE KEY partner,  
          lt\_t023t  TYPE HASHED TABLE OF ty\_t023t WITH UNIQUE KEY matkl,  
          lt\_rseg   TYPE SORTED TABLE OF ty\_rseg WITH NON-UNIQUE KEY belnr gjahr,  
          lt\_vbfa   TYPE SORTED TABLE OF ty\_vbfa WITH NON-UNIQUE KEY vbeln posnn,  
          lt\_lips   TYPE SORTED TABLE OF ty\_lips WITH NON-UNIQUE KEY vgbel vgpos,  
          lt\_vbrp   TYPE SORTED TABLE OF ty\_vbrp WITH NON-UNIQUE KEY vgbel vgpos,  
          lt\_nast   TYPE SORTED TABLE OF nast WITH NON-UNIQUE KEY objky vstat,  
          ls\_nast   TYPE nast,  
          lt\_toa01  TYPE SORTED TABLE OF toa01 WITH NON-UNIQUE KEY object\_id.  
  
    DATA: lt\_zdys\_t\_001 TYPE zdys\_tt\_011,  
          lt\_zdys\_t\_016 TYPE SORTED TABLE OF zdys\_t\_016 WITH NON-UNIQUE KEY dysid,  
          lt\_zdys\_t\_002 TYPE SORTED TABLE OF zdys\_t\_002 WITH NON-UNIQUE KEY dysid.  
  
    DATA: lt\_/kspack/ef001 TYPE HASHED TABLE OF /kspack/ef001 WITH UNIQUE KEY eguid.  
  
    DATA: lt\_dysid\_unmatched TYPE zdys\_tt\_003,  
          lt\_dysid\_attach    TYPE zdys\_tt\_003,  
          lt\_matkl           TYPE TABLE OF matkl\_struct,  
          lr\_dysid           TYPE RANGE OF zdys\_t\_001-dysid,  
          lr\_awtyp           TYPE RANGE OF zdys\_t\_001-awtyp,  
          lr\_awkey           TYPE RANGE OF zdys\_t\_001-awkey,  
          lr\_matkl           TYPE mmpur\_t\_mat\_grp,  
          lt\_matkl\_det       TYPE zfi\_tt\_003,  
          ls\_matkl\_det       TYPE zfi\_s\_003,  
          lv\_completed\_rmrp  TYPE xfeld,  
          lv\_conf\_chk        TYPE xfeld.  
  
    DATA: lt\_lines TYPE TABLE OF tline,  
          lv\_name  TYPE thead-tdname,  
          lv\_tabix TYPE sy-tabix,  
          lv\_islno TYPE zdys\_e\_islno,  
          lv\_bolnr TYPE likp-bolnr.  
  
    BREAK-POINT ID zdys\_fatura.  
  
    IF mt\_dysid\_selected[] IS INITIAL.  
      APPEND LINES OF s\_dysid TO lr\_dysid.  
    ELSE.  
      " ALV'de seçili satırın yenilenmesi için eklendi  
      LOOP AT mt\_dysid\_selected INTO DATA(lv\_dysid\_selected).  
        APPEND VALUE #( sign = 'I' option = 'EQ' low = lv\_dysid\_selected ) TO lr\_dysid.  
      ENDLOOP.  
    ENDIF.  
  
    " SAP fatura belge numarası filtreleme  
    IF s\_blgno[] IS NOT INITIAL.  
      LOOP AT s\_blgno INTO DATA(ls\_blgno).  
        IF ls\_blgno-low(1) = '9'. " Satış fatura no  
          APPEND VALUE #( sign   = 'I'  
                          option = 'EQ'  
                          low    = 'VBRK' ) TO lr\_awtyp.  
          APPEND VALUE #( sign   = ls\_blgno-sign  
                          option = ls\_blgno-option  
                          low    = ls\_blgno-low ) TO lr\_awkey.  
        ELSE.  
          IF ls\_blgno-low(2) = '51'. " Satın alma fatura belge no  
            APPEND VALUE #( sign   = 'I'  
                            option = 'EQ'  
                            low    = 'RMRP' ) TO lr\_awtyp.  
          ELSE. " FI Muhasebe belge no  
            APPEND VALUE #( sign   = 'I'  
                            option = 'EQ'  
                            low    = 'BKPFF' ) TO lr\_awtyp.  
          ENDIF.  
          IF ls\_blgno-option = 'EQ'.  
            APPEND VALUE #( sign   = ls\_blgno-sign  
                            option = 'CP'  
                            low    = ls\_blgno-low && `\*` ) TO lr\_awkey.  
          ELSE.  
            APPEND VALUE #( sign   = ls\_blgno-sign  
                            option = ls\_blgno-option  
                            low    = ls\_blgno-low && `\*`  
                            high   = ls\_blgno-high && `\*` ) TO lr\_awkey.  
          ENDIF.  
        ENDIF.  
      ENDLOOP.  
      SORT lr\_awtyp.  
      DELETE ADJACENT DUPLICATES FROM lr\_awtyp.  
    ENDIF.  
  
    SELECT \* FROM zdys\_t\_001  
             INTO TABLE lt\_zdys\_t\_001  
             WHERE dysid IN lr\_dysid  
             AND rol IN mr\_rol  
             AND partner IN s\_partn  
             AND fatno IN s\_fatno  
             AND fattr IN s\_fattr  
             AND fatur IN s\_fatur  
             AND marka IN s\_marka  
             AND matkl IN s\_matkl  
             AND rol   IN s\_rol  
             AND tasbl IN s\_tasbl  
             AND fsorm IN s\_fsorm  
             AND bimno IN s\_bimno  
             AND awtyp IN lr\_awtyp  
             AND awkey IN lr\_awkey  
             AND crdat IN s\_crdat  
             AND crnam IN s\_crnam.  
  
    " Gider faturalarını görüntüleme için yetki kontrolü  
    check\_auth\_gdr( CHANGING ct\_zdys\_t\_001 = lt\_zdys\_t\_001 ).  
  
    " Fatura belgeleri ve DYS durumları son güncel hali ile güncellenir  
    zdys\_cl\_general=>adjust\_dys\_values( EXPORTING iv\_date\_check = '12'  " Son 12 ayın fatura durumlarını baştan güncelle  
                                        CHANGING  ct\_item       = lt\_zdys\_t\_001 ).  
    IF s\_durum[] IS NOT INITIAL.  
      DELETE lt\_zdys\_t\_001 WHERE durum NOT IN s\_durum.  
    ENDIF.  
    rt\_item[] = CORRESPONDING #( lt\_zdys\_t\_001[] ).  
  
    IF rt\_item[] IS NOT INITIAL.  
      SELECT \*  
             FROM zdys\_t\_016  
             INTO TABLE lt\_zdys\_t\_016  
             FOR ALL ENTRIES IN rt\_item  
             WHERE dysid = rt\_item-dysid.  
    ENDIF.  
  
    " Toplu olarak çekilecek tablolar belirleniyor  
    LOOP AT rt\_item INTO DATA(ls\_item).  
      IF ls\_item-partner IS NOT INITIAL.  
        READ TABLE lt\_but000 TRANSPORTING NO FIELDS WITH TABLE KEY partner = ls\_item-partner.  
        IF sy-subrc <> 0.  
          INSERT VALUE #( partner = ls\_item-partner ) INTO TABLE lt\_but000.  
        ENDIF.  
      ENDIF.  
      IF ls\_item-matkl IS NOT INITIAL.  
        READ TABLE lt\_t023t TRANSPORTING NO FIELDS WITH TABLE KEY matkl = ls\_item-matkl.  
        IF sy-subrc <> 0.  
          INSERT VALUE #( matkl = ls\_item-matkl ) INTO TABLE lt\_t023t.  
        ENDIF.  
      ENDIF.  
  
      " Doküman bilgileri  
      APPEND ls\_item-dysid TO lt\_dysid\_attach.  
  
      " Yapılan işlem detayları  
      READ TABLE lt\_zdys\_t\_002 TRANSPORTING NO FIELDS WITH KEY dysid = ls\_item-dysid.  
      IF sy-subrc <> 0.  
        INSERT VALUE #( dysid = ls\_item-dysid ) INTO TABLE lt\_zdys\_t\_002.  
      ENDIF.  
  
      " E-fatura bilgisi  
      IF ls\_item-fatur = 'EF' AND ls\_item-ettn IS NOT INITIAL.  
        READ TABLE lt\_/kspack/ef001 TRANSPORTING NO FIELDS WITH TABLE KEY eguid = ls\_item-ettn.  
        IF sy-subrc <> 0.  
          INSERT VALUE #( eguid = ls\_item-ettn ) INTO TABLE lt\_/kspack/ef001.  
        ENDIF.  
      ENDIF.  
    ENDLOOP.  
  
    IF lt\_but000[] IS NOT INITIAL.  
      SELECT partner name\_org1 name\_org2  
             FROM but000  
             INTO TABLE lt\_but000  
             FOR ALL ENTRIES IN lt\_but000  
             WHERE partner = lt\_but000-partner.  
    ENDIF.  
    IF lt\_t023t[] IS NOT INITIAL.  
      SELECT matkl wgbez  
             FROM t023t  
             INTO TABLE lt\_t023t  
             FOR ALL ENTRIES IN lt\_t023t  
             WHERE spras = sy-langu  
             AND matkl = lt\_t023t-matkl.  
    ENDIF.  
  
    DATA(lt\_toa01\_tmp) = zdys\_cl\_attachment=>get\_toa01\_values( EXPORTING it\_dysid = lt\_dysid\_attach ).  
    INSERT LINES OF lt\_toa01\_tmp INTO TABLE lt\_toa01.  
  
    IF lt\_zdys\_t\_002[] IS NOT INITIAL.  
      SELECT \*  
             FROM zdys\_t\_002  
             INTO TABLE lt\_zdys\_t\_002  
             FOR ALL ENTRIES IN lt\_zdys\_t\_002  
             WHERE dysid = lt\_zdys\_t\_002-dysid.  
    ENDIF.  
    IF lt\_/kspack/ef001 IS NOT INITIAL.  
      SELECT \*  
             FROM /kspack/ef001  
             INTO TABLE lt\_/kspack/ef001  
             FOR ALL ENTRIES IN lt\_/kspack/ef001  
             WHERE eguid = lt\_/kspack/ef001-eguid  
             AND direct = '2'.  
    ENDIF.  
  
    LOOP AT rt\_item ASSIGNING FIELD-SYMBOL(<ls\_item>).  
      " Ek butonları  
      READ TABLE lt\_toa01 TRANSPORTING NO FIELDS WITH KEY object\_id = <ls\_item>-dysid.  
      IF sy-subrc = 0.  
        lv\_tabix = sy-tabix.  
        LOOP AT lt\_toa01 INTO DATA(ls\_toa01) FROM lv\_tabix.  
          IF ls\_toa01-object\_id = <ls\_item>-dysid.  
            CASE ls\_toa01-ar\_object.  
              WHEN zdys\_cl\_attachment=>mc\_ar\_object\_fatura.  
                IF <ls\_item>-btn\_fat IS INITIAL.  
                  <ls\_item>-btn\_fat = icon\_gos\_services\_attachment.  
                  INSERT VALUE #( fieldname = 'BTN\_FAT'  
                                  style     = cl\_gui\_alv\_grid=>mc\_style\_button ) INTO TABLE <ls\_item>-cellstyle.  
                ENDIF.  
              WHEN zdys\_cl\_attachment=>mc\_ar\_object\_faturaek.  
                IF <ls\_item>-btn\_fatek IS INITIAL.  
                  <ls\_item>-btn\_fatek = icon\_gos\_services\_attachment.  
                  INSERT VALUE #( fieldname = 'BTN\_FATEK'  
                                  style     = cl\_gui\_alv\_grid=>mc\_style\_button ) INTO TABLE <ls\_item>-cellstyle.  
                ENDIF.  
            ENDCASE.  
          ELSE.  
            EXIT.  
          ENDIF.  
        ENDLOOP.  
      ENDIF.  
  
      " E-fatura bilgisi  
      IF <ls\_item>-fatur = 'EF'.  
        READ TABLE lt\_/kspack/ef001 INTO DATA(ls\_/kspack/ef001) WITH TABLE KEY eguid = <ls\_item>-ettn.  
        IF sy-subrc = 0.  
          <ls\_item>-eguid = ls\_/kspack/ef001-eguid.  
          IF <ls\_item>-btn\_fat IS INITIAL.  
            <ls\_item>-btn\_fat = icon\_gos\_services\_attachment.  
            INSERT VALUE #( fieldname = 'BTN\_FAT'  
                            style     = cl\_gui\_alv\_grid=>mc\_style\_button ) INTO TABLE <ls\_item>-cellstyle.  
          ENDIF.  
        ENDIF.  
      ENDIF.  
  
      " Muhatap adı  
      IF <ls\_item>-partner IS NOT INITIAL.  
        READ TABLE lt\_but000 INTO DATA(ls\_but000) WITH TABLE KEY partner = <ls\_item>-partner.  
        IF sy-subrc = 0.  
          <ls\_item>-partnert = ls\_but000-name\_org1 && ` ` && ls\_but000-name\_org2.  
        ENDIF.  
      ENDIF.  
  
      " Fatura tipi  
      get\_domvalue\_text( EXPORTING iv\_domname    = 'ZDYS\_D\_FATTP'  
                                   iv\_domvalue\_l = <ls\_item>-fattp  
                         IMPORTING ev\_ddtext     = <ls\_item>-fattpt ).  
  
      " Marka tanımı  
      IF <ls\_item>-marka IS NOT INITIAL.  
        READ TABLE mt\_zdys\_t\_004 INTO DATA(ls\_zdys\_t\_004) WITH TABLE KEY marka = <ls\_item>-marka.  
        IF sy-subrc = 0.  
          <ls\_item>-markat = ls\_zdys\_t\_004-markat.  
          IF ls\_zdys\_t\_004-matkl IS NOT INITIAL.  
            COLLECT VALUE matkl\_struct( matkl = ls\_zdys\_t\_004-matkl ) INTO lt\_matkl.  
          ENDIF.  
        ENDIF.  
      ENDIF.  
  
      " Mal grubu tanımı  
      IF <ls\_item>-matkl IS NOT INITIAL.  
        READ TABLE lt\_t023t INTO DATA(ls\_t023t) WITH TABLE KEY matkl = <ls\_item>-matkl.  
        IF sy-subrc = 0.  
          <ls\_item>-wgbez = ls\_t023t-wgbez.  
        ENDIF.  
        COLLECT VALUE matkl\_struct( matkl = <ls\_item>-matkl ) INTO lt\_matkl.  
      ENDIF.  
  
      " Durum tanımı  
      get\_domvalue\_text( EXPORTING iv\_domname    = 'ZDYS\_D\_DURUM'  
                                   iv\_domvalue\_l = <ls\_item>-durum  
                         IMPORTING ev\_ddtext     = <ls\_item>-durumt ).  
  
      " Fatura türü  
      get\_domvalue\_text( EXPORTING iv\_domname    = 'ZDYS\_D\_FATUR'  
                                   iv\_domvalue\_l = <ls\_item>-fatur  
                         IMPORTING ev\_ddtext     = <ls\_item>-faturt ).  
  
      " E-imza süreci  
      get\_domvalue\_text( EXPORTING iv\_domname    = 'ZDYS\_D\_EIMZS'  
                                   iv\_domvalue\_l = <ls\_item>-eimzs  
                         IMPORTING ev\_ddtext     = <ls\_item>-eimzst ).  
  
      " E-imza durumu  
      IF <ls\_item>-eimzs IS NOT INITIAL.  
        get\_domvalue\_text( EXPORTING iv\_domname    = 'ZDYS\_D\_EIMZD'  
                                     iv\_domvalue\_l = <ls\_item>-eimzd  
                           IMPORTING ev\_ddtext     = <ls\_item>-eimzdt ).  
      ENDIF.  
  
      " Gider tipi tanımı  
      IF <ls\_item>-gdrtp IS NOT INITIAL.  
        READ TABLE mt\_zdys\_t\_007 INTO DATA(ls\_zdys\_t\_007) WITH TABLE KEY gdrtp = <ls\_item>-gdrtp.  
        IF sy-subrc = 0.  
          <ls\_item>-gdrtpt = ls\_zdys\_t\_007-gdrtpt.  
        ENDIF.  
      ENDIF.  
  
      zdys\_cl\_general=>get\_awkey\_values(  
        EXPORTING  
          iv\_awtyp     = <ls\_item>-awtyp  
          iv\_awkey     = <ls\_item>-awkey  
        IMPORTING  
          ev\_belnr\_mm  = DATA(lv\_belnr\_mm)  
          ev\_gjahr\_mm  = DATA(lv\_gjahr\_mm)  
          ev\_blgno     = <ls\_item>-blgno  
          ev\_yilno     = <ls\_item>-yilno ).  
      IF <ls\_item>-blgno IS INITIAL.  
        APPEND <ls\_item>-dysid TO lt\_dysid\_unmatched.  
      ELSE.  
        <ls\_item>-matched\_tutar = 'X'.  
        <ls\_item>-matched\_tarih = 'X'.  
      ENDIF.  
  
      " İade nedeni uzun açıklaması  
      IF <ls\_item>-iadenk IS NOT INITIAL.  
        lv\_name = <ls\_item>-dysid.  
        REFRESH lt\_lines.  
        CALL FUNCTION 'READ\_TEXT'  
          EXPORTING  
            id                      = 'Z002'  
            language                = sy-langu  
            name                    = lv\_name  
            object                  = zdys\_cl\_general=>mc\_tdobject  
          TABLES  
            lines                   = lt\_lines  
          EXCEPTIONS  
            id                      = 1  
            language                = 2  
            name                    = 3  
            not\_found               = 4  
            object                  = 5  
            reference\_check         = 6  
            wrong\_access\_to\_archive = 7  
            OTHERS                  = 8.  
        IF sy-subrc = 0 AND lt\_lines[] IS NOT INITIAL.  
          LOOP AT lt\_lines INTO DATA(ls\_lines).  
            IF <ls\_item>-iadent IS INITIAL.  
              <ls\_item>-iadent = ls\_lines-tdline.  
            ELSE.  
              CONCATENATE <ls\_item>-iadent ls\_lines-tdline  
                     INTO <ls\_item>-iadent SEPARATED BY space.  
            ENDIF.  
            CONDENSE <ls\_item>-iadent.  
          ENDLOOP.  
        ENDIF.  
      ENDIF.  
  
      " Sipariş numarası bilgisi (İlk sipariş no listede görüntülenir)  
      READ TABLE lt\_zdys\_t\_016 INTO DATA(ls\_zdys\_t\_016) WITH KEY dysid = <ls\_item>-dysid.  
      IF sy-subrc = 0.  
        <ls\_item>-sipno = ls\_zdys\_t\_016-sipno.  
      ENDIF.  
  
      " Son işlem yapan kullanıcı  
      READ TABLE lt\_zdys\_t\_002 TRANSPORTING NO FIELDS WITH KEY dysid = <ls\_item>-dysid.  
      IF sy-subrc = 0.  
        lv\_tabix = sy-tabix.  
        CLEAR lv\_islno.  
        LOOP AT lt\_zdys\_t\_002 INTO DATA(ls\_zdys\_t\_002) FROM lv\_tabix.  
          IF ls\_zdys\_t\_002-dysid = <ls\_item>-dysid.  
            IF ls\_zdys\_t\_002-islno > lv\_islno.  
              lv\_islno = ls\_zdys\_t\_002-islno.  
              <ls\_item>-uname\_last = ls\_zdys\_t\_002-uname.  
            ENDIF.  
          ELSE.  
            EXIT.  
          ENDIF.  
        ENDLOOP.  
      ENDIF.  
    ENDLOOP.  
  
    " Satın alma faturası ile eşleştirilememiş DYS faturalarını eşleştir  
    IF lt\_dysid\_unmatched[] IS NOT INITIAL.  
      zdys\_cl\_general=>match\_invoices( EXPORTING it\_dysid  = lt\_dysid\_unmatched  
                                       IMPORTING et\_data   = DATA(lt\_data) ).  
      LOOP AT lt\_data INTO DATA(ls\_data).  
        READ TABLE rt\_item ASSIGNING <ls\_item> WITH KEY dysid = ls\_data-dysid.  
        IF sy-subrc = 0.  
          <ls\_item>-blgno = ls\_data-belnr.  
          <ls\_item>-yilno = ls\_data-gjahr.  
          <ls\_item>-awkey = ls\_data-awkey.  
          <ls\_item>-awtyp = ls\_data-awtyp.  
          <ls\_item>-matched\_tutar = ls\_data-matched\_tutar.  
          <ls\_item>-matched\_tarih = ls\_data-matched\_tarih.  
        ENDIF.  
      ENDLOOP.  
    ENDIF.  
  
\*    " SAP fatura belge numarası filtreleme  
\*    IF s\_blgno[] IS NOT INITIAL.  
\*      DELETE rt\_item WHERE blgno NOT IN s\_blgno.  
\*    ENDIF.  
  
    LOOP AT rt\_item INTO ls\_item WHERE awtyp = 'RMRP'  
                                 AND awkey IS NOT INITIAL.  
      READ TABLE lt\_rseg TRANSPORTING NO FIELDS WITH KEY belnr = ls\_item-blgno  
                                                         gjahr = ls\_item-yilno.  
      IF sy-subrc <> 0.  
        INSERT VALUE #( belnr = ls\_item-blgno  
                        gjahr = ls\_item-yilno ) INTO TABLE lt\_rseg.  
      ENDIF.  
    ENDLOOP.  
    IF lt\_rseg[] IS NOT INITIAL.  
      SELECT rbkp~rbstat  
             rseg~belnr rseg~gjahr rseg~buzei rseg~ebeln rseg~ebelp  
             FROM rbkp  
             INNER JOIN rseg  
             ON rbkp~belnr = rseg~belnr  
             AND rbkp~gjahr = rseg~gjahr  
             INTO CORRESPONDING FIELDS OF TABLE lt\_rseg  
             FOR ALL ENTRIES IN lt\_rseg  
             WHERE rbkp~belnr = lt\_rseg-belnr  
             AND rbkp~gjahr = lt\_rseg-gjahr.  
    ENDIF.  
    LOOP AT lt\_rseg INTO DATA(ls\_rseg).  
      READ TABLE lt\_vbfa TRANSPORTING NO FIELDS WITH KEY vbeln = ls\_rseg-ebeln  
                                                         posnn = ls\_rseg-ebelp.  
      IF sy-subrc <> 0.  
        INSERT VALUE #( vbeln = ls\_rseg-ebeln  
                        posnn = ls\_rseg-ebelp  
                       ) INTO TABLE lt\_vbfa.  
      ENDIF.  
    ENDLOOP.  
    IF lt\_vbfa[] IS NOT INITIAL.  
      SELECT vbelv posnv vbeln posnn vbtyp\_n  
             FROM vbfa  
             INTO TABLE lt\_vbfa  
             FOR ALL ENTRIES IN lt\_vbfa  
             WHERE vbeln = lt\_vbfa-vbeln  
             AND posnn = lt\_vbfa-posnn  
             AND vbtyp\_n = 'V'.  
    ENDIF.  
    IF lt\_vbfa[] IS NOT INITIAL.  
      SELECT likp~faksk likp~bolnr  
             lips~vbeln lips~posnr lips~vgbel lips~vgpos lips~faksp lips~fksta  
             FROM likp  
             INNER JOIN lips  
             ON likp~vbeln = lips~vbeln  
             INTO CORRESPONDING FIELDS OF TABLE lt\_lips  
             FOR ALL ENTRIES IN lt\_vbfa  
             WHERE lips~vgbel = lt\_vbfa-vbelv  
             AND lips~vgpos = lt\_vbfa-posnv.  
    ENDIF.  
    IF lt\_lips[] IS NOT INITIAL.  
      SELECT vbrp~vbeln vbrp~posnr vbrp~vgbel vbrp~vgpos vbrp~vbeln  
             FROM vbrp  
             INNER JOIN vbrk  
             ON vbrp~vbeln = vbrk~vbeln  
             INTO TABLE lt\_vbrp  
             FOR ALL ENTRIES IN lt\_lips  
             WHERE vbrp~vgbel = lt\_lips-vbeln  
             AND vbrp~vgpos = lt\_lips-posnr  
             AND vbrk~sfakn = ''  
             AND vbrk~fksto = ''.  
    ENDIF.  
    IF lt\_vbrp[] IS NOT INITIAL.  
      SELECT \*  
             FROM nast  
             INTO TABLE lt\_nast  
             FOR ALL ENTRIES IN lt\_vbrp  
             WHERE kappl = 'V3'  
             AND objky = lt\_vbrp-objky.  
    ENDIF.  
  
    " Müdür ve yönetici bilgileri  
    LOOP AT lt\_matkl INTO DATA(ls\_matkl) WHERE matkl IS NOT INITIAL.  
      APPEND VALUE #( sign = 'I'  
                      option = 'EQ'  
                      low    = ls\_matkl-matkl  
                     ) TO lr\_matkl.  
    ENDLOOP.  
    CALL FUNCTION 'ZFI\_KES3\_MATKL\_DETAIL'  
      EXPORTING  
        ir\_matkl     = lr\_matkl  
      IMPORTING  
        et\_matkl\_det = lt\_matkl\_det.  
  
    LOOP AT rt\_item ASSIGNING <ls\_item>.  
      " Satış fatura durum bilgileri  
      CLEAR lv\_completed\_rmrp.  
      DO 1 TIMES.  
        CHECK <ls\_item>-awtyp = 'RMRP' AND <ls\_item>-awkey IS NOT INITIAL.  
        CLEAR: lv\_belnr\_mm, lv\_gjahr\_mm.  
        zdys\_cl\_general=>get\_awkey\_values( EXPORTING iv\_awtyp     = <ls\_item>-awtyp  
                                                     iv\_awkey     = <ls\_item>-awkey  
                                           IMPORTING ev\_belnr\_mm  = lv\_belnr\_mm  
                                                     ev\_gjahr\_mm  = lv\_gjahr\_mm ).  
        READ TABLE lt\_rseg INTO ls\_rseg WITH KEY belnr = lv\_belnr\_mm  
                                                 gjahr = lv\_gjahr\_mm.  
        CHECK sy-subrc = 0.  
        <ls\_item>-rbstat = ls\_rseg-rbstat.  
        READ TABLE lt\_vbfa INTO DATA(ls\_vbfa) WITH KEY vbeln = ls\_rseg-ebeln  
                                                       posnn = ls\_rseg-ebelp.  
        IF sy-subrc <> 0. " Satış siparişi bulunamadı  
          <ls\_item>-sfatd = '00'. " İlişkili değil  
          lv\_completed\_rmrp = 'X'.  
          EXIT.  
        ENDIF.  
  
        lv\_bolnr = ls\_rseg-belnr && `-` && ls\_rseg-gjahr.  
        READ TABLE lt\_lips INTO DATA(ls\_lips) WITH KEY vgbel = ls\_vbfa-vbelv  
                                                       vgpos = ls\_vbfa-posnv  
                                                       bolnr = lv\_bolnr.  
        IF sy-subrc = 0.  
          <ls\_item>-faksk = ls\_lips-faksk.  
          <ls\_item>-faksp = ls\_lips-faksp.  
          <ls\_item>-fksta = ls\_lips-fksta.  
        ELSE. " Siparişin teslimatı bulunamadı  
          <ls\_item>-sfatd = '00'. " İlişkili değil  
          lv\_completed\_rmrp = 'X'.  
          EXIT.  
        ENDIF.  
  
        " Teslimat üzerinde blokaj var mı kontrolü  
        IF <ls\_item>-faksk IS NOT INITIAL OR <ls\_item>-faksp IS NOT INITIAL.  
          <ls\_item>-sfatd = '00'. " İlişkili değil  
          lv\_completed\_rmrp = 'X'.  
          EXIT.  
        ENDIF.  
  
        " Satış fatutası var mı kontrolü  
        READ TABLE lt\_vbrp INTO DATA(ls\_vbrp) WITH KEY vgbel = ls\_lips-vbeln  
                                                       vgpos = ls\_lips-posnr.  
        IF sy-subrc <> 0.  
          <ls\_item>-sfatd = '01'. "  İşlenmedi  
          EXIT.  
        ENDIF.  
  
        " Çıktı işleme durumu  
        ls\_nast-vstat = '0'.  
        DO 1 TIMES.  
          READ TABLE lt\_nast INTO ls\_nast WITH KEY objky = ls\_vbrp-objky  
                                                   vstat = '1'. " İşlendi  
          CHECK sy-subrc <> 0.  
          READ TABLE lt\_nast INTO ls\_nast WITH KEY objky = ls\_vbrp-objky  
                                                   vstat = '2'. " Hatalı işlendi  
        ENDDO.  
        <ls\_item>-vstat = ls\_nast-vstat.  
        CASE <ls\_item>-vstat.  
          WHEN '0'. "  İşlenmedi  
            <ls\_item>-sfatd = '04'. " Çıktı işlenmedi  
          WHEN '1'. " İşlendi  
            lv\_completed\_rmrp = 'X'.  
            CASE <ls\_item>-fksta.  
              WHEN 'A'.  " İşlenmedi  
                <ls\_item>-sfatd = '01'.  
              WHEN 'B'.  " Kısmen işlendi  
                <ls\_item>-sfatd = '02'.  
              WHEN 'C'.  " Tamamen işlendi  
                <ls\_item>-sfatd = '03'.  
            ENDCASE.  
          WHEN '2'. "  Hatalı işlendi  
            <ls\_item>-sfatd = '05'. " Çıktı hatalı işlendi  
        ENDCASE.  
      ENDDO.  
  
      " Fatura tutarı 0 olan faturaların satış fatura durumu 'İlişkili değil' olarak gösterilir  
      " Manuel tamamlandıya çekilen faturaların durumu 'İlişkili değil' olarak gösterilir  
      IF <ls\_item>-fattu = 0 OR <ls\_item>-cmpmn = 'X'.  
        <ls\_item>-sfatd = '00'. " İlişkili değil  
      ENDIF.  
  
      get\_domvalue\_text( EXPORTING iv\_domname    = 'ZDYS\_D\_SFATD'  
                                   iv\_domvalue\_l = <ls\_item>-sfatd  
                         IMPORTING ev\_ddtext     = <ls\_item>-sfatdt ).  
      IF s\_sfatd[] IS NOT INITIAL AND <ls\_item>-sfatd NOT IN s\_sfatd.  
        DELETE rt\_item.  
        CONTINUE.  
      ENDIF.  
  
      " Fatura iş listesinde tamamlanan faturalar görüntülenmemeli  
      " Ancak satış ilişkili faturalarda DYS'nin durumu 'Başarıyla tamamlandı' olsa  
      " bile satış faturası tamamlanmadıysa(çıktı işleme dahil) ekrana getirilmeli!  
      IF p\_report = ''.  
        IF <ls\_item>-durum = '99'.  
          CASE <ls\_item>-awtyp.  
            WHEN 'RMRP'.  
              IF lv\_completed\_rmrp = 'X'.  
                DELETE rt\_item.  
                CONTINUE.  
              ENDIF.  
            WHEN OTHERS.  
              DELETE rt\_item.  
              CONTINUE.  
          ENDCASE.  
        ELSEIF <ls\_item>-durum IN zdys\_cl\_general=>get\_completed\_status( ).  
          DELETE rt\_item.  
          CONTINUE.  
        ENDIF.  
      ENDIF.  
  
      " Fatura belge numarasının eşleştirme durumuna göre renklendirilmesi  
      IF <ls\_item>-blgno IS NOT INITIAL AND  
         <ls\_item>-matched\_tutar = 'X' AND <ls\_item>-matched\_tarih = 'X'.  
        APPEND VALUE #( fname = 'BLGNO'         color-col = '5' ) TO <ls\_item>-cellcolor.  
        APPEND VALUE #( fname = 'MATCHED\_TUTAR' color-col = '5' ) TO <ls\_item>-cellcolor.  
        APPEND VALUE #( fname = 'MATCHED\_TARIH' color-col = '5' ) TO <ls\_item>-cellcolor.  
      ELSEIF <ls\_item>-blgno IS INITIAL AND <ls\_item>-yilno IS INITIAL AND  
               <ls\_item>-matched\_tutar = '' AND <ls\_item>-matched\_tarih = ''.  
        APPEND VALUE #( fname = 'BLGNO'         color-col = '3' ) TO <ls\_item>-cellcolor.  
        APPEND VALUE #( fname = 'MATCHED\_TUTAR' color-col = '3' ) TO <ls\_item>-cellcolor.  
        APPEND VALUE #( fname = 'MATCHED\_TARIH' color-col = '3' ) TO <ls\_item>-cellcolor.  
      ELSE.  
        APPEND VALUE #( fname = 'BLGNO' color-col = '6' ) TO <ls\_item>-cellcolor.  
        IF <ls\_item>-matched\_tutar = 'X'.  
          APPEND VALUE #( fname = 'MATCHED\_TUTAR' color-col = '5' ) TO <ls\_item>-cellcolor.  
        ELSE.  
          APPEND VALUE #( fname = 'MATCHED\_TUTAR' color-col = '6' ) TO <ls\_item>-cellcolor.  
        ENDIF.  
        IF <ls\_item>-matched\_tarih = 'X'.  
          APPEND VALUE #( fname = 'MATCHED\_TARIH' color-col = '5' ) TO <ls\_item>-cellcolor.  
        ELSE.  
          APPEND VALUE #( fname = 'MATCHED\_TARIH' color-col = '6' ) TO <ls\_item>-cellcolor.  
        ENDIF.  
      ENDIF.  
  
      " İşlem tipi  
      get\_domvalue\_text( EXPORTING iv\_domname    = 'ZDYS\_D\_ISLTP'  
                                   iv\_domvalue\_l = <ls\_item>-isltp  
                         IMPORTING ev\_ddtext     = <ls\_item>-isltpt ).  
      " Alt işlem tipi  
      get\_domvalue\_text( EXPORTING iv\_domname    = 'ZDYS\_D\_ISLTP\_ALT'  
                                   iv\_domvalue\_l = <ls\_item>-isltp\_alt  
                         IMPORTING ev\_ddtext     = <ls\_item>-isltp\_altt ).  
      " Fatura işlem tipi  
      get\_domvalue\_text( EXPORTING iv\_domname    = 'ZDYS\_D\_ISLTP\_FAT'  
                                   iv\_domvalue\_l = <ls\_item>-isltp\_fat  
                         IMPORTING ev\_ddtext     = <ls\_item>-isltp\_fatt ).  
  
      " Açıklamalar butonu  
      <ls\_item>-btn\_txt = icon\_annotation.  
      INSERT VALUE #( fieldname = 'BTN\_TXT'  
                      style     = cl\_gui\_alv\_grid=>mc\_style\_button ) INTO TABLE <ls\_item>-cellstyle.  
  
      " Onay butonlarını ekleme koşulu  
      CLEAR lv\_conf\_chk.  
      IF ( <ls\_item>-rol = 'GOS' OR <ls\_item>-rol = 'FAOS') AND  
           zcl\_ot\_tool=>check\_appr\_hist( <ls\_item>-dysid ) = 0.  
        lv\_conf\_chk = 'X'.  
      ELSEIF <ls\_item>-rol = 'FOS'.  
        BREAK-POINT ID zdys\_rebilling.  
        IF sy-sysid = 'ZHP'.  
          SELECT SINGLE aprid  
                 FROM zmmgen\_t\_001  
                 INTO @DATA(lv\_aprid)  
                 WHERE dysid = @<ls\_item>-dysid.  
        ELSE.  
          SELECT SINGLE aprid  
                 FROM zdys\_t\_031  
                 INTO lv\_aprid  
                 WHERE dysid = <ls\_item>-dysid.  
        ENDIF.  
        IF sy-subrc = 0 AND lv\_aprid IS NOT INITIAL.  
          lv\_conf\_chk = 'X'.  
        ENDIF.  
      ENDIF.  
  
      IF lv\_conf\_chk = 'X'.  
        " Onay hiyerarşi butonu  
        <ls\_item>-btn\_onyh = icon\_dependents.  
        INSERT VALUE #( fieldname = 'BTN\_ONYH'  
                        style     = cl\_gui\_alv\_grid=>mc\_style\_button ) INTO TABLE <ls\_item>-cellstyle.  
        " Onay tarihçe butonu  
        <ls\_item>-btn\_onyt = icon\_booking\_ok.  
        INSERT VALUE #( fieldname = 'BTN\_ONYT'  
                        style     = cl\_gui\_alv\_grid=>mc\_style\_button ) INTO TABLE <ls\_item>-cellstyle.  
      ENDIF.  
  
      " GIB XML butonu  
      <ls\_item>-btn\_xml = icon\_xml\_doc.  
      INSERT VALUE #( fieldname = 'BTN\_XML'  
                      style     = cl\_gui\_alv\_grid=>mc\_style\_button ) INTO TABLE <ls\_item>-cellstyle.  
  
      " Müdür / yönetici bilgisi  
      CLEAR ls\_matkl\_det.  
      IF <ls\_item>-matkl IS NOT INITIAL.  
        READ TABLE lt\_matkl\_det INTO ls\_matkl\_det WITH KEY matkl = <ls\_item>-matkl.  
      ELSEIF <ls\_item>-marka IS NOT INITIAL.  
        READ TABLE mt\_zdys\_t\_004 INTO ls\_zdys\_t\_004 WITH TABLE KEY marka = <ls\_item>-marka.  
        IF sy-subrc = 0 AND ls\_zdys\_t\_004-matkl IS NOT INITIAL.  
          READ TABLE lt\_matkl\_det INTO ls\_matkl\_det WITH KEY matkl = ls\_zdys\_t\_004-matkl.  
        ENDIF.  
      ENDIF.  
      <ls\_item>-mudur = ls\_matkl\_det-mudur.  
      IF s\_mudur[] IS NOT INITIAL AND <ls\_item>-mudur NOT IN s\_mudur.  
        DELETE rt\_item.  
        CONTINUE.  
      ENDIF.  
  
      " Ulusal para biriminde KDV'siz tutar değeri  
      <ls\_item>-hwaer = ms\_t001-waers.  
      IF <ls\_item>-waers = <ls\_item>-hwaer.  
        <ls\_item>-ksztu\_upb = <ls\_item>-ksztu.  
      ELSE.  
        CALL FUNCTION 'CONVERT\_TO\_LOCAL\_CURRENCY'  
          EXPORTING  
            date             = sy-datum  
            foreign\_amount   = <ls\_item>-ksztu  
            foreign\_currency = <ls\_item>-waers  
            local\_currency   = <ls\_item>-hwaer  
          IMPORTING  
            local\_amount     = <ls\_item>-ksztu\_upb  
          EXCEPTIONS  
            no\_rate\_found    = 1  
            overflow         = 2  
            no\_factors\_found = 3  
            no\_spread\_found  = 4  
            derived\_2\_times  = 5  
            OTHERS           = 6.  
      ENDIF.  
    ENDLOOP.  
  
    " Ek kriterlerine göre listeyi filtreleme (Sadece ZDYS0009 için)  
    IF p\_report = 'X'.  
      " Satın alma faturası eki  
      CASE 'X'.  
        WHEN p\_saoln. " Eki olanlar  
          DELETE rt\_item WHERE btn\_fatek IS INITIAL.  
        WHEN p\_saolm. " Eki olmayanlar  
          DELETE rt\_item WHERE btn\_fatek IS NOT INITIAL.  
      ENDCASE.  
    ENDIF.  
  ENDMETHOD.  
  
  METHOD get\_data\_oub.  
    DATA lt\_rsparams TYPE TABLE OF rsparams.  
  
    LOOP AT s\_fatno INTO DATA(ls\_fatno).  
      APPEND VALUE #( selname = 'S\_FATNO'  
                      kind    = 'S'  
                      sign    = ls\_fatno-sign  
                      option  = ls\_fatno-option  
                      low     = ls\_fatno-low  
                      high    = ls\_fatno-high  
                     ) TO lt\_rsparams.  
    ENDLOOP.  
    LOOP AT s\_blgno INTO DATA(ls\_blgno).  
      APPEND VALUE #( selname = 'S\_BLGNO'  
                      kind    = 'S'  
                      sign    = ls\_blgno-sign  
                      option  = ls\_blgno-option  
                      low     = ls\_blgno-low  
                      high    = ls\_blgno-high  
                     ) TO lt\_rsparams.  
    ENDLOOP.  
    LOOP AT s\_matkl INTO DATA(ls\_matkl).  
      APPEND VALUE #( selname = 'S\_MATKL'  
                      kind    = 'S'  
                      sign    = ls\_matkl-sign  
                      option  = ls\_matkl-option  
                      low     = ls\_matkl-low  
                      high    = ls\_matkl-high  
                     ) TO lt\_rsparams.  
    ENDLOOP.  
    LOOP AT s\_partn INTO DATA(ls\_partn).  
      APPEND VALUE #( selname = 'S\_PARTN'  
                      kind    = 'S'  
                      sign    = ls\_partn-sign  
                      option  = ls\_partn-option  
                      low     = ls\_partn-low  
                      high    = ls\_partn-high  
                     ) TO lt\_rsparams.  
    ENDLOOP.  
    LOOP AT s\_fattr INTO DATA(ls\_fattr).  
      APPEND VALUE #( selname = 'S\_FATTR'  
                      kind    = 'S'  
                      sign    = ls\_fattr-sign  
                      option  = ls\_fattr-option  
                      low     = ls\_fattr-low  
                      high    = ls\_fattr-high  
                     ) TO lt\_rsparams.  
    ENDLOOP.  
    LOOP AT s\_waers INTO DATA(ls\_waers).  
      APPEND VALUE #( selname = 'S\_WAERS'  
                      kind    = 'S'  
                      sign    = ls\_waers-sign  
                      option  = ls\_waers-option  
                      low     = ls\_waers-low  
                      high    = ls\_waers-high  
                     ) TO lt\_rsparams.  
    ENDLOOP.  
    LOOP AT s\_crdat INTO DATA(ls\_crdat).  
      APPEND VALUE #( selname = 'S\_CRDAT'  
                      kind    = 'S'  
                      sign    = ls\_crdat-sign  
                      option  = ls\_crdat-option  
                      low     = ls\_crdat-low  
                      high    = ls\_crdat-high  
                     ) TO lt\_rsparams.  
    ENDLOOP.  
    LOOP AT s\_crnam INTO DATA(ls\_crnam).  
      APPEND VALUE #( selname = 'S\_CRNAM'  
                      kind    = 'S'  
                      sign    = ls\_crnam-sign  
                      option  = ls\_crnam-option  
                      low     = ls\_crnam-low  
                      high    = ls\_crnam-high  
                     ) TO lt\_rsparams.  
    ENDLOOP.  
    APPEND VALUE #( selname = 'P\_SDFAT'  
                    kind    = 'P'  
                    sign    = 'I'  
                    option  = 'EQ'  
                    low     = 'X'  
                   ) TO lt\_rsparams.  
    APPEND VALUE #( selname = 'P\_MMFAT'  
                    kind    = 'P'  
                    sign    = 'I'  
                    option  = 'EQ'  
                    low     = ''  
                   ) TO lt\_rsparams.  
  
    SUBMIT zzerotgen\_r0007 WITH SELECTION-TABLE lt\_rsparams  
                           AND RETURN.  
  ENDMETHOD.  
  
  METHOD get\_domvalue\_text.  
    READ TABLE mt\_dd07t INTO DATA(ls\_dd07t) WITH KEY domname = iv\_domname  
                                                     domvalue\_l = iv\_domvalue\_l.  
    IF sy-subrc <> 0.  
      SELECT \*  
             FROM dd07t  
             INTO ls\_dd07t  
             UP TO 1 ROWS  
             WHERE domname = iv\_domname  
             AND ddlanguage = sy-langu  
             AND domvalue\_l = iv\_domvalue\_l  
             ORDER BY valpos.  
        INSERT ls\_dd07t INTO TABLE mt\_dd07t.  
        EXIT.  
      ENDSELECT.  
    ENDIF.  
    ev\_ddtext = ls\_dd07t-ddtext.  
  ENDMETHOD.  
  
  METHOD check\_input\_partner.  
    CLEAR cv\_partnert.  
    IF cv\_partner IS NOT INITIAL.  
      SELECT SINGLE name\_org1,name\_org2  
             FROM but000  
             INTO (@DATA(lv\_name\_org1), @DATA(lv\_name\_org2) )  
             WHERE partner = @cv\_partner.  
      IF sy-subrc = 0.  
        cv\_partnert = lv\_name\_org1 && ` ` && lv\_name\_org2.  
      ELSE.  
        MESSAGE i013 DISPLAY LIKE 'E' WITH cv\_partner.  
        CLEAR cv\_partner.  
      ENDIF.  
    ENDIF.  
  ENDMETHOD.  
  
  METHOD check\_input\_marka.  
    CLEAR: cv\_markat.  
    IF cv\_marka IS NOT INITIAL.  
      SELECT SINGLE markat  
             INTO cv\_markat  
             FROM zdys\_t\_004  
             WHERE marka = cv\_marka.  
      IF sy-subrc <> 0.  
        MESSAGE i014 DISPLAY LIKE 'E' WITH cv\_marka.  
        CLEAR cv\_marka.  
      ENDIF.  
    ENDIF.  
  ENDMETHOD.  
  
  METHOD check\_input\_matkl.  
    CLEAR cv\_wgbez.  
    IF cv\_matkl IS NOT INITIAL.  
      SELECT SINGLE matkl  
             INTO @DATA(lv\_matkl)  
             FROM t023  
             WHERE matkl = @cv\_matkl.  
      IF sy-subrc = 0.  
        SELECT SINGLE wgbez  
               INTO cv\_wgbez  
               FROM t023t  
               WHERE spras = sy-langu  
               AND matkl = cv\_matkl.  
      ELSE.  
        MESSAGE i015 DISPLAY LIKE 'E' WITH cv\_matkl.  
        CLEAR cv\_matkl.  
      ENDIF.  
    ENDIF.  
  ENDMETHOD.  
  
  METHOD check\_stcd2\_inconsistency.  
    CHECK iv\_partner IS NOT INITIAL.  
    CHECK iv\_stcd2 IS NOT INITIAL.  
  
    SELECT SINGLE taxnum  
           FROM dfkkbptaxnum  
           INTO @DATA(lv\_taxnum)  
           WHERE partner = @iv\_partner  
           AND taxtype = 'TR2'.  
    IF sy-subrc = 0 AND iv\_stcd2 = lv\_taxnum.  
      rv\_matched = 'X'.  
    ELSE.  
      zcl\_util\_001=>add\_message(  
        EXPORTING  
          i\_msgty   = 'E'  
          i\_msgid   = mc\_msgid  
          i\_msgno   = '037'  
          i\_msgv1   = iv\_partner  
          i\_msgv2   = lv\_taxnum  
          i\_msgv3   = iv\_stcd2  
          i\_msgv4   = iv\_dysid  
        IMPORTING  
          et\_return = et\_return ).  
      IF iv\_popup\_msg = 'X'.  
        MESSAGE i037 WITH iv\_partner lv\_taxnum iv\_stcd2 iv\_dysid DISPLAY LIKE 'E'.  
      ENDIF.  
    ENDIF.  
  ENDMETHOD.  
  
  METHOD check\_auth\_gdr.  
    DATA: lr\_uname TYPE RANGE OF sy-uname.  
  
    " İş akışı raporunda bu eleme yapılmaz  
    CHECK p\_report = ''.  
  
    " Süper kullanıcı ise eleme yapılmaz (tam yetki)  
    CHECK ms\_zdys\_t\_010-uname IS INITIAL.  
  
    " Login olan kullanıcı  
    APPEND VALUE #( sign = 'I' option = 'EQ' low = sy-uname ) TO lr\_uname.  
  
    " Login olan kullanıcının vekalet ettiği kullanıcılar  
    LOOP AT mt\_zdys\_t\_027 INTO DATA(ls\_zdys\_t\_027).  
      APPEND VALUE #( sign = 'I' option = 'EQ' low = ls\_zdys\_t\_027-vkedl ) TO lr\_uname.  
    ENDLOOP.  
  
    " Kullanıcının hepsinin görüntüleme(DYS tam vekalet) yetkisi var mı  
    " Vekalet edilen kullanıcı(VKEDL) boş ise tam DYS yetkisi var demektir  
    READ TABLE mt\_zdys\_t\_027 TRANSPORTING NO FIELDS WITH KEY vkedl = ''.  
    IF sy-subrc = 0.  
      EXIT.  
    ENDIF.  
  
    " Sadece fatura sorumlularının ilgili gider faturalarını görüntüleyebilmesi  
    " Süper kullanıcı hepsini görüntüler  
    " FSA rolünde FSA yetkisine sahip olanlar görüntüleyebilir  
    " GOS rolünde istisna tablosundaki kullanıcılar ve fatura sorumlusu görüntüleyebilir  
    " FAOS rolünde FAOS yetkisi olan kullanıcılar ve fatura sorumlusu görüntüleyebilir  
    LOOP AT ct\_zdys\_t\_001 INTO DATA(ls\_zdys\_t\_001) WHERE isaks = 'GDR'  
                                                   AND rol <> 'FAOS'  
                                                   AND fsorm NOT IN lr\_uname.  
      CASE ls\_zdys\_t\_001-rol.  
        WHEN 'FSA'.  
          IF ls\_zdys\_t\_001-fsorm IS NOT INITIAL.  
            DELETE ct\_zdys\_t\_001.  
          ENDIF.  
        WHEN OTHERS.  
          DELETE ct\_zdys\_t\_001.  
      ENDCASE.  
    ENDLOOP.  
  ENDMETHOD.  
  
  METHOD pbo\_0100\_screen.  
    DATA: lt\_fcode  TYPE TABLE OF sy-ucomm,  
          lt\_sort   TYPE lvc\_t\_sort,  
          lv\_tittle TYPE text50.  
  
    IF p\_report = 'X'.  
      APPEND '&EXEC\_PROC' TO lt\_fcode.  
      APPEND '&UPDATEINV' TO lt\_fcode.  
      APPEND '&MATCH\_INV' TO lt\_fcode.  
      APPEND '&UPD\_EIMZA' TO lt\_fcode.  
      APPEND '&RUNEXPINV' TO lt\_fcode.  
      APPEND '&RETREASON' TO lt\_fcode.  
      APPEND '&UPDATE\_MD' TO lt\_fcode.  
      APPEND '&RETRNAUTO' TO lt\_fcode.  
      APPEND '&CALL\_YNST' TO lt\_fcode.  
      APPEND '&MATCH\_DYS' TO lt\_fcode.  
    ELSE.  
      APPEND '&DSPEXPINV' TO lt\_fcode.  
    ENDIF.  
  
    DATA(lv\_row\_count) = lines( gt\_item ).  
    WRITE lv\_row\_count TO lv\_tittle LEFT-JUSTIFIED.  
    lv\_tittle = TEXT-003 && ` ` && lv\_tittle.  
    SET PF-STATUS '0100' EXCLUDING lt\_fcode.  
    IF p\_report = 'X'.  
      SET TITLEBAR '0100' WITH TEXT-032 lv\_tittle.  
    ELSE.  
      SET TITLEBAR '0100' WITH TEXT-031 lv\_tittle.  
    ENDIF.  
  
    IF go\_dock\_cont\_0100 IS INITIAL.  
      CREATE OBJECT go\_dock\_cont\_0100  
        EXPORTING  
          repid     = sy-repid  
          dynnr     = '100'  
          extension = 99999.  
      CREATE OBJECT go\_alv\_grid\_0100  
        EXPORTING  
          i\_parent = go\_dock\_cont\_0100.  
  
      get\_fieldcatalog( EXPORTING iv\_alv\_name = '0100'  
                        IMPORTING et\_fieldcat = DATA(lt\_fieldcat) ).  
      get\_layout( EXPORTING iv\_alv\_name = '0100'  
                  IMPORTING es\_layout = DATA(ls\_layout) ).  
  
      APPEND VALUE #( spos      = 1  
                      fieldname = 'DYSID'  
                      up        = 'X'  
                     ) TO lt\_sort.  
  
      CALL METHOD go\_alv\_grid\_0100->set\_table\_for\_first\_display  
        EXPORTING  
          is\_layout                     = ls\_layout  
          i\_bypassing\_buffer            = 'X'  
          i\_save                        = 'A'  
          i\_default                     = 'X'  
          is\_variant                    = ms\_variant  
        CHANGING  
          it\_outtab                     = gt\_item  
          it\_fieldcatalog               = lt\_fieldcat  
          it\_sort                       = lt\_sort  
        EXCEPTIONS  
          invalid\_parameter\_combination = 1  
          program\_error                 = 2  
          too\_many\_lines                = 3  
          OTHERS                        = 4.  
      IF sy-subrc NE 0.  
        MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno  
              WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
      ENDIF.  
  
      CALL METHOD go\_alv\_grid\_0100->register\_edit\_event  
        EXPORTING  
          i\_event\_id = cl\_gui\_alv\_grid=>mc\_evt\_modified  
        EXCEPTIONS  
          error      = 1  
          OTHERS     = 2.  
      set\_event\_handlers( ).  
      IF go\_alv\_grid\_0100 IS NOT INITIAL.  
        go\_alv\_grid\_0100->set\_toolbar\_interactive( ).  
      ENDIF.  
    ELSE.  
      refresh\_0100\_screen( ).  
    ENDIF.  
  
  ENDMETHOD.  
  
  METHOD pai\_0100\_screen.  
  
    REFRESH mt\_dysid\_selected.  
  
    CASE gv\_ucomm\_0100.  
      WHEN '&F01' OR '&F02'.  
        LEAVE TO SCREEN 0.  
  
      WHEN '&EXEC\_PROC'.  " İşlemi yürüt  
        execute\_process( ).  
  
      WHEN '&REFRESH'.    " Listeyi yenile  
        refresh\_list( ).  
  
      WHEN '&LIST\_DETA'.  " İşlem detayları  
        list\_process\_details( ).  
  
      WHEN '&UPDATEINV'.  " Faturayı güncelle  
        go\_alv\_grid\_0100->get\_selected\_rows( IMPORTING et\_row\_no = DATA(lt\_row\_no) ).  
        IF lines( lt\_row\_no ) = 1.  
          update\_invoice\_data\_single( ).  
        ELSE.  
          update\_invoice\_data\_multi( ).  
        ENDIF.  
  
      WHEN '&RUNEXPINV'. " Gider faturası düzenle  
        run\_expense\_invoice( ).  
  
      WHEN '&DSPEXPINV'.  " Gider faturası görüntüle  
        run\_expense\_invoice( iv\_dispmode = 'X' ).  
  
      WHEN '&UPD\_EIMZA'.  " E-imza düzenle  
        run\_esignature\_process( ).  
  
      WHEN '&ATTACHMEN'.  " Ek işlemleri  
        display\_attachments( ).  
  
      WHEN '&WORKFLOW'.   " İş akışı  
        workflow\_process( ).  
  
      WHEN '&RETREASON'.  " İade nedeni  
        IF run\_return\_reason( ) = ''.  
          REFRESH mt\_dysid\_selected.  
        ENDIF.  
  
      WHEN '&RETRNAUTO'.  " Otomatik iade  
        IF run\_auto\_return( ) = ''.  
          REFRESH mt\_dysid\_selected.  
        ENDIF.  
  
      WHEN '&UPDATE\_MD'.  " Ana veri güncelle  
        IF run\_update\_md( ) = ''.  
          REFRESH mt\_dysid\_selected.  
        ENDIF.  
  
      WHEN '&CALL\_YNST'.  " Fatura yansıtma  
        IF call\_rebilling\_screen( ) = ''.  
          REFRESH mt\_dysid\_selected.  
        ENDIF.  
  
      WHEN '&DOWNL\_ZIP'.  " Toplu indir  
        download\_as\_zip( ).  
  
      WHEN '&MATCH\_DYS'.  " Faturayı eşleştir  
        match\_dys( ).  
    ENDCASE.  
    CLEAR gv\_ucomm\_0100.  
  
    refresh\_data\_inb( ).  
  
  ENDMETHOD.  
  
  METHOD refresh\_0100\_screen.  
    DATA ls\_stable TYPE lvc\_s\_stbl.  
    get\_layout( EXPORTING iv\_alv\_name = '0100'  
                IMPORTING es\_layout = DATA(ls\_layout) ).  
    go\_alv\_grid\_0100->set\_frontend\_layout( is\_layout = ls\_layout ).  
    ls\_stable-row = 'X'.  
    ls\_stable-col = 'X'.  
    go\_alv\_grid\_0100->refresh\_table\_display( EXPORTING is\_stable = ls\_stable ).  
  ENDMETHOD.  
  
  METHOD pbo\_0200\_screen.  
    DATA: lt\_sort    TYPE lvc\_t\_sort,  
          ls\_variant TYPE disvariant.  
  
    SET PF-STATUS '0200'.  
    SET TITLEBAR '0200' WITH ms\_item-fatno ms\_item-dysid.  
  
    IF go\_dock\_cont\_0200 IS INITIAL.  
      CREATE OBJECT go\_dock\_cont\_0200  
        EXPORTING  
          repid     = sy-repid  
          dynnr     = '200'  
          extension = 99999.  
      CREATE OBJECT go\_alv\_grid\_0200  
        EXPORTING  
          i\_parent = go\_dock\_cont\_0200.  
  
      get\_fieldcatalog( EXPORTING iv\_alv\_name = '0200'  
                        IMPORTING et\_fieldcat = DATA(lt\_fieldcat) ).  
      get\_layout( EXPORTING iv\_alv\_name = '0200'  
                  IMPORTING es\_layout = DATA(ls\_layout) ).  
      ls\_variant-report = sy-repid.  
      ls\_variant-handle = 'H2'.  
  
      CALL METHOD go\_alv\_grid\_0200->set\_table\_for\_first\_display  
        EXPORTING  
          is\_layout                     = ls\_layout  
          i\_bypassing\_buffer            = 'X'  
          i\_save                        = 'A'  
          i\_default                     = 'X'  
          is\_variant                    = ls\_variant  
        CHANGING  
          it\_outtab                     = mt\_detail  
          it\_fieldcatalog               = lt\_fieldcat  
          it\_sort                       = lt\_sort  
        EXCEPTIONS  
          invalid\_parameter\_combination = 1  
          program\_error                 = 2  
          too\_many\_lines                = 3  
          OTHERS                        = 4.  
      IF sy-subrc NE 0.  
        MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno  
              WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
      ENDIF.  
  
      CALL METHOD go\_alv\_grid\_0200->register\_edit\_event  
        EXPORTING  
          i\_event\_id = cl\_gui\_alv\_grid=>mc\_evt\_modified  
        EXCEPTIONS  
          error      = 1  
          OTHERS     = 2.  
      set\_event\_handlers( ).  
      IF go\_alv\_grid\_0200 IS NOT INITIAL.  
        go\_alv\_grid\_0200->set\_toolbar\_interactive( ).  
      ENDIF.  
    ELSE.  
      refresh\_0200\_screen( ).  
    ENDIF.  
  ENDMETHOD.  
  
  METHOD pai\_0200\_screen.  
    CASE gv\_ucomm\_0200.  
      WHEN '&F01' OR '&F02'.  
        LEAVE TO SCREEN 0.  
    ENDCASE.  
    CLEAR gv\_ucomm\_0200.  
  ENDMETHOD.  
  
  METHOD refresh\_0200\_screen.  
    get\_layout( EXPORTING iv\_alv\_name = '0200'  
                IMPORTING es\_layout = DATA(ls\_layout) ).  
    go\_alv\_grid\_0200->set\_frontend\_layout( is\_layout = ls\_layout ).  
    go\_alv\_grid\_0200->refresh\_table\_display( ).  
  ENDMETHOD.  
  
  METHOD pbo\_0300\_screen.  
    DATA: lt\_toolbar\_excluding TYPE ui\_functions,  
          lv\_editable          TYPE xfeld,  
          lt\_fcode             TYPE TABLE OF sy-ucomm,  
          lv\_fname             TYPE char50.  
  
    CONSTANTS: lc\_key\_val TYPE char08 VALUE 'GS\_0300-'.  
  
    DATA(lv\_key\_len) = strlen( lc\_key\_val ).  
  
    IF mt\_zdys\_t\_006[] IS NOT INITIAL AND  
       zdys\_cl\_general=>check\_if\_inv\_completed( iv\_durum = gs\_0300-durum ) = ''.  
      LOOP AT SCREEN.  
        IF strlen( screen-name ) > lv\_key\_len AND screen-name(lv\_key\_len) = lc\_key\_val.  
          lv\_fname = screen-name+lv\_key\_len(\*).  
          READ TABLE mt\_zdys\_t\_006 TRANSPORTING NO FIELDS WITH KEY fname = lv\_fname.  
          IF sy-subrc = 0.  
            lv\_editable = 'X'.  
            screen-input = 1.  
            MODIFY SCREEN.  
          ENDIF.  
        ENDIF.  
      ENDLOOP.  
    ENDIF.  
  
    " Değiştirilebilir fatura verisine yetki yok ise 'Kaydet' butonu gizlenir  
    IF lv\_editable = ''.  
      APPEND '&SAVE' TO lt\_fcode.  
    ENDIF.  
  
    SET PF-STATUS '0300' EXCLUDING lt\_fcode.  
    SET TITLEBAR '0300' WITH gs\_0300-fatno gs\_0300-dysid.  
  
    IF go\_cust\_cont\_0300 IS INITIAL.  
      CREATE OBJECT go\_cust\_cont\_0300  
        EXPORTING  
          container\_name = 'CC\_ORD'.  
      CREATE OBJECT go\_alv\_grid\_0300  
        EXPORTING  
          i\_parent = go\_cust\_cont\_0300.  
      get\_fieldcatalog( EXPORTING iv\_alv\_name = '0300'  
                        IMPORTING et\_fieldcat = DATA(lt\_fieldcat) ).  
      get\_layout( EXPORTING iv\_alv\_name = '0300'  
                  IMPORTING es\_layout = DATA(ls\_layout) ).  
      APPEND cl\_gui\_alv\_grid=>mc\_mb\_variant   TO lt\_toolbar\_excluding.  
      APPEND cl\_gui\_alv\_grid=>mc\_mb\_sum       TO lt\_toolbar\_excluding.  
      APPEND cl\_gui\_alv\_grid=>mc\_mb\_filter    TO lt\_toolbar\_excluding.  
      APPEND cl\_gui\_alv\_grid=>mc\_mb\_view      TO lt\_toolbar\_excluding.  
      APPEND cl\_gui\_alv\_grid=>mc\_fc\_detail    TO lt\_toolbar\_excluding.  
      APPEND cl\_gui\_alv\_grid=>mc\_fc\_print     TO lt\_toolbar\_excluding.  
      APPEND cl\_gui\_alv\_grid=>mc\_fc\_info      TO lt\_toolbar\_excluding.  
      APPEND cl\_gui\_alv\_grid=>mc\_mb\_export    TO lt\_toolbar\_excluding.  
      APPEND cl\_gui\_alv\_grid=>mc\_fc\_find\_more TO lt\_toolbar\_excluding.  
  
      CALL METHOD go\_alv\_grid\_0300->set\_table\_for\_first\_display  
        EXPORTING  
          is\_layout                     = ls\_layout  
          it\_toolbar\_excluding          = lt\_toolbar\_excluding  
          i\_bypassing\_buffer            = 'X'  
        CHANGING  
          it\_outtab                     = mt\_zdys\_t\_016  
          it\_fieldcatalog               = lt\_fieldcat  
        EXCEPTIONS  
          invalid\_parameter\_combination = 1  
          program\_error                 = 2  
          too\_many\_lines                = 3  
          OTHERS                        = 4.  
      IF sy-subrc NE 0.  
        MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno  
              WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
      ENDIF.  
  
      CALL METHOD go\_alv\_grid\_0300->register\_edit\_event  
        EXPORTING  
          i\_event\_id = cl\_gui\_alv\_grid=>mc\_evt\_modified  
        EXCEPTIONS  
          error      = 1  
          OTHERS     = 2.  
      set\_event\_handlers( ).  
      IF go\_alv\_grid\_0300 IS NOT INITIAL.  
        go\_alv\_grid\_0300->set\_toolbar\_interactive( ).  
      ENDIF.  
    ELSE.  
      refresh\_0300\_screen( ).  
    ENDIF.  
  
    LOOP AT SCREEN.  
      CASE screen-name.  
        WHEN 'GS\_0300-KURSFM' OR '\*GS\_0300-KURSFM'.  
          IF gs\_0300-isaks = 'GDR'.  
            screen-active = 1.  
          ELSE.  
            screen-active = 0.  
          ENDIF.  
          MODIFY SCREEN.  
      ENDCASE.  
    ENDLOOP.  
  ENDMETHOD.  
  
  METHOD pai\_0300\_screen.  
    CASE gv\_ucomm\_0300.  
      WHEN '&F01' OR '&F02'.  
        LEAVE TO SCREEN 0.  
      WHEN '&SAVE'.  
        save\_screen\_0300( ).  
        IF gs\_0300-leave\_screen = 'X'.  
          LEAVE TO SCREEN 0.  
        ENDIF.  
    ENDCASE.  
    CLEAR gv\_ucomm\_0300.  
  ENDMETHOD.  
  
  METHOD refresh\_0300\_screen.  
    get\_layout( EXPORTING iv\_alv\_name = '0300'  
                IMPORTING es\_layout = DATA(ls\_layout) ).  
    go\_alv\_grid\_0300->set\_frontend\_layout( is\_layout = ls\_layout ).  
    go\_alv\_grid\_0300->refresh\_table\_display( ).  
  ENDMETHOD.  
  
  METHOD pbo\_0400\_screen.  
    DATA: lt\_values TYPE vrm\_values,  
          lt\_dd07t  TYPE TABLE OF dd07t,  
          ls\_dd07t  TYPE dd07t.  
  
    " İşlem tipi değeri  
    REFRESH lt\_values.  
    IF mt\_zdys\_t\_021[] IS NOT INITIAL.  
      SELECT \*  
             FROM dd07t  
             INTO TABLE lt\_dd07t  
             WHERE domname = 'ZDYS\_D\_ISLTP'  
             AND ddlanguage = sy-langu.  
    ENDIF.  
    LOOP AT mt\_zdys\_t\_021 INTO DATA(ls\_zdys\_t\_021).  
      CLEAR ls\_dd07t.  
      READ TABLE lt\_dd07t INTO ls\_dd07t WITH KEY domvalue\_l = ls\_zdys\_t\_021-isltp.  
      APPEND VALUE #( key  = ls\_dd07t-domvalue\_l  
                      text = ls\_dd07t-ddtext ) TO lt\_values.  
    ENDLOOP.  
    CALL FUNCTION 'VRM\_SET\_VALUES'  
      EXPORTING  
        id              = 'GS\_0400-ISLTP'  
        values          = lt\_values  
      EXCEPTIONS  
        id\_illegal\_name = 1  
        OTHERS          = 2.  
  
    " Alt işlem tipi  
    REFRESH lt\_values.  
    IF gs\_0400-isltp IS NOT INITIAL.  
      READ TABLE mt\_zdys\_t\_022 TRANSPORTING NO FIELDS WITH KEY isltp = gs\_0400-isltp.  
      IF sy-subrc = 0.  
        SELECT \*  
               FROM dd07t  
               INTO TABLE lt\_dd07t  
               WHERE domname = 'ZDYS\_D\_ISLTP\_ALT'  
               AND ddlanguage = sy-langu.  
      ENDIF.  
      LOOP AT mt\_zdys\_t\_022 INTO DATA(ls\_zdys\_t\_022) WHERE isltp = gs\_0400-isltp.  
        CLEAR ls\_dd07t.  
        READ TABLE lt\_dd07t INTO ls\_dd07t WITH KEY domvalue\_l = ls\_zdys\_t\_022-isltp\_alt.  
        APPEND VALUE #( key  = ls\_dd07t-domvalue\_l  
                        text = ls\_dd07t-ddtext ) TO lt\_values.  
      ENDLOOP.  
    ENDIF.  
    CALL FUNCTION 'VRM\_SET\_VALUES'  
      EXPORTING  
        id              = 'GS\_0400-ISLTP\_ALT'  
        values          = lt\_values  
      EXCEPTIONS  
        id\_illegal\_name = 1  
        OTHERS          = 2.  
  
    IF go\_cust\_cont\_0400 IS INITIAL.  
      CREATE OBJECT go\_cust\_cont\_0400  
        EXPORTING  
          container\_name = 'C\_T400'.  
      CREATE OBJECT go\_textedit\_0400  
        EXPORTING  
          parent            = go\_cust\_cont\_0400  
          wordwrap\_mode     = 2  
          wordwrap\_position = 100.  
      CALL METHOD go\_textedit\_0400->set\_toolbar\_mode  
        EXPORTING  
          toolbar\_mode = 0.  
      CALL METHOD go\_textedit\_0400->set\_statusbar\_mode  
        EXPORTING  
          statusbar\_mode = 0.  
      CALL METHOD go\_textedit\_0400->set\_text\_as\_stream  
        EXPORTING  
          text = gs\_0400-text.  
    ELSE.  
      CALL METHOD go\_textedit\_0400->set\_text\_as\_stream  
        EXPORTING  
          text = gs\_0400-text.  
    ENDIF.  
  
    SET PF-STATUS '0400'.  
    SET TITLEBAR '0400'.  
  ENDMETHOD.  
  
  METHOD pai\_0400\_screen.  
    CASE gv\_ucomm\_0400.  
      WHEN '&F01' OR '&F02'.  
        LEAVE TO SCREEN 0.  
      WHEN 'FC\_ISLTP'.  
        CLEAR gs\_0400-isltp\_alt.  
        IF gs\_0400-isltp IS NOT INITIAL.  
          SELECT SINGLE isltp\_alt  
                 FROM zdys\_t\_022  
                 INTO @DATA(lv\_isltp\_alt)  
                 WHERE isltp = @gs\_0400-isltp.  
          IF sy-subrc <> 0.  
            MESSAGE i118 WITH gs\_0400-isltp DISPLAY LIKE 'E'.  
          ENDIF.  
        ENDIF.  
      WHEN '&SAVE'.  
        save\_screen\_0400( ).  
        IF gs\_0400-leave\_screen = 'X'.  
          LEAVE TO SCREEN 0.  
        ENDIF.  
    ENDCASE.  
    CLEAR gv\_ucomm\_0400.  
  ENDMETHOD.  
  
  METHOD pbo\_0500\_screen.  
    CONSTANTS: lc\_key\_val TYPE char08 VALUE 'GS\_0500-'.  
  
    DATA: lv\_fname TYPE char50.  
  
    SET PF-STATUS '0500'.  
    SET TITLEBAR '0500'.  
  
    IF mt\_zdys\_t\_006[] IS NOT INITIAL.  
      DATA(lv\_key\_len) = strlen( lc\_key\_val ).  
      LOOP AT SCREEN.  
        IF strlen( screen-name ) > lv\_key\_len AND screen-name(lv\_key\_len) = lc\_key\_val.  
          lv\_fname = screen-name+lv\_key\_len(\*).  
          READ TABLE mt\_zdys\_t\_006 TRANSPORTING NO FIELDS WITH KEY fname = lv\_fname.  
          IF sy-subrc = 0.  
            screen-input = 1.  
            MODIFY SCREEN.  
          ENDIF.  
        ENDIF.  
      ENDLOOP.  
    ENDIF.  
  ENDMETHOD.  
  
  METHOD pai\_0500\_screen.  
    CASE gv\_ucomm\_0500.  
      WHEN '&F01' OR '&F02'.  
        LEAVE TO SCREEN 0.  
      WHEN '&SAVE'.  
        save\_screen\_0500( ).  
        IF gs\_0500-leave\_screen = 'X'.  
          LEAVE TO SCREEN 0.  
        ENDIF.  
    ENDCASE.  
    CLEAR gv\_ucomm\_0500.  
  ENDMETHOD.  
  
  METHOD pbo\_0600\_screen.  
    DATA: lt\_text\_600 TYPE STANDARD TABLE OF tdline,  
          lt\_lines    TYPE TABLE OF tline,  
          lv\_tittle   TYPE text50,  
          lv\_name     TYPE thead-tdname.  
  
    IF lines( gs\_0600-dysid\_selected ) = 1.  
      READ TABLE gs\_0600-dysid\_selected INTO DATA(lv\_dysid) INDEX 1.  
      lv\_tittle = lv\_dysid && ` ` && TEXT-037.  
      SELECT SINGLE iadenk  
             FROM zdys\_t\_001  
             INTO gs\_0600-iadenk  
             WHERE dysid = lv\_dysid.  
  
      lv\_name = lv\_dysid.  
      CALL FUNCTION 'READ\_TEXT'  
        EXPORTING  
          id                      = 'Z002'  
          language                = sy-langu  
          name                    = lv\_name  
          object                  = zdys\_cl\_general=>mc\_tdobject  
        TABLES  
          lines                   = lt\_lines  
        EXCEPTIONS  
          id                      = 1  
          language                = 2  
          name                    = 3  
          not\_found               = 4  
          object                  = 5  
          reference\_check         = 6  
          wrong\_access\_to\_archive = 7  
          OTHERS                  = 8.  
      IF sy-subrc = 0.  
        LOOP AT lt\_lines INTO DATA(ls\_lines).  
          APPEND ls\_lines-tdline TO lt\_text\_600.  
        ENDLOOP.  
      ENDIF.  
    ELSE.  
      lv\_tittle = TEXT-036.  
    ENDIF.  
  
    IF go\_cust\_cont\_0600 IS INITIAL.  
      CREATE OBJECT go\_cust\_cont\_0600  
        EXPORTING  
          container\_name = 'C\_T600'.  
      CREATE OBJECT go\_textedit\_0600  
        EXPORTING  
          parent            = go\_cust\_cont\_0600  
          wordwrap\_mode     = 2  
          wordwrap\_position = 100.  
      CALL METHOD go\_textedit\_0600->set\_toolbar\_mode  
        EXPORTING  
          toolbar\_mode = 0.  
      CALL METHOD go\_textedit\_0600->set\_statusbar\_mode  
        EXPORTING  
          statusbar\_mode = 0.  
      CALL METHOD go\_textedit\_0600->set\_text\_as\_stream  
        EXPORTING  
          text = lt\_text\_600.  
    ELSE.  
      CALL METHOD go\_textedit\_0600->set\_text\_as\_stream  
        EXPORTING  
          text = lt\_text\_600.  
    ENDIF.  
  
    SET PF-STATUS '0600'.  
    SET TITLEBAR '0600' WITH lv\_tittle.  
  ENDMETHOD.  
  
  METHOD pai\_0600\_screen.  
    CASE gv\_ucomm\_0600.  
      WHEN '&F01' OR '&F02'.  
        LEAVE TO SCREEN 0.  
      WHEN '&SAVE'.  
        save\_screen\_0600( ).  
        IF gs\_0600-leave\_screen = 'X'.  
          LEAVE TO SCREEN 0.  
        ENDIF.  
    ENDCASE.  
    CLEAR gv\_ucomm\_0600.  
  ENDMETHOD.  
  
  METHOD pbo\_0700\_screen.  
    DATA: lt\_toolbar\_excluding TYPE ui\_functions.  
  
    " Açıklamanın görüntülendiği editör  
    IF go\_cust\_cont\_0700\_1 IS INITIAL.  
      CREATE OBJECT go\_cust\_cont\_0700\_1  
        EXPORTING  
          container\_name = 'C\_T700'.  
      CREATE OBJECT go\_textedit\_0700  
        EXPORTING  
          parent            = go\_cust\_cont\_0700\_1  
          wordwrap\_mode     = 1  
          wordwrap\_position = 100.  
      CALL METHOD go\_textedit\_0700->set\_toolbar\_mode  
        EXPORTING  
          toolbar\_mode = 0.  
      CALL METHOD go\_textedit\_0700->set\_statusbar\_mode  
        EXPORTING  
          statusbar\_mode = 0.  
      CALL METHOD go\_textedit\_0700->set\_enable  
        EXPORTING  
          enable = 'X'.  
      CALL METHOD go\_textedit\_0700->set\_readonly\_mode  
        EXPORTING  
          readonly\_mode = 1.  
      CALL METHOD go\_textedit\_0700->set\_text\_as\_stream  
        EXPORTING  
          text = gs\_0700-text.  
    ELSE.  
      CALL METHOD go\_textedit\_0700->set\_text\_as\_stream  
        EXPORTING  
          text = gs\_0700-text.  
    ENDIF.  
  
    " Girilen açıklamaların listesi  
    IF go\_cust\_cont\_0700\_2 IS INITIAL.  
      CREATE OBJECT go\_cust\_cont\_0700\_2  
        EXPORTING  
          container\_name = 'CC\_TXT'.  
      CREATE OBJECT go\_alv\_grid\_0700  
        EXPORTING  
          i\_parent = go\_cust\_cont\_0700\_2.  
      get\_fieldcatalog( EXPORTING iv\_alv\_name = '0700'  
                        IMPORTING et\_fieldcat = DATA(lt\_fieldcat) ).  
      get\_layout( EXPORTING iv\_alv\_name = '0700'  
                  IMPORTING es\_layout = DATA(ls\_layout) ).  
      APPEND cl\_gui\_alv\_grid=>mc\_mb\_variant   TO lt\_toolbar\_excluding.  
      APPEND cl\_gui\_alv\_grid=>mc\_mb\_sum       TO lt\_toolbar\_excluding.  
      APPEND cl\_gui\_alv\_grid=>mc\_mb\_filter    TO lt\_toolbar\_excluding.  
      APPEND cl\_gui\_alv\_grid=>mc\_mb\_view      TO lt\_toolbar\_excluding.  
      APPEND cl\_gui\_alv\_grid=>mc\_fc\_detail    TO lt\_toolbar\_excluding.  
      APPEND cl\_gui\_alv\_grid=>mc\_fc\_print     TO lt\_toolbar\_excluding.  
      APPEND cl\_gui\_alv\_grid=>mc\_fc\_info      TO lt\_toolbar\_excluding.  
      APPEND cl\_gui\_alv\_grid=>mc\_mb\_export    TO lt\_toolbar\_excluding.  
      APPEND cl\_gui\_alv\_grid=>mc\_fc\_find\_more TO lt\_toolbar\_excluding.  
  
      CALL METHOD go\_alv\_grid\_0700->set\_table\_for\_first\_display  
        EXPORTING  
          is\_layout                     = ls\_layout  
          it\_toolbar\_excluding          = lt\_toolbar\_excluding  
          i\_bypassing\_buffer            = 'X'  
        CHANGING  
          it\_outtab                     = mt\_ltxt  
          it\_fieldcatalog               = lt\_fieldcat  
        EXCEPTIONS  
          invalid\_parameter\_combination = 1  
          program\_error                 = 2  
          too\_many\_lines                = 3  
          OTHERS                        = 4.  
      IF sy-subrc NE 0.  
        MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno  
              WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
      ENDIF.  
  
      CALL METHOD go\_alv\_grid\_0700->register\_edit\_event  
        EXPORTING  
          i\_event\_id = cl\_gui\_alv\_grid=>mc\_evt\_modified  
        EXCEPTIONS  
          error      = 1  
          OTHERS     = 2.  
      set\_event\_handlers( ).  
      IF go\_alv\_grid\_0700 IS NOT INITIAL.  
        go\_alv\_grid\_0700->set\_toolbar\_interactive( ).  
      ENDIF.  
    ELSE.  
      refresh\_0700\_screen( ).  
    ENDIF.  
  
    SET PF-STATUS '0700'.  
    SET TITLEBAR '0700' WITH gs\_0700-item-fatno gs\_0700-item-dysid.  
  ENDMETHOD.  
  
  METHOD pai\_0700\_screen.  
    CASE gv\_ucomm\_0700.  
      WHEN '&F01' OR '&F02'.  
        LEAVE TO SCREEN 0.  
      WHEN '&CREATETXT'.  
        IF create\_new\_ltxt( iv\_dysid = gs\_0700-item-dysid ) = 'X'.  
          get\_ltxt\_data( iv\_dysid = gs\_0700-item-dysid ).  
          refresh\_0700\_screen( ).  
        ENDIF.  
    ENDCASE.  
    CLEAR gv\_ucomm\_0700.  
  ENDMETHOD.  
  
  METHOD refresh\_0700\_screen.  
    get\_layout( EXPORTING iv\_alv\_name = '0700'  
                IMPORTING es\_layout = DATA(ls\_layout) ).  
    go\_alv\_grid\_0700->set\_frontend\_layout( is\_layout = ls\_layout ).  
    go\_alv\_grid\_0700->refresh\_table\_display( ).  
  ENDMETHOD.  
  
  METHOD get\_fieldcatalog.  
    DEFINE modif\_fcat.  
      &1-scrtext\_l = &2.  
      &1-scrtext\_m = &2.  
      &1-scrtext\_s = &2.  
      &1-reptext = &2.  
      &1-colddictxt = 'L'.  
    END-OF-DEFINITION.  
  
    DATA lv\_structure\_name TYPE tabname.  
  
    CASE iv\_alv\_name.  
      WHEN '0100'.  
        lv\_structure\_name = 'ZDYS\_S\_003'.  
      WHEN '0200'.  
        lv\_structure\_name = 'ZDYS\_S\_005'.  
      WHEN '0300'.  
        lv\_structure\_name = 'ZDYS\_T\_016'.  
      WHEN '0700'.  
        lv\_structure\_name = 'ZDYS\_S\_019'.  
      WHEN 'WFLOG'.  
        lv\_structure\_name = 'ZDYS\_S\_009'.  
      WHEN 'CHIER'.  
        lv\_structure\_name = 'ZSD\_ES\_S038'.  
      WHEN 'CHIST'.  
        lv\_structure\_name = 'ZSD\_ES\_S037'.  
    ENDCASE.  
  
    CALL FUNCTION 'LVC\_FIELDCATALOG\_MERGE'  
      EXPORTING  
        i\_structure\_name       = lv\_structure\_name  
        i\_client\_never\_display = 'X'  
        i\_bypassing\_buffer     = 'X'  
      CHANGING  
        ct\_fieldcat            = et\_fieldcat[]  
      EXCEPTIONS  
        inconsistent\_interface = 1  
        program\_error          = 2  
        OTHERS                 = 3.  
    CHECK sy-subrc = 0.  
  
    CASE iv\_alv\_name.  
      WHEN '0100'.  
        LOOP AT et\_fieldcat ASSIGNING FIELD-SYMBOL(<ls\_fieldcat>).  
          CASE <ls\_fieldcat>-fieldname.  
            WHEN 'DYSID'.  
              <ls\_fieldcat>-key = 'X'.  
            WHEN 'BTN\_FAT'.  
              modif\_fcat <ls\_fieldcat> TEXT-023.  
              <ls\_fieldcat>-col\_pos = change\_col\_pos( EXPORTING iv\_fieldname = 'DYSID'  
                                                                it\_fieldcat  = et\_fieldcat ).  
            WHEN 'BTN\_FATEK'.  
              modif\_fcat <ls\_fieldcat> TEXT-024.  
              <ls\_fieldcat>-col\_pos = change\_col\_pos( EXPORTING iv\_fieldname = 'DYSID'  
                                                                it\_fieldcat  = et\_fieldcat ).  
            WHEN 'BTN\_TXT'.  
              modif\_fcat <ls\_fieldcat> TEXT-021.  
              <ls\_fieldcat>-col\_pos = change\_col\_pos( EXPORTING iv\_fieldname = 'DYSID'  
                                                                it\_fieldcat  = et\_fieldcat ).  
            WHEN 'BTN\_ONYH'.  
              modif\_fcat <ls\_fieldcat> TEXT-052.  
            WHEN 'BTN\_ONYT'.  
              modif\_fcat <ls\_fieldcat> TEXT-053.  
            WHEN 'BTN\_XML'.  
              modif\_fcat <ls\_fieldcat> TEXT-039.  
            WHEN 'PARTNERT'.  
              <ls\_fieldcat>-col\_pos = change\_col\_pos( EXPORTING iv\_fieldname = 'PARTNER'  
                                                                it\_fieldcat  = et\_fieldcat ).  
            WHEN 'FATTP'.  
              <ls\_fieldcat>-no\_out = 'X'.  
            WHEN 'FATTPT'.  
              <ls\_fieldcat>-col\_pos = change\_col\_pos( EXPORTING iv\_fieldname = 'FATTP'  
                                                                it\_fieldcat  = et\_fieldcat ).  
              modif\_fcat <ls\_fieldcat> TEXT-006.  
            WHEN 'MARKAT'.  
              <ls\_fieldcat>-col\_pos = change\_col\_pos( EXPORTING iv\_fieldname = 'MARKA'  
                                                                it\_fieldcat  = et\_fieldcat ).  
            WHEN 'WGBEZ'.  
              <ls\_fieldcat>-col\_pos = change\_col\_pos( EXPORTING iv\_fieldname = 'MATKL'  
                                                                it\_fieldcat  = et\_fieldcat ).  
            WHEN 'ORTID'.  
              <ls\_fieldcat>-no\_out = 'X'.  
            WHEN 'SIRNO'.  
              <ls\_fieldcat>-no\_out = 'X'.  
            WHEN 'DURUM'.  
              <ls\_fieldcat>-no\_out = 'X'.  
            WHEN 'DURUMT'.  
              <ls\_fieldcat>-col\_pos = change\_col\_pos( EXPORTING iv\_fieldname = 'DURUM'  
                                                                it\_fieldcat  = et\_fieldcat ).  
            WHEN 'FATUR'.  
              <ls\_fieldcat>-no\_out = 'X'.  
            WHEN 'FATURT'.  
              <ls\_fieldcat>-col\_pos = change\_col\_pos( EXPORTING iv\_fieldname = 'FATUR'  
                                                                it\_fieldcat  = et\_fieldcat ).  
              modif\_fcat <ls\_fieldcat> TEXT-022.  
            WHEN 'EIMZS'.  
              <ls\_fieldcat>-no\_out = 'X'.  
            WHEN 'EIMZST'.  
              <ls\_fieldcat>-col\_pos = change\_col\_pos( EXPORTING iv\_fieldname = 'EIMZS'  
                                                                it\_fieldcat  = et\_fieldcat ).  
              modif\_fcat <ls\_fieldcat> TEXT-025.  
            WHEN 'EIMZD'.  
              <ls\_fieldcat>-no\_out = 'X'.  
            WHEN 'EIMZDT'.  
              <ls\_fieldcat>-col\_pos = change\_col\_pos( EXPORTING iv\_fieldname = 'EIMZD'  
                                                                it\_fieldcat  = et\_fieldcat ).  
              modif\_fcat <ls\_fieldcat> TEXT-026.  
            WHEN 'GDRTPT'.  
              <ls\_fieldcat>-col\_pos = change\_col\_pos( EXPORTING iv\_fieldname = 'GDRTP'  
                                                                it\_fieldcat  = et\_fieldcat ).  
            WHEN 'MANUE'.  
              <ls\_fieldcat>-checkbox = 'X'.  
            WHEN 'SOZLS'.  
              <ls\_fieldcat>-checkbox = 'X'.  
            WHEN 'DONMS'.  
              <ls\_fieldcat>-checkbox = 'X'.  
            WHEN 'AWTYP' OR 'AWKEY'.  
              <ls\_fieldcat>-no\_out = 'X'.  
            WHEN 'BLGNO'.  
              <ls\_fieldcat>-col\_pos = change\_col\_pos( EXPORTING iv\_fieldname = 'WAERS'  
                                                                it\_fieldcat  = et\_fieldcat ).  
              <ls\_fieldcat>-hotspot = 'X'.  
            WHEN 'YILNO'.  
              <ls\_fieldcat>-col\_pos = change\_col\_pos( EXPORTING iv\_fieldname = 'BLGNO'  
                                                                it\_fieldcat  = et\_fieldcat ).  
            WHEN 'RBSTAT'.  
              <ls\_fieldcat>-col\_pos = change\_col\_pos( EXPORTING iv\_fieldname = 'WAERS'  
                                                                it\_fieldcat  = et\_fieldcat ).  
              <ls\_fieldcat>-colddictxt = 'M'.  
            WHEN 'FAKSK' OR 'FAKSP' OR 'FKSTA' OR 'VSTAT'.  
              DELETE et\_fieldcat.  
              CONTINUE.  
            WHEN 'SFATDT'.  
              modif\_fcat <ls\_fieldcat> TEXT-038.  
            WHEN 'KURSF'.  
              <ls\_fieldcat>-col\_pos = change\_col\_pos( EXPORTING iv\_fieldname = 'FATTR'  
                                                                it\_fieldcat  = et\_fieldcat ).  
            WHEN 'IADENT'.  
              <ls\_fieldcat>-col\_pos = change\_col\_pos( EXPORTING iv\_fieldname = 'IADENK'  
                                                                it\_fieldcat  = et\_fieldcat ).  
            WHEN 'ISLTP'.  
              <ls\_fieldcat>-no\_out = 'X'.  
            WHEN 'ISLTPT'.  
              modif\_fcat <ls\_fieldcat> TEXT-076.  
              <ls\_fieldcat>-col\_pos = change\_col\_pos( EXPORTING iv\_fieldname = 'ISLTP'  
                                                                it\_fieldcat  = et\_fieldcat ).  
            WHEN 'ISLTP\_ALT'.  
              <ls\_fieldcat>-no\_out = 'X'.  
            WHEN 'ISLTP\_ALTT'.  
              modif\_fcat <ls\_fieldcat> TEXT-077.  
              <ls\_fieldcat>-col\_pos = change\_col\_pos( EXPORTING iv\_fieldname = 'ISLTP\_ALT'  
                                                                it\_fieldcat  = et\_fieldcat ).  
            WHEN 'ISLTP\_FAT'.  
              <ls\_fieldcat>-no\_out = 'X'.  
            WHEN 'ISLTP\_FATT'.  
              modif\_fcat <ls\_fieldcat> TEXT-078.  
              <ls\_fieldcat>-col\_pos = change\_col\_pos( EXPORTING iv\_fieldname = 'ISLTP\_FAT'  
                                                                it\_fieldcat  = et\_fieldcat ).  
            WHEN 'MATCHED\_TUTAR'.  
              <ls\_fieldcat>-col\_pos = change\_col\_pos( EXPORTING iv\_fieldname = 'GJAHR'  
                                                                it\_fieldcat  = et\_fieldcat ).  
              <ls\_fieldcat>-checkbox = 'X'.  
              <ls\_fieldcat>-colddictxt = 'L'.  
            WHEN 'MATCHED\_TARIH'.  
              <ls\_fieldcat>-col\_pos = change\_col\_pos( EXPORTING iv\_fieldname = 'GJAHR'  
                                                                it\_fieldcat  = et\_fieldcat ).  
              <ls\_fieldcat>-checkbox = 'X'.  
              <ls\_fieldcat>-colddictxt = 'L'.  
            WHEN 'CRDAT'.  
              modif\_fcat <ls\_fieldcat> TEXT-055.  
            WHEN 'CRZET'.  
              modif\_fcat <ls\_fieldcat> TEXT-060.  
            WHEN 'XMLRCD'.  
              modif\_fcat <ls\_fieldcat> TEXT-056.  
            WHEN 'XMLRCT'.  
              modif\_fcat <ls\_fieldcat> TEXT-057.  
            WHEN 'XMLRLD'.  
              modif\_fcat <ls\_fieldcat> TEXT-058.  
            WHEN 'XMLRLT'.  
              modif\_fcat <ls\_fieldcat> TEXT-059.  
            WHEN 'XMLSCD' OR 'XMLSCT'.  
              <ls\_fieldcat>-no\_out = 'X'.  
            WHEN 'KSZTU\_UPB'.  
              <ls\_fieldcat>-colddictxt = 'M'.  
            WHEN 'HWAER'.  
              DELETE et\_fieldcat.  
              CONTINUE.  
            WHEN 'MUDUR'.  
              modif\_fcat <ls\_fieldcat> TEXT-004.  
          ENDCASE.  
        ENDLOOP.  
  
      WHEN '0200'.  
        LOOP AT et\_fieldcat ASSIGNING <ls\_fieldcat>.  
          CASE <ls\_fieldcat>-fieldname.  
            WHEN 'DYSID' OR 'ISLNO' OR 'ISLTP' OR 'UNAME' OR 'FNAME'.  
              <ls\_fieldcat>-no\_out = 'X'.  
            WHEN 'ISLTPT'.  
              modif\_fcat <ls\_fieldcat> TEXT-018.  
            WHEN 'REPTEXT'.  
              modif\_fcat <ls\_fieldcat> TEXT-019.  
            WHEN 'ISDTY'.  
              modif\_fcat <ls\_fieldcat> TEXT-020.  
            WHEN 'ACKLM'.  
              modif\_fcat <ls\_fieldcat> TEXT-021.  
          ENDCASE.  
        ENDLOOP.  
  
      WHEN '0300'.  
        LOOP AT et\_fieldcat ASSIGNING <ls\_fieldcat>.  
          CASE <ls\_fieldcat>-fieldname.  
            WHEN 'DYSID' OR 'UNAME' OR 'UDATE' OR 'UTIME'.  
              <ls\_fieldcat>-no\_out = 'X'.  
          ENDCASE.  
        ENDLOOP.  
  
      WHEN '0700'.  
        LOOP AT et\_fieldcat ASSIGNING <ls\_fieldcat>.  
          CASE <ls\_fieldcat>-fieldname.  
            WHEN 'TDOBJECT' OR 'TDNAME' OR 'TDID' OR 'TDSPRAS'.  
              <ls\_fieldcat>-no\_out = 'X'.  
            WHEN 'ISLNO'.  
              modif\_fcat <ls\_fieldcat> TEXT-048.  
            WHEN 'ACKLM'.  
              <ls\_fieldcat>-hotspot = 'X'.  
          ENDCASE.  
        ENDLOOP.  
  
      WHEN 'WFLOG'.  
        LOOP AT et\_fieldcat ASSIGNING <ls\_fieldcat>.  
          CASE <ls\_fieldcat>-fieldname.  
            WHEN 'DYSID' OR 'KONTROL' OR 'ISLTP'.  
              <ls\_fieldcat>-no\_out = 'X'.  
            WHEN 'KONTROLT'.  
              <ls\_fieldcat>-col\_pos = change\_col\_pos( EXPORTING iv\_fieldname = 'KONTROL'  
                                                                it\_fieldcat  = et\_fieldcat ).  
              modif\_fcat <ls\_fieldcat> TEXT-011.  
            WHEN 'ISLTPT'.  
              <ls\_fieldcat>-col\_pos = change\_col\_pos( EXPORTING iv\_fieldname = 'ISLTP'  
                                                                it\_fieldcat  = et\_fieldcat ).  
              modif\_fcat <ls\_fieldcat> TEXT-012.  
          ENDCASE.  
        ENDLOOP.  
  
      WHEN 'CHIER'.  
        LOOP AT et\_fieldcat ASSIGNING <ls\_fieldcat>.  
          CASE <ls\_fieldcat>-fieldname.  
            WHEN 'USNAM' OR 'INFTP' OR 'APRTP'.  
              <ls\_fieldcat>-tech = 'X'.  
          ENDCASE.  
        ENDLOOP.  
      WHEN 'CHIST'.  
        LOOP AT et\_fieldcat ASSIGNING <ls\_fieldcat>.  
          CASE <ls\_fieldcat>-fieldname.  
            WHEN 'USNAM' OR 'USRNM' OR 'APRST' OR 'APRAV'.  
              <ls\_fieldcat>-tech = 'X'.  
          ENDCASE.  
        ENDLOOP.  
    ENDCASE.  
  ENDMETHOD.  
  
  METHOD change\_col\_pos.  
    DATA lt\_fieldcat TYPE lvc\_t\_fcat.  
  
    " Bu method ilgili alanı ALV'de istenen alanın yanına getirmek için yazıldı  
    lt\_fieldcat[] = it\_fieldcat.  
  
    READ TABLE lt\_fieldcat INTO DATA(ls\_fieldcat) WITH KEY fieldname = iv\_fieldname.  
    IF sy-subrc = 0.  
      rv\_col\_pos = ls\_fieldcat-col\_pos.  
    ENDIF.  
  ENDMETHOD.  
  
  METHOD get\_layout.  
    CASE iv\_alv\_name.  
      WHEN '0100'.  
        es\_layout-cwidth\_opt = 'X'.  
        es\_layout-zebra      = 'X'.  
        es\_layout-no\_rowins  = 'X'.  
        es\_layout-sel\_mode   = 'D'.  
        es\_layout-no\_merging = 'X'.  
        es\_layout-ctab\_fname = 'CELLCOLOR'.  
        es\_layout-stylefname = 'CELLSTYLE'.  
      WHEN '0200'.  
        es\_layout-cwidth\_opt = 'X'.  
        es\_layout-zebra      = 'X'.  
        es\_layout-no\_rowins  = 'X'.  
        es\_layout-ctab\_fname = 'CELLCOLOR'.  
        es\_layout-stylefname = 'CELLSTYLE'.  
      WHEN '0300'.  
        es\_layout-cwidth\_opt = 'X'.  
        es\_layout-zebra      = 'X'.  
        es\_layout-no\_rowins  = 'X'.  
        es\_layout-sel\_mode   = 'D'.  
      WHEN '0700'.  
        es\_layout-cwidth\_opt = 'X'.  
        es\_layout-zebra      = 'X'.  
        es\_layout-no\_rowins  = 'X'.  
      WHEN 'WFLOG'.  
        es\_layout-cwidth\_opt = 'X'.  
        es\_layout-zebra      = 'X'.  
        es\_layout-no\_rowins  = 'X'.  
      WHEN 'CHIER'.  
        es\_layout-cwidth\_opt = 'X'.  
        es\_layout-zebra      = 'X'.  
        es\_layout-no\_rowins  = 'X'.  
      WHEN 'CHIST'.  
        es\_layout-cwidth\_opt = 'X'.  
        es\_layout-zebra      = 'X'.  
        es\_layout-no\_rowins  = 'X'.  
    ENDCASE.  
  ENDMETHOD.  
  
  METHOD set\_event\_handlers.  
    SET HANDLER handle\_button\_click FOR ALL INSTANCES.  
    SET HANDLER handle\_hotspot\_click FOR ALL INSTANCES.  
    SET HANDLER handle\_toolbar FOR ALL INSTANCES.  
    SET HANDLER handle\_user\_command FOR ALL INSTANCES.  
    SET HANDLER handle\_double\_click FOR ALL INSTANCES.  
    SET HANDLER handle\_data\_changed\_finished FOR ALL INSTANCES.  
  ENDMETHOD.  
  
  METHOD check\_selected\_rows.  
    REFRESH mt\_dysid\_selected.  
  
    CASE iv\_sel\_method.  
      WHEN 'S'. " Single (Sadece 1 satır seçilebilen durum)  
        go\_alv\_grid\_0100->get\_selected\_rows( IMPORTING et\_row\_no = DATA(lt\_row\_no) ).  
        CASE lines( lt\_row\_no ).  
          WHEN 0.  
            rv\_error = 'X'.  
            MESSAGE i002 DISPLAY LIKE 'E'.  
            EXIT.  
          WHEN 1.  
            LOOP AT lt\_row\_no INTO DATA(ls\_row\_no).  
              READ TABLE gt\_item INTO DATA(ls\_item) INDEX ls\_row\_no-row\_id.  
              CHECK sy-subrc = 0.  
              APPEND ls\_item-dysid TO mt\_dysid\_selected.  
              EXIT.  
            ENDLOOP.  
          WHEN OTHERS.  
            rv\_error = 'X'.  
            MESSAGE i003 DISPLAY LIKE 'E'.  
            EXIT.  
        ENDCASE.  
  
      WHEN 'M'. " Multi  (En az 1 satır seçilebilen durum)  
        go\_alv\_grid\_0100->get\_selected\_rows( IMPORTING et\_row\_no = lt\_row\_no ).  
        IF lines( lt\_row\_no ) = 0.  
          rv\_error = 'X'.  
          MESSAGE i001 DISPLAY LIKE 'E'.  
          EXIT.  
        ELSE.  
          LOOP AT lt\_row\_no INTO ls\_row\_no.  
            READ TABLE gt\_item INTO ls\_item INDEX ls\_row\_no-row\_id.  
            CHECK sy-subrc = 0.  
            APPEND ls\_item-dysid TO mt\_dysid\_selected.  
          ENDLOOP.  
        ENDIF.  
    ENDCASE.  
  ENDMETHOD.  
  
  METHOD refresh\_list.  
    go\_alv\_grid\_0100->get\_selected\_rows( IMPORTING et\_row\_no = DATA(lt\_row\_no) ).  
    IF lt\_row\_no[] IS INITIAL.  
      APPEND '\*' TO mt\_dysid\_selected.  
    ELSE.  
      LOOP AT lt\_row\_no INTO DATA(ls\_row\_no).  
        READ TABLE gt\_item INTO DATA(ls\_item) INDEX ls\_row\_no-row\_id.  
        CHECK sy-subrc = 0.  
        APPEND ls\_item-dysid TO mt\_dysid\_selected.  
      ENDLOOP.  
    ENDIF.  
  ENDMETHOD.  
  
  METHOD refresh\_data\_inb.  
    DATA: lr\_dysid TYPE RANGE OF zdys\_t\_001-dysid.  
  
    CHECK mt\_dysid\_selected[] IS NOT INITIAL.  
  
    LOOP AT mt\_dysid\_selected INTO DATA(lv\_dysid\_selected).  
      IF lv\_dysid\_selected = '\*'.  
        REFRESH mt\_dysid\_selected.  
        REFRESH lr\_dysid.  
        EXIT.  
      ELSE.  
        APPEND VALUE #( sign = 'I' option = 'EQ' low = lv\_dysid\_selected ) TO lr\_dysid.  
      ENDIF.  
    ENDLOOP.  
  
    DELETE gt\_item WHERE dysid IN lr\_dysid.  
  
    DATA(lt\_item\_updated) = get\_data\_inb( ).  
    APPEND LINES OF lt\_item\_updated TO gt\_item.  
    SORT gt\_item BY dysid.  
    REFRESH mt\_dysid\_selected.  
  
    refresh\_0100\_screen( ).  
  ENDMETHOD.  
  
  METHOD execute\_process.  
    TYPES: BEGIN OF ty\_rol,  
             rol TYPE zdys\_e\_rol,  
           END OF ty\_rol.  
    DATA: lt\_return    TYPE bapiret2\_t,  
          lt\_rol       TYPE TABLE OF ty\_rol,  
          lr\_isltp\_alt TYPE RANGE OF zdys\_t\_022-isltp\_alt,  
          lv\_count     TYPE int2.  
  
    CHECK check\_selected\_rows( EXPORTING iv\_sel\_method = 'M' ) = ''.  
  
    CLEAR: gs\_0400, gv\_ucomm\_0400.  
    REFRESH: mt\_zdys\_t\_021, mt\_zdys\_t\_022.  
  
    DO 1 TIMES.  
      LOOP AT mt\_dysid\_selected INTO DATA(lv\_dysid\_selected).  
        READ TABLE gt\_item INTO DATA(ls\_item) WITH KEY dysid = lv\_dysid\_selected.  
        CHECK sy-subrc = 0.  
  
        " 'Onay bekliyor' durum kontrolü  
        IF ls\_item-durum = '02'.  " Onay bekliyor  
          zcl\_util\_001=>add\_message(  
            EXPORTING  
              i\_msgty   = 'E'  
              i\_msgid   = mc\_msgid  
              i\_msgno   = '130'  
              i\_msgv1   = ls\_item-dysid  
            IMPORTING  
              et\_return = lt\_return ).  
          IF 1 = 2. MESSAGE e130 WITH ls\_item-dysid. ENDIF.  
        ENDIF.  
  
        " DYS durum kontrolü  
        IF zdys\_cl\_general=>check\_if\_inv\_completed( iv\_durum = ls\_item-durum ) = 'X'.  
          zcl\_util\_001=>add\_message(  
            EXPORTING  
              i\_msgty   = 'E'  
              i\_msgid   = mc\_msgid  
              i\_msgno   = '062'  
              i\_msgv1   = ls\_item-dysid  
            IMPORTING  
              et\_return = lt\_return ).  
          IF 1 = 2. MESSAGE e062 WITH ls\_item-dysid. ENDIF.  
        ENDIF.  
  
        " İşlem yürüt işlevi için iş akışı kontrolü  
        IF ls\_item-isaks = 'ORT' AND ls\_item-rol <> 'TTS'.  
          zcl\_util\_001=>add\_message(  
            EXPORTING  
              i\_msgty   = 'E'  
              i\_msgid   = mc\_msgid  
              i\_msgno   = '122'  
              i\_msgv1   = ls\_item-dysid  
            IMPORTING  
              et\_return = lt\_return ).  
          IF 1 = 2. MESSAGE e122 WITH ls\_item-dysid. ENDIF.  
        ENDIF.  
  
        " İmza onayı olmayan fatura için işlem yapılamaz  
        IF ls\_item-eimzs IS NOT INITIAL AND NOT ( ls\_item-eimzd = 'O' OR ls\_item-eimzd = 'M' ).  
          zcl\_util\_001=>add\_message(  
            EXPORTING  
              i\_msgty   = 'E'  
              i\_msgid   = mc\_msgid  
              i\_msgno   = '125'  
              i\_msgv1   = ls\_item-dysid  
            IMPORTING  
              et\_return = lt\_return ).  
          IF 1 = 2. MESSAGE e125 WITH ls\_item-dysid. ENDIF.  
        ENDIF.  
  
        COLLECT VALUE ty\_rol( rol = ls\_item-rol ) INTO lt\_rol.  
        APPEND ls\_item-dysid TO gs\_0400-dysid\_selected.  
      ENDLOOP.  
  
      " Seçilen DYS'ler içinde işlenen(lockl'lı) var mı kontrolü  
      DATA(lt\_return\_lock) = zdys\_cl\_general=>check\_if\_dysid\_locked( EXPORTING it\_dysid  = gs\_0400-dysid\_selected ).  
      APPEND LINES OF lt\_return\_lock TO lt\_return.  
  
      CHECK zcl\_util\_001=>check\_error\_msg( EXPORTING it\_return = lt\_return ) = ''.  
  
      " Farklı rol seçimi kontrolü  
      IF lines( lt\_rol ) = 1.  
        READ TABLE lt\_rol INTO DATA(ls\_rol) INDEX 1.  
        gs\_0400-rol = ls\_rol-rol.  
        SELECT \*  
               FROM zdys\_t\_021  
               INTO TABLE mt\_zdys\_t\_021  
               WHERE rol = gs\_0400-rol.  
        IF sy-subrc <> 0.  
          zcl\_util\_001=>add\_message(  
            EXPORTING  
              i\_msgty   = 'E'  
              i\_msgid   = mc\_msgid  
              i\_msgno   = '117'  
              i\_msgv1   = gs\_0400-rol  
            IMPORTING  
              et\_return = lt\_return ).  
          IF 1 = 2. MESSAGE e117 WITH gs\_0400-rol. ENDIF.  
        ENDIF.  
      ELSE.  
        zcl\_util\_001=>add\_message(  
          EXPORTING  
            i\_msgty   = 'E'  
            i\_msgid   = mc\_msgid  
            i\_msgno   = '114'  
          IMPORTING  
            et\_return = lt\_return ).  
        IF 1 = 2. MESSAGE e114. ENDIF.  
      ENDIF.  
    ENDDO.  
  
    IF zcl\_util\_001=>check\_error\_msg( EXPORTING it\_return = lt\_return ) = 'X'.  
      zcl\_util\_001=>show\_messages( EXPORTING it\_return = lt\_return ).  
      EXIT.  
    ENDIF.  
  
    " 'A24-Satış faturasıyla onayla' alt işlem tipi sadece 'FOS' ve 'SAOS' rollerinde girilebilir  
    IF NOT ( gs\_0400-rol = 'FOS' OR gs\_0400-rol = 'SAOS' ).  
      APPEND VALUE #( sign = 'E' option = 'EQ' low = 'A24' ) TO lr\_isltp\_alt. " Satış faturasıyla onayla  
    ENDIF.  
    SELECT \*  
           FROM zdys\_t\_022  
           INTO TABLE mt\_zdys\_t\_022  
           WHERE isltp\_alt IN lr\_isltp\_alt.  
  
    " Ekran başlık bilgisi  
    gs\_0400-header\_txt = lines( gs\_0400-dysid\_selected ).  
    CONDENSE gs\_0400-header\_txt.  
    gs\_0400-header\_txt = `Seçilen ` && gs\_0400-header\_txt &&  ` fatura için işlem tipi seçiniz`.  
    IF lines( gs\_0400-dysid\_selected ) = 1.  
      READ TABLE gs\_0400-dysid\_selected INTO lv\_dysid\_selected INDEX 1.  
      gs\_0400-header\_txt = gs\_0400-header\_txt && ` (DYS ID: ` && lv\_dysid\_selected && `)`.  
    ENDIF.  
  
    " Seçili rol için tek işlem tipi atanmış ise bu atanan işlem tipi ekrana default getirilir  
    IF lines( mt\_zdys\_t\_021 ) = 1.  
      READ TABLE mt\_zdys\_t\_021 INTO DATA(ls\_zdys\_t\_021) INDEX 1.  
      gs\_0400-isltp = ls\_zdys\_t\_021-isltp.  
    ELSE.  
      " Role 1'den fazla işlem tipi atandıysa ve '11-Onayla(hatasız)'  
      " işlem tipi de bu role atanmış ise ekrana default getirilir  
      READ TABLE mt\_zdys\_t\_021 INTO ls\_zdys\_t\_021 WITH KEY isltp = '11'.  " Onayla(hatasız)  
      IF sy-subrc = 0.  
        gs\_0400-isltp = ls\_zdys\_t\_021-isltp.  
      ENDIF.  
    ENDIF.  
  
    IF gs\_0400-isltp IS NOT INITIAL.  
      CLEAR lv\_count.  
      LOOP AT mt\_zdys\_t\_022 INTO DATA(ls\_zdys\_t\_022) WHERE isltp = gs\_0400-isltp.  
        ADD 1 TO lv\_count.  
      ENDLOOP.  
      " Seçili işlem tipine 1 alt işlem tipi atandıysa ekrana bu alt işlem tipi default getirilir  
      IF lv\_count = 1.  
        gs\_0400-isltp\_alt = ls\_zdys\_t\_022-isltp\_alt.  
      ELSE.  
        " Seçili işlem tipine 1'den fazla alt işlem tipi atanmış ve 'A01-Onayla'  
        " alt işlem tipi bu işlem tipine atamış ise ekrana default getirilir  
        READ TABLE mt\_zdys\_t\_022 INTO ls\_zdys\_t\_022 WITH KEY isltp = gs\_0400-isltp  
                                                             isltp\_alt = 'A01'.  " Onayla  
        gs\_0400-isltp\_alt = ls\_zdys\_t\_022-isltp\_alt.  
      ENDIF.  
    ENDIF.  
  
    CALL SCREEN 0400 STARTING AT 5 5 ENDING AT 67 15.  
  ENDMETHOD.  
  
  METHOD save\_screen\_0400.  
    DATA: lt\_return\_all   TYPE bapiret2\_t,  
          lt\_return       TYPE bapiret2\_t,  
          lt\_itf\_text     TYPE tlinet,  
          lt\_soli\_text    TYPE soli\_tab,  
          lt\_toa01        TYPE zdys\_tt\_010,  
          lv\_fname        TYPE thead-tdname,  
          lv\_text          TYPE text200,  
          lv\_res\_secs     TYPE i,  
          lv\_res\_secs\_chk TYPE i.  
  
    DATA: ls\_zdys\_t\_001     TYPE zdys\_t\_001,  
          lv\_islno\_max\_new TYPE zdys\_t\_002-islno.  
  
    CONSTANTS: lc\_7day TYPE i VALUE 7.  " 7 gün limit fatura kontrolü  
  
    CALL METHOD go\_textedit\_0400->get\_text\_as\_stream  
      IMPORTING  
        text = gs\_0400-text.  
    CALL FUNCTION 'CONVERT\_STREAM\_TO\_ITF\_TEXT'  
      EXPORTING  
        language    = 'T'  
      TABLES  
        text\_stream = gs\_0400-text  
        itf\_text    = lt\_itf\_text.  
  
    " GOS rolünde işlemi yürüt onayla dediğinde açıklama girmek zorunlu  
    IF gs\_0400-rol = 'GOS' AND gs\_0400-isltp = '11' AND lt\_itf\_text[] IS INITIAL.  
      MESSAGE i127 DISPLAY LIKE 'E'.  
      EXIT.  
    ENDIF.  
  
    DATA(ls\_zdys\_t\_012) = zdys\_cl\_attachment=>get\_archive\_disc( ).  
  
    LOOP AT gs\_0400-dysid\_selected INTO DATA(lv\_dysid).  
      REFRESH lt\_return.  
      DO 1 TIMES.  
        DATA(lo\_dys) = NEW zdys\_cl\_general( ).  
  
        READ TABLE gt\_item INTO DATA(ls\_item) WITH KEY dysid = lv\_dysid.  
        CHECK sy-subrc = 0.  
  
\* Seçilen işlem tipini yürütmeden önce yapılan kontroller  
        CASE gs\_0400-isltp.  
          WHEN '11'.  " Onayla(hatasız)  
            " Gelen fatura satın alma faturası ile eşleşti mi kontrolü  
            IF ls\_item-fattp = 'SA' AND  
             ( ls\_item-isaks = 'MLZ' OR ls\_item-isaks = 'MCR' ) AND  
             NOT ( gs\_0400-rol = 'SY' OR gs\_0400-rol = 'SAOS' ).  
              IF ls\_item-matched\_tutar = '' AND ls\_item-matched\_tarih = ''.  
                zcl\_util\_001=>add\_message(  
                  EXPORTING  
                    i\_msgty   = 'E'  
                    i\_msgid   = mc\_msgid  
                    i\_msgno   = '027'  
                    i\_msgv1   = ls\_item-dysid  
                  IMPORTING  
                    et\_return = lt\_return ).  
                IF 1 = 2. MESSAGE e027 WITH ls\_item-dysid. ENDIF.  
              ELSEIF ls\_item-matched\_tutar = 'X' AND ls\_item-matched\_tarih = ''.  
                zcl\_util\_001=>add\_message(  
                  EXPORTING  
                    i\_msgty   = 'E'  
                    i\_msgid   = mc\_msgid  
                    i\_msgno   = '126'  
                    i\_msgv1   = ls\_item-dysid  
                    i\_msgv2   = 'tarih'  
                  IMPORTING  
                    et\_return = lt\_return ).  
                IF 1 = 2. MESSAGE e126 WITH ls\_item-dysid 'tarih'. ENDIF.  
              ELSEIF ls\_item-matched\_tutar = '' AND ls\_item-matched\_tarih = 'X'.  
                zcl\_util\_001=>add\_message(  
                  EXPORTING  
                    i\_msgty   = 'E'  
                    i\_msgid   = mc\_msgid  
                    i\_msgno   = '126'  
                    i\_msgv1   = ls\_item-dysid  
                    i\_msgv2   = 'tutar'  
                  IMPORTING  
                    et\_return = lt\_return ).  
                IF 1 = 2. MESSAGE e126 WITH ls\_item-dysid 'tutar'. ENDIF.  
              ENDIF.  
            ENDIF.  
  
            " Gider faturaları kontrolleri  
            IF ls\_item-isaks = 'GDR'.  
              IF gs\_0400-rol = 'GOS'.  
                " Fatura sorumlusu kontrolü  
                IF ls\_item-fsorm IS NOT INITIAL AND ls\_item-fsorm <> sy-uname.  
                  " Fatura sorumlusuna vekalet edebilecek kullanıcı mı kontrolü  
                  LOOP AT mt\_zdys\_t\_027 TRANSPORTING NO FIELDS WHERE vkedl = ''  
                                                               OR vkedl = ls\_item-fsorm.  
                    EXIT.  
                  ENDLOOP.  
                  IF sy-subrc = 0.  " Vekalet edebilen bir kullanıcı ise  
                    IF zcl\_util\_001=>confirm\_popup( iv\_text = TEXT-061 ) <> '1'.  
                      zcl\_util\_001=>add\_message(  
                        EXPORTING  
                          i\_msgty   = 'E'  
                          i\_msgid   = mc\_msgid  
                          i\_msgno   = '162'  
                          i\_msgv1   = ls\_item-dysid  
                        IMPORTING  
                          et\_return = lt\_return ).  
                      IF 1 = 2. MESSAGE e162 WITH ls\_item-dysid. ENDIF.  
                    ENDIF.  
                  ELSE. " Vekalet edebilen bir kullanıcı değil ise  
                    zcl\_util\_001=>add\_message(  
                      EXPORTING  
                        i\_msgty   = 'E'  
                        i\_msgid   = mc\_msgid  
                        i\_msgno   = '128'  
                        i\_msgv1   = ls\_item-dysid  
                      IMPORTING  
                        et\_return = lt\_return ).  
                    IF 1 = 2. MESSAGE e128 WITH ls\_item-dysid. ENDIF.  
                  ENDIF.  
                ENDIF.  
  
                IF gs\_0400-rol = 'GOS' OR gs\_0400-rol = 'FAOS'.  
                  " Fatura sorumlusu zorunlu alan kontrolü  
                  IF ls\_item-fsorm IS INITIAL.  
                    zcl\_util\_001=>add\_message(  
                      EXPORTING  
                        i\_msgty   = 'E'  
                        i\_msgid   = mc\_msgid  
                        i\_msgno   = '131'  
                        i\_msgv1   = ls\_item-dysid  
                      IMPORTING  
                        et\_return = lt\_return ).  
                    IF 1 = 2. MESSAGE e131 WITH ls\_item-dysid. ENDIF.  
                  ENDIF.  
                  " Gider tipi zorunlu alan kontrolü  
                  IF ls\_item-gdrtp IS INITIAL.  
                    zcl\_util\_001=>add\_message(  
                         EXPORTING  
                           i\_msgty   = 'E'  
                           i\_msgid   = mc\_msgid  
                           i\_msgno   = '228'  
                           i\_msgv1   = gs\_0400-rol  
                           i\_msgv2   = ls\_item-dysid  
                         IMPORTING  
                           et\_return = lt\_return ).  
                    IF 1 = 2. MESSAGE e228 WITH gs\_0400-rol ls\_item-dysid. ENDIF.  
                  ENDIF.  
                  " Bütçe kontrolü zorunlu alan kontrolü  
                  IF ls\_item-butkn IS INITIAL.  
                    zcl\_util\_001=>add\_message(  
                         EXPORTING  
                           i\_msgty   = 'E'  
                           i\_msgid   = mc\_msgid  
                           i\_msgno   = '248'  
                           i\_msgv1   = gs\_0400-rol  
                           i\_msgv2   = ls\_item-dysid  
                         IMPORTING  
                           et\_return = lt\_return ).  
                    IF 1 = 2. MESSAGE e248 WITH gs\_0400-rol ls\_item-dysid. ENDIF.  
                  ENDIF.  
                ENDIF.  
              ENDIF.  
            ENDIF.  
  
            " FAOS rolünde ek onaycı dolu ise  
            IF gs\_0400-rol = 'FAOS' AND ls\_item-ekony IS NOT INITIAL.  
              DATA(ls\_zerony\_ret) = zcl\_ot\_tool=>zer\_onay\_kontrol( ls\_item-dysid ).  
              IF ls\_zerony\_ret-rc = 0.  
                CASE ls\_zerony\_ret-durum.  
                  WHEN '1'." Onayda değişiklik yok  
                  WHEN '3'." Onayda değişiklik var onay süreci tetiklenir  
                    zcl\_util\_001=>add\_message(  
                      EXPORTING  
                        i\_msgty   = 'W'  
                        i\_msgid   = mc\_msgid  
                        i\_msgno   = '163'  
                        i\_msgv1   = ls\_item-dysid  
                        i\_msgv2   = ls\_item-ekony  
                      IMPORTING  
                        et\_return = lt\_return ).  
                    IF 1 = 2. MESSAGE w163 WITH ls\_item-dysid ls\_item-ekony. ENDIF.  
                    gs\_0400-leave\_screen = 'X'.  
                    CONTINUE.  
                ENDCASE.  
              ENDIF.  
            ENDIF.  
  
          WHEN '12'.  " Eksik/hatalı bilgi  
          WHEN '13'.  " Satış kontrolü  
          WHEN '14'.  " E-fatura reddet  
            " 7 gün limit kontrolü  
            lv\_res\_secs\_chk = lc\_7day \* 24 \* 3600.  " Saniye olarak 7 gün  
            TRY.  
                CALL METHOD cl\_abap\_tstmp=>td\_subtract  
                  EXPORTING  
                    date1    = sy-datum  
                    time1    = sy-uzeit  
                    date2    = ls\_item-xmlrcd  
                    time2    = ls\_item-xmlrct  
                  IMPORTING  
                    res\_secs = lv\_res\_secs.  
              CATCH cx\_parameter\_invalid\_type INTO DATA(lo\_parameter\_invalid\_type).  
              CATCH cx\_parameter\_invalid\_range INTO DATA(lo\_parameter\_invalid\_range).  
            ENDTRY.  
            IF lv\_res\_secs\_chk > lv\_res\_secs\_chk.  
              zcl\_util\_001=>add\_message(  
                EXPORTING  
                  i\_msgty   = 'E'  
                  i\_msgid   = mc\_msgid  
                  i\_msgno   = '142'  
                  i\_msgv1   = lc\_7day  
                  i\_msgv2   = ls\_item-dysid  
                IMPORTING  
                  et\_return = lt\_return ).  
              IF 1 = 2. MESSAGE e142 WITH lc\_7day ls\_item-dysid. ENDIF.  
            ENDIF.  
  
            " Red işlemi için açıklama girilmiş mi kontrolü  
            IF lt\_itf\_text[] IS INITIAL.  
              zcl\_util\_001=>add\_message(  
                EXPORTING  
                  i\_msgty   = 'E'  
                  i\_msgid   = mc\_msgid  
                  i\_msgno   = '143'  
                  i\_msgv1   = ls\_item-dysid  
                IMPORTING  
                  et\_return = lt\_return ).  
              IF 1 = 2. MESSAGE e143 WITH ls\_item-dysid. ENDIF.  
            ENDIF.  
  
            " Temel fatura reddelimez kontrolü  
            " (Bu kontrol E-fatura reddet işlem tipi kontrollerinde  
            "  en sonda olmalı çünkü IDOC durumu güncelleniyor!)  
            IF ls\_item-fatur = 'EF'.  
              SELECT SINGLE profileid  
                     FROM /kspack/ef001  
                     INTO @DATA(lv\_profileid)  
                     WHERE eguid = @ls\_item-ettn  
                     AND direct = '2'.  
              IF lv\_profileid = 'TEMELFATURA'.  
                zcl\_util\_001=>add\_message(  
                  EXPORTING  
                    i\_msgty   = 'E'  
                    i\_msgid   = mc\_msgid  
                    i\_msgno   = '144'  
                    i\_msgv1   = ls\_item-dysid  
                  IMPORTING  
                    et\_return = lt\_return ).  
                IF 1 = 2. MESSAGE e144 WITH ls\_item-dysid. ENDIF.  
              ELSE.  
                update\_idoc\_statu( EXPORTING is\_item   = ls\_item  
                                             iv\_status = '80'  
                                             it\_text   = lt\_itf\_text  
                                   IMPORTING et\_return = lt\_return ).  
              ENDIF.  
            ENDIF.  
  
          WHEN '15'.  " Kep red işlemi  
            " Fatura eki kontrolü  
            REFRESH lt\_toa01.  
            SELECT \*  
                   FROM toa01  
                   INTO TABLE lt\_toa01  
                   WHERE sap\_object = zdys\_cl\_attachment=>mc\_sap\_object  
                   AND object\_id = ls\_item-dysid  
                   AND archiv\_id = ls\_zdys\_t\_012-arcdisk  
                   AND ar\_object = zdys\_cl\_attachment=>mc\_ar\_object\_faturaek.  
            zdys\_cl\_attachment=>eliminate\_deleted\_files( CHANGING ct\_toa01 = lt\_toa01 ).  
            IF lt\_toa01[] IS INITIAL.  
              MESSAGE i145 WITH ls\_item-dysid INTO lv\_text.  
              IF zcl\_util\_001=>confirm\_popup( iv\_text = lv\_text ) <> '1'.  
                zcl\_util\_001=>add\_message(  
                  EXPORTING  
                    i\_msgty   = 'E'  
                    i\_msgid   = mc\_msgid  
                    i\_msgno   = '146'  
                    i\_msgv1   = ls\_item-dysid  
                  IMPORTING  
                    et\_return = lt\_return ).  
                IF 1 = 2. MESSAGE e146 WITH ls\_item-dysid. ENDIF.  
              ENDIF.  
            ENDIF.  
  
  
          WHEN '17'.  " Tedarikçiyi kontrol et  
            IF gs\_0400-rol = 'TTS'.  
              IF ls\_item-partner IS INITIAL AND ls\_item-stcd2 IS NOT INITIAL.  
                SELECT SINGLE partner  
                       FROM dfkkbptaxnum  
                       INTO ls\_item-partner  
                       WHERE taxnum = ls\_item-stcd2  
                       AND taxtype = 'TR2'.  
                IF sy-subrc = 0.  
                  UPDATE zdys\_t\_001  
                         SET partner = ls\_item-partner  
                         WHERE dysid =  ls\_item-dysid.  
                  COMMIT WORK AND WAIT.  
                  WAIT UP TO 1 SECONDS.  
                ENDIF.  
              ENDIF.  
              MOVE-CORRESPONDING ls\_item TO ls\_zdys\_t\_001.  
              IF lo\_dys->check\_if\_vendor\_ok( EXPORTING is\_item   = ls\_zdys\_t\_001  
                                             IMPORTING et\_return = lt\_return ) = lo\_dys->mc\_isltp\_yes.  
              ELSE.  
                LOOP AT lt\_return ASSIGNING FIELD-SYMBOL(<ls\_return>) WHERE type = 'W'.  
                  <ls\_return>-type = 'E'.  
                ENDLOOP.  
              ENDIF.  
            ENDIF.  
        ENDCASE.  
  
        CHECK zcl\_util\_001=>check\_error\_msg( EXPORTING it\_return = lt\_return ) = ''.  
        gs\_0400-leave\_screen = 'X'.  
  
        " İş akışını başlat  
        REFRESH lo\_dys->mt\_return.  
        " İş akışı TTS rolünde onaylandığında gelecek yeni iş akışının  
        " 'İşlem yürüt' kontrolünü otomatik işlememesi için eklendi  
        IF NOT ( gs\_0400-isltp = '17' ).  
          lo\_dys->mv\_isltp\_selected = gs\_0400-isltp.  
        ENDIF.  
        lo\_dys->run\_workflow( EXPORTING iv\_dysid  = ls\_item-dysid ).  
        " İş akışında gösterilmelesi gereken iletilerin ekranda gösterilmesi için eklendi  
        APPEND LINES OF lo\_dys->mt\_return TO lt\_return\_all.  
  
        IF zcl\_util\_001=>check\_error\_msg( EXPORTING it\_return = lt\_return ) = ''.  
          " İşlem tipi değişikliği ve log bilgisi  
          UPDATE zdys\_t\_001  
                 SET isltp = gs\_0400-isltp  
                 WHERE dysid = ls\_item-dysid.  
          zdys\_cl\_general=>log\_dys\_process(  
            iv\_dysid     = ls\_item-dysid  
            iv\_isltp     = gs\_0400-isltp  
            iv\_fname     = 'ISLTP'  
            iv\_value\_old = ''  
            iv\_value\_new = gs\_0400-isltp ).  
  
          " Alt işlem tipi değişikliği ve log bilgisi  
          IF ls\_item-isltp\_alt <> gs\_0400-isltp\_alt.  
            UPDATE zdys\_t\_001  
                   SET isltp\_alt = gs\_0400-isltp\_alt  
                   WHERE dysid = ls\_item-dysid.  
            zdys\_cl\_general=>log\_dys\_process(  
              iv\_dysid     = ls\_item-dysid  
              iv\_isltp     = '03'  
              iv\_fname     = 'ISLTP\_ALT'  
              iv\_value\_old = ls\_item-isltp\_alt  
              iv\_value\_new = gs\_0400-isltp\_alt ).  
          ENDIF.  
  
          " Fatura işlem tipi değişikliği ve log bilgisi  
          IF ls\_item-isltp\_fat <> gs\_0400-isltp\_fat.  
            UPDATE zdys\_t\_001  
                   SET isltp\_fat = gs\_0400-isltp\_fat  
                   WHERE dysid = ls\_item-dysid.  
            zdys\_cl\_general=>log\_dys\_process(  
              iv\_dysid     = ls\_item-dysid  
              iv\_isltp     = '03'  
              iv\_fname     = 'ISLTP\_FAT'  
              iv\_value\_old = ls\_item-isltp\_fat  
              iv\_value\_new = gs\_0400-isltp\_fat ).  
          ENDIF.  
  
          " Özet işlem durumu değişikliği ve log bilgisi  
          IF ls\_item-isldr\_ozt <> gs\_0400-isldr\_ozt.  
            UPDATE zdys\_t\_001  
                   SET isldr\_ozt = gs\_0400-isldr\_ozt  
                   WHERE dysid = ls\_item-dysid.  
            zdys\_cl\_general=>log\_dys\_process(  
              iv\_dysid     = ls\_item-dysid  
              iv\_isltp     = '03'  
              iv\_fname     = 'ISLDR\_OZT'  
              iv\_value\_old = ls\_item-isldr\_ozt  
              iv\_value\_new = gs\_0400-isldr\_ozt ).  
          ENDIF.  
  
          " Satış faturasıyla onayla seçildiyse satış fatura oluşturm  
          " ve satış çıktısının alınması işlemi de yapılır  
          DO 1 TIMES.  
            CHECK gs\_0400-isltp = '11' AND gs\_0400-isltp\_alt = 'A24'.  
            " Satın alma faturasının muhasebeleştirilmiş olması gerekir  
            IF ls\_item-awtyp = 'RMRP'.  
              zdys\_cl\_general=>get\_awkey\_values(  
                EXPORTING  
                  iv\_awtyp    = ls\_item-awtyp  
                  iv\_awkey    = ls\_item-awkey  
                IMPORTING  
                  ev\_belnr\_mm = DATA(lv\_belnr\_mm)  
                  ev\_gjahr\_mm = DATA(lv\_gjahr\_mm) ).  
              CHECK lv\_belnr\_mm IS NOT INITIAL AND lv\_gjahr\_mm IS NOT INITIAL.  
              SELECT SINGLE rbstat  
                     FROM rbkp  
                     INTO @DATA(lv\_rbstat)  
                     WHERE belnr = @lv\_belnr\_mm  
                     AND gjahr = @lv\_gjahr\_mm.  
              CHECK sy-subrc = 0 AND lv\_rbstat = '5'.  
            ENDIF.  
            create\_sales\_inv( EXPORTING is\_item   = ls\_item  
                              CHANGING  ct\_return = lt\_return\_all ).  
          ENDDO.  
  
          " Onaya gönderme işlemi  
          IF gs\_0400-isltp = '11' AND  " Onayla(hatasız)  
             gs\_0400-rol = 'GOS'.  
            LOOP AT lt\_itf\_text INTO DATA(ls\_itf\_text).  
              APPEND VALUE #( line = ls\_itf\_text-tdline ) TO lt\_soli\_text.  
            ENDLOOP.  
            DATA(ls\_zeronybas\_ret) = zcl\_ot\_tool=>zer\_onay\_baslat( i\_dysid = ls\_item-dysid  
                                                                   it\_text = lt\_soli\_text[] ).  
          ENDIF.  
        ENDIF.  
  
        IF lt\_itf\_text[] IS NOT INITIAL.  
          " İş akışı yürütüldükten sonraki son işlem numarası  
          CLEAR lv\_islno\_max\_new.  
          SELECT MAX( islno )  
                 FROM zdys\_t\_002  
                 INTO lv\_islno\_max\_new  
                 WHERE dysid = ls\_item-dysid.  
          IF lv\_islno\_max\_new IS NOT INITIAL.  
            " Girilen not eklenen son işlem numarası ile ilişkilendirilir  
            lv\_fname = ls\_item-dysid && lv\_islno\_max\_new.  
            CALL FUNCTION 'CREATE\_TEXT'  
              EXPORTING  
                fid         = 'Z001'  
                flanguage   = 'T'  
                fname       = lv\_fname  
                fobject     = zdys\_cl\_general=>mc\_tdobject  
                save\_direct = 'X'  
              TABLES  
                flines      = lt\_itf\_text  
              EXCEPTIONS  
                no\_init     = 1  
                no\_save     = 2  
                OTHERS      = 3.  
          ENDIF.  
        ENDIF.  
      ENDDO.  
      APPEND LINES OF lt\_return TO lt\_return\_all.  
    ENDLOOP.  
  
    zcl\_util\_001=>show\_messages( EXPORTING it\_return = lt\_return\_all ).  
  ENDMETHOD.  
  
  METHOD save\_screen\_0500.  
    DATA: lt\_fname\_changeable TYPE tty\_fname,  
          lt\_fname\_updated    TYPE tty\_fname,  
          lt\_return\_all       TYPE bapiret2\_t,  
          ls\_zdys\_t\_001\_new   TYPE zdys\_t\_001,  
          lv\_set\_param        TYPE char50,  
          lv\_updated          TYPE xfeld.  
  
    IF gs\_0500-marka IS INITIAL AND gs\_0500-matkl IS INITIAL AND  
       gs\_0500-fattp IS INITIAL AND gs\_0500-partner IS INITIAL.  
      MESSAGE i134 DISPLAY LIKE 'E'.  
      EXIT.  
    ENDIF.  
  
    IF gs\_0500-marka IS NOT INITIAL AND gs\_0500-matkl IS NOT INITIAL.  
      MESSAGE i025 DISPLAY LIKE 'E'.  
      EXIT.  
    ENDIF.  
  
    " Girilen marka / mal grubu için iş akışı tanımlı mı kontrolü  
    IF gs\_0500-marka IS NOT INITIAL OR gs\_0500-matkl IS NOT INITIAL.  
      zdys\_cl\_general=>get\_wf\_id\_from\_md( EXPORTING iv\_marka = gs\_0500-marka  
                                                    iv\_matkl = gs\_0500-matkl  
                                          IMPORTING et\_return = DATA(lt\_return) ).  
      READ TABLE lt\_return INTO DATA(ls\_return) WITH KEY type = 'E'.  
      IF sy-subrc = 0.  
        MESSAGE ID ls\_return-id TYPE 'I' NUMBER ls\_return-number  
                WITH ls\_return-message\_v1 ls\_return-message\_v2  
                     ls\_return-message\_v3 ls\_return-message\_v4  
                DISPLAY LIKE 'E'.  
        EXIT.  
      ENDIF.  
    ENDIF.  
  
    CHECK zcl\_util\_001=>confirm\_popup( iv\_text = TEXT-002 ) = '1'.  
  
    " Kullanıcının değiştirmeye yetkisi olduğu alan isimleri  
    LOOP AT mt\_zdys\_t\_006 INTO DATA(ls\_zdys\_t\_006) WHERE fname IS NOT INITIAL.  
      COLLECT VALUE ty\_fname( fname = ls\_zdys\_t\_006-fname ) INTO lt\_fname\_changeable.  
    ENDLOOP.  
  
    LOOP AT gs\_0500-dysid\_selected INTO DATA(lv\_dysid).  
      SELECT SINGLE \*  
             FROM zdys\_t\_001  
             INTO @DATA(ls\_zdys\_t\_001\_old)  
             WHERE dysid = @lv\_dysid.  
      CHECK sy-subrc = 0.  
  
      " Değiştirilen muhatabın VKN'si DYS VKN ile uyumlu mu kontrolü  
      IF gs\_0500-partner IS NOT INITIAL AND gs\_0500-partner <> ls\_zdys\_t\_001\_old-partner.  
        CHECK check\_stcd2\_inconsistency( EXPORTING iv\_dysid   = lv\_dysid  
                                                   iv\_partner = gs\_0500-partner  
                                                   iv\_stcd2   = ls\_zdys\_t\_001\_old-stcd2  
                                         IMPORTING et\_return  = lt\_return\_all ) = 'X'.  
      ENDIF.  
  
      REFRESH lt\_fname\_updated.  
      LOOP AT lt\_fname\_changeable INTO DATA(ls\_fname\_changeable).  
        ASSIGN COMPONENT ls\_fname\_changeable-fname OF STRUCTURE ls\_zdys\_t\_001\_old TO FIELD-SYMBOL(<lv\_value\_old>).  
        CHECK sy-subrc = 0.  
        ASSIGN COMPONENT ls\_fname\_changeable-fname OF STRUCTURE gs\_0500 TO FIELD-SYMBOL(<lv\_value\_new>).  
        " Toplu güncelleme ekranında ilgili alan için yeni bir değer girildiyse(dolu olmalı!)  
        CHECK sy-subrc = 0 AND <lv\_value\_new> IS NOT INITIAL.  
        " Herhangi bir değişiklik yapıldıysa  
        CHECK <lv\_value\_old> <> <lv\_value\_new>.  
  
        REFRESH lt\_return.  
        lv\_set\_param = ls\_fname\_changeable-fname && ` = <LV\_VALUE\_NEW>`.  
        UPDATE zdys\_t\_001  
               SET (lv\_set\_param)  
               WHERE dysid = lv\_dysid.  
        COMMIT WORK.  
        IF sy-subrc = 0.  
          lv\_updated = 'X'.  
          " Güncelleme yapılan alan isimleri  
          APPEND VALUE #( fname = ls\_fname\_changeable-fname ) TO lt\_fname\_updated.  
          " İşlem detayı(yapılan değişiklik bilgisi loglanıyor)  
          zdys\_cl\_general=>log\_dys\_process(  
            iv\_dysid     = lv\_dysid  
            iv\_isltp     = '03' " Ana veri güncelle  
            iv\_fname     = ls\_fname\_changeable-fname  
            iv\_value\_old = <lv\_value\_old>  
            iv\_value\_new = <lv\_value\_new> ).  
        ENDIF.  
  
        " Yapılan değişikliklere göre diğer DYS verileri güncellenir  
        ls\_zdys\_t\_001\_new = ls\_zdys\_t\_001\_old.  
        MOVE-CORRESPONDING gs\_0500 TO ls\_zdys\_t\_001\_new.  
        adjust\_updated\_inv\_values( EXPORTING it\_fname\_updated = lt\_fname\_updated  
                                             is\_item\_old      = ls\_zdys\_t\_001\_old  
                                   IMPORTING et\_return        = lt\_return  
                                   CHANGING  cs\_item\_new      = ls\_zdys\_t\_001\_new ).  
        APPEND LINES OF lt\_return TO lt\_return\_all.  
      ENDLOOP.  
    ENDLOOP.  
  
    zcl\_util\_001=>show\_messages( EXPORTING it\_return = lt\_return\_all ).  
    " Herhangi bir kaydetme işlemi yapıldı mı kontrolü  
    IF lv\_updated = 'X'.  
      gs\_0500-leave\_screen = 'X'.  
      MESSAGE s135.  
    ELSE.  
      MESSAGE i026 DISPLAY LIKE 'W'.  
    ENDIF.  
  ENDMETHOD.  
  
  METHOD list\_process\_details.  
  
    CHECK check\_selected\_rows( EXPORTING iv\_sel\_method = 'S' ) = ''.  
  
    CLEAR ms\_item.  
    LOOP AT mt\_dysid\_selected INTO DATA(lv\_dysid\_selected).  
      READ TABLE gt\_item INTO ms\_item WITH KEY dysid = lv\_dysid\_selected.  
    ENDLOOP.  
  
    REFRESH mt\_detail.  
    zdys\_cl\_general=>get\_process\_details(  
      EXPORTING  
        iv\_dysid   = ms\_item-dysid  
      IMPORTING  
        et\_details = mt\_detail ).  
  
    CALL SCREEN 0200 STARTING AT 5 5 ENDING AT 170 25.  
  
  ENDMETHOD.  
  
  METHOD check\_before\_update\_invoice.  
  
    " 'Onay bekliyor' durum kontrolü  
    IF is\_item-durum = '02'.  " Onay bekliyor  
      zcl\_util\_001=>add\_message(  
        EXPORTING  
          i\_msgty   = 'E'  
          i\_msgid   = mc\_msgid  
          i\_msgno   = '130'  
          i\_msgv1   = is\_item-dysid  
        IMPORTING  
          et\_return = et\_return ).  
      IF 1 = 2. MESSAGE e130 WITH is\_item-dysid. ENDIF.  
    ENDIF.  
  
  ENDMETHOD.  
  
  METHOD get\_auth\_input\_for\_update.  
    REFRESH mt\_zdys\_t\_006.  
    " Manuel girilen faturalarda tüm alanlar değişikliğe açık olmalıdır  
    IF gs\_0300-manue = 'X'.  
      DATA(lt\_fields) = CAST cl\_abap\_structdescr( cl\_abap\_typedescr=>describe\_by\_name( 'ZDYS\_T\_001' ) )->get\_ddic\_field\_list( ).  
      LOOP AT lt\_fields INTO DATA(ls\_fields).  
        APPEND VALUE #( fname = ls\_fields-fieldname ) TO mt\_zdys\_t\_006.  
      ENDLOOP.  
    ELSE.  
      SELECT \*  
             FROM zdys\_t\_006  
             INTO TABLE mt\_zdys\_t\_006  
             WHERE rol IN mr\_rol  
             OR uname = sy-uname              "#EC CI\_ALL\_FIELDS\_NEEDED  
             OR ( rol = '' AND uname = '' ).    "ROL ve UNAME değeri boş olan fatura alanlarını herkes değiştirebilir  
    ENDIF.  
  ENDMETHOD.  
  
  METHOD update\_invoice\_data\_single.  
  
    CHECK check\_selected\_rows( EXPORTING iv\_sel\_method = 'S' ) = ''.  
  
    CLEAR ms\_item.  
    LOOP AT mt\_dysid\_selected INTO DATA(lv\_dysid\_selected).  
      READ TABLE gt\_item INTO ms\_item WITH KEY dysid = lv\_dysid\_selected.  
    ENDLOOP.  
  
    CHECK zdys\_cl\_general=>lock\_dysid( ms\_item-dysid ) = ''.  
  
    check\_before\_update\_invoice( EXPORTING is\_item   = ms\_item  
                                 IMPORTING et\_return = DATA(lt\_return) ).  
  
    IF zcl\_util\_001=>check\_error\_msg( EXPORTING it\_return = lt\_return ) = 'X'.  
      zcl\_util\_001=>show\_messages( EXPORTING it\_return = lt\_return ).  
      EXIT.  
    ENDIF.  
  
    CLEAR: gs\_0300, gv\_ucomm\_0300.  
    MOVE-CORRESPONDING ms\_item TO gs\_0300.  
    MOVE-CORRESPONDING ms\_item TO gs\_0300-old\_values.  
  
    " Müşteri sipariş bilgileri  
    REFRESH mt\_zdys\_t\_016.  
    SELECT \*  
           FROM zdys\_t\_016  
           INTO TABLE mt\_zdys\_t\_016  
           WHERE dysid = gs\_0300-dysid.  
  
    " Fatura ana verisinde değiştirilebilir alanların yetkileri  
    get\_auth\_input\_for\_update( ).  
  
    CALL SCREEN 0300 STARTING AT 5 5 ENDING AT 135 18.  
  
    zdys\_cl\_general=>unlock\_dysid( ms\_item-dysid ).  
  
  ENDMETHOD.  
  
  METHOD update\_invoice\_data\_multi.  
  
    CHECK check\_selected\_rows( EXPORTING iv\_sel\_method = 'M' ) = ''.  
  
    CLEAR: gs\_0500, gv\_ucomm\_0500.  
  
    LOOP AT mt\_dysid\_selected INTO DATA(lv\_dysid\_selected).  
      READ TABLE gt\_item INTO DATA(ls\_item) WITH KEY dysid = lv\_dysid\_selected.  
      CHECK sy-subrc = 0.  
      check\_before\_update\_invoice( EXPORTING is\_item   = ls\_item  
                                   IMPORTING et\_return = DATA(lt\_return) ).  
      APPEND ls\_item-dysid TO gs\_0500-dysid\_selected.  
    ENDLOOP.  
  
    DATA(lt\_return\_lock) = zdys\_cl\_general=>check\_if\_dysid\_locked( EXPORTING it\_dysid  = gs\_0500-dysid\_selected ).  
    APPEND LINES OF lt\_return\_lock TO lt\_return.  
  
    IF zcl\_util\_001=>check\_error\_msg( EXPORTING it\_return = lt\_return ) = 'X'.  
      zcl\_util\_001=>show\_messages( EXPORTING it\_return = lt\_return ).  
      EXIT.  
    ENDIF.  
  
    " Fatura ana verisinde değiştirilebilir alanların yetkileri  
    get\_auth\_input\_for\_update( ).  
  
    CALL SCREEN 0500 STARTING AT 5 5 ENDING AT 105 10.  
  
  ENDMETHOD.  
  
  METHOD adjust\_updated\_inv\_values.  
    DATA: lv\_stcd2         TYPE stcd2,  
          lv\_isaks\_restart TYPE xfeld.  
  
    IF cs\_item\_new-marka IS INITIAL AND cs\_item\_new-matkl IS INITIAL.  
      cs\_item\_new-isaks = zdys\_cl\_general=>mc\_isaks\_start.  
    ELSE.  
      cs\_item\_new-isaks = zdys\_cl\_general=>get\_wf\_id\_from\_md( EXPORTING iv\_marka = cs\_item\_new-marka  
                                                                        iv\_matkl = cs\_item\_new-matkl ).  
    ENDIF.  
  
\*    " Eğer tedarikçi güncellendi ise tabloda vergi numarası da güncellenir  
\*    IF line\_exists( it\_fname\_updated[ fname = 'PARTNER' ] ).  
\*      CLEAR lv\_stcd2.  
\*      IF cs\_item\_new-partner IS NOT INITIAL.  
\*        SELECT SINGLE taxnum  
\*               FROM dfkkbptaxnum  
\*               INTO lv\_stcd2  
\*               WHERE partner = cs\_item\_new-partner  
\*               AND taxtype = 'TR2'.  
\*      ENDIF.  
\*      UPDATE zdys\_t\_001  
\*             SET stcd2 = lv\_stcd2  
\*             WHERE dysid = cs\_item\_new-dysid.  
\*      COMMIT WORK.  
\*    ENDIF.  
  
    " Eğer marka ya da mal grubu güncellendi ise  
    IF line\_exists( it\_fname\_updated[ fname = 'MARKA' ] ) OR  
       line\_exists( it\_fname\_updated[ fname = 'MATKL' ] ).  
      " İş akışı değiştiyse yeni iş akışı baştan başlatılır  
      IF ( is\_item\_old-isaks <> cs\_item\_new-isaks ) OR  
         ( is\_item\_old-isaks = cs\_item\_new-isaks AND  
           is\_item\_old-sirno <> zdys\_cl\_general=>mc\_sirno\_start ).  
        lv\_isaks\_restart = 'X'.  
      ENDIF.  
    ENDIF.  
  
    " Eğer marka güncellendiyse(değiştirildiyse) ve mal grubu değeri dolu ise, mal grubu değeri silinir  
    " Bu işlemi marka alanı boş değilse yapmalı ( Aynı anda markanın silinip mal grubunun oldurulması)  
    IF line\_exists( it\_fname\_updated[ fname = 'MARKA' ] ).  
      SELECT SINGLE marka, matkl  
             FROM zdys\_t\_001  
             INTO (@DATA(lv\_marka\_old), @DATA(lv\_matkl\_old))  
             WHERE dysid = @cs\_item\_new-dysid.  
      IF sy-subrc = 0 AND lv\_matkl\_old IS NOT INITIAL AND lv\_marka\_old IS NOT INITIAL.  
        UPDATE zdys\_t\_001  
               SET matkl = ''  
               WHERE dysid = cs\_item\_new-dysid.  
        COMMIT WORK AND WAIT.  
        zdys\_cl\_general=>log\_dys\_process(  
          iv\_dysid     = cs\_item\_new-dysid  
          iv\_isltp     = '03' " Ana veri güncelle  
          iv\_fname     = 'MATKL'  
          iv\_value\_old = lv\_matkl\_old  
          iv\_value\_new = '' ).  
      ENDIF.  
    ENDIF.  
  
    " Eğer mal grubu güncellendiyse(değiştirildiyse) ve marka değeri dolu ise, marka değeri silinir  
    " Bu işlemi mal grubu alanı boş değilse yapmalı ( Aynı anda mal grubunun silinip markanın doldurulması)  
    IF line\_exists( it\_fname\_updated[ fname = 'MATKL' ] ).  
      CLEAR: lv\_marka\_old, lv\_matkl\_old.  
      SELECT SINGLE marka matkl  
             FROM zdys\_t\_001  
             INTO (lv\_marka\_old, lv\_matkl\_old)  
             WHERE dysid = cs\_item\_new-dysid.  
      IF sy-subrc = 0 AND lv\_marka\_old IS NOT INITIAL AND lv\_matkl\_old IS NOT INITIAL.  
        UPDATE zdys\_t\_001  
               SET marka = ''  
               WHERE dysid = cs\_item\_new-dysid.  
        COMMIT WORK AND WAIT.  
        zdys\_cl\_general=>log\_dys\_process(  
          iv\_dysid     = cs\_item\_new-dysid  
          iv\_isltp     = '03' " Ana veri güncelle  
          iv\_fname     = 'MARKA'  
          iv\_value\_old = lv\_marka\_old  
          iv\_value\_new = '' ).  
      ENDIF.  
    ENDIF.  
  
    " Eğer mevcut muhatap silindiyse, doldurulduysa ya da boş ise  
    " iş akışı ortak giderler (ORT) iş akışından baştan başlar  
    IF line\_exists( it\_fname\_updated[ fname = 'PARTNER' ] ) AND  
       ( ( is\_item\_old-partner IS NOT INITIAL AND cs\_item\_new-partner IS INITIAL ) OR  
         ( is\_item\_old-partner IS INITIAL AND cs\_item\_new-partner IS NOT INITIAL ) ).  
      lv\_isaks\_restart  = 'X'.  
      cs\_item\_new-isaks = zdys\_cl\_general=>mc\_isaks\_start.  
    ELSEIF is\_item\_old-partner IS INITIAL AND cs\_item\_new-partner IS INITIAL.  
      lv\_isaks\_restart  = 'X'.  
      cs\_item\_new-isaks = zdys\_cl\_general=>mc\_isaks\_start.  
    ENDIF.  
  
    " Eğer fatura tipi değiştiyse (ORT) iş akışından iş akışı baştan başlar  
    IF line\_exists( it\_fname\_updated[ fname = 'FATTP' ] ).  
      lv\_isaks\_restart  = 'X'.  
      cs\_item\_new-isaks = zdys\_cl\_general=>mc\_isaks\_start.  
    ENDIF.  
  
    " Muhatap/tarih/tutar/BIM no(BIM no ile çalışan tedarikçilerde)  
    " değiştirildiyse belge numara eşleştirilmesi yeniden yapılır  
    IF cs\_item\_new-awkey IS NOT INITIAL AND  
       ( line\_exists( it\_fname\_updated[ fname = 'PARTNER' ] ) OR  
         line\_exists( it\_fname\_updated[ fname = 'FATTR' ] )   OR  
         line\_exists( it\_fname\_updated[ fname = 'FATTU' ] )   OR  
         ( line\_exists( it\_fname\_updated[ fname = 'BIMNO' ] ) AND  
           zdys\_cl\_general=>check\_if\_vendor\_use\_bimno( iv\_partner = cs\_item\_new-partner  
                                                       iv\_marka   = cs\_item\_new-marka  
                                                       iv\_matkl   = cs\_item\_new-matkl ) = 'X' ) ).  
      UPDATE zdys\_t\_001  
             SET awtyp = ''  
                 awkey = ''  
             WHERE dysid = cs\_item\_new-dysid.  
      COMMIT WORK AND WAIT.  
      zdys\_cl\_general=>match\_invoices( iv\_dysid = cs\_item\_new-dysid ).  
    ENDIF.  
  
    " İlgili iş akışı yeniden başlatılıyorsa iş akış ve sıra no değerleri güncelleniyor  
    IF lv\_isaks\_restart = 'X'.  
      UPDATE zdys\_t\_001  
             SET isaks = cs\_item\_new-isaks  
                 sirno = zdys\_cl\_general=>mc\_sirno\_start  
             WHERE dysid = cs\_item\_new-dysid.  
      COMMIT WORK.  
    ENDIF.  
  
    " İş akışında herhangi bir değişiklik varsa iş akışı değişti logu atılır  
    IF is\_item\_old-isaks <> cs\_item\_new-isaks.  
      zdys\_cl\_general=>log\_dys\_process(  
        iv\_dysid     = gs\_0300-dysid  
        iv\_isltp     = '07' " İş akışı değişikliği  
        iv\_fname     = 'ISAKS'  
        iv\_value\_old = is\_item\_old-isaks  
        iv\_value\_new = cs\_item\_new-isaks ).  
    ENDIF.  
  
    IF lv\_isaks\_restart = 'X'.  
      WAIT UP TO 1 SECONDS. " Güncellenen son degerlere göre iş akışının başlatılması için eklendi  
      zdys\_cl\_general=>run\_workflow(  
        EXPORTING  
          iv\_dysid  = cs\_item\_new-dysid  
        IMPORTING  
          et\_return = et\_return ).  
    ENDIF.  
  ENDMETHOD.  
  
  METHOD save\_screen\_0300.  
    DATA: lt\_fname\_changeable TYPE tty\_fname,  
          lt\_fname\_updated    TYPE tty\_fname,  
          ls\_zdys\_t\_001       TYPE zdys\_t\_001,  
          lv\_set\_param        TYPE char50.  
  
    IF gs\_0300-marka IS NOT INITIAL AND gs\_0300-matkl IS NOT INITIAL.  
      MESSAGE i025 DISPLAY LIKE 'E'.  
      EXIT.  
    ENDIF.  
  
    " Girilen marka / mal grubu için iş akışı tanımlı mı kontrolü  
    IF gs\_0300-marka IS NOT INITIAL OR gs\_0300-matkl IS NOT INITIAL.  
      zdys\_cl\_general=>get\_wf\_id\_from\_md( EXPORTING iv\_marka = gs\_0300-marka  
                                                    iv\_matkl = gs\_0300-matkl  
                                          IMPORTING et\_return = DATA(lt\_return) ).  
      READ TABLE lt\_return INTO DATA(ls\_return) WITH KEY type = 'E'.  
      IF sy-subrc = 0.  
        MESSAGE ID ls\_return-id TYPE 'I' NUMBER ls\_return-number  
                WITH ls\_return-message\_v1 ls\_return-message\_v2  
                     ls\_return-message\_v3 ls\_return-message\_v4  
                DISPLAY LIKE 'E'.  
        EXIT.  
      ENDIF.  
    ENDIF.  
  
    CHECK zcl\_util\_001=>confirm\_popup( iv\_text = TEXT-002 ) = '1'.  
  
    " Kullanıcının değiştirmeye yetkisi olduğu alan isimleri  
    LOOP AT mt\_zdys\_t\_006 INTO DATA(ls\_zdys\_t\_006) WHERE fname IS NOT INITIAL.  
      COLLECT VALUE ty\_fname( fname = ls\_zdys\_t\_006-fname ) INTO lt\_fname\_changeable.  
    ENDLOOP.  
  
    LOOP AT lt\_fname\_changeable INTO DATA(ls\_fname\_changeable).  
      ASSIGN COMPONENT ls\_fname\_changeable-fname OF STRUCTURE gs\_0300-old\_values TO FIELD-SYMBOL(<lv\_value\_old>).  
      CHECK sy-subrc = 0.  
      ASSIGN COMPONENT ls\_fname\_changeable-fname OF STRUCTURE gs\_0300 TO FIELD-SYMBOL(<lv\_value\_new>).  
      CHECK sy-subrc = 0.  
      " Herhangi bir değişiklik yapıldıysa  
      CHECK <lv\_value\_old> <> <lv\_value\_new>.  
  
      lv\_set\_param = ls\_fname\_changeable-fname && ` = <LV\_VALUE\_NEW>`.  
      UPDATE zdys\_t\_001  
             SET (lv\_set\_param)  
             WHERE dysid = gs\_0300-dysid.  
      COMMIT WORK.  
      IF sy-subrc = 0.  
        " Güncelleme yapılan alan isimleri  
        APPEND VALUE #( fname = ls\_fname\_changeable-fname ) TO lt\_fname\_updated.  
        " İşlem detayı(yapılan değişiklik bilgisi loglanıyor)  
        zdys\_cl\_general=>log\_dys\_process(  
          iv\_dysid     = gs\_0300-dysid  
          iv\_isltp     = '03' " Ana veri güncelle  
          iv\_fname     = ls\_fname\_changeable-fname  
          iv\_value\_old = <lv\_value\_old>  
          iv\_value\_new = <lv\_value\_new> ).  
      ENDIF.  
    ENDLOOP.  
  
    " Müşteri siparişlerinde bir değişiklik yapıldıysa güncellenir  
    IF gs\_0300-order\_changed = 'X'.  
      DELETE FROM zdys\_t\_016 WHERE dysid = gs\_0300-dysid.  
      LOOP AT mt\_zdys\_t\_016 ASSIGNING FIELD-SYMBOL(<ls\_zdys\_t\_016>).  
        <ls\_zdys\_t\_016>-dysid = gs\_0300-dysid.  
        <ls\_zdys\_t\_016>-uname = sy-uname.  
        <ls\_zdys\_t\_016>-udate = sy-datum.  
        <ls\_zdys\_t\_016>-utime = sy-uzeit.  
      ENDLOOP.  
      IF mt\_zdys\_t\_016[] IS NOT INITIAL.  
        MODIFY zdys\_t\_016 FROM TABLE mt\_zdys\_t\_016.  
      ENDIF.  
      gs\_0300-leave\_screen = 'X'.  
    ENDIF.  
  
    " Yapılan değişikliklere göre diğer DYS verileri güncellenir  
    MOVE-CORRESPONDING gs\_0300 TO ls\_zdys\_t\_001.  
    adjust\_updated\_inv\_values( EXPORTING it\_fname\_updated = lt\_fname\_updated  
                                         is\_item\_old      = gs\_0300-old\_values  
                               IMPORTING et\_return        = lt\_return  
                               CHANGING  cs\_item\_new      = ls\_zdys\_t\_001 ).  
  
    " Herhangi bir kaydetme işlemi yapıldı mı kontrolü  
    IF lt\_fname\_updated[] IS NOT INITIAL.  
      gs\_0300-leave\_screen = 'X'.  
      zcl\_util\_001=>show\_messages( EXPORTING it\_return = lt\_return ).  
    ELSEIF gs\_0300-order\_changed = 'X'.  
      gs\_0300-leave\_screen = 'X'.  
      MESSAGE s079.  
    ELSE.  
      MESSAGE i026 DISPLAY LIKE 'W'.  
    ENDIF.  
  ENDMETHOD.  
  
  METHOD save\_screen\_0600.  
    DATA: lt\_itf\_text TYPE STANDARD TABLE OF tline,  
          lt\_text\_600 TYPE STANDARD TABLE OF tdline,  
          lv\_fname    TYPE thead-tdname,  
          lv\_ettn     TYPE zdys\_e\_ettn.  
  
    CHECK zcl\_util\_001=>confirm\_popup( iv\_text = TEXT-002 ) = '1'.  
  
    CALL METHOD go\_textedit\_0600->get\_text\_as\_stream  
      IMPORTING  
        text = lt\_text\_600.  
    IF lt\_text\_600[] IS NOT INITIAL.  
      CALL FUNCTION 'CONVERT\_STREAM\_TO\_ITF\_TEXT'  
        EXPORTING  
          language    = 'T'  
        TABLES  
          text\_stream = lt\_text\_600  
          itf\_text    = lt\_itf\_text.  
    ENDIF.  
  
    LOOP AT gs\_0600-dysid\_selected INTO DATA(lv\_dysid).  
      " Otomatik iade nedeni kodu  
      SELECT SINGLE ettn, partner, fatno  
             INTO @DATA(ls\_zdys\_t\_001)  
             FROM zdys\_t\_001  
             WHERE dysid = @lv\_dysid.  
      IF sy-subrc = 0.  
        CLEAR lv\_ettn.  
        IF ls\_zdys\_t\_001-ettn IS INITIAL.  
          lv\_ettn = ls\_zdys\_t\_001-partner && `-` && ls\_zdys\_t\_001-fatno.  
        ELSE.  
          lv\_ettn = ls\_zdys\_t\_001-ettn.  
        ENDIF.  
        IF lv\_ettn IS NOT INITIAL.  
          UPDATE ziade\_t\_004  
                 SET kategoridetay = gs\_0600-iadenk  
                 WHERE ettn = lv\_ettn.  
        ENDIF.  
      ENDIF.  
      " DYS iade nedeni kodu  
      UPDATE zdys\_t\_001  
             SET iadenk = gs\_0600-iadenk  
             WHERE dysid = lv\_dysid.  
      " İade nedeni uzun açıklama  
      IF lt\_itf\_text[] IS NOT INITIAL.  
        lv\_fname = lv\_dysid.  
        CALL FUNCTION 'CREATE\_TEXT'  
          EXPORTING  
            fid         = 'Z002'  
            flanguage   = 'T'  
            fname       = lv\_fname  
            fobject     = zdys\_cl\_general=>mc\_tdobject  
            save\_direct = 'X'  
          TABLES  
            flines      = lt\_itf\_text  
          EXCEPTIONS  
            no\_init     = 1  
            no\_save     = 2  
            OTHERS      = 3.  
        IF sy-subrc <> 0.  
\* Implement suitable error handling here  
        ENDIF.  
      ENDIF.  
    ENDLOOP.  
    COMMIT WORK AND WAIT.  
    MESSAGE s098.  
    gs\_0600-modified = 'X'.  
    gs\_0600-leave\_screen = 'X'.  
  ENDMETHOD.  
  
  METHOD restart\_workflow.  
    DATA: lt\_return    TYPE bapiret2\_t,  
          lv\_restarted TYPE xfeld.  
  
    " İş akışı restart yetkisi  
    AUTHORITY-CHECK OBJECT 'ZDYS\_RESTR' FOR USER sy-uname  
    ID 'ZDYS\_YTVAR' FIELD 'X'.  
    IF sy-subrc <> 0.  
      MESSAGE w024 DISPLAY LIKE 'E' WITH 'ZDYS\_RESTR'.  
      EXIT.  
    ENDIF.  
  
    CHECK check\_selected\_rows( EXPORTING iv\_sel\_method = 'M' ) = ''.  
  
    CHECK zcl\_util\_001=>confirm\_popup( iv\_text = TEXT-010 ) = '1'.  
  
    LOOP AT mt\_dysid\_selected INTO DATA(lv\_dysid\_selected).  
      READ TABLE gt\_item INTO DATA(ls\_item) WITH KEY dysid = lv\_dysid\_selected.  
      CHECK sy-subrc = 0.  
  
      CLEAR lv\_restarted.  
      DO 1 TIMES.  
        UPDATE zdys\_t\_001  
               SET isaks = zdys\_cl\_general=>mc\_isaks\_start  
                   sirno = zdys\_cl\_general=>mc\_sirno\_start  
               WHERE dysid = ls\_item-dysid.  
        CHECK sy-subrc = 0.  
        COMMIT WORK.  
        lv\_restarted = 'X'.  
  
        zdys\_cl\_general=>log\_dys\_process(  
          iv\_dysid = ls\_item-dysid  
          iv\_isltp = '06' ).  
  
        DELETE FROM zdys\_t\_003 WHERE dysid = ls\_item-dysid.  
        COMMIT WORK AND WAIT.  
  
        zdys\_cl\_general=>mv\_isltp\_selected = '06'.  " İş akışını yeniden başlat  
        zdys\_cl\_general=>run\_workflow(  
          EXPORTING  
            iv\_dysid  = ls\_item-dysid  
          IMPORTING  
            et\_return = lt\_return ).  
      ENDDO.  
  
      IF lv\_restarted = 'X'.  
        zcl\_util\_001=>add\_message(  
          EXPORTING  
            i\_msgty   = 'S'  
            i\_msgid   = mc\_msgid  
            i\_msgno   = '045'  
            i\_msgv1   = ls\_item-dysid  
          IMPORTING  
            et\_return = lt\_return ).  
        IF 1 = 2. MESSAGE s045 WITH ls\_item-dysid. ENDIF.  
      ELSE.  
        zcl\_util\_001=>add\_message(  
          EXPORTING  
            i\_msgty   = 'E'  
            i\_msgid   = mc\_msgid  
            i\_msgno   = '046'  
            i\_msgv1   = ls\_item-dysid  
          IMPORTING  
            et\_return = lt\_return ).  
        IF 1 = 2. MESSAGE e046 WITH ls\_item-dysid. ENDIF.  
      ENDIF.  
    ENDLOOP.  
  
    zcl\_util\_001=>show\_messages( EXPORTING it\_return = lt\_return ).  
  ENDMETHOD.  
  
  METHOD complete\_workflow\_manually.  
    CONSTANTS: lc\_durum\_comp TYPE zdys\_e\_durum VALUE '99'.  
  
    DATA: lt\_return TYPE bapiret2\_t.  
  
    " Durumu manuel tamamlama yetkisi var mı (Fatura sil yetkisi ile aynı nesne kullanıldı)  
    AUTHORITY-CHECK OBJECT 'ZDYS\_SIL'  
    ID 'ACTVT' FIELD '1'.  
    IF sy-subrc <> 0.  
      MESSAGE w223 DISPLAY LIKE 'E' WITH 'ZDYS\_SIL'.  
      EXIT.  
    ENDIF.  
  
    CHECK check\_selected\_rows( EXPORTING iv\_sel\_method = 'M' ) = ''.  
  
    CHECK zcl\_util\_001=>confirm\_popup( iv\_text = TEXT-071 ) = '1'.  
  
    LOOP AT mt\_dysid\_selected INTO DATA(lv\_dysid\_selected).  
      READ TABLE gt\_item INTO DATA(ls\_item) WITH KEY dysid = lv\_dysid\_selected.  
      CHECK sy-subrc = 0.  
  
      IF ls\_item-durum IN zdys\_cl\_general=>get\_completed\_status( ).  
        zcl\_util\_001=>add\_message(  
          EXPORTING  
            i\_msgty   = 'E'  
            i\_msgid   = mc\_msgid  
            i\_msgno   = '222'  
            i\_msgv1   = ls\_item-dysid  
          IMPORTING  
            et\_return = lt\_return ).  
        IF 1 = 2. MESSAGE e222 WITH ls\_item-dysid. ENDIF.  
        CONTINUE.  
      ENDIF.  
  
      UPDATE zdys\_t\_001  
             SET durum = lc\_durum\_comp  
                 cmpmn = 'X'  
             WHERE dysid = ls\_item-dysid.  
      zdys\_cl\_general=>log\_dys\_process( iv\_dysid     = ls\_item-dysid  
                                        iv\_isltp     = '03'  
                                        iv\_fname     = 'DURUM'  
                                        iv\_value\_old = ls\_item-durum  
                                        iv\_value\_new = lc\_durum\_comp ).  
      zdys\_cl\_general=>log\_dys\_process( iv\_dysid     = ls\_item-dysid  
                                        iv\_isltp     = '03'  
                                        iv\_fname     = 'CMPMN'  
                                        iv\_value\_old = ''  
                                        iv\_value\_new = 'X' ).  
      zcl\_util\_001=>add\_message(  
        EXPORTING  
          i\_msgty   = 'S'  
          i\_msgid   = mc\_msgid  
          i\_msgno   = '221'  
          i\_msgv1   = ls\_item-dysid  
        IMPORTING  
          et\_return = lt\_return ).  
      IF 1 = 2. MESSAGE s221 WITH ls\_item-dysid. ENDIF.  
      COMMIT WORK.  
    ENDLOOP.  
    zcl\_util\_001=>show\_messages( EXPORTING it\_return = lt\_return ).  
  ENDMETHOD.  
  
  METHOD display\_workflow\_log.  
    DATA: lt\_sort       TYPE lvc\_t\_sort,  
          ls\_variant    TYPE disvariant,  
          lv\_grid\_title TYPE lvc\_title.  
  
    CHECK check\_selected\_rows( EXPORTING iv\_sel\_method = 'S' ) = ''.  
  
    CLEAR ms\_item.  
    LOOP AT mt\_dysid\_selected INTO DATA(lv\_dysid\_selected).  
      READ TABLE gt\_item INTO ms\_item WITH KEY dysid = lv\_dysid\_selected.  
    ENDLOOP.  
  
    REFRESH mt\_wf\_log.  
    SELECT \*  
           FROM zdys\_t\_003  
           INTO CORRESPONDING FIELDS OF TABLE mt\_wf\_log  
           WHERE dysid = ms\_item-dysid.  
    LOOP AT mt\_wf\_log ASSIGNING FIELD-SYMBOL(<ls\_wf\_log>).  
      get\_domvalue\_text( EXPORTING iv\_domname    = 'ZDYS\_D\_KONTROL'  
                                   iv\_domvalue\_l = <ls\_wf\_log>-kontrol  
                         IMPORTING ev\_ddtext     = <ls\_wf\_log>-kontrolt ).  
      get\_domvalue\_text( EXPORTING iv\_domname    = 'ZDYS\_D\_ISLTP'  
                                   iv\_domvalue\_l = <ls\_wf\_log>-isltp  
                         IMPORTING ev\_ddtext     = <ls\_wf\_log>-isltpt ).  
    ENDLOOP.  
  
    get\_fieldcatalog( EXPORTING iv\_alv\_name = 'WFLOG'  
                      IMPORTING et\_fieldcat = DATA(lt\_fieldcat) ).  
    get\_layout( EXPORTING iv\_alv\_name = 'WFLOG'  
                IMPORTING es\_layout = DATA(ls\_layout) ).  
    ls\_variant-report = sy-repid.  
    ls\_variant-handle = 'HW'.  
  
    APPEND VALUE #( spos      = 1  
                    fieldname = 'ISLNO'  
                    up        = 'X'  
                   ) TO lt\_sort.  
  
    lv\_grid\_title = ms\_item-dysid && ` ` && TEXT-013.  
  
    CALL FUNCTION 'REUSE\_ALV\_GRID\_DISPLAY\_LVC'  
      EXPORTING  
        i\_bypassing\_buffer    = 'X'  
        i\_callback\_program    = sy-repid  
        i\_grid\_title          = lv\_grid\_title  
        is\_layout\_lvc         = ls\_layout  
        it\_fieldcat\_lvc       = lt\_fieldcat[]  
        it\_sort\_lvc           = lt\_sort[]  
        i\_save                = 'A'  
        is\_variant            = ls\_variant  
        i\_screen\_start\_column = 5  
        i\_screen\_start\_line   = 5  
        i\_screen\_end\_column   = 90  
        i\_screen\_end\_line     = 30  
      TABLES  
        t\_outtab              = mt\_wf\_log[]  
      EXCEPTIONS  
        program\_error         = 1  
        OTHERS                = 2.  
  ENDMETHOD.  
  
  METHOD display\_attachments.  
    DATA lv\_ar\_object TYPE saeobjart.  
  
    CHECK check\_selected\_rows( EXPORTING iv\_sel\_method = 'S' ) = ''.  
  
    CLEAR ms\_item.  
    LOOP AT mt\_dysid\_selected INTO DATA(lv\_dysid\_selected).  
      READ TABLE gt\_item INTO ms\_item WITH KEY dysid = lv\_dysid\_selected.  
    ENDLOOP.  
  
    " Sadece manuel eklemelerde fatura ve fatura eki ekleme çıkarma yapılabilir  
    IF ms\_item-manue = ''.  
      lv\_ar\_object = zdys\_cl\_attachment=>mc\_ar\_object\_faturaek.  
    ENDIF.  
  
    DATA(lo\_attachment) = NEW zdys\_cl\_attachment( iv\_dysid     = ms\_item-dysid  
                                                  iv\_ar\_object = lv\_ar\_object ).  
    lo\_attachment->call\_attachment\_screen( ).  
    IF lo\_attachment->ms\_data-save = 'X'.  
      lo\_attachment->save\_files\_to\_sap( IMPORTING et\_return = DATA(lt\_return) ).  
    ENDIF.  
    zcl\_util\_001=>show\_messages( EXPORTING it\_return = lt\_return ).  
  ENDMETHOD.  
  
  METHOD workflow\_process.  
    DATA: lt\_spopli TYPE TABLE OF spopli,  
          lv\_answer TYPE char01.  
  
    APPEND VALUE #( varoption = TEXT-009 selflag = 'X' ) TO lt\_spopli.  
    APPEND VALUE #( varoption = TEXT-008 ) TO lt\_spopli.  
    APPEND VALUE #( varoption = TEXT-070 ) TO lt\_spopli.  
    APPEND VALUE #( varoption = TEXT-033 ) TO lt\_spopli.  
  
    CALL FUNCTION 'POPUP\_TO\_DECIDE\_LIST'  
      EXPORTING  
        textline1          = TEXT-069  
        titel              = TEXT-007  
      IMPORTING  
        answer             = lv\_answer  
      TABLES  
        t\_spopli           = lt\_spopli  
      EXCEPTIONS  
        not\_enough\_answers = 1  
        too\_much\_answers   = 2  
        too\_much\_marks     = 3  
        OTHERS             = 4.  
    CHECK sy-subrc = 0.  
  
    CASE lv\_answer.  
      WHEN '1'. " İş akışı sürecini görüntüle  
        display\_workflow\_log( ).  
      WHEN '2'. " İş akışını yeniden başlat  
        restart\_workflow( ).  
      WHEN '3'. " İş akışını manuel tamamla  
        complete\_workflow\_manually( ).  
      WHEN '4'. " Fatura sil  
        delete\_invoice( ).  
    ENDCASE.  
  ENDMETHOD.  
  
  METHOD run\_esignature\_process.  
    DATA: lt\_item\_update TYPE TABLE OF zdys\_s\_003,  
          lt\_return      TYPE bapiret2\_t,  
          lt\_spopli      TYPE TABLE OF spopli,  
          ls\_spopli      TYPE spopli,  
          lv\_answer      TYPE c,  
          lv\_update      TYPE xfeld.  " Güncelleme yapılabilir  
  
    CHECK check\_selected\_rows( EXPORTING iv\_sel\_method = 'M' ) = ''.  
  
    LOOP AT mt\_dysid\_selected INTO DATA(lv\_dysid\_selected).  
      READ TABLE gt\_item INTO DATA(ls\_item) WITH KEY dysid = lv\_dysid\_selected.  
      " E-imza süreci değeri dolu faturalarda e-imza durumu güncellenebilir  
      IF ls\_item-eimzs IS INITIAL.  
        zcl\_util\_001=>add\_message(  
          EXPORTING  
            i\_msgty   = 'E'  
            i\_msgid   = mc\_msgid  
            i\_msgno   = '063'  
            i\_msgv1   = ls\_item-dysid  
          IMPORTING  
            et\_return = lt\_return ).  
        IF 1 = 2. MESSAGE e063 WITH ls\_item-dysid. ENDIF.  
        CONTINUE.  
      ENDIF.  
  
      " E-İmza onaylama işlemi sadece e-arşiv faturalarda yapılabilir  
      IF ls\_item-fatur <> 'EA'.  
        zcl\_util\_001=>add\_message(  
          EXPORTING  
            i\_msgty   = 'E'  
            i\_msgid   = mc\_msgid  
            i\_msgno   = '123'  
            i\_msgv1   = ls\_item-dysid  
          IMPORTING  
            et\_return = lt\_return ).  
        IF 1 = 2. MESSAGE e123 WITH ls\_item-dysid. ENDIF.  
        CONTINUE.  
      ENDIF.  
  
      " E-İmza onaylama işlemi sadece onaylama yapılmamış faturalarda yapılabilir  
      IF ls\_item-eimzd <> ''. " Henüz onaylanmadı  
        zcl\_util\_001=>add\_message(  
          EXPORTING  
            i\_msgty   = 'E'  
            i\_msgid   = mc\_msgid  
            i\_msgno   = '124'  
            i\_msgv1   = ls\_item-dysid  
          IMPORTING  
            et\_return = lt\_return ).  
        IF 1 = 2. MESSAGE e124 WITH ls\_item-dysid. ENDIF.  
        CONTINUE.  
      ENDIF.  
  
      APPEND ls\_item TO lt\_item\_update.  
    ENDLOOP.  
  
    IF zcl\_util\_001=>check\_error\_msg( EXPORTING it\_return = lt\_return ) = 'X'.  
      zcl\_util\_001=>show\_messages( EXPORTING it\_return = lt\_return ).  
      EXIT.  
    ENDIF.  
  
    SELECT \*  
           FROM dd07t  
           INTO TABLE @DATA(lt\_dd07t)  
           WHERE domname = 'ZDYS\_D\_EIMZD'  
           AND ddlanguage = 'T'.  
  
    LOOP AT lt\_dd07t INTO DATA(ls\_dd07t).  
      CLEAR ls\_spopli.  
      ls\_spopli-varoption = ls\_dd07t-ddtext.  
      APPEND ls\_spopli TO lt\_spopli.  
    ENDLOOP.  
  
    CALL FUNCTION 'POPUP\_TO\_DECIDE\_LIST'  
      EXPORTING  
        textline1          = TEXT-028  
        titel              = TEXT-027  
      IMPORTING  
        answer             = lv\_answer  
      TABLES  
        t\_spopli           = lt\_spopli  
      EXCEPTIONS  
        not\_enough\_answers = 1  
        too\_much\_answers   = 2  
        too\_much\_marks     = 3  
        OTHERS             = 4.  
    CHECK sy-subrc = 0.  
  
    READ TABLE lt\_spopli INTO ls\_spopli WITH KEY selflag = 'X'.  
    CHECK sy-subrc = 0.  
    READ TABLE lt\_dd07t INTO ls\_dd07t WITH KEY ddtext = ls\_spopli-varoption.  
    CHECK sy-subrc = 0.  
  
    CASE ls\_dd07t-domvalue\_l.  
      WHEN ' '.  " Henüz onaylanmadı  
        lv\_update = 'X'.  
      WHEN 'M'.  " Manuel onaylandı  
        lv\_update = 'X'.  
      WHEN 'O'.  " Otomatik onaylandı  
        lv\_update = 'X'.  
      WHEN 'R'.  " Reddedildi  
        " Reddetme işlemi varsa seçili faturalardan fatura tarihi 7  
        " günü geçmiş olan var mı kontrol edilip teyit alınır  
        lv\_update = 'X'.  
      WHEN OTHERS.  
        EXIT.  
    ENDCASE.  
  
    IF lv\_update = 'X'.  
      LOOP AT lt\_item\_update INTO DATA(ls\_item\_update).  
        UPDATE zdys\_t\_001  
               SET eimzd = ls\_dd07t-domvalue\_l  
               WHERE dysid = ls\_item\_update-dysid.  
      ENDLOOP.  
      COMMIT WORK AND WAIT.  
      MESSAGE s067 WITH ls\_dd07t-ddtext.  
    ELSE.  
      MESSAGE s068 WITH ls\_dd07t-ddtext DISPLAY LIKE 'E'.  
    ENDIF.  
  ENDMETHOD.  
  
  METHOD run\_expense\_invoice.  
  
    CHECK check\_selected\_rows( EXPORTING iv\_sel\_method = 'S' ) = ''.  
  
    CLEAR ms\_item.  
    LOOP AT mt\_dysid\_selected INTO DATA(lv\_dysid\_selected).  
      READ TABLE gt\_item INTO ms\_item WITH KEY dysid = lv\_dysid\_selected.  
    ENDLOOP.  
  
    IF ms\_item-isaks <> 'GDR'.  
      MESSAGE i080 DISPLAY LIKE 'E' WITH ms\_item-dysid.  
      EXIT.  
    ENDIF.  
  
    IF iv\_dispmode = 'X'. " Görüntüle modu  
    ELSE.  
      IF zdys\_cl\_general=>check\_if\_inv\_completed( ms\_item-durum  ) = 'X'.  
        MESSAGE i069 WITH ms\_item-dysid DISPLAY LIKE 'E'.  
        EXIT.  
      ENDIF.  
      " 'Onay bekliyor' durum kontrolü  
      IF ms\_item-durum = '02'.  " Onay bekliyor  
        MESSAGE i130 WITH ms\_item-dysid DISPLAY LIKE 'E'.  
        EXIT.  
      ENDIF.  
    ENDIF.  
  
    CALL FUNCTION 'ZDYS\_CALL\_EXPENSE\_SCREEN'  
      EXPORTING  
        iv\_dysid    = ms\_item-dysid  
        iv\_dispmode = iv\_dispmode.  
  
  ENDMETHOD.  
  
  METHOD add\_order.  
    DATA: lt\_fields     TYPE TABLE OF sval,  
          lv\_returncode TYPE c.  
  
    APPEND VALUE #( tabname   = 'ZDYS\_T\_016'  
                    fieldname = 'SIPNO'  
                    field\_obl = 'X'  
                   ) TO lt\_fields.  
    CALL FUNCTION 'POPUP\_GET\_VALUES'  
      EXPORTING  
        popup\_title     = TEXT-035  
      IMPORTING  
        returncode      = lv\_returncode  
      TABLES  
        fields          = lt\_fields  
      EXCEPTIONS  
        error\_in\_fields = 1  
        OTHERS          = 2.  
    CHECK lv\_returncode <> 'A'.  
    READ TABLE lt\_fields INTO DATA(ls\_fields) WITH KEY fieldname = 'SIPNO'.  
    CHECK sy-subrc = 0.  
  
    READ TABLE mt\_zdys\_t\_016 TRANSPORTING NO FIELDS WITH KEY sipno = ls\_fields-value.  
    IF sy-subrc = 0.  
      MESSAGE i078 DISPLAY LIKE 'E' WITH ls\_fields-value.  
      EXIT.  
    ELSE.  
      APPEND VALUE #( sipno = ls\_fields-value ) TO mt\_zdys\_t\_016.  
    ENDIF.  
  
    rv\_modified = 'X'.  
  ENDMETHOD.  
  
  METHOD delete\_order.  
    go\_alv\_grid\_0300->get\_selected\_rows( IMPORTING et\_row\_no = DATA(lt\_row\_no) ).  
    IF lines( lt\_row\_no ) = 0.  
      MESSAGE i002 DISPLAY LIKE 'E'.  
      EXIT.  
    ENDIF.  
  
    LOOP AT lt\_row\_no INTO DATA(ls\_row\_no).  
      DELETE mt\_zdys\_t\_016 INDEX ls\_row\_no-row\_id.  
    ENDLOOP.  
    rv\_modified = 'X'.  
  ENDMETHOD.  
  
  METHOD delete\_invoice.  
    DATA: lt\_return     TYPE bapiret2\_t,  
          ls\_zdys\_t\_030 TYPE zdys\_t\_030.  
  
    " DYS silme yetkisi var mı  
    AUTHORITY-CHECK OBJECT 'ZDYS\_SIL'  
    ID 'ACTVT' FIELD '1'.  
    IF sy-subrc <> 0.  
      MESSAGE w155 DISPLAY LIKE 'E' WITH 'ZDYS\_SIL'.  
      EXIT.  
    ENDIF.  
  
    CHECK check\_selected\_rows( EXPORTING iv\_sel\_method = 'S' ) = ''.  
  
    CLEAR ms\_item.  
    LOOP AT mt\_dysid\_selected INTO DATA(lv\_dysid\_selected).  
      READ TABLE gt\_item INTO ms\_item WITH KEY dysid = lv\_dysid\_selected.  
    ENDLOOP.  
    CHECK ms\_item-dysid IS NOT INITIAL.  
  
    CHECK zcl\_util\_001=>confirm\_popup( iv\_text = TEXT-034 ) = '1'.  
  
    DELETE FROM zdys\_t\_001 WHERE dysid = ms\_item-dysid.  
    IF sy-subrc = 0.  
      CLEAR ls\_zdys\_t\_030.  
      MOVE-CORRESPONDING ms\_item TO ls\_zdys\_t\_030.  
      ls\_zdys\_t\_030-dldat = sy-datum.  
      ls\_zdys\_t\_030-dlzet = sy-uzeit.  
      ls\_zdys\_t\_030-dlnam = sy-uname.  
      MODIFY zdys\_t\_030 FROM ls\_zdys\_t\_030.  
      COMMIT WORK AND WAIT.  
      zcl\_util\_001=>add\_message(  
        EXPORTING  
          i\_msgty   = 'S'  
          i\_msgid   = mc\_msgid  
          i\_msgno   = '084'  
          i\_msgv1   = ms\_item-dysid  
        IMPORTING  
          et\_return = lt\_return ).  
      IF 1 = 2. MESSAGE e084 WITH ms\_item-dysid. ENDIF.  
    ENDIF.  
  
    zcl\_util\_001=>show\_messages( EXPORTING it\_return = lt\_return ).  
  ENDMETHOD.  
  
  METHOD run\_return\_reason.  
    DATA lt\_return TYPE bapiret2\_t.  
  
    CHECK check\_selected\_rows( EXPORTING iv\_sel\_method = 'M' ) = ''.  
  
    CLEAR: gs\_0600, gv\_ucomm\_0600.  
    LOOP AT mt\_dysid\_selected INTO DATA(lv\_dysid\_selected).  
      READ TABLE gt\_item INTO DATA(ls\_item) WITH KEY dysid = lv\_dysid\_selected.  
      CHECK sy-subrc = 0.  
      IF ls\_item-fattp = 'MI'.  
        APPEND ls\_item-dysid TO gs\_0600-dysid\_selected.  
      ELSE.  
        zcl\_util\_001=>add\_message(  
          EXPORTING  
            i\_msgty   = 'E'  
            i\_msgid   = mc\_msgid  
            i\_msgno   = '099'  
            i\_msgv1   = ls\_item-dysid  
          IMPORTING  
            et\_return = lt\_return ).  
        IF 1 = 2. MESSAGE e099 WITH ls\_item-dysid. ENDIF.  
      ENDIF.  
    ENDLOOP.  
  
    IF zcl\_util\_001=>check\_error\_msg( EXPORTING it\_return = lt\_return ) = 'X'.  
      zcl\_util\_001=>show\_messages( EXPORTING it\_return = lt\_return ).  
      EXIT.  
    ENDIF.  
  
    CALL SCREEN 0600 STARTING AT 5 5 ENDING AT 65 15.  
    rv\_modified = gs\_0600-modified.  
  ENDMETHOD.  
  
  METHOD run\_auto\_return.  
    DATA: lr\_dysid TYPE RANGE OF zdys\_e\_dysid,  
          lr\_fattr TYPE RANGE OF zdys\_e\_fattr.  
  
    DATA: lv\_no\_popup\_7days TYPE xfeld,  
          lv\_no\_popup       TYPE xfeld.  
  
    CHECK check\_selected\_rows( EXPORTING iv\_sel\_method = 'M' ) = ''.  
  
    APPEND VALUE #( sign = 'I' option = 'BT' low = '20000101' high = '99991231' ) TO lr\_fattr.  
    LOOP AT mt\_dysid\_selected INTO DATA(lv\_dysid\_selected).  
      READ TABLE gt\_item INTO DATA(ls\_item) WITH KEY dysid = lv\_dysid\_selected.  
      CHECK sy-subrc = 0.  
      APPEND VALUE #( sign = 'I' option = 'EQ' low = ls\_item-dysid ) TO lr\_dysid.  
    ENDLOOP.  
  
    " MIR4 ile çağırılan batch input ile çağırılan Z\_SDGEN\_EF\_VERGI\_KONTROL  
    " fonksiyonu içinde popup'ı çıkartmaması için memory'e atılan parametre  
    lv\_no\_popup = 'X'.  
    EXPORT lv\_no\_popup FROM lv\_no\_popup TO MEMORY ID 'LV\_NO\_POPUP'.  
  
    " ZINVOICE\_UPDATE badisi CHANGE\_AT\_SAVE methodu içinde yapılan  
    " 'Fatura tarihinden 7 gün öncesine ait olma durumu' kontrolündeki pop-up  
    " batch-input'u bozduğu için verilen değişkenle bu pop-up atlatıldı.  
    " ZDYS0001'den onaylanan faturalar bu 7 gün kontrolü için kullanıcı inisiyatifindedir  
    lv\_no\_popup\_7days = 'X'.  
    EXPORT lv\_no\_popup\_7days FROM lv\_no\_popup\_7days TO MEMORY ID 'LV\_NO\_POPUP\_7DAYS'.  
  
    SUBMIT ziade\_p\_0001 USING SELECTION-SCREEN '1000'  
                        WITH s\_fattr IN lr\_fattr  
                        WITH s\_dysid IN lr\_dysid  
                        AND RETURN.  
    rv\_modified = 'X'.  
  ENDMETHOD.  
  
  METHOD run\_update\_md.  
    TYPES: BEGIN OF ty\_sel\_scr,  
             p\_bimno TYPE xfeld,  
             p\_tasbl TYPE xfeld,  
             p\_sipno TYPE xfeld,  
           END OF ty\_sel\_scr.  
    DATA: lr\_dysid   TYPE RANGE OF zdys\_t\_001-dysid,  
          lt\_spopli  TYPE TABLE OF spopli,  
          ls\_sel\_scr TYPE ty\_sel\_scr,  
          lv\_answer  TYPE char01.  
  
    CONSTANTS: lc\_tcode TYPE tstc-tcode VALUE 'ZDYS0004'.  
  
    CALL FUNCTION 'AUTHORITY\_CHECK\_TCODE'  
      EXPORTING  
        tcode  = lc\_tcode  
      EXCEPTIONS  
        ok     = 1  
        not\_ok = 2.  
    IF sy-subrc = 2.  
      MESSAGE i077(s#) WITH lc\_tcode DISPLAY LIKE 'E'.  
      EXIT.  
    ENDIF.  
  
    CHECK check\_selected\_rows( EXPORTING iv\_sel\_method = 'M' ) = ''.  
  
    LOOP AT mt\_dysid\_selected INTO DATA(lv\_dysid\_selected).  
      READ TABLE gt\_item INTO DATA(ls\_item) WITH KEY dysid = lv\_dysid\_selected.  
      CHECK sy-subrc = 0.  
      APPEND VALUE #( sign   = 'I'  
                      option = 'EQ'  
                      low    = ls\_item-dysid ) TO lr\_dysid.  
    ENDLOOP.  
  
    APPEND VALUE #( selflag = 'X' varoption = TEXT-045 ) TO lt\_spopli.  
    APPEND VALUE #( selflag = ' ' varoption = TEXT-046 ) TO lt\_spopli.  
    APPEND VALUE #( selflag = ' ' varoption = TEXT-047 ) TO lt\_spopli.  
  
    CALL FUNCTION 'POPUP\_TO\_DECIDE\_LIST'  
      EXPORTING  
        textline1          = TEXT-043  
        titel              = TEXT-044  
      IMPORTING  
        answer             = lv\_answer  
      TABLES  
        t\_spopli           = lt\_spopli  
      EXCEPTIONS  
        not\_enough\_answers = 1  
        too\_much\_answers   = 2  
        too\_much\_marks     = 3  
        OTHERS             = 4.  
    CASE lv\_answer.  
      WHEN '1'. " BIM Numarası  
        ls\_sel\_scr-p\_bimno = 'X'.  
      WHEN '2'. " Taşıma Belgesi  
        ls\_sel\_scr-p\_tasbl = 'X'.  
      WHEN '3'. " Sipariş Numarası  
        ls\_sel\_scr-p\_sipno = 'X'.  
      WHEN OTHERS.  
        EXIT.  
    ENDCASE.  
  
    SUBMIT zdys\_p\_0004 VIA SELECTION-SCREEN  
                       WITH p\_submit = 'X'  
                       WITH s\_dysid IN lr\_dysid  
                       WITH p\_bimno = ls\_sel\_scr-p\_bimno  
                       WITH p\_tasbl = ls\_sel\_scr-p\_tasbl  
                       WITH p\_sipno = ls\_sel\_scr-p\_sipno.  
    rv\_modified = 'X'.  
  ENDMETHOD.  
  
  METHOD call\_rebilling\_screen.  
  
    BREAK-POINT ID zdys\_rebilling.  
    IF sy-sysid = 'ZHP'.  
      DATA: lr\_dysid TYPE RANGE OF zdys\_t\_001-dysid.  
  
      CHECK check\_selected\_rows( EXPORTING iv\_sel\_method = 'S' ) = ''.  
  
      CLEAR ms\_item.  
      LOOP AT mt\_dysid\_selected INTO DATA(lv\_dysid\_selected).  
        READ TABLE gt\_item INTO ms\_item WITH KEY dysid = lv\_dysid\_selected.  
      ENDLOOP.  
  
      " İş akışı kontrolü  
      IF ms\_item-isaks = 'GDR' OR ms\_item-isaks = 'ORT'.  
        MESSAGE w136 WITH ms\_item-isaks DISPLAY LIKE 'E'.  
        EXIT.  
      ENDIF.  
  
      APPEND VALUE #( sign = 'I'  
                      option = 'EQ'  
                      low = ms\_item-dysid  
                     ) TO lr\_dysid.  
      SUBMIT zmmgen\_p\_0003 USING SELECTION-SCREEN '1000'  
                           WITH s\_dysid IN lr\_dysid  
                           AND RETURN.  
      rv\_called = 'X'.  
  
    ELSE.  
  
      DATA: lt\_spopli TYPE TABLE OF spopli,  
            lt\_return TYPE bapiret2\_t,  
            lt\_dysid  TYPE zdys\_tt\_003,  
            ls\_param  TYPE zdys\_s\_027,  
            lv\_answer TYPE char01.  
  
      APPEND VALUE #( varoption = TEXT-064 selflag = 'X' ) TO lt\_spopli.  
      APPEND VALUE #( varoption = TEXT-065 ) TO lt\_spopli.  
  
      CALL FUNCTION 'POPUP\_TO\_DECIDE\_LIST'  
        EXPORTING  
          textline1          = TEXT-072  
          titel              = TEXT-063  
        IMPORTING  
          answer             = lv\_answer  
        TABLES  
          t\_spopli           = lt\_spopli  
        EXCEPTIONS  
          not\_enough\_answers = 1  
          too\_much\_answers   = 2  
          too\_much\_marks     = 3  
          OTHERS             = 4.  
      CHECK sy-subrc = 0.  
  
      CASE lv\_answer.  
        WHEN '1'.  " Tekil işlem yansıtma detay ekranı  
          CHECK check\_selected\_rows( EXPORTING iv\_sel\_method = 'S' ) = ''.  
          READ TABLE mt\_dysid\_selected INTO lv\_dysid\_selected INDEX 1.  
          READ TABLE gt\_item INTO DATA(ls\_item) WITH KEY dysid = lv\_dysid\_selected.  
          IF zdys\_cl\_rebilling=>check\_if\_isaks\_ok( EXPORTING iv\_dysid  = ls\_item-dysid  
                                                             iv\_isaks  = ls\_item-isaks  
                                                   IMPORTING et\_return = lt\_return ) = ''.  
            CALL FUNCTION 'ZDYS\_CALL\_REBILLING\_SCREEN'  
              EXPORTING  
                iv\_dysid = lv\_dysid\_selected  
                iv\_popup = 'X'.  
            rv\_called = 'X'.  
          ELSE.  
            zcl\_util\_001=>show\_messages( EXPORTING it\_return = lt\_return ).  
          ENDIF.  
  
        WHEN '2'.  " Excel ile toplu aktarım yansıtma ekranı  
          REFRESH mt\_dysid\_selected.  
          go\_alv\_grid\_0100->get\_selected\_rows( IMPORTING et\_row\_no = DATA(lt\_row\_no) ).  
          LOOP AT lt\_row\_no INTO DATA(ls\_row\_no).  
            READ TABLE gt\_item INTO ls\_item INDEX ls\_row\_no-row\_id.  
            CHECK sy-subrc = 0.  
            IF zdys\_cl\_rebilling=>check\_if\_isaks\_ok( EXPORTING iv\_dysid  = ls\_item-dysid  
                                                               iv\_isaks  = ls\_item-isaks  
                                                     IMPORTING et\_return = lt\_return ) = ''.  
              APPEND ls\_item-dysid TO mt\_dysid\_selected.  
            ENDIF.  
          ENDLOOP.  
          IF zcl\_util\_001=>check\_error\_msg( EXPORTING it\_return = lt\_return ) = ''.  
            ls\_param-t\_dysid = mt\_dysid\_selected.  
            CALL FUNCTION 'ZDYS\_CALL\_EXCEL\_PARAM\_SCREEN'  
              CHANGING  
                cs\_param = ls\_param.  
            IF ls\_param-ok = 'X'.  
              CASE 'X'.  
                WHEN ls\_param-download.  
                  zdys\_cl\_rebilling=>download\_excel\_template( EXPORTING it\_dysid  = mt\_dysid\_selected  
                                                                        is\_param  = ls\_param  
                                                              IMPORTING et\_return = lt\_return ).  
                WHEN ls\_param-upload.  
                  zdys\_cl\_rebilling=>upload\_excel\_template( EXPORTING is\_param  = ls\_param  
                                                            IMPORTING et\_return = lt\_return ).  
              ENDCASE.  
              zcl\_util\_001=>show\_messages( EXPORTING it\_return = lt\_return ).  
              rv\_called = 'X'.  
            ENDIF.  
          ELSE.  
            zcl\_util\_001=>show\_messages( EXPORTING it\_return = lt\_return ).  
          ENDIF.  
      ENDCASE.  
    ENDIF.  
  
  
  ENDMETHOD.  
  
  METHOD call\_ltxt\_screen.  
    CLEAR gs\_0700.  
    gs\_0700-item = is\_item.  
    get\_ltxt\_data( iv\_dysid = gs\_0700-item-dysid ).  
    CALL SCREEN 0700 STARTING AT 5 5 ENDING AT 150 20.  
  ENDMETHOD.  
  
  METHOD get\_ltxt\_data.  
    DATA: lt\_lines  TYPE TABLE OF tline,  
          lv\_tdname TYPE tdobname,  
          lv\_islno  TYPE zdys\_e\_islno.  
  
    REFRESH mt\_ltxt.  
    lv\_tdname = iv\_dysid && '%'.  
    SELECT \*  
           FROM stxh  
           INTO CORRESPONDING FIELDS OF TABLE mt\_ltxt  
           WHERE tdobject = zdys\_cl\_general=>mc\_tdobject  
           AND tdname LIKE lv\_tdname  
           AND tdid LIKE 'Z%'.  
    SORT mt\_ltxt BY tdfdate tdftime.  
  
    LOOP AT mt\_ltxt ASSIGNING FIELD-SYMBOL(<ls\_ltxt>).  
      ADD 1 TO lv\_islno.  
      <ls\_ltxt>-islno = lv\_islno.  
      REFRESH lt\_lines.  
      CALL FUNCTION 'READ\_TEXT'  
        EXPORTING  
          id                      = <ls\_ltxt>-tdid  
          language                = <ls\_ltxt>-tdspras  
          name                    = <ls\_ltxt>-tdname  
          object                  = <ls\_ltxt>-tdobject  
        TABLES  
          lines                   = lt\_lines  
        EXCEPTIONS  
          id                      = 1  
          language                = 2  
          name                    = 3  
          not\_found               = 4  
          object                  = 5  
          reference\_check         = 6  
          wrong\_access\_to\_archive = 7  
          OTHERS                  = 8.  
      READ TABLE lt\_lines INTO DATA(ls\_lines) INDEX 1.  
      IF sy-subrc = 0.  
        <ls\_ltxt>-acklm = ls\_lines-tdline(30) && `...`.  
      ENDIF.  
    ENDLOOP.  
  ENDMETHOD.  
  
  METHOD create\_new\_ltxt.  
    DATA: lt\_txwnote TYPE TABLE OF txw\_note,  
          lt\_lines   TYPE TABLE OF tline,  
          ls\_header  TYPE thead,  
          lv\_tdname  TYPE tdobname,  
          lv\_islno   TYPE zdys\_e\_islno.  
  
    CALL FUNCTION 'TXW\_TEXTNOTE\_EDIT'  
      EXPORTING  
        edit\_mode = 'X'  
      TABLES  
        t\_txwnote = lt\_txwnote.  
  
    CHECK lt\_txwnote[] IS NOT INITIAL.  
    LOOP AT lt\_txwnote INTO DATA(ls\_txwnote).  
      APPEND VALUE #( tdline = ls\_txwnote-line ) TO lt\_lines.  
    ENDLOOP.  
  
    lv\_tdname = iv\_dysid && '%'.  
    SELECT MAX( tdname )  
           FROM stxh  
           INTO @DATA(lv\_tdname\_max)  
           WHERE tdobject = @zdys\_cl\_general=>mc\_tdobject  
           AND tdname LIKE @lv\_tdname  
           AND tdid = 'Z003'.  
    IF lv\_tdname\_max IS NOT INITIAL.  
      lv\_islno = lv\_tdname\_max+10(\*).  
    ENDIF.  
    ADD 1 TO lv\_islno.  
  
    ls\_header-tdobject = zdys\_cl\_general=>mc\_tdobject.  
    ls\_header-tdid     = 'Z003'.  
    ls\_header-tdname   = iv\_dysid && lv\_islno.  
    ls\_header-tdspras  = 'TR'.  
    CALL FUNCTION 'SAVE\_TEXT'  
      EXPORTING  
        header   = ls\_header  
      TABLES  
        lines    = lt\_lines  
      EXCEPTIONS  
        id       = 1  
        language = 2  
        name     = 3  
        object   = 4  
        OTHERS   = 5.  
    IF sy-subrc = 0.  
      rv\_modified = 'X'.  
      CALL FUNCTION 'COMMIT\_TEXT'.  
      MESSAGE s137.  
    ENDIF.  
  ENDMETHOD.  
  
  METHOD display\_conf\_hierarchy.  
    DATA: lo\_appr       TYPE REF TO zcl\_es\_appr,  
          lt\_hier        TYPE zsd\_es\_s038\_tab,  
          ls\_variant    TYPE disvariant,  
          lv\_grid\_title TYPE lvc\_title.  
  
    SELECT SINGLE MAX( aprid )  
           INTO @DATA(lv\_aprid)  
           FROM zsd\_es\_t030h  
           WHERE awref = 'DYSID'  
          AND awkey = @iv\_dysid.  
    IF lv\_aprid IS INITIAL.  
      BREAK-POINT ID zdys\_rebilling.  
      IF sy-sysid = 'ZHP'.  
        SELECT SINGLE MAX( aprid )  
               INTO lv\_aprid  
               FROM zmmgen\_t\_001  
               WHERE dysid = iv\_dysid.  
      ELSE.  
        SELECT SINGLE MAX( aprid )  
               INTO lv\_aprid  
               FROM zdys\_t\_031  
               WHERE dysid = iv\_dysid.  
      ENDIF.  
    ENDIF.  
    CHECK lv\_aprid IS NOT INITIAL.  
    CREATE OBJECT lo\_appr  
      EXPORTING  
        i\_aprid = lv\_aprid.  
    lo\_appr->get\_approval\_hierarchy( IMPORTING et\_hier = lt\_hier[] ).  
  
    get\_fieldcatalog( EXPORTING iv\_alv\_name = 'CHIER'  
                      IMPORTING et\_fieldcat = DATA(lt\_fieldcat) ).  
    get\_layout( EXPORTING iv\_alv\_name = 'CHIER'  
                IMPORTING es\_layout = DATA(ls\_layout) ).  
    ls\_variant-report = sy-repid.  
    ls\_variant-handle = 'HR'.  
    lv\_grid\_title = ms\_item-dysid && ` ` && TEXT-050.  
  
    CALL FUNCTION 'REUSE\_ALV\_GRID\_DISPLAY\_LVC'  
      EXPORTING  
        i\_bypassing\_buffer    = 'X'  
        i\_callback\_program    = sy-repid  
        i\_grid\_title          = lv\_grid\_title  
        is\_layout\_lvc         = ls\_layout  
        it\_fieldcat\_lvc       = lt\_fieldcat[]  
        i\_save                = 'A'  
        is\_variant            = ls\_variant  
        i\_screen\_start\_column = 5  
        i\_screen\_start\_line   = 5  
        i\_screen\_end\_column   = 120  
        i\_screen\_end\_line     = 30  
      TABLES  
        t\_outtab              = lt\_hier[]  
      EXCEPTIONS  
        program\_error         = 1  
        OTHERS                = 2.  
  ENDMETHOD.  
  
  METHOD display\_conf\_history.  
    DATA: lo\_appr       TYPE REF TO zcl\_es\_appr,  
          lt\_hist       TYPE TABLE OF zsd\_es\_s037,  
          ls\_variant    TYPE disvariant,  
          lv\_grid\_title TYPE lvc\_title.  
  
    SELECT SINGLE MAX( aprid )  
           INTO @DATA(lv\_aprid)  
           FROM zsd\_es\_t030h  
           WHERE awref = 'DYSID'  
          AND awkey = @iv\_dysid.  
    IF lv\_aprid IS INITIAL.  
      BREAK-POINT ID zdys\_rebilling.  
      IF sy-sysid = 'ZHP'.  
        SELECT SINGLE MAX( aprid )  
               INTO lv\_aprid  
               FROM zmmgen\_t\_001  
               WHERE dysid = iv\_dysid.  
      ELSE.  
        SELECT SINGLE MAX( aprid )  
               INTO lv\_aprid  
               FROM zdys\_t\_031  
               WHERE dysid = iv\_dysid.  
      ENDIF.  
    ENDIF.  
    CHECK lv\_aprid IS NOT INITIAL.  
    CREATE OBJECT lo\_appr  
      EXPORTING  
        i\_aprid = lv\_aprid.  
    lo\_appr->get\_approval\_log( IMPORTING et\_log = lt\_hist[] ).  
  
    get\_fieldcatalog( EXPORTING iv\_alv\_name = 'CHIST'  
                      IMPORTING et\_fieldcat = DATA(lt\_fieldcat) ).  
    get\_layout( EXPORTING iv\_alv\_name = 'CHIST'  
                IMPORTING es\_layout = DATA(ls\_layout) ).  
    ls\_variant-report = sy-repid.  
    ls\_variant-handle = 'HS'.  
    lv\_grid\_title = ms\_item-dysid && ` ` && TEXT-051.  
  
    CALL FUNCTION 'REUSE\_ALV\_GRID\_DISPLAY\_LVC'  
      EXPORTING  
        i\_bypassing\_buffer    = 'X'  
        i\_callback\_program    = sy-repid  
        i\_grid\_title          = lv\_grid\_title  
        is\_layout\_lvc         = ls\_layout  
        it\_fieldcat\_lvc       = lt\_fieldcat[]  
        i\_save                = 'A'  
        is\_variant            = ls\_variant  
        i\_screen\_start\_column = 5  
        i\_screen\_start\_line   = 5  
        i\_screen\_end\_column   = 120  
        i\_screen\_end\_line     = 30  
      TABLES  
        t\_outtab              = lt\_hist[]  
      EXCEPTIONS  
        program\_error         = 1  
        OTHERS                = 2.  
  ENDMETHOD.  
  
  METHOD display\_gib\_xml.  
    DATA: lt\_rsparams TYPE TABLE OF rsparams,  
          lv\_docnum   TYPE edi\_docnum.  
  
    CASE is\_item-fatur.  
      WHEN 'EA'.  " E-arşiv fatura  
        SELECT SINGLE docnum  
               FROM /kspack/ea001  
               INTO lv\_docnum  
               WHERE eguid = is\_item-ettn  
               AND direct = '2'.  
      WHEN 'EF'.  " E-fatura  
        SELECT SINGLE docnum  
               FROM /kspack/ef001  
               INTO lv\_docnum  
               WHERE eguid = is\_item-ettn  
               AND direct = '2'.  
    ENDCASE.  
  
    CHECK lv\_docnum IS NOT INITIAL.  
  
    APPEND VALUE #( selname = 'DOCNUM'  
                    kind    = 'S'  
                    sign    = 'I'  
                    option  = 'EQ'  
                    low     = lv\_docnum  
                   ) TO lt\_rsparams.  
    APPEND VALUE #( selname = 'CREDAT'  
                    kind    = 'S'  
                    sign    = 'I'  
                    option  = 'BT'  
                    low     = '19000101'  
                    high    = '99991231'  
                   ) TO lt\_rsparams.  
  
    SUBMIT rseidoc2 WITH SELECTION-TABLE lt\_rsparams  
                    AND RETURN.  
  ENDMETHOD.  
  
  METHOD update\_idoc\_statu.  
    DATA: lt\_text\_tab TYPE catsxt\_longtext\_itab,  
          ls\_result   TYPE /kspack/ef\_status\_messages,  
          lt\_lines    TYPE trtexts.  
  
    DATA: lt\_idoc\_header TYPE TABLE OF edidc,  
          lt\_idoc\_data   TYPE TABLE OF edidd,  
          ls\_idoc\_header TYPE edidc,  
          lv\_docnum      TYPE edidc-docnum.  
  
    SELECT SINGLE \*  
           FROM /kspack/ef001  
           INTO @DATA(ls\_/kspack/ef001)  
           WHERE eguid = @is\_item-ettn  
           AND direct = '2'.  
    CHECK sy-subrc = 0  AND ls\_/kspack/ef001-profileid <> 'TEMELFATURA'.  
  
    LOOP AT it\_text INTO DATA(ls\_text).  
      REFRESH lt\_lines.  
      CALL FUNCTION 'TR\_SPLIT\_TEXT'  
        EXPORTING  
          iv\_text  = ls\_text-tdline  
          iv\_len   = 72  
        IMPORTING  
          et\_lines = lt\_lines.  
      LOOP AT lt\_lines INTO DATA(lv\_lines).  
        APPEND lv\_lines TO lt\_text\_tab.  
      ENDLOOP.  
    ENDLOOP.  
  
    CALL FUNCTION 'ZZEREFGEN\_F006'  
      EXPORTING  
        i\_bukrs     = ls\_/kspack/ef001-bukrs  
        i\_awtyp     = ls\_/kspack/ef001-awtyp  
        i\_awkey     = ls\_/kspack/ef001-awkey  
        i\_xblnr     = ls\_/kspack/ef001-xblnr  
        i\_docnum    = ls\_/kspack/ef001-docnum  
        i\_eguid     = ls\_/kspack/ef001-eguid  
        i\_status    = iv\_status  
        i\_profileid = ls\_/kspack/ef001-profileid  
        i\_direct    = ls\_/kspack/ef001-direct  
      IMPORTING  
        e\_result    = ls\_result  
      TABLES  
        text\_tab    = lt\_text\_tab.  
    IF ls\_result-message\_type = 'S' AND  
       ls\_result-message\_id   = '/KSPACK/EF' AND ls\_result-message\_number = '046'.  
      lv\_docnum = ls\_result-message\_v1.  
      zcl\_util\_001=>add\_message(  
        EXPORTING  
          i\_msgty   = 'S'  
          i\_msgid   = mc\_msgid  
          i\_msgno   = '147'  
          i\_msgv1   = lv\_docnum  
          i\_msgv2   = iv\_status  
          i\_msgv3   = is\_item-dysid  
        IMPORTING  
          et\_return = et\_return ).  
      IF 1 = 2. MESSAGE s147 WITH lv\_docnum iv\_status is\_item-dysid. ENDIF.  
  
      SELECT SINGLE \*  
             FROM edidc  
             INTO @DATA(ls\_edidc)  
             WHERE docnum = @lv\_docnum.  
      MOVE-CORRESPONDING ls\_edidc TO ls\_idoc\_header.  
      APPEND ls\_idoc\_header TO lt\_idoc\_header.  
  
      CALL FUNCTION 'EDI\_OUTPUT\_NEW'  
        EXPORTING  
          onl\_option = 'B'  
        TABLES  
          i\_edidc    = lt\_idoc\_header  
          i\_edidd    = lt\_idoc\_data.  
      COMMIT WORK.  
      CALL FUNCTION 'DEQUEUE\_ALL'  
        EXCEPTIONS  
          OTHERS = 1.  
    ELSE.  
      zcl\_util\_001=>add\_message(  
        EXPORTING  
          i\_msgty   = 'E'  
          i\_msgid   = mc\_msgid  
          i\_msgno   = '148'  
          i\_msgv1   = is\_item-dysid  
          i\_msgv2   = iv\_status  
        IMPORTING  
          et\_return = et\_return ).  
      IF 1 = 2. MESSAGE e148 WITH is\_item-dysid iv\_status. ENDIF.  
    ENDIF.  
  ENDMETHOD.  
  
  METHOD download\_as\_zip.  
    TYPES: BEGIN OF ty\_edocu,  
             dysid TYPE zdys\_e\_dysid,  
             fatno TYPE zdys\_e\_fatno,  
             ettn  TYPE zdys\_e\_ettn,  
             url   TYPE char2048,  
           END OF ty\_edocu.  
  
    DATA: lt\_toa01           TYPE zdys\_tt\_010,  
          lt\_toaat           TYPE HASHED TABLE OF toaat WITH UNIQUE KEY arc\_doc\_id,  
          lt\_content\_bin     TYPE TABLE OF sdokcntbin,  
          lt\_edocu           TYPE TABLE OF ty\_edocu,  
          lv\_name             TYPE string,  
          lv\_xstring         TYPE xstring,  
          lv\_window\_title    TYPE string,  
          lv\_selected\_folder TYPE string.  
  
    DATA: lt\_comps         TYPE TABLE OF scms\_stinf,  
          lt\_binary\_tab    TYPE STANDARD TABLE OF solisti1,  
          lv\_output\_length TYPE  i,  
          lv\_input\_length  TYPE  i,  
          lv\_filename       TYPE string,  
          lv\_path           TYPE string,  
          lv\_fullpath       TYPE string.  
  
    DATA: lo\_http\_client TYPE REF TO if\_http\_client,  
          lv\_url         TYPE string.  
  
    CHECK check\_selected\_rows( EXPORTING iv\_sel\_method = 'M' ) = ''.  
  
    LOOP AT mt\_dysid\_selected INTO DATA(lv\_dysid\_selected).  
      READ TABLE gt\_item INTO DATA(ls\_item) WITH KEY dysid = lv\_dysid\_selected.  
      CHECK sy-subrc = 0.  
      APPEND VALUE #( object\_id = ls\_item-dysid ) TO lt\_toa01.  
  
      IF ls\_item-ettn IS NOT INITIAL.  
        APPEND VALUE #( dysid = ls\_item-dysid  
                        fatno = ls\_item-fatno  
                        ettn  = ls\_item-ettn ) TO lt\_edocu.  
      ENDIF.  
    ENDLOOP.  
  
    " E-fatura/ e-arşiv fatura aslı bilgileri  
    LOOP AT lt\_edocu ASSIGNING FIELD-SYMBOL(<ls\_edocu>).  
      <ls\_edocu>-url = zdys\_cl\_general=>get\_url\_of\_einvoice( EXPORTING iv\_ettn  = <ls\_edocu>-ettn ).  
      IF <ls\_edocu>-url IS INITIAL.  
        DELETE lt\_edocu.  
        CONTINUE.  
      ENDIF.  
    ENDLOOP.  
  
    " Fatura ekleri  
    IF lt\_toa01[] IS NOT INITIAL.  
      DATA(ls\_zdys\_t\_012) = zdys\_cl\_attachment=>get\_archive\_disc( ).  
      SELECT \*  
             FROM toa01  
             INTO TABLE lt\_toa01  
             FOR ALL ENTRIES IN lt\_toa01  
             WHERE sap\_object = zdys\_cl\_attachment=>mc\_sap\_object  
             AND object\_id = lt\_toa01-object\_id  
             AND archiv\_id = ls\_zdys\_t\_012-arcdisk  
             AND ( ar\_object = zdys\_cl\_attachment=>mc\_ar\_object\_fatura OR  
                   ar\_object = zdys\_cl\_attachment=>mc\_ar\_object\_faturaek ).  
      zdys\_cl\_attachment=>eliminate\_deleted\_files( CHANGING ct\_toa01 = lt\_toa01 ).  
    ENDIF.  
  
    IF lt\_toa01[] IS INITIAL AND lt\_edocu[] IS INITIAL.  
      MESSAGE i153.  
      EXIT.  
    ENDIF.  
  
    IF lt\_toa01[] IS NOT INITIAL.  
      SELECT \*  
             FROM toaat  
             INTO TABLE lt\_toaat  
             FOR ALL ENTRIES IN lt\_toa01  
             WHERE arc\_doc\_id = lt\_toa01-arc\_doc\_id.  
    ENDIF.  
  
    DATA(lo\_zip) = NEW cl\_abap\_zip( ).  
  
    " E-fatura/ e-arşiv faturalar zip dosyaya ekleniyor  
    LOOP AT lt\_edocu INTO DATA(ls\_edocu).  
      lv\_url = ls\_edocu-url.  
      CLEAR lo\_http\_client.  
      CALL METHOD cl\_http\_client=>create\_by\_url  
        EXPORTING  
          url                        = lv\_url  
        IMPORTING  
          client                     = lo\_http\_client  
        EXCEPTIONS  
          argument\_not\_found         = 1  
          plugin\_not\_active          = 2  
          internal\_error             = 3  
          pse\_not\_found              = 4  
          pse\_not\_distrib            = 5  
          pse\_errors                 = 6  
          oa2c\_set\_token\_error       = 7  
          oa2c\_missing\_authorization = 8  
          oa2c\_invalid\_config        = 9  
          oa2c\_invalid\_parameters    = 10  
          oa2c\_invalid\_scope         = 11  
          oa2c\_invalid\_grant         = 12  
          OTHERS                     = 13.  
      CHECK sy-subrc = 0.  
      lo\_http\_client->send( ).  
      CHECK sy-subrc = 0.  
      lo\_http\_client->receive( ).  
      CHECK sy-subrc = 0.  
      lo\_http\_client->response->get\_status(  
        IMPORTING  
          code   = DATA(lv\_code)  
          reason = DATA(lv\_reason) ).  
      CHECK sy-subrc = 0 AND lv\_code = '200'.  
      CLEAR lv\_xstring.  
      lv\_xstring = lo\_http\_client->response->get\_data( ).  
      IF lv\_xstring IS NOT INITIAL.  
        lv\_name = ls\_edocu-fatno && '.' && 'HTML'.  
        lo\_zip->add( name    = lv\_name  
                     content = lv\_xstring ).  
      ENDIF.  
      lo\_http\_client->close( ).  
    ENDLOOP.  
  
    " Fatura ekleri zip dosyaya ekleniyor  
    LOOP AT lt\_toa01 INTO DATA(ls\_toa01).  
      REFRESH lt\_comps.  
      CALL FUNCTION 'SCMS\_DOC\_INFO'  
        EXPORTING  
          stor\_cat              = ''  
          crep\_id               = ls\_toa01-archiv\_id  
          doc\_id                = ls\_toa01-arc\_doc\_id  
        TABLES  
          comps                 = lt\_comps  
        EXCEPTIONS  
          bad\_storage\_type      = 1  
          bad\_request           = 2  
          unauthorized          = 3  
          not\_found             = 4  
          conflict              = 5  
          internal\_server\_error = 6  
          error\_http            = 7  
          error\_signature       = 8  
          error\_config          = 9  
          error\_hierarchy       = 10  
          error\_parameter       = 11  
          error                 = 12  
          OTHERS                = 13.  
      CHECK sy-subrc = 0.  
  
      REFRESH lt\_content\_bin.  
      CALL FUNCTION 'SCMS\_DOC\_READ'  
        EXPORTING  
          stor\_cat              = ''  
          crep\_id               = ls\_toa01-archiv\_id  
          doc\_id                = ls\_toa01-arc\_doc\_id  
        TABLES  
          content\_bin           = lt\_content\_bin  
        EXCEPTIONS  
          bad\_storage\_type      = 1  
          bad\_request           = 2  
          unauthorized          = 3  
          comp\_not\_found        = 4  
          not\_found             = 5  
          forbidden             = 6  
          conflict              = 7  
          internal\_server\_error = 8  
          error\_http            = 9  
          error\_signature       = 10  
          error\_config          = 11  
          error\_format          = 12  
          error\_parameter       = 13  
          error                 = 14  
          OTHERS                = 15.  
      CHECK sy-subrc = 0.  
  
      CLEAR: lv\_input\_length, lv\_xstring.  
      READ TABLE lt\_comps INTO DATA(ls\_comps) INDEX 1.  
      IF sy-subrc = 0.  
        lv\_input\_length = ls\_comps-comp\_size.  
      ENDIF.  
      CALL FUNCTION 'SCMS\_BINARY\_TO\_XSTRING'  
        EXPORTING  
          input\_length = lv\_input\_length  
        IMPORTING  
          buffer       = lv\_xstring  
        TABLES  
          binary\_tab   = lt\_content\_bin  
        EXCEPTIONS  
          failed       = 1  
          OTHERS       = 2.  
      CHECK sy-subrc = 0 AND lv\_xstring IS NOT INITIAL.  
  
      CLEAR lv\_name.  
      READ TABLE lt\_toaat INTO DATA(ls\_toaat) WITH TABLE KEY arc\_doc\_id = ls\_toa01-arc\_doc\_id.  
      IF sy-subrc = 0.  
        lv\_name = ls\_toaat-filename && '.' && ls\_toa01-reserve.  
      ENDIF.  
      lo\_zip->add( name    = lv\_name  
                   content = lv\_xstring ).  
    ENDLOOP.  
  
    " Zip klasörü kaydediliyor  
    DATA(lv\_zip) = lo\_zip->save( ).  
  
    CALL FUNCTION 'SCMS\_XSTRING\_TO\_BINARY'  
      EXPORTING  
        buffer        = lv\_zip  
      IMPORTING  
        output\_length = lv\_output\_length  
      TABLES  
        binary\_tab    = lt\_binary\_tab.  
  
    lv\_window\_title = TEXT-005.  
    cl\_gui\_frontend\_services=>file\_save\_dialog(  
      EXPORTING  
        window\_title         = lv\_window\_title  
        file\_filter          = '(\*.zip)|\*.zip|'  
      CHANGING  
        filename             = lv\_filename  
        path                 = lv\_path  
        fullpath             = lv\_fullpath  
      EXCEPTIONS  
        cntl\_error           = 1  
        error\_no\_gui         = 2  
        not\_supported\_by\_gui = 3  
        OTHERS               = 4 ).  
  
    CALL FUNCTION 'GUI\_DOWNLOAD'  
      EXPORTING  
        bin\_filesize            = lv\_output\_length  
        filename                = lv\_fullpath  
        filetype                = 'BIN'  
      TABLES  
        data\_tab                = lt\_binary\_tab  
      EXCEPTIONS  
        file\_write\_error        = 1  
        no\_batch                = 2  
        gui\_refuse\_filetransfer = 3  
        invalid\_type            = 4  
        no\_authority            = 5  
        unknown\_error           = 6  
        header\_not\_allowed      = 7  
        separator\_not\_allowed   = 8  
        filesize\_not\_allowed    = 9  
        header\_too\_long         = 10  
        dp\_error\_create         = 11  
        dp\_error\_send           = 12  
        dp\_error\_write          = 13  
        unknown\_dp\_error        = 14  
        access\_denied           = 15  
        dp\_out\_of\_memory        = 16  
        disk\_full               = 17  
        dp\_timeout              = 18  
        file\_not\_found          = 19  
        dataprovider\_exception  = 20  
        control\_flush\_error     = 21  
        OTHERS                  = 22.  
    IF sy-subrc = 0.  
      MESSAGE s154.  
    ELSE.  
      MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno  
            WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
    ENDIF.  
  
  ENDMETHOD.  
  
  METHOD match\_dys.  
    DATA: lt\_return  TYPE bapiret2\_t,  
          lt\_spopli  TYPE TABLE OF spopli,  
          lr\_gjahr   TYPE RANGE OF bkpf-gjahr,  
          lv\_answer  TYPE char01,  
          lv\_wrbtr   TYPE wrbtr,  
          lv\_awtyp   TYPE awtyp,  
          lv\_awkey   TYPE awkey,  
          lv\_matched TYPE xfeld.  
  
    CHECK check\_selected\_rows( EXPORTING iv\_sel\_method = 'M' ) = ''.  
  
    APPEND VALUE #( selflag   = 'X'  
                    varoption = TEXT-081  
                   ) TO lt\_spopli.  
    APPEND VALUE #( selflag   = ''  
                    varoption = TEXT-082  
                    inactive  = 'X'  
                   ) TO lt\_spopli.  
    APPEND VALUE #( selflag   = ''  
                    varoption = TEXT-083  
                    inactive  = 'X'  
                   ) TO lt\_spopli.  
    CALL FUNCTION 'POPUP\_TO\_DECIDE\_LIST'  
      EXPORTING  
        textline1          = ''  
        titel              = TEXT-079  
      IMPORTING  
        answer             = lv\_answer  
      TABLES  
        t\_spopli           = lt\_spopli  
      EXCEPTIONS  
        not\_enough\_answers = 1  
        too\_much\_answers   = 2  
        too\_much\_marks     = 3  
        OTHERS             = 4.  
    CHECK sy-subrc = 0 AND lv\_answer <> 'A'.  
  
    READ TABLE lt\_spopli INTO DATA(ls\_spopli) WITH KEY selflag = 'X'.  
    CHECK sy-subrc = 0.  
  
    LOOP AT mt\_dysid\_selected INTO DATA(lv\_dysid\_selected).  
      READ TABLE gt\_item INTO DATA(ls\_item) WITH KEY dysid = lv\_dysid\_selected.  
      CHECK sy-subrc = 0.  
  
      IF ls\_item-awtyp IS NOT INITIAL AND ls\_item-awkey IS NOT INITIAL.  
        zcl\_util\_001=>add\_message(  
          EXPORTING  
            i\_msgty   = 'E'  
            i\_msgid   = mc\_msgid  
            i\_msgno   = '254'  
            i\_msgv1   = ls\_item-dysid  
            i\_msgv2   = ls\_item-blgno  
          IMPORTING  
            et\_return = lt\_return ).  
        IF 1 = 2. MESSAGE e254 WITH ls\_item-dysid ls\_item-blgno. ENDIF.  
      ELSE.  
        CASE ls\_spopli-varoption.  
          WHEN TEXT-081.  " FI faturası eşleştir  
            IF ls\_item-isaks <> 'GDR'.  
              zcl\_util\_001=>add\_message(  
                EXPORTING  
                  i\_msgty   = 'E'  
                  i\_msgid   = mc\_msgid  
                  i\_msgno   = '080'  
                  i\_msgv1   = ls\_item-dysid  
                IMPORTING  
                  et\_return = lt\_return ).  
              IF 1 = 2. MESSAGE e080 WITH ls\_item-dysid. ENDIF.  
              CONTINUE.  
            ENDIF.  
  
            APPEND VALUE #( sign   = 'I'  
                            option = 'EQ'  
                            low    = ls\_item-fattr(4)  
                          ) TO lr\_gjahr.  
            APPEND VALUE #( sign   = 'I'  
                            option = 'EQ'  
                            low    = ls\_item-fattr(4) + 1  
                          ) TO lr\_gjahr.  
            CLEAR lv\_matched.  
            SELECT bukrs,belnr,gjahr,bstat  
                   FROM bkpf  
                   INTO @DATA(ls\_bkpf)  
                   WHERE bukrs = @ls\_item-bukrs  
                   AND gjahr IN @lr\_gjahr  
                   AND bldat = @ls\_item-fattr  
                   AND xblnr = @ls\_item-fatno  
                   AND waers = @ls\_item-waers  
                   AND stblg = ''.  
              CLEAR lv\_wrbtr.  
              IF ls\_bkpf-bstat = 'V'.  
                SELECT SINGLE wrbtr  
                       FROM vbsegk  
                       INTO lv\_wrbtr  
                       WHERE belnr = ls\_bkpf-belnr  
                       AND gjahr = ls\_bkpf-gjahr  
                       AND bukrs = ls\_bkpf-bukrs  
                       AND lifnr = ls\_item-partner.  
                CHECK sy-subrc = 0.  
              ELSE.  
                SELECT SINGLE wrbtr  
                       FROM bseg  
                       INTO lv\_wrbtr  
                       WHERE bukrs = ls\_bkpf-bukrs  
                       AND belnr = ls\_bkpf-belnr  
                       AND gjahr = ls\_bkpf-gjahr  
                       AND lifnr = ls\_item-partner  
                       AND koart = 'K'.  
                CHECK sy-subrc = 0.  
              ENDIF.  
              IF lv\_wrbtr = ls\_item-fattu.  
                lv\_awtyp = 'BKPFF'.  
                lv\_awkey = ls\_bkpf-belnr && ls\_bkpf-bukrs && ls\_bkpf-gjahr.  
                SELECT SINGLE dysid  
                       FROM zdys\_t\_001  
                       INTO @DATA(lv\_dysid\_matched)  
                       WHERE awtyp = @lv\_awtyp  
                       AND awkey = @lv\_awkey.  
                CHECK sy-subrc <> 0.  
                lv\_matched = 'X'.  
                UPDATE zdys\_t\_001  
                       SET durum = '99' " Başarıyla tamamlandı  
                           awtyp = lv\_awtyp  
                           awkey = lv\_awkey  
                       WHERE dysid = ls\_item-dysid.  
                zdys\_cl\_general=>log\_dys\_process(  
                  iv\_dysid     = ls\_item-dysid  
                  iv\_isltp     = '08' " Fatura muhasebeleştirme  
                  iv\_fname     = 'BELNR'  
                  iv\_value\_old = ''  
                  iv\_value\_new = lv\_awkey ).  
                zdys\_cl\_general=>log\_dys\_process(  
                  iv\_dysid     = ls\_item-dysid  
                  iv\_isltp     = '03' " Ana veri güncelle  
                  iv\_fname     = 'DURUM'  
                  iv\_value\_old = ls\_item-durum  
                  iv\_value\_new = '99' ).  
                EXIT.  
              ENDIF.  
            ENDSELECT.  
  
            IF lv\_matched = 'X'.  
              COMMIT WORK.  
              zcl\_util\_001=>add\_message(  
                EXPORTING  
                  i\_msgty   = 'S'  
                  i\_msgid   = mc\_msgid  
                  i\_msgno   = '255'  
                  i\_msgv1   = ls\_item-dysid  
                  i\_msgv2   = lv\_awkey  
                IMPORTING  
                  et\_return = lt\_return ).  
              IF 1 = 2. MESSAGE s255 WITH ls\_item-dysid lv\_awkey. ENDIF.  
            ELSE.  
              zcl\_util\_001=>add\_message(  
                EXPORTING  
                  i\_msgty   = 'E'  
                  i\_msgid   = mc\_msgid  
                  i\_msgno   = '256'  
                  i\_msgv1   = ls\_item-dysid  
                IMPORTING  
                  et\_return = lt\_return ).  
              IF 1 = 2. MESSAGE e256 WITH ls\_item-dysid. ENDIF.  
            ENDIF.  
  
          WHEN TEXT-082.  " Satın alma faturası eşleştir  
          WHEN TEXT-083.  " Satış faturası eşleştir  
        ENDCASE.  
      ENDIF.  
    ENDLOOP.  
  
    zcl\_util\_001=>show\_messages( EXPORTING it\_return = lt\_return ).  
  ENDMETHOD.  
  
  METHOD display\_edocu.  
    DATA(lv\_url) = zdys\_cl\_general=>get\_url\_of\_einvoice( EXPORTING iv\_ettn  = iv\_ettn ).  
  
    CALL FUNCTION 'CALL\_BROWSER'  
      EXPORTING  
        url                    = lv\_url  
      EXCEPTIONS  
        frontend\_not\_supported = 1  
        frontend\_error         = 2  
        prog\_not\_found         = 3  
        no\_batch               = 4  
        unspecified\_error      = 5  
        OTHERS                 = 6.  
    IF sy-subrc <> 0.  
      MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno  
            WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
    ENDIF.  
  ENDMETHOD.  
  
  METHOD create\_sales\_inv.  
    TYPES: BEGIN OF ty\_vbfa,  
             vbeln TYPE vbfa-vbeln,  
             posnn TYPE vbfa-posnn,  
           END OF ty\_vbfa.  
    TYPES: BEGIN OF ty\_vbap,  
             vbeln TYPE vbap-vbeln,  
             posnr TYPE vbap-posnr,  
           END OF ty\_vbap.  
    TYPES: BEGIN OF ty\_docno,  
             docno TYPE char10,  
           END OF ty\_docno.  
  
    DATA: lt\_lips\_key   TYPE shp\_vl10\_lips\_key\_t,  
          lt\_vbeln\_vf    TYPE tt\_vbeln\_vf,  
          lt\_return     TYPE bapiret2\_t,  
          lt\_vbfa       TYPE TABLE OF ty\_vbfa,  
          lt\_vbap       TYPE TABLE OF ty\_vbap,  
          lt\_docno      TYPE TABLE OF ty\_docno,  
          ls\_zdys\_t\_001 TYPE zdys\_t\_001,  
          lv\_bolnr      TYPE likp-bolnr.  
  
    CHECK is\_item-awtyp = 'RMRP' AND is\_item-awkey IS NOT INITIAL.  
    zdys\_cl\_general=>get\_awkey\_values( EXPORTING iv\_awtyp = is\_item-awtyp  
                                                 iv\_awkey = is\_item-awkey  
                                       IMPORTING ev\_belnr\_mm = DATA(lv\_belnr\_mm)  
                                                 ev\_gjahr\_mm = DATA(lv\_gjahr\_mm) ).  
    CHECK lv\_belnr\_mm IS NOT INITIAL AND lv\_gjahr\_mm IS NOT INITIAL.  
    " Satın alma siparişi  
    SELECT ebeln ebelp  
           FROM rseg  
           INTO TABLE lt\_vbfa  
           WHERE belnr = lv\_belnr\_mm  
           AND gjahr = lv\_gjahr\_mm  
           AND ebeln <> ''.  
    CHECK sy-subrc = 0 AND lt\_vbfa[] IS NOT INITIAL.  
    REFRESH lt\_docno.  
    LOOP AT lt\_vbfa INTO DATA(ls\_vbfa).  
      COLLECT VALUE ty\_docno( docno = ls\_vbfa-vbeln ) INTO lt\_docno.  
    ENDLOOP.  
    IF lines( lt\_docno ) > 1.  
      zcl\_util\_001=>add\_message(  
        EXPORTING  
          i\_msgty   = 'E'  
          i\_msgid   = mc\_msgid  
          i\_msgno   = '203'  
          i\_msgv1   = lv\_belnr\_mm  
          i\_msgv2   = TEXT-066  
        IMPORTING  
          et\_return = ct\_return ).  
      IF 1 = 2. MESSAGE e203 WITH lv\_belnr\_mm TEXT-066. ENDIF.  
      EXIT.  
    ENDIF.  
  
    " Satış siparişi  
    SELECT vbelv posnv  
           FROM vbfa  
           INTO TABLE lt\_vbap  
           FOR ALL ENTRIES IN lt\_vbfa  
           WHERE vbeln = lt\_vbfa-vbeln  
           AND posnn = lt\_vbfa-posnn  
           AND vbtyp\_n = 'V'.  
    CHECK sy-subrc = 0 AND lt\_vbap[] IS NOT INITIAL.  
    REFRESH lt\_docno.  
    LOOP AT lt\_vbap INTO DATA(ls\_vbap).  
      COLLECT VALUE ty\_docno( docno = ls\_vbap-vbeln ) INTO lt\_docno.  
    ENDLOOP.  
    IF lines( lt\_docno ) > 1.  
      zcl\_util\_001=>add\_message(  
        EXPORTING  
          i\_msgty   = 'E'  
          i\_msgid   = mc\_msgid  
          i\_msgno   = '203'  
          i\_msgv1   = lv\_belnr\_mm  
          i\_msgv2   = TEXT-067  
        IMPORTING  
          et\_return = ct\_return ).  
      IF 1 = 2. MESSAGE e203 WITH lv\_belnr\_mm TEXT-067. ENDIF.  
      EXIT.  
    ENDIF.  
  
    " Teslimat  
    lv\_bolnr = lv\_belnr\_mm && '-' && lv\_gjahr\_mm.  
    SELECT lips~vbeln,lips~posnr,lips~fksta  
           FROM likp  
           INNER JOIN lips  
           ON likp~vbeln = lips~vbeln  
           INTO TABLE @DATA(lt\_lips)  
           FOR ALL ENTRIES IN @lt\_vbap  
           WHERE likp~bolnr = @lv\_bolnr  
           AND lips~vgbel = @lt\_vbap-vbeln  
           AND lips~vgpos = @lt\_vbap-posnr.  
    " Henüz faturalandırılmamış teslimatlar  
    LOOP AT lt\_lips INTO DATA(ls\_lips) WHERE fksta <> 'C'.  
      APPEND VALUE #( vbeln = ls\_lips-vbeln  
                      posnr = ls\_lips-posnr  
                     ) TO lt\_lips\_key.  
    ENDLOOP.  
  
    REFRESH lt\_docno.  
    LOOP AT lt\_lips\_key INTO DATA(ls\_lips\_key).  
      COLLECT VALUE ty\_docno( docno = ls\_lips\_key-vbeln ) INTO lt\_docno.  
    ENDLOOP.  
    IF lines( lt\_docno ) > 1.  
      zcl\_util\_001=>add\_message(  
        EXPORTING  
          i\_msgty   = 'E'  
          i\_msgid   = mc\_msgid  
          i\_msgno   = '203'  
          i\_msgv1   = lv\_belnr\_mm  
          i\_msgv2   = TEXT-068  
        IMPORTING  
          et\_return = ct\_return ).  
      IF 1 = 2. MESSAGE e203 WITH lv\_belnr\_mm TEXT-068. ENDIF.  
      EXIT.  
    ENDIF.  
  
    IF lt\_lips\_key[] IS INITIAL.  
      zcl\_util\_001=>add\_message(  
        EXPORTING  
          i\_msgty   = 'E'  
          i\_msgid   = mc\_msgid  
          i\_msgno   = '204'  
          i\_msgv1   = lv\_belnr\_mm  
        IMPORTING  
          et\_return = ct\_return ).  
      IF 1 = 2. MESSAGE e204 WITH lv\_belnr\_mm. ENDIF.  
      EXIT.  
    ELSE.  
      REFRESH lt\_return.  
      MOVE-CORRESPONDING is\_item TO ls\_zdys\_t\_001.  
      zdys\_cl\_general=>create\_sales\_invoice( EXPORTING it\_lips\_key = lt\_lips\_key  
                                                       is\_item     = ls\_zdys\_t\_001  
                                                       it\_dysid\_selected = gs\_0400-dysid\_selected  "SİLİNECEK  
                                             IMPORTING et\_vbeln\_vf = lt\_vbeln\_vf  
                                                       et\_return   = lt\_return ).  
      APPEND LINES OF lt\_return TO ct\_return.  
      LOOP AT lt\_vbeln\_vf INTO DATA(lv\_vbeln\_vf).  
        zdys\_cl\_general=>log\_dys\_process(  
          iv\_dysid     = ls\_zdys\_t\_001-dysid  
          iv\_isltp     = '20' " Satış faturası oluşturma  
          iv\_fname     = 'VBELN'  
          iv\_value\_old = ''  
          iv\_value\_new = lv\_vbeln\_vf ).  
      ENDLOOP.  
    ENDIF.  
    " Daha önceden faturalandırılmış teslimatlar  
    LOOP AT lt\_lips INTO ls\_lips WHERE fksta = 'C'.  
      SELECT SINGLE vbeln  
             FROM vbrp  
             INTO @DATA(lv\_vbeln)  
             WHERE vgbel = @ls\_lips-vbeln  
             AND vgpos = @ls\_lips-posnr.  
      IF sy-subrc = 0.  
        APPEND lv\_vbeln TO lt\_vbeln\_vf.  
      ENDIF.  
    ENDLOOP.  
    IF lt\_vbeln\_vf[] IS NOT INITIAL.  
      WAIT UP TO 2 SECONDS.  
      REFRESH lt\_return.  
      SORT lt\_vbeln\_vf.  
      DELETE ADJACENT DUPLICATES FROM lt\_vbeln\_vf.  
      zdys\_cl\_general=>print\_sales\_invoice( EXPORTING it\_vbeln\_vf = lt\_vbeln\_vf  
                                            IMPORTING et\_return   = lt\_return ).  
      LOOP AT lt\_return INTO DATA(ls\_return) WHERE type = 'S'  
                                             AND id = zdys\_cl\_general=>mc\_msgid  
                                             AND number = '184'.  
        zdys\_cl\_general=>log\_dys\_process(  
          iv\_dysid     = ls\_zdys\_t\_001-dysid  
          iv\_isltp     = '21' " Satış faturası çıktısı alma  
          iv\_fname     = 'VBELN'  
          iv\_value\_old = ''  
          iv\_value\_new = ls\_return-message\_v1 ).  
      ENDLOOP.  
      APPEND LINES OF lt\_return TO ct\_return.  
    ENDIF.  
  ENDMETHOD.  
  
  METHOD f4\_variant.  
    DATA: ls\_variant TYPE disvariant,  
          lv\_exit    TYPE c.  
  
    CALL FUNCTION 'REUSE\_ALV\_VARIANT\_F4'  
      EXPORTING  
        is\_variant    = ms\_variant  
        i\_save        = 'A'  
      IMPORTING  
        e\_exit        = lv\_exit  
        es\_variant    = ls\_variant  
      EXCEPTIONS  
        not\_found     = 1  
        program\_error = 2  
        OTHERS        = 3.  
    IF sy-subrc = 0.  
      IF lv\_exit = ''.  
        rv\_variant = ls\_variant-variant.  
      ENDIF.  
    ELSE.  
      MESSAGE ID sy-msgid TYPE 'I' NUMBER sy-msgno  
              WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4  
              DISPLAY LIKE 'E'.  
    ENDIF.  
  ENDMETHOD.  
  
  METHOD run\_bg\_process.  
    DATA: lt\_return     TYPE bapiret2\_t,  
          ls\_zdys\_t\_017 TYPE zdys\_t\_017,  
          lv\_matkl\_rseg TYPE matkl,  
          lv\_wfcod      TYPE zdys\_e\_wfcod,  
          lv\_wfcot      TYPE zdys\_e\_wfcot,  
          lv\_msgv       TYPE symsgv.  
  
    CASE p\_updty.  
      WHEN '01'. " Satış fatura durumu güncelle  
        LOOP AT gt\_item INTO DATA(ls\_item).  
          CLEAR ls\_zdys\_t\_017.  
          SELECT SINGLE \*  
                 FROM zdys\_t\_017  
                 INTO ls\_zdys\_t\_017  
                 WHERE dysid = ls\_item-dysid.  
          IF sy-subrc = 0.  
            UPDATE zdys\_t\_017  
                   SET sfatd = ls\_item-sfatd  
                   WHERE dysid = ls\_item-dysid.  
          ELSE.  
            ls\_zdys\_t\_017-dysid = ls\_item-dysid.  
            ls\_zdys\_t\_017-sfatd = ls\_item-sfatd.  
            MODIFY zdys\_t\_017 FROM ls\_zdys\_t\_017.  
          ENDIF.  
          zcl\_util\_001=>add\_message(  
            EXPORTING  
              i\_msgty   = 'S'  
              i\_msgid   = mc\_msgid  
              i\_msgno   = '196'  
              i\_msgv1   = ls\_item-dysid  
              i\_msgv2   = TEXT-073  
              i\_msgv3   = ls\_item-sfatd  
            IMPORTING  
              et\_return = lt\_return ).  
          IF 1 = 2. MESSAGE s196 WITH ls\_item-dysid TEXT-073 ls\_item-sfatd. ENDIF.  
        ENDLOOP.  
        COMMIT WORK.  
  
      WHEN '02'.  " İş akışı tamamlama sürelerini güncelle  
        DATA(lr\_durum\_completed) = zdys\_cl\_general=>get\_completed\_status( ).  
        LOOP AT gt\_item INTO ls\_item.  
          CLEAR: ls\_zdys\_t\_017, lv\_wfcod, lv\_wfcot, lv\_msgv.  
          SELECT udate utime  
                 FROM zdys\_t\_002  
                 INTO ( lv\_wfcod, lv\_wfcot )  
                 UP TO 1 ROWS  
                 WHERE dysid = ls\_item-dysid  
                 AND fname = 'DURUM'  
                 AND value\_new IN lr\_durum\_completed  
                 ORDER BY islno DESCENDING.  
            EXIT.  
          ENDSELECT.  
          SELECT SINGLE \*  
                 FROM zdys\_t\_017  
                 INTO ls\_zdys\_t\_017  
                 WHERE dysid = ls\_item-dysid.  
          IF sy-subrc = 0.  
            UPDATE zdys\_t\_017  
                   SET wfcod = lv\_wfcod  
                       wfcot = lv\_wfcot  
                   WHERE dysid = ls\_item-dysid.  
          ELSE.  
            ls\_zdys\_t\_017-dysid = ls\_item-dysid.  
            ls\_zdys\_t\_017-wfcod = lv\_wfcod.  
            ls\_zdys\_t\_017-wfcot = lv\_wfcot.  
            MODIFY zdys\_t\_017 FROM ls\_zdys\_t\_017.  
          ENDIF.  
          WRITE lv\_wfcod TO lv\_msgv DD/MM/YYYY.  
          lv\_msgv = lv\_msgv && ` ` && lv\_wfcot(2) && `:` && lv\_wfcot+2(2) && `:` && lv\_wfcot+4(2).  
          zcl\_util\_001=>add\_message(  
            EXPORTING  
              i\_msgty   = 'S'  
              i\_msgid   = mc\_msgid  
              i\_msgno   = '196'  
              i\_msgv1   = ls\_item-dysid  
              i\_msgv2   = TEXT-075  
              i\_msgv3   = lv\_msgv  
            IMPORTING  
              et\_return = lt\_return ).  
          IF 1 = 2. MESSAGE s196 WITH ls\_item-dysid TEXT-075 lv\_msgv. ENDIF.  
        ENDLOOP.  
        COMMIT WORK.  
  
      WHEN '03'.  " MTS rolünde hatalı mal gruplarını güncelle  
        LOOP AT gt\_item INTO ls\_item WHERE rol = 'MTS'  
                                     AND marka IS INITIAL  
                                     AND awtyp = 'RMRP'.  
          CLEAR lv\_matkl\_rseg.  
          lv\_matkl\_rseg = zdys\_cl\_general=>get\_matkl\_of\_incoming\_invoice(  
                            iv\_belnr = ls\_item-blgno  
                            iv\_gjahr = ls\_item-yilno ).  
          CHECK lv\_matkl\_rseg IS NOT INITIAL.  
          CHECK lv\_matkl\_rseg <> ls\_item-matkl.  
          UPDATE zdys\_t\_001  
                 SET matkl = lv\_matkl\_rseg  
                 WHERE dysid = ls\_item-dysid.  
          zdys\_cl\_general=>log\_dys\_process( iv\_dysid     = ls\_item-dysid  
                                            iv\_isltp     = '03'  
                                            iv\_fname     = 'MATKL'  
                                            iv\_value\_old = ls\_item-matkl  
                                            iv\_value\_new = lv\_matkl\_rseg ).  
          zcl\_util\_001=>add\_message(  
            EXPORTING  
              i\_msgty   = 'S'  
              i\_msgid   = mc\_msgid  
              i\_msgno   = '196'  
              i\_msgv1   = ls\_item-dysid  
              i\_msgv2   = TEXT-074  
              i\_msgv3   = lv\_matkl\_rseg  
            IMPORTING  
              et\_return = lt\_return ).  
          IF 1 = 2. MESSAGE s196 WITH ls\_item-dysid TEXT-074 lv\_matkl\_rseg. ENDIF.  
          COMMIT WORK AND WAIT.  
          zdys\_cl\_general=>run\_workflow( EXPORTING iv\_dysid = ls\_item-dysid ).  
        ENDLOOP.  
  
    ENDCASE.  
  
    IF sy-batch = 'X'.  
      zcl\_util\_001=>write\_messages( it\_return = lt\_return ).  
    ELSE.  
      zcl\_util\_001=>show\_messages( EXPORTING it\_return = lt\_return ).  
    ENDIF.  
  ENDMETHOD.  
ENDCLASS.