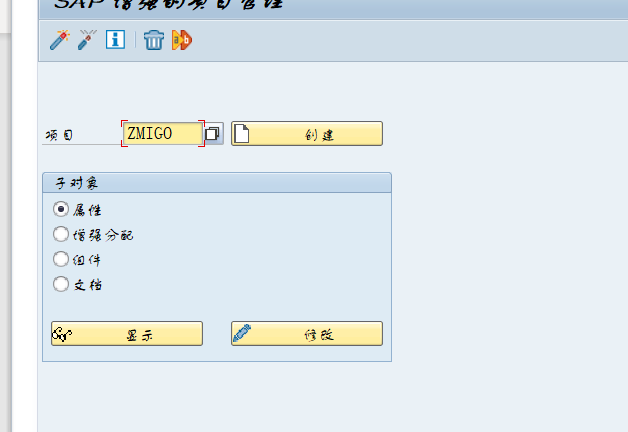
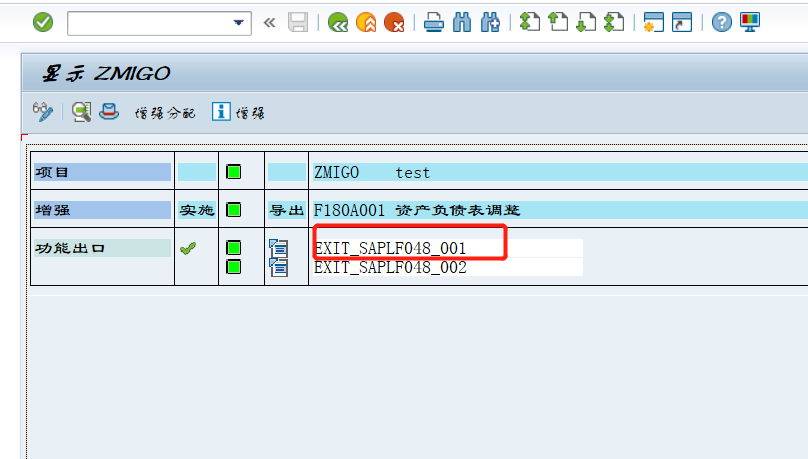
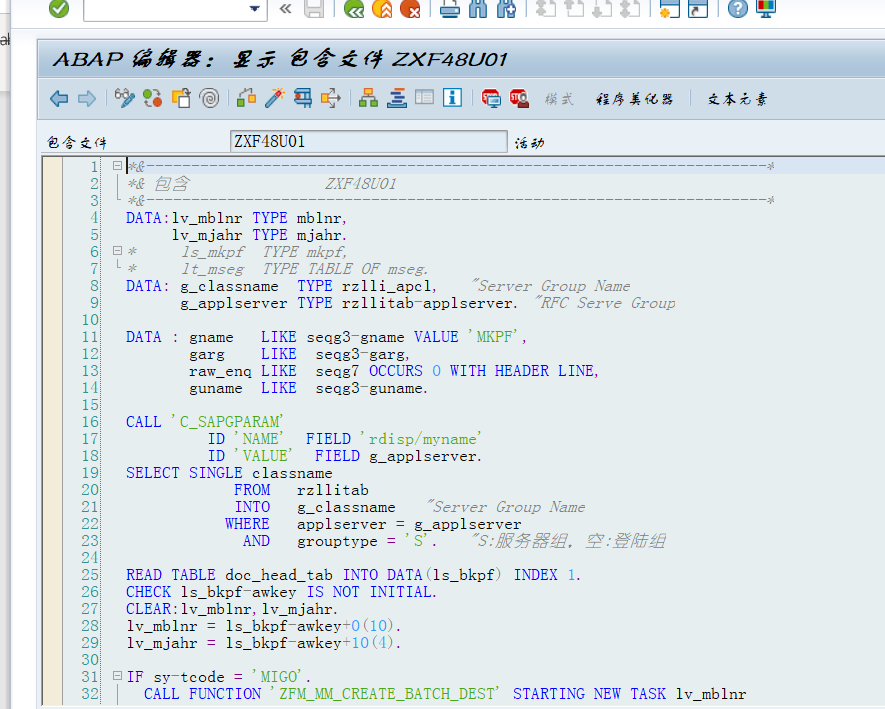
业务要求migo产生批次后 再进行该凭证得批次更新 ，难度很大

前台操作时migo收获操作







DATA:lv\_mblnr TYPE mblnr,  
     lv\_mjahr TYPE mjahr.  
\*     ls\_mkpf  TYPE mkpf,  
\*     lt\_mseg  TYPE TABLE OF mseg.  
DATA: g\_classname  TYPE rzlli\_apcl,   "Server Group Name  
      g\_applserver TYPE rzllitab-applserver. "RFC Serve Group  
  
DATA : gname   LIKE seqg3-gname VALUE 'MKPF',  
       garg    LIKE  seqg3-garg,  
       raw\_enq LIKE  seqg7 OCCURS 0 WITH HEADER LINE,  
       guname  LIKE  seqg3-guname.  
  
CALL 'C\_SAPGPARAM'  
         ID 'NAME'  FIELD 'rdisp/myname'  
         ID 'VALUE'  FIELD g\_applserver.  
SELECT SINGLE classname  
            FROM   rzllitab  
            INTO   g\_classname   "Server Group Name  
           WHERE   applserver = g\_applserver  
             AND   grouptype = 'S'.   "S:服务器组，空:登陆组  
  
READ TABLE doc\_head\_tab INTO DATA(ls\_bkpf) INDEX 1.  
CHECK ls\_bkpf-awkey IS NOT INITIAL.  
CLEAR:lv\_mblnr,lv\_mjahr.  
lv\_mblnr = ls\_bkpf-awkey+0(10).  
lv\_mjahr = ls\_bkpf-awkey+10(4).  
  
IF sy-tcode = 'MIGO'.  
  CALL FUNCTION 'ZFM\_MM\_CREATE\_BATCH\_DEST' STARTING NEW TASK lv\_mblnr  
    DESTINATION IN GROUP g\_classname  
    PERFORMING frm\_batch\_upload ON END OF TASK  
    EXPORTING  
      iv\_mblnr = lv\_mblnr  
      iv\_mjahr = lv\_mjahr.  
ENDIF.

FORM frm\_batch\_upload USING lv\_mblnr.  
  RECEIVE RESULTS FROM FUNCTION 'ZFM\_MM\_CREATE\_BATCH\_DEST'.  
ENDFORM.

Rfc内容

FUNCTION zfm\_mm\_create\_batch\_dest.  
\*"----------------------------------------------------------------------  
\*"\*"本地接口：  
\*"  IMPORTING  
\*"     VALUE(IV\_MBLNR) TYPE  MBLNR OPTIONAL  
\*"     VALUE(IV\_MJAHR) TYPE  MJAHR OPTIONAL  
\*"----------------------------------------------------------------------  
  DATA:iv\_matnr  TYPE matnr,  
       iv\_werks  TYPE werks\_d,  
       iv\_charg  TYPE charg\_d,  
       e\_objek   TYPE cuobn,  
       e\_obtab   TYPE tabelle,  
       e\_klart   TYPE klassenart,  
       e\_class   TYPE klasse\_d,  
       lt\_num    TYPE TABLE OF bapi1003\_alloc\_values\_num,  
       lt\_char   TYPE TABLE OF bapi1003\_alloc\_values\_char,  
       lt\_curr   TYPE TABLE OF bapi1003\_alloc\_values\_curr,  
       it\_mseg   TYPE TABLE OF mseg,  
       is\_mkpf   TYPE mkpf,  
       lt\_return TYPE TABLE OF bapiret2.  
  
  
  CLEAR:is\_mkpf,it\_mseg[].  
  DO 10 TIMES.  
    SELECT SINGLE \* INTO CORRESPONDING FIELDS OF is\_mkpf FROM mkpf WHERE mblnr = iv\_mblnr AND mjahr = iv\_mjahr.  
    IF sy-subrc = 0.  
      EXIT.  
    ELSE.  
      WAIT UP TO 1 SECONDS.  
    ENDIF.  
  ENDDO.  
  
  IF is\_mkpf IS  INITIAL.  
    EXIT.  
  ENDIF.  
  
  SELECT \* INTO CORRESPONDING FIELDS OF TABLE it\_mseg FROM mseg WHERE mblnr = iv\_mblnr AND mjahr = iv\_mjahr.  
  
\*       ls\_catt   TYPE bapibatchatt.  
  "产地 生产厂家编码 生产产家  
  SELECT FROM ekpo AS a  
  INNER JOIN ztcqm002 AS b  
  ON a~werks = b~werks AND  
     a~matnr = b~matnr AND  
     a~idnlf = b~zid  
  FIELDS  
  a~ebeln,a~ebelp,a~matnr,a~idnlf,  
  b~zid,b~zscsmc,b~zchandi  
  FOR ALL ENTRIES IN @it\_mseg[]  
  WHERE ebeln = @it\_mseg-ebeln  
    AND ebelp = @it\_mseg-ebelp  
    AND zdel = ''  
   INTO TABLE @DATA(lt\_ekpo).  
  SORT lt\_ekpo BY ebeln ebelp matnr.  
  DELETE ADJACENT DUPLICATES FROM lt\_ekpo COMPARING ebeln ebelp matnr.  
\* 上市许可持有人：证书类型Z032  
\* 第二类医疗器械经营备案凭证：证书类型Z015  
\* 医疗器械经营许可证：证书类型Z017  
  SELECT FROM ztgsp\_lic\_mas AS a  
    INNER JOIN t001k AS b  
    ON a~bukrs = b~bukrs  
    INNER JOIN ztgsp\_bukrs AS c  
    ON a~bukrs = c~bukrs  
    FIELDS  
    a~zzlicid,b~bwkey AS werks,a~bukrs,a~matnr,a~zzholder,a~zzlictyp,a~zzlicno  
    FOR ALL ENTRIES IN @it\_mseg  
    WHERE matnr = @it\_mseg-matnr  
      AND zzlictyp IN ('Z032','Z015','Z017')  
      AND zzvalto > @sy-datum  
      AND b~bwkey = @it\_mseg-werks  
     INTO TABLE @DATA(lt\_lic\_mas).  
  SORT lt\_lic\_mas BY werks matnr zzlictyp.  
  DELETE ADJACENT DUPLICATES FROM lt\_lic\_mas COMPARING werks matnr zzlictyp.  
\*注册证/备案号取数逻辑：  
  SELECT FROM ztmm\_master\_bas  
    FIELDS  
    matnr,  
    zzcba  
    FOR ALL ENTRIES IN @it\_mseg  
    WHERE matnr = @it\_mseg-matnr  
    INTO TABLE @DATA(lt\_master\_bas).  
  SORT lt\_master\_bas BY matnr.  
  LOOP AT it\_mseg ASSIGNING FIELD-SYMBOL(<fs\_mseg>) WHERE bwart = '101'  
                                                      AND  kzbew = 'B' AND charg IS NOT INITIAL.  
    CLEAR:  
       iv\_matnr,  
       iv\_werks,  
       iv\_charg,  
       e\_objek,  
       e\_obtab,  
       e\_klart,  
       e\_class,  
\*       ls\_catt,  
       lt\_num[],  
       lt\_char[],  
       lt\_curr[],  
       lt\_return[].  
    iv\_matnr = <fs\_mseg>-matnr.  
    iv\_werks = <fs\_mseg>-werks.  
    iv\_charg = <fs\_mseg>-charg.  
    CALL FUNCTION 'VB\_BATCH\_2\_CLASS\_OBJECT'  
      EXPORTING  
        i\_matnr = iv\_matnr  
        i\_charg = iv\_charg  
        i\_werks = iv\_werks  
      IMPORTING  
        e\_objek = e\_objek  
        e\_obtab = e\_obtab  
        e\_klart = e\_klart  
        e\_class = e\_class.  
    CALL FUNCTION 'BAPI\_OBJCL\_GETDETAIL'  
      EXPORTING  
\*       OBJECTKEY       =  
        objecttable     = e\_obtab  
        classnum        = e\_class  
        classtype       = e\_klart  
\*       KEYDATE         = SY-DATUM  
\*       UNVALUATED\_CHARS       = ' '  
\*       LANGUAGE        = SY-LANGU  
        objectkey\_long  = e\_objek  
\* IMPORTING  
\*       STATUS          =  
\*       STANDARDCLASS   =  
      TABLES  
        allocvaluesnum  = lt\_num  
        allocvalueschar = lt\_char  
        allocvaluescurr = lt\_curr  
        return          = lt\_return.  
    READ TABLE lt\_ekpo ASSIGNING FIELD-SYMBOL(<fs\_ekpo>) WITH KEY ebeln = <fs\_mseg>-ebeln  
                                                                  ebelp = <fs\_mseg>-ebelp  
                                                                  matnr = <fs\_mseg>-matnr  
                                                                  BINARY SEARCH.  
    IF sy-subrc = 0.  
      IF e\_class = 'Z\_BATCH01'.  
        " 生产厂家编号  
        IF <fs\_ekpo>-zid IS NOT INITIAL.  
          READ TABLE lt\_char ASSIGNING FIELD-SYMBOL(<fs\_char>) WITH KEY charact = 'Z\_DATA21'.  
          IF sy-subrc = 0.  
            <fs\_char>-value\_char     = <fs\_ekpo>-zid.  
            <fs\_char>-value\_neutral  = <fs\_ekpo>-zid.  
          ELSE.  
            APPEND INITIAL LINE TO lt\_char ASSIGNING <fs\_char>.  
            <fs\_char>-charact        = 'Z\_DATA21'.  
            <fs\_char>-value\_char     = <fs\_ekpo>-zid.  
            <fs\_char>-value\_neutral  = <fs\_ekpo>-zid.  
          ENDIF.  
        ENDIF.  
        "生产厂家  
        IF <fs\_ekpo>-zscsmc IS NOT INITIAL.  
          IF <fs\_char> IS ASSIGNED.  
            UNASSIGN <fs\_char>.  
          ENDIF.  
          READ TABLE lt\_char ASSIGNING <fs\_char> WITH KEY charact = 'Z\_DATA05'.  
          IF sy-subrc = 0.  
            <fs\_char>-value\_char     = <fs\_ekpo>-zscsmc.  
            <fs\_char>-value\_neutral  = <fs\_ekpo>-zscsmc.  
          ELSE.  
            APPEND INITIAL LINE TO lt\_char ASSIGNING <fs\_char>.  
            <fs\_char>-charact        = 'Z\_DATA05'.  
            <fs\_char>-value\_char     = <fs\_ekpo>-zscsmc.  
            <fs\_char>-value\_neutral  = <fs\_ekpo>-zscsmc.  
          ENDIF.  
        ENDIF.  
        "产地  
        IF <fs\_ekpo>-zchandi IS NOT INITIAL.  
          IF <fs\_char> IS ASSIGNED.  
            UNASSIGN <fs\_char>.  
          ENDIF.  
          READ TABLE lt\_char ASSIGNING <fs\_char> WITH KEY charact = 'Z\_DATA23'.  
          IF sy-subrc = 0.  
            <fs\_char>-value\_char     = <fs\_ekpo>-zchandi.  
            <fs\_char>-value\_neutral  = <fs\_ekpo>-zchandi.  
          ELSE.  
            APPEND INITIAL LINE TO lt\_char ASSIGNING <fs\_char>.  
            <fs\_char>-charact        = 'Z\_DATA23'.  
            <fs\_char>-value\_char     = <fs\_ekpo>-zchandi.  
            <fs\_char>-value\_neutral  = <fs\_ekpo>-zchandi.  
          ENDIF.  
        ENDIF.  
      ENDIF.  
    ENDIF.  
    IF e\_class = 'Z\_BATCH03'.  
      READ TABLE lt\_lic\_mas ASSIGNING FIELD-SYMBOL(<fs\_lic\_mas>) WITH KEY werks = <fs\_mseg>-werks"4.2.6  上市许可持有人  
                                                                          matnr = <fs\_mseg>-matnr  
                                                                          zzlictyp = 'Z032'  
                                                                          BINARY SEARCH.  
      IF sy-subrc = 0.  
        READ TABLE lt\_char ASSIGNING <fs\_char> WITH KEY charact = 'Z\_DATA43'.  
        IF sy-subrc = 0.  
          <fs\_char>-value\_char     = <fs\_lic\_mas>-zzlicno.  
          <fs\_char>-value\_neutral  = <fs\_lic\_mas>-zzlicno.  
        ELSE.  
          APPEND INITIAL LINE TO lt\_char ASSIGNING <fs\_char>.  
          <fs\_char>-charact        = 'Z\_DATA43'.  
          <fs\_char>-value\_char     = <fs\_lic\_mas>-zzlicno.  
          <fs\_char>-value\_neutral  = <fs\_lic\_mas>-zzlicno.  
        ENDIF.  
      ENDIF.  
      READ TABLE lt\_lic\_mas ASSIGNING <fs\_lic\_mas> WITH KEY werks    = <fs\_mseg>-werks  
                                                            matnr    = <fs\_mseg>-matnr  
                                                            zzlictyp = 'Z015'  
                                                            BINARY SEARCH.  
      IF sy-subrc = 0.  
        READ TABLE lt\_char ASSIGNING <fs\_char> WITH KEY charact = 'Z\_DATA45'."4.2.8  医疗器械经营许可证  
        IF sy-subrc = 0.  
          <fs\_char>-value\_char     = <fs\_lic\_mas>-zzlicno.  
          <fs\_char>-value\_neutral  = <fs\_lic\_mas>-zzlicno.  
        ELSE.  
          APPEND INITIAL LINE TO lt\_char ASSIGNING <fs\_char>.  
          <fs\_char>-charact        = 'Z\_DATA45'.  
          <fs\_char>-value\_char     = <fs\_lic\_mas>-zzlicno.  
          <fs\_char>-value\_neutral  = <fs\_lic\_mas>-zzlicno.  
        ENDIF.  
      ENDIF.  
      READ TABLE lt\_lic\_mas ASSIGNING <fs\_lic\_mas> WITH KEY werks    = <fs\_mseg>-werks  
                                                            matnr    = <fs\_mseg>-matnr  
                                                            zzlictyp = 'Z017'  
                                                            BINARY SEARCH.  
      IF sy-subrc = 0.  
        READ TABLE lt\_char ASSIGNING <fs\_char> WITH KEY charact = 'Z\_DATA46'."4.2.8  医疗器械经营许可证  
        IF sy-subrc = 0.  
          <fs\_char>-value\_char     = <fs\_lic\_mas>-zzlicno.  
          <fs\_char>-value\_neutral  = <fs\_lic\_mas>-zzlicno.  
        ELSE.  
          APPEND INITIAL LINE TO lt\_char ASSIGNING <fs\_char>.  
          <fs\_char>-charact        = 'Z\_DATA46'.  
          <fs\_char>-value\_char     = <fs\_lic\_mas>-zzlicno.  
          <fs\_char>-value\_neutral  = <fs\_lic\_mas>-zzlicno.  
        ENDIF.  
      ENDIF.  
      READ TABLE lt\_master\_bas ASSIGNING FIELD-SYMBOL(<fs\_master\_bas>) WITH KEY matnr = <fs\_mseg>-matnr BINARY SEARCH.  
      IF sy-subrc = 0.  
        READ TABLE lt\_char ASSIGNING <fs\_char> WITH KEY charact = 'Z\_DATA44'.  
        IF sy-subrc = 0.  
          <fs\_char>-value\_char     = <fs\_master\_bas>-zzcba.  
          <fs\_char>-value\_neutral  = <fs\_master\_bas>-zzcba.  
        ELSE.  
          APPEND INITIAL LINE TO lt\_char ASSIGNING <fs\_char>.  
          <fs\_char>-charact        = 'Z\_DATA44'.  
          <fs\_char>-value\_char     = <fs\_master\_bas>-zzcba.  
          <fs\_char>-value\_neutral  = <fs\_master\_bas>-zzcba.  
        ENDIF.  
      ENDIF.  
    ENDIF.  
    CALL FUNCTION 'BAPI\_OBJCL\_CHANGE' "对特性值进行一次更改  
      EXPORTING  
\*       OBJECTKEY          = E\_OBJEK  
        objecttable        = e\_obtab  
        classnum           = e\_class  
        classtype          = e\_klart  
        objectkey\_long     = e\_objek  
\*      IMPORTING  
\*       CLASSIF\_STATUS     = V\_STATUS  
      TABLES  
        allocvaluesnumnew  = lt\_num  
        allocvaluescharnew = lt\_char  
        allocvaluescurrnew = lt\_curr  
        return             = lt\_return  
      EXCEPTIONS  
        OTHERS             = 01.  
  
    DATA:lv\_check TYPE c.  
    CLEAR:lv\_check.  
    LOOP AT lt\_return ASSIGNING FIELD-SYMBOL(<lfs\_rettab>) WHERE type CA 'AEX'.  
      ROLLBACK WORK.  
      lv\_check = 'X'.  
      EXIT.  
    ENDLOOP.  
  
    IF lv\_check = ''.  
      COMMIT WORK .  
    ENDIF.  
  ENDLOOP.  
ENDFUNCTION.