说明：

|  |
| --- |
| 两步执出货，并且PGI，最终开具发票 |

界面

|  |
| --- |
|  |

界面设计-1000

|  |
| --- |
|  |
| PROCESS BEFORE OUTPUT.   MODULE STATUS\_1000.   LOOP AT IT\_DATA WITH CONTROL TABCONT01.     MODULE SET\_STATUS\_DATA.     MODULE ENABLE\_FIELDS.   ENDLOOP.  PROCESS AFTER INPUT.   LOOP AT IT\_DATA.     CHAIN.       FIELD IT\_DATA-SEL.       FIELD IT\_DATA-LIGHT.       MODULE RES\_INPUTS\_1000 ON CHAIN-REQUEST.     ENDCHAIN.   ENDLOOP.   MODULE USER\_COMMAND\_1000. |
|  |
|  |

界面设计-1001

|  |
| --- |
|  |
| PROCESS BEFORE OUTPUT.  MODULE STATUS\_1001.  PROCESS AFTER INPUT.  MODULE USER\_COMMAND\_1001. |
|  |
|  |
|  |

界面设计-1002

|  |
| --- |
|  |
| PROCESS BEFORE OUTPUT.   MODULE STATUS\_1002.  PROCESS AFTER INPUT.   MODULE USER\_COMMAND\_1002. |
|  |
|  |
|  |

源代码

|  |
| --- |
| *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** *\*& Report  ZRPWM046* *\*& 描述:* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** *\*&   创建人:Joseph.l* *\*&   创建时间:* *\*&   目的：用于批量导入BARCODE抄磅信息过账开发票* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** *\* REPORT NAME :报表名称及格式* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** *"程序名称* REPORT ZRPWM046.  *\*线型池引用* TYPE-POOLS: SLIS,ICON,SYM,LINE,COL.  *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** *\*引入 :ZPUB\_UPLOAD* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** INCLUDE ZPUB\_UPLOAD.  *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** *\*TABLES :声明报表中使用到的数据表* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** TABLES: LIKP, LIPS, VBAK, VBAP.  *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** *\* DATA :定义报表中使用到的变量及内表* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** *"DATA CONTROL 行數* DATA:GV\_ROW TYPE I. *"DATA CONTROL 字段* DATA:GV\_FIELD TYPE CHAR30. *"定義提示信息* DATA:GV\_MESSAGE TYPE STRING VALUE ''. *"过滤的文件类型* DATA:GV\_FILTER TYPE STRING. *"文件选择* DATA:GV\_FILE  LIKE RLGRAP-FILENAME.  DATA:LV\_DNDATE LIKE LIKP-LFDAT. DATA:LV\_PLANT  LIKE LIKP-VSTEL VALUE '1200'. DATA:TM\_PLANT  LIKE LIKP-VSTEL. DATA:LV\_DRIVER LIKE KNA1-SORTL. DATA:LV\_RECORD TYPE STRING VALUE '0'. DATA:LV\_SSNUM  TYPE INT4 VALUE 1. DATA:LV\_ADDRESS\_NUM LIKE ADRC-ADDRNUMBER.  *"接口调用状态和返回信息* DATA:LV\_STATUS TYPE CHAR1. DATA:LV\_MESSAGE TYPE CHAR100.  *"判断是否已经更新抄磅信息-DN 关键信息修改历史* DATA:LV\_BRGEW LIKE CDPOS-FNAME. DATA:LV\_NTGEW LIKE CDPOS-FNAME. DATA:LV\_UMVKN LIKE CDPOS-FNAME. DATA:LV\_UMVKZ LIKE CDPOS-FNAME. DATA:LV\_VOLUM LIKE CDPOS-FNAME.  *"1002屏幕比那里* DATA:LV\_DNNUM\_1002 LIKE LIKP-VBELN. DATA:LV\_KUNNR\_1002 LIKE KNA1-KUNNR. DATA:LV\_NAME1\_1002 LIKE KNA1-NAME1.  DATA: BEGIN OF IT\_DATA OCCURS 0,         SEL        TYPE CHAR1,         STATUS     TYPE CHAR1,         LIGHT      LIKE ICONS-TEXT,         SSNUM      TYPE INT4,         DN\_HOUSE   LIKE LTAK-LGNUM,         DN\_PLANT   LIKE LIPS-WERKS,         DN\_GROUP   LIKE VBSS-SAMMG,         DN\_NUM     LIKE LIKP-VBELN,         DN\_DATE    LIKE LIKP-LFDAT,         DN\_TYPE    LIKE LIKP-LFART,         SO\_NUM     LIKE VBAK-VBELN,         SO\_TYPE    LIKE VBAK-AUART,         IV\_NUM     LIKE VBRK-VBELN,         SHIPTO\_NUM LIKE KNA1-KUNNR,         SHIPTO\_NAM LIKE KNA1-NAME1,         SJZ1\_NUM   LIKE KNA1-KUNNR,         SJZ1\_NAM   LIKE KNA1-NAME1,         SJZ3\_NUM   LIKE KNA1-KUNNR,         SJZ3\_NAM   LIKE KNA1-NAME2,         SJZ4\_NUM   LIKE KNA1-KUNNR,         SJZ4\_NAM   LIKE KNA1-SORTL,         HAVECWT    TYPE CHAR10,         UPDATCWT   TYPE CHAR10,         1STTO      LIKE LTAK-TANUM,         1STCFM     TYPE CHAR10,         2NDTO      TYPE CHAR10,         2NDCFM     TYPE CHAR10,         DNPGI      TYPE CHAR10,         LAST       TYPE CHAR10,         MESAGE     TYPE STRING,       END OF IT\_DATA.  *"抄磅数据* DATA: BEGIN OF IT\_WEIGHT OCCURS 0,         LFDAT    LIKE LIKP-LFDAT,         DN\_PLANT LIKE LIPS-WERKS,         DN\_NUM   LIKE LIKP-VBELN,         DN\_ITEM  LIKE LIPS-POSNR,         SO\_NUM   LIKE VBAK-VBELN,         SO\_ITEM  LIKE VBAP-POSNR,         MATNR    LIKE VBAP-MATNR,         SJZ1\_NAM LIKE KNA1-NAME1,         SJZ1\_NUM LIKE KNA1-KUNNR,         QUANTITY LIKE LIPS-LFIMG,         WEIGHT   LIKE LIPS-NTGEW,         UNIT     LIKE LIPS-GEWEI,       END OF IT\_WEIGHT.  *"合作伙伴信息* DATA:IT\_VBPA LIKE VBPA OCCURS 0 WITH HEADER LINE. DATA:IT\_KNA1 LIKE KNA1 OCCURS 0 WITH HEADER LINE. DATA:IT\_CWDN LIKE IT\_WEIGHT OCCURS 0 WITH HEADER LINE. *"中间处理表* DATA:IT\_TEMP LIKE IT\_DATA OCCURS 0 WITH HEADER LINE. *"司机最后一站* DATA: IT\_ZWM41 LIKE ZTWM041 OCCURS 0 WITH HEADER LINE.  *"IMENU* DATA: BEGIN OF IMENU OCCURS 0,         MENU LIKE SY-UCOMM,       END OF IMENU.  *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** *\* FIELD SYMBOLS/定义指针* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** FIELD-SYMBOLS: <IT\_DATA> LIKE IT\_DATA.  *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** *\* CONSTANTS :定义产量* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** CONTROLS: TABCONT01 TYPE TABLEVIEW USING SCREEN 1000. DATA:G\_TABCONT01\_LINES  LIKE SY-LOOPC.  *"中间处理行数255字符* DATA:LT\_TEXT TYPE STANDARD TABLE OF CHAR255. *"文本内表* DATA:LS\_TEXT LIKE LINE OF LT\_TEXT.  *"自定义长文本框变量* DATA: GTM\_COMPONENT\_LTEXT TYPE REF TO CL\_GUI\_CUSTOM\_CONTAINER,        *"长文本控件对象*       GUI\_COMPONENT\_LTEXT TYPE SCRFNAME VALUE 'CATCHWEIGHT\_TEXT',     *"GUI层面控件*       GUI\_LONG\_TEXTEDITOR TYPE REF TO CL\_GUI\_TEXTEDIT.                *"最终显示用户的空间对象*  *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** *\*SELECTION-SCREEN :定义选择屏幕中的选择字段* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**   *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** *\* INITIALIZATION:程序初始化时候执行的代码* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** INITIALIZATION.   PERFORM FRM\_INITAL.  *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** *\*AT SELECTION-SCREEN:在选择屏幕上执行的代码* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** AT SELECTION-SCREEN.   *""""[CODE]*  *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** *\*START-OF-SELECTION:程序运行所处理的代码* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** START-OF-SELECTION.   *"記錄使用次數*   CALL FUNCTION 'ZRFSY002'     EXPORTING       I\_UNAME = SY-UNAME       I\_TCODE = SY-TCODE       I\_PRONM = SY-REPID.   *"執行程序*   PERFORM FRM\_INITAL.   CALL SCREEN 1000.  *\*&---------------------------------------------------------------------\** *\*&      Form  FRM\_INITAL* *\*&---------------------------------------------------------------------\** FORM FRM\_INITAL.   *"默认日期*   LV\_DNDATE = SY-DATUM.    *"文件类型*   GV\_FILTER = 'EXCEL-Files (\*.XLSX)|\*.XLSX|All Files(\*.\*)|\*.\*|'. ENDFORM.  *\*&---------------------------------------------------------------------\** *\*&      Module  SET\_STATUS\_DATA  OUTPUT* *\*&---------------------------------------------------------------------\** MODULE SET\_STATUS\_DATA OUTPUT.   READ TABLE IT\_DATA  INDEX TABCONT01-CURRENT\_LINE.   IF SY-SUBRC = 0.     IF IT\_DATA-STATUS = 'S'.       CALL FUNCTION 'ICON\_CREATE'         EXPORTING           NAME                  = 'ICON\_GREEN\_LIGHT'           INFO                  = 'S'           ADD\_STDINF            = 'X'         IMPORTING           RESULT                = IT\_DATA-LIGHT         EXCEPTIONS           ICON\_NOT\_FOUND        = 1           OUTPUTFIELD\_TOO\_SHORT = 2           OTHERS                = 3.     ELSEIF IT\_DATA-STATUS = 'E'.       CALL FUNCTION 'ICON\_CREATE'         EXPORTING           NAME                  = 'ICON\_RED\_LIGHT'           INFO                  = 'E'           ADD\_STDINF            = 'X'         IMPORTING           RESULT                = IT\_DATA-LIGHT         EXCEPTIONS           ICON\_NOT\_FOUND        = 1           OUTPUTFIELD\_TOO\_SHORT = 2           OTHERS                = 3.     ELSE.       CALL FUNCTION 'ICON\_CREATE'         EXPORTING           NAME                  = 'ICON\_LIGHT\_OUT'           INFO                  = 'S'           ADD\_STDINF            = 'X'         IMPORTING           RESULT                = IT\_DATA-LIGHT         EXCEPTIONS           ICON\_NOT\_FOUND        = 1           OUTPUTFIELD\_TOO\_SHORT = 2           OTHERS                = 3.     ENDIF.   ENDIF. ENDMODULE.  *\*&---------------------------------------------------------------------\** *\*&      Module  ENABLE\_FIELDS  OUTPUT* *\*&---------------------------------------------------------------------\** MODULE ENABLE\_FIELDS OUTPUT.   G\_TABCONT01\_LINES = SY-LOOPC. ENDMODULE.  *\*&---------------------------------------------------------------------\** *\*&      Module  RES\_INPUTS\_1000  INPUT* *\*&---------------------------------------------------------------------\** MODULE RES\_INPUTS\_1000 INPUT. *\*  MOVE-CORRESPONDING TABCONT01 TO IT\_DATA.*   MODIFY IT\_DATA INDEX TABCONT01-CURRENT\_LINE. ENDMODULE.  *\*&---------------------------------------------------------------------\** *\*&      Module  STATUS\_1000  OUTPUT* *\*&---------------------------------------------------------------------\** *\*       text* *\*----------------------------------------------------------------------\** MODULE STATUS\_1000 OUTPUT.   SET PF-STATUS 'STATUS\_1000'.   SET TITLEBAR 'TITLE\_1000'. ENDMODULE.  *\*&---------------------------------------------------------------------\** *\*&      Module  USER\_COMMAND\_1000  INPUT* *\*&---------------------------------------------------------------------\** *\*       text* *\*----------------------------------------------------------------------\** MODULE USER\_COMMAND\_1000 INPUT.   *"判斷事件對象*   CASE SY-UCOMM.     WHEN 'TEMPLATE'.       PERFORM FRM\_DOWNLOAD\_TMP USING '' 'ZWM46'.     WHEN 'LOAD'.       PERFORM FRM\_GET\_DATA.     WHEN 'SELALL'.       PERFORM FRM\_SELECT\_1000 USING 'ALL'.     WHEN 'SELNO'.       PERFORM FRM\_SELECT\_1000 USING 'NO'.     WHEN 'SELOP'.       PERFORM FRM\_SELECT\_1000 USING 'OP'.     WHEN 'LASTSTOP'.       PERFORM FRM\_SAVE\_LAST\_STOP.     WHEN 'PRNT\_INV'.       PERFORM FRM\_PRINT\_INVOICE.     WHEN 'DBCLICK'.       GET CURSOR LINE GV\_ROW.       GV\_ROW = TABCONT01-TOP\_LINE + GV\_ROW - 1.       GET CURSOR FIELD GV\_FIELD.        *"显示凭证-DN Group* *\*      IF ( GV\_FIELD = 'IT\_DATA-DN\_GROUP').* *\*        READ TABLE IT\_DATA INDEX GV\_ROW.* *\*        IF SY-SUBRC = 0 AND IT\_DATA-DN\_GROUP <> ''.* *\*          SET PARAMETER ID 'GRN' FIELD IT\_DATA-DN\_GROUP.* *\*          CALL TRANSACTION 'VG03' AND SKIP FIRST SCREEN.* *\*        ENDIF.* *\*      ENDIF.*        *"显示凭证-WM 2 STEP PICK*       IF ( GV\_FIELD = 'IT\_DATA-DN\_GROUP').         READ TABLE IT\_DATA INDEX GV\_ROW.         IF SY-SUBRC = 0 AND IT\_DATA-DN\_GROUP <> ''.           DATA: LT\_SELTAB TYPE STANDARD TABLE OF RSPARAMS WITH HEADER LINE.           LT\_SELTAB-SELNAME = 'REFNR'.           LT\_SELTAB-SIGN    = 'I'.           LT\_SELTAB-OPTION  = 'EQ'.           LT\_SELTAB-LOW     = IT\_DATA-DN\_GROUP.           APPEND LT\_SELTAB.            LT\_SELTAB-SELNAME = 'DATUM'.           LT\_SELTAB-SIGN    = 'I'.           LT\_SELTAB-OPTION  = 'BT'.           LT\_SELTAB-LOW     = '20151101'.           LT\_SELTAB-HIGH    = LV\_DNDATE.           APPEND LT\_SELTAB.            SUBMIT RL2STK00 WITH LGNUM EQ IT\_DATA-DN\_HOUSE                           WITH SELECTION-TABLE LT\_SELTAB                           AND RETURN.         ENDIF.       ENDIF.        *"显示凭证-销售订单*       IF ( GV\_FIELD = 'IT\_DATA-SO\_NUM').         READ TABLE IT\_DATA INDEX GV\_ROW.         IF SY-SUBRC = 0 AND IT\_DATA-SO\_NUM <> ''.           SET PARAMETER ID 'AUN' FIELD IT\_DATA-SO\_NUM.           CALL TRANSACTION 'VA03' AND SKIP FIRST SCREEN.         ENDIF.       ENDIF.       *"显示凭证-交货单*       IF ( GV\_FIELD = 'IT\_DATA-DN\_NUM').         READ TABLE IT\_DATA INDEX GV\_ROW.         IF SY-SUBRC = 0 AND IT\_DATA-DN\_NUM <> ''.           SET PARAMETER ID 'VL' FIELD IT\_DATA-DN\_NUM.           CALL TRANSACTION 'VL03N' AND SKIP FIRST SCREEN.         ENDIF.       ENDIF.       *"显示凭证-发票*       IF ( GV\_FIELD = 'IT\_DATA-IV\_NUM').         READ TABLE IT\_DATA INDEX GV\_ROW.         IF SY-SUBRC = 0 AND IT\_DATA-IV\_NUM <> ''.           SET PARAMETER ID 'VF' FIELD IT\_DATA-IV\_NUM.           CALL TRANSACTION 'VF03' AND SKIP FIRST SCREEN.         ENDIF.       ENDIF.        *"显示抄磅数据*       IF ( GV\_FIELD = 'IT\_DATA-UPDATCWT').         READ TABLE IT\_DATA INDEX GV\_ROW.         IF SY-SUBRC = 0 AND IT\_DATA-UPDATCWT <> ''.            *"1002屏幕比那里*           CLEAR: LV\_DNNUM\_1002.           CLEAR: LV\_KUNNR\_1002.           CLEAR: LV\_NAME1\_1002.           LV\_DNNUM\_1002 = IT\_DATA-DN\_NUM.           LV\_KUNNR\_1002 = IT\_DATA-SHIPTO\_NUM.           LV\_NAME1\_1002 = IT\_DATA-SHIPTO\_NAM.            *"读取DN记录的抄磅信息*           PERFORM FRM\_GET\_DN\_CATCH\_WEIGHT USING IT\_DATA-DN\_NUM.            CALL SCREEN 1002 STARTING AT 30 10 ENDING   AT 180   30.         ENDIF.       ENDIF.     WHEN 'TOCFM'.       PERFORM FRM\_TO\_CONFIRM.     WHEN 'UPTCW'.       DATA: LV\_ANS TYPE C.       CLEAR LV\_ANS.       LV\_MESSAGE = '导入抄磅数据前，请先确认第一步TO和第二步TO已经确认完毕！'.       CALL FUNCTION 'POPUP\_TO\_CONFIRM'         EXPORTING           TITLEBAR              = 'CONFIRMATION'           TEXT\_QUESTION         = LV\_MESSAGE           TEXT\_BUTTON\_1         = '是,继续导入'           ICON\_BUTTON\_1         = 'ICON\_OKAY'           TEXT\_BUTTON\_2         = '否,取消导入'           ICON\_BUTTON\_2         = 'ICON\_CANCEL'           DEFAULT\_BUTTON        = '2'           DISPLAY\_CANCEL\_BUTTON = ' '         IMPORTING           ANSWER                = LV\_ANS         EXCEPTIONS           TEXT\_NOT\_FOUND        = 1           OTHERS                = 2.       IF LV\_ANS = '1'.         CALL SCREEN 1001 STARTING AT 30 10 ENDING   AT 80   10.       ENDIF.     WHEN 'DNPGI'.       PERFORM FRM\_DN\_PGI.     WHEN 'INVOICE'.       PERFORM FRM\_DN\_INVOICE.     WHEN 'REFRESH'.       PERFORM FRM\_REFRESH\_DATA.     WHEN 'BACK' OR 'EXIT'.       LEAVE PROGRAM.   ENDCASE.   CLEAR SY-UCOMM. ENDMODULE.  *\*&---------------------------------------------------------------------\** *\*&      Form  FRM\_SELECT\_1000* *\*&---------------------------------------------------------------------\** FORM FRM\_SELECT\_1000 USING STROPTION TYPE CHAR10.   LOOP AT IT\_DATA ASSIGNING <IT\_DATA>.     IF STROPTION = 'ALL'.       <IT\_DATA>-SEL = 'X'.     ELSEIF STROPTION = 'NO'.       <IT\_DATA>-SEL = ''.     ELSEIF STROPTION = 'OP'.       IF <IT\_DATA>-SEL = 'X'.         <IT\_DATA>-SEL = ''.       ELSE.         <IT\_DATA>-SEL = 'X'.       ENDIF.     ENDIF.   ENDLOOP. ENDFORM.  *\*&---------------------------------------------------------------------\** *\*&      Form  FRM\_GET\_DATA* *\*&---------------------------------------------------------------------\** FORM FRM\_GET\_DATA.   CLEAR IT\_DATA.   REFRESH IT\_DATA.    DATA:LV\_CHECK TYPE CHAR1 VALUE ''.   IF LV\_DNDATE = '00000000'.     GV\_MESSAGE = '查询条件：Delivery Date 不能为空！'.     MESSAGE GV\_MESSAGE  TYPE 'I' DISPLAY LIKE 'E'.     LV\_CHECK = 'X'.   ENDIF.    IF LV\_PLANT = ''.     GV\_MESSAGE = '查询条件：Delivery Plant 不能为空！'.     MESSAGE GV\_MESSAGE  TYPE 'I' DISPLAY LIKE 'E'.     LV\_CHECK = 'X'.   ENDIF.  *\*  IF LV\_DRIVER = ''.* *\*    GV\_MESSAGE = '查询条件：Driver Name 不能为空！'.* *\*    MESSAGE GV\_MESSAGE  TYPE 'I' DISPLAY LIKE 'E'.* *\*    LV\_CHECK = 'X'.* *\*  ENDIF.*    IF LV\_CHECK = 'X'.     *"默認可控制行数*     TABCONT01-LINES = LINES( IT\_DATA ).     LV\_RECORD = LINES( IT\_DATA ).     RETURN.   ENDIF.    *"查询所有的订单之后再过滤数据*   SELECT     VSTEL AS DN\_PLANT     VBELN AS DN\_NUM     LFART AS DN\_TYPE     LGNUM AS DN\_HOUSE     LFDAT AS DN\_DATE     INTO CORRESPONDING FIELDS OF TABLE IT\_DATA     FROM LIKP     WHERE LFDAT = LV\_DNDATE     AND VSTEL = LV\_PLANT     AND ( LFART = 'LF' OR LFART = 'LR' )     AND VKORG = '1000'     AND VSBED <> '02'. *"排除上门PickUp订单*    *"简单排序*   SORT IT\_DATA BY DN\_NUM ASCENDING.    *"查询合作伙伴信息*   IF LINES( IT\_DATA ) > 0.     SELECT VBELN PARVW KUNNR ADRNR       INTO CORRESPONDING FIELDS OF TABLE IT\_VBPA       FROM VBPA       FOR ALL ENTRIES IN IT\_DATA        WHERE VBELN = IT\_DATA-DN\_NUM        AND ( PARVW = 'WE' OR PARVW = 'Z1' OR PARVW = 'Z3' OR PARVW = 'Z4'  ).     *"排序*     SORT IT\_VBPA BY VBELN PARVW ASCENDING.      *"查询所有派单主数据信息1~2000*     IF LINES( IT\_VBPA ) > 0.       SELECT KUNNR NAME1 NAME2 SORTL       INTO CORRESPONDING FIELDS OF TABLE IT\_KNA1         FROM KNA1         WHERE KUNNR BETWEEN '0000000001' AND '0000002000'.       *"排序*       SORT IT\_KNA1 BY KUNNR ASCENDING .     ENDIF.   ENDIF.    *"修改数据*   DATA:LV\_MATNR LIKE LIPS-MATNR.   LOOP AT IT\_DATA ASSIGNING <IT\_DATA>.     *"DN\_GROUP*     SELECT SINGLE SAMMG INTO <IT\_DATA>-DN\_GROUP FROM VBSS WHERE VBELN = <IT\_DATA>-DN\_NUM.      *"SO\_NUM*     <IT\_DATA>-SO\_NUM = ''.     SELECT SINGLE VGBEL INTO <IT\_DATA>-SO\_NUM FROM LIPS WHERE VBELN = <IT\_DATA>-DN\_NUM.     *"SO TYPE*     IF <IT\_DATA>-SO\_NUM <> ''.       SELECT SINGLE AUART INTO <IT\_DATA>-SO\_TYPE FROM VBAK WHERE VBELN = <IT\_DATA>-SO\_NUM.     ENDIF.      *"判断交货单中是否有抄磅物料*     CLEAR LV\_MATNR.     SELECT SINGLE MATNR INTO LV\_MATNR  FROM LIPS WHERE VBELN = <IT\_DATA>-DN\_NUM AND MATNR LIKE '%CW%'.     IF LV\_MATNR IS NOT INITIAL.       <IT\_DATA>-HAVECWT = 'YES'.     ELSE.       <IT\_DATA>-HAVECWT = ''.     ENDIF.      *"读取SHIP TO信息*     CLEAR LV\_ADDRESS\_NUM.     CLEAR IT\_VBPA.     READ TABLE IT\_VBPA WITH KEY VBELN = <IT\_DATA>-DN\_NUM PARVW = 'WE' BINARY SEARCH.     IF SY-SUBRC = 0.       <IT\_DATA>-SHIPTO\_NUM = IT\_VBPA-KUNNR.       LV\_ADDRESS\_NUM = IT\_VBPA-ADRNR.     ENDIF.     IF NOT LV\_ADDRESS\_NUM IS INITIAL.       SELECT SINGLE NAME1  INTO (<IT\_DATA>-SHIPTO\_NAM ) FROM ADRC WHERE ADDRNUMBER = LV\_ADDRESS\_NUM.     ENDIF.      *"读取司机信息*     CLEAR IT\_VBPA.     READ TABLE IT\_VBPA WITH KEY VBELN = <IT\_DATA>-DN\_NUM PARVW = 'Z1' BINARY SEARCH.     IF SY-SUBRC = 0.       <IT\_DATA>-SJZ1\_NUM = IT\_VBPA-KUNNR.       READ TABLE IT\_KNA1 WITH  KEY KUNNR = <IT\_DATA>-SJZ1\_NUM BINARY SEARCH.       IF SY-SUBRC = 0.         <IT\_DATA>-SJZ1\_NAM = IT\_KNA1-SORTL.       ENDIF.     ENDIF.      *"读取站数信息*     CLEAR IT\_VBPA.     READ TABLE IT\_VBPA WITH KEY VBELN = <IT\_DATA>-DN\_NUM PARVW = 'Z3' BINARY SEARCH.     IF SY-SUBRC = 0.       <IT\_DATA>-SJZ3\_NUM = IT\_VBPA-KUNNR.       READ TABLE IT\_KNA1 WITH  KEY KUNNR = <IT\_DATA>-SJZ3\_NUM BINARY SEARCH.       IF SY-SUBRC = 0.         <IT\_DATA>-SJZ3\_NAM = IT\_KNA1-NAME2.       ENDIF.     ENDIF.      *"读取货车信息*     CLEAR IT\_VBPA.     READ TABLE IT\_VBPA WITH KEY VBELN = <IT\_DATA>-DN\_NUM PARVW = 'Z4' BINARY SEARCH.     IF SY-SUBRC = 0.       <IT\_DATA>-SJZ4\_NUM = IT\_VBPA-KUNNR.       READ TABLE IT\_KNA1 WITH  KEY KUNNR = <IT\_DATA>-SJZ4\_NUM BINARY SEARCH.       IF SY-SUBRC = 0.         <IT\_DATA>-SJZ4\_NAM = IT\_KNA1-SORTL.       ENDIF.     ENDIF.      IF <IT\_DATA>-SHIPTO\_NUM <> ''.       CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_OUTPUT'         EXPORTING           INPUT  = <IT\_DATA>-SHIPTO\_NUM         IMPORTING           OUTPUT = <IT\_DATA>-SHIPTO\_NUM.     ENDIF.      IF <IT\_DATA>-SJZ1\_NUM <> ''.       CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_OUTPUT'         EXPORTING           INPUT  = <IT\_DATA>-SJZ1\_NUM         IMPORTING           OUTPUT = <IT\_DATA>-SJZ1\_NUM.     ENDIF.   ENDLOOP.    *"按司机过滤数据*   IF LV\_DRIVER <> ''.     DELETE IT\_DATA WHERE SJZ1\_NAM <> LV\_DRIVER.   ENDIF.    *"最终排序*   SORT IT\_DATA BY SJZ1\_NUM SJZ3\_NUM ASCENDING.    LV\_SSNUM = 1.   *"修改数据*   LOOP AT IT\_DATA ASSIGNING <IT\_DATA>.     *"序号*     <IT\_DATA>-SSNUM = LV\_SSNUM.     LV\_SSNUM = LV\_SSNUM + 1.      *"判断是否已经PGI*     <IT\_DATA>-DNPGI = ''.     SELECT SINGLE WBSTK INTO <IT\_DATA>-DNPGI FROM VBUK WHERE VBELN = <IT\_DATA>-DN\_NUM.     IF <IT\_DATA>-DNPGI = 'C'.       <IT\_DATA>-DNPGI = 'PGI'.     ELSE.       <IT\_DATA>-DNPGI = ''.     ENDIF.      *"判断是否已经Confirm 1st TO*     <IT\_DATA>-1STTO = ''.     <IT\_DATA>-1STCFM = ''.     IF <IT\_DATA>-DN\_NUM <> '' AND <IT\_DATA>-DN\_TYPE = 'LF'.       SELECT SINGLE TANUM KQUIT INTO (<IT\_DATA>-1STTO,<IT\_DATA>-1STCFM) FROM LTAK         WHERE LGNUM = <IT\_DATA>-DN\_HOUSE  AND REFNR = <IT\_DATA>-DN\_GROUP AND VBELN = ''.       IF <IT\_DATA>-1STCFM = 'X'.         <IT\_DATA>-1STCFM = 'Confirm'.       ENDIF.     ENDIF.     IF <IT\_DATA>-DN\_TYPE = 'LR'.       <IT\_DATA>-1STCFM = 'No Need'.     ENDIF.      *"判断是否已经Confirm 2nd TO*     <IT\_DATA>-2NDTO = ''.     IF <IT\_DATA>-DN\_NUM <> '' AND <IT\_DATA>-DN\_TYPE = 'LF'.       SELECT SINGLE VBELN INTO <IT\_DATA>-2NDTO FROM VBFA WHERE VBELV = <IT\_DATA>-DN\_NUM AND VBTYP\_N = 'Q'.       IF <IT\_DATA>-2NDTO <> ''.         SELECT SINGLE KQUIT INTO <IT\_DATA>-2NDCFM FROM LTAK WHERE LGNUM = <IT\_DATA>-DN\_HOUSE  AND TANUM = <IT\_DATA>-2NDTO.         IF <IT\_DATA>-2NDCFM = 'X'.           <IT\_DATA>-2NDCFM = 'Confirm'.         ENDIF.       ENDIF.     ENDIF.     IF <IT\_DATA>-DN\_TYPE = 'LR'.       <IT\_DATA>-2NDCFM = 'No Need'.     ENDIF.      *"关联是否有发票号*     IF <IT\_DATA>-DN\_NUM <> ''.       SELECT SINGLE VBELN INTO <IT\_DATA>-IV\_NUM FROM VBFA WHERE VBELV = <IT\_DATA>-DN\_NUM AND VBTYP\_N = 'M'.     ENDIF.      *"退货订单发票号*     IF <IT\_DATA>-DN\_TYPE = 'LR'.       SELECT SINGLE VBELN INTO <IT\_DATA>-IV\_NUM FROM VBFA WHERE VBELV = <IT\_DATA>-SO\_NUM AND VBTYP\_N = 'O'.     ENDIF.      *"关联是否最后一张*     <IT\_DATA>-LAST = ''.     SELECT SINGLE SORTL INTO <IT\_DATA>-LAST FROM ZTWM041 WHERE SDATE = <IT\_DATA>-DN\_DATE AND SORTL = <IT\_DATA>-SJZ1\_NAM AND LSTOP = <IT\_DATA>-SJZ3\_NAM.     IF <IT\_DATA>-LAST <> ''.       <IT\_DATA>-LAST = 'Last'.     ENDIF.   ENDLOOP.    *"判断是否已经更新抄磅信息-DN 关键信息修改历史*   LOOP AT IT\_DATA ASSIGNING <IT\_DATA> WHERE HAVECWT = 'YES'.     CLEAR LV\_BRGEW.     SELECT SINGLE FNAME INTO LV\_BRGEW FROM CDPOS WHERE OBJECTCLAS = 'LIEFERUNG' AND OBJECTID = <IT\_DATA>-DN\_NUM       AND TABNAME ='LIPS' AND FNAME = 'BRGEW' AND CHNGIND = 'U'.     CLEAR LV\_NTGEW.     SELECT SINGLE FNAME INTO LV\_NTGEW FROM CDPOS WHERE OBJECTCLAS = 'LIEFERUNG' AND OBJECTID = <IT\_DATA>-DN\_NUM       AND TABNAME ='LIPS' AND FNAME = 'NTGEW' AND CHNGIND = 'U'.     CLEAR LV\_UMVKN.     SELECT SINGLE FNAME INTO LV\_UMVKN FROM CDPOS WHERE OBJECTCLAS = 'LIEFERUNG' AND OBJECTID = <IT\_DATA>-DN\_NUM       AND TABNAME ='LIPS' AND FNAME = 'UMVKN' AND CHNGIND = 'U'.     CLEAR LV\_UMVKZ.     SELECT SINGLE FNAME INTO LV\_UMVKZ FROM CDPOS WHERE OBJECTCLAS = 'LIEFERUNG' AND OBJECTID = <IT\_DATA>-DN\_NUM       AND TABNAME ='LIPS' AND FNAME = 'UMVKZ' AND CHNGIND = 'U'.     CLEAR LV\_VOLUM.     SELECT SINGLE FNAME INTO LV\_VOLUM FROM CDPOS WHERE OBJECTCLAS = 'LIEFERUNG' AND OBJECTID = <IT\_DATA>-DN\_NUM       AND TABNAME ='LIPS' AND FNAME = 'VOLUM' AND CHNGIND = 'U'.      <IT\_DATA>-UPDATCWT = ''.     IF LV\_BRGEW <> '' AND LV\_NTGEW <> '' AND LV\_UMVKZ <> '' .*"AND LV\_VOLUM <> ''. "AND LV\_UMVKN <> ''*       <IT\_DATA>-UPDATCWT = 'YES'.     ENDIF.   ENDLOOP.    *"默認可控制行数*   TABCONT01-LINES = LINES( IT\_DATA ).   LV\_RECORD = LINES( IT\_DATA ). ENDFORM.   *\*&---------------------------------------------------------------------\** *\*&      Form  FRM\_CHECK\_ORDER* *\*&---------------------------------------------------------------------\** FORM FRM\_CHECK\_ORDER USING       P\_MATNR LIKE MARC-MATNR       P\_RETURN TYPE INT4.  ENDFORM.  *\*&---------------------------------------------------------------------\** *\*&      Form  FRM\_1ST\_CONFIRM* *\*&---------------------------------------------------------------------\** FORM FRM\_TO\_CONFIRM.   *"检查是否选择订单*   READ TABLE IT\_DATA WITH KEY SEL = 'X'.   IF SY-SUBRC <> 0.     MESSAGE '请先选中需要处理的交货订单!' TYPE 'I' DISPLAY LIKE 'E'.   ENDIF.    DATA:IT\_DN\_LIST LIKE ZSWM035 OCCURS 0 WITH HEADER LINE.   CLEAR IT\_DN\_LIST.   REFRESH IT\_DN\_LIST.   LOOP AT IT\_DATA WHERE SEL = 'X'.     IT\_DN\_LIST-DN\_NUM = IT\_DATA-DN\_NUM.     APPEND IT\_DN\_LIST.   ENDLOOP.    IF LINES( IT\_DN\_LIST ) > 0.     READ TABLE IT\_DATA INDEX 1.     IF SY-SUBRC = 0.       TM\_PLANT = IT\_DATA-DN\_PLANT.     ENDIF.      CLEAR LV\_STATUS.     CLEAR LV\_MESSAGE.     *"调用接口批量确认TO*     CALL FUNCTION 'ZRFWM049'       EXPORTING         I\_PLANT    = TM\_PLANT       IMPORTING         O\_STATUS   = LV\_STATUS         O\_MESSAGE  = LV\_MESSAGE       TABLES         IT\_DN\_LIST = IT\_DN\_LIST.   ENDIF.     *"根棍数据状态*   LOOP AT IT\_DATA ASSIGNING <IT\_DATA> WHERE SEL = 'X'.     *"判断是否已经Confirm 1st TO*     <IT\_DATA>-1STTO = ''.     <IT\_DATA>-1STCFM = ''.     IF <IT\_DATA>-DN\_NUM <> '' AND <IT\_DATA>-DN\_TYPE = 'LF'.       SELECT SINGLE TANUM KQUIT INTO (<IT\_DATA>-1STTO,<IT\_DATA>-1STCFM) FROM LTAK         WHERE LGNUM = <IT\_DATA>-DN\_HOUSE  AND REFNR = <IT\_DATA>-DN\_GROUP AND VBELN = ''.       IF <IT\_DATA>-1STCFM = 'X'.         <IT\_DATA>-1STCFM = 'Confirm'.       ENDIF.     ENDIF.     IF <IT\_DATA>-DN\_TYPE = 'LR'.       <IT\_DATA>-1STCFM = 'No Need'.     ENDIF.      *"判断是否已经Confirm 2nd TO*     <IT\_DATA>-2NDTO = ''.     IF <IT\_DATA>-DN\_NUM <> '' AND <IT\_DATA>-DN\_TYPE = 'LF'.       SELECT SINGLE VBELN INTO <IT\_DATA>-2NDTO FROM VBFA WHERE VBELV = <IT\_DATA>-DN\_NUM AND VBTYP\_N = 'Q'.       IF <IT\_DATA>-2NDTO <> ''.         SELECT SINGLE KQUIT INTO <IT\_DATA>-2NDCFM FROM LTAK WHERE LGNUM = <IT\_DATA>-DN\_HOUSE  AND TANUM = <IT\_DATA>-2NDTO.         IF <IT\_DATA>-2NDCFM = 'X'.           <IT\_DATA>-2NDCFM = 'Confirm'.         ENDIF.       ENDIF.     ENDIF.     IF <IT\_DATA>-DN\_TYPE = 'LR'.       <IT\_DATA>-2NDCFM = 'No Need'.     ENDIF.   ENDLOOP.  ENDFORM.   *\*&---------------------------------------------------------------------\** *\*&      Form  FRM\_DN\_PGI Delivery Order Post Goods Issue* *\*&---------------------------------------------------------------------\** FORM FRM\_DN\_PGI.   *"检查是否选择订单*   READ TABLE IT\_DATA WITH KEY SEL = 'X'.   IF SY-SUBRC <> 0.     MESSAGE '请先选中需要处理的交货订单!' TYPE 'I' DISPLAY LIKE 'E'.   ENDIF.    DATA:IT\_DN\_LIST LIKE ZSWM038 OCCURS 0 WITH HEADER LINE.   *"调用接口批量PGI 排除退货订单*   LOOP AT IT\_DATA WHERE SEL = 'X' AND DN\_TYPE <> 'LR'.     IT\_DN\_LIST-DN\_NUM = IT\_DATA-DN\_NUM.     APPEND IT\_DN\_LIST.   ENDLOOP.    IF LINES( IT\_DN\_LIST ) > 0.     CLEAR LV\_STATUS.     CLEAR LV\_MESSAGE.     CALL FUNCTION 'ZRFWM051'       IMPORTING         O\_STATUS   = LV\_STATUS         O\_MESSAGE  = LV\_MESSAGE       TABLES         IT\_DN\_LIST = IT\_DN\_LIST.   ENDIF.    WAIT UP TO 3 SECONDS.    *"根棍数据状态*   LOOP AT IT\_DATA ASSIGNING <IT\_DATA> WHERE SEL = 'X'.     *"判断是否已经PGI*     <IT\_DATA>-DNPGI = ''.     SELECT SINGLE WBSTK INTO <IT\_DATA>-DNPGI FROM VBUK WHERE VBELN = <IT\_DATA>-DN\_NUM.     IF <IT\_DATA>-DNPGI = 'C'.       <IT\_DATA>-DNPGI = 'PGI'.     ELSE.       <IT\_DATA>-DNPGI = ''.     ENDIF.   ENDLOOP.  ENDFORM.  *\*&---------------------------------------------------------------------\** *\*&      Form  FRM\_DN\_INVOICE Delivery Invoice* *\*&---------------------------------------------------------------------\** FORM FRM\_DN\_INVOICE.   *"检查是否选择订单*   READ TABLE IT\_DATA WITH KEY SEL = 'X'.   IF SY-SUBRC <> 0.     MESSAGE '请先选中需要处理的交货订单!' TYPE 'I' DISPLAY LIKE 'E'.   ENDIF.    DATA:IT\_DN\_LIST LIKE ZSWM039 OCCURS 0 WITH HEADER LINE.   *"调用接口批量INVOICE 排除退货订单开发票*   LOOP AT IT\_DATA WHERE SEL = 'X' AND DN\_TYPE <> 'LR'.     IT\_DN\_LIST-DN\_NUM = IT\_DATA-DN\_NUM.     APPEND IT\_DN\_LIST.   ENDLOOP.    IF LINES( IT\_DN\_LIST ) > 0.     CLEAR LV\_STATUS.     CLEAR LV\_MESSAGE.     CALL FUNCTION 'ZRFWM052'       IMPORTING         O\_STATUS   = LV\_STATUS         O\_MESSAGE  = LV\_MESSAGE       TABLES         IT\_DN\_LIST = IT\_DN\_LIST.   ENDIF.    *"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**   *"如果是Yamimeal订单，则调用WEB API推送数据-抄磅信息和订单状态*   DATA: LV\_OFFON  TYPE CHAR1 VALUE ''.   SELECT SINGLE OFFON INTO LV\_OFFON FROM ZTSY001 WHERE APPNM = 'ZRPWM046'.   IF LV\_OFFON = 'X'.     PERFORM FRM\_PUSH\_WEB\_API\_YAMIMEAL.   ENDIF.   *"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**    *"根棍数据状态*   LOOP AT IT\_DATA ASSIGNING <IT\_DATA> WHERE SEL = 'X'.     *"关联是否有发票号*     IF <IT\_DATA>-DN\_NUM <> ''.       SELECT SINGLE VBELN INTO <IT\_DATA>-IV\_NUM FROM VBFA WHERE VBELV = <IT\_DATA>-DN\_NUM AND VBTYP\_N = 'M'.     ENDIF.      *"退货订单发票号*     IF <IT\_DATA>-DN\_TYPE = 'LR'.       SELECT SINGLE VBELN INTO <IT\_DATA>-IV\_NUM FROM VBFA WHERE VBELV = <IT\_DATA>-SO\_NUM AND VBTYP\_N = 'O'.     ENDIF.   ENDLOOP. ENDFORM.  *\*&---------------------------------------------------------------------\** *\*&      Module  STATUS\_1001  OUTPUT* *\*&---------------------------------------------------------------------\** *\*       text* *\*----------------------------------------------------------------------\** MODULE STATUS\_1001 OUTPUT.   SET PF-STATUS 'STATUS\_1001'. *\*  SET TITLEBAR 'xxx'.* ENDMODULE.  *\*&---------------------------------------------------------------------\** *\*&      Module  USER\_COMMAND\_1001  INPUT* *\*&---------------------------------------------------------------------\** *\*       text* *\*----------------------------------------------------------------------\** MODULE USER\_COMMAND\_1001 INPUT.   *"判斷事件對象*   CASE SY-UCOMM.     WHEN 'CHOOSE'.       PERFORM FRM\_SELECT\_FILE USING GV\_FILTER GV\_FILE.     WHEN 'IMPORT'.       IF GV\_FILE <>''.         PERFORM FRM\_GET\_EXCEL\_DATA USING GV\_FILE                 '1'                 *"开始列*                 '1'                 *"开始行*                 '12'                *"结束列*                 '1001'.             *"结束行*         PERFORM FRM\_IMPORT\_DATA.    *"导入数据*         *"修改DN订单的抄磅数据*         PERFORM FRM\_UPT\_DN\_CWEIGHT.       ENDIF.       CLEAR GV\_FILE.       LEAVE TO SCREEN 0.     WHEN 'EXIT'.       CLEAR GV\_FILE.       LEAVE TO SCREEN 0.   ENDCASE.   CLEAR SY-UCOMM. ENDMODULE.  *\*&---------------------------------------------------------------------\** *\*&      Form  FRM\_IMPORT\_DATA* *\*&---------------------------------------------------------------------\** FORM FRM\_IMPORT\_DATA.   CLEAR IT\_WEIGHT..   REFRESH IT\_WEIGHT..    LOOP AT IT\_UPLOAD\_DATA.     IT\_WEIGHT-LFDAT      = IT\_UPLOAD\_DATA-DATA01.     IT\_WEIGHT-DN\_PLANT   = IT\_UPLOAD\_DATA-DATA02.     CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'       EXPORTING         INPUT  = IT\_UPLOAD\_DATA-DATA03       IMPORTING         OUTPUT = IT\_WEIGHT-DN\_NUM.     CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'       EXPORTING         INPUT  = IT\_UPLOAD\_DATA-DATA04       IMPORTING         OUTPUT = IT\_WEIGHT-DN\_ITEM.     CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'       EXPORTING         INPUT  = IT\_UPLOAD\_DATA-DATA05       IMPORTING         OUTPUT = IT\_WEIGHT-SO\_NUM.     CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'       EXPORTING         INPUT  = IT\_UPLOAD\_DATA-DATA06       IMPORTING         OUTPUT = IT\_WEIGHT-SO\_ITEM.     CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'       EXPORTING         INPUT  = IT\_UPLOAD\_DATA-DATA07       IMPORTING         OUTPUT = IT\_WEIGHT-MATNR.     IT\_WEIGHT-SJZ1\_NAM   = IT\_UPLOAD\_DATA-DATA08.     IT\_WEIGHT-SJZ1\_NUM   = IT\_UPLOAD\_DATA-DATA09.     IT\_WEIGHT-QUANTITY   = IT\_UPLOAD\_DATA-DATA10.     IT\_WEIGHT-WEIGHT     = IT\_UPLOAD\_DATA-DATA11.     IT\_WEIGHT-UNIT       = IT\_UPLOAD\_DATA-DATA12.     APPEND IT\_WEIGHT.     CLEAR IT\_WEIGHT.   ENDLOOP.    SORT  IT\_WEIGHT BY DN\_NUM DN\_ITEM ASCENDING.    CLEAR IT\_UPLOAD\_DATA.   REFRESH IT\_UPLOAD\_DATA. ENDFORM.  *\*&---------------------------------------------------------------------\** *\*&      Form  FRM\_UPT\_DN\_CWEIGHT* *\*&---------------------------------------------------------------------\** FORM FRM\_UPT\_DN\_CWEIGHT.   *"统计订单个数*   IT\_CWDN[] = IT\_WEIGHT[].   SORT  IT\_CWDN BY DN\_NUM ASCENDING.   DELETE ADJACENT DUPLICATES FROM IT\_CWDN COMPARING DN\_NUM.    *"循环修改磅数*   DATA:IT\_DN\_CATCH LIKE ZSWM036 OCCURS 0 WITH HEADER LINE.   LOOP AT IT\_CWDN.     CLEAR IT\_DN\_CATCH.     REFRESH IT\_DN\_CATCH.      LOOP AT IT\_WEIGHT WHERE DN\_NUM = IT\_CWDN-DN\_NUM.       IT\_DN\_CATCH-VBELN = IT\_WEIGHT-DN\_NUM.       IT\_DN\_CATCH-POSNR = IT\_WEIGHT-DN\_ITEM.       IT\_DN\_CATCH-CATCH = IT\_WEIGHT-WEIGHT.       IT\_DN\_CATCH-MEINS = IT\_WEIGHT-UNIT.       APPEND IT\_DN\_CATCH.     ENDLOOP.      IF LINES( IT\_DN\_CATCH ) > 0.       *"抄磅信息由小到大排序输入*       SORT IT\_DN\_CATCH BY VBELN POSNR CATCH ASCENDING.        CLEAR LV\_STATUS.       CLEAR LV\_MESSAGE.       CALL FUNCTION 'ZRFWM050'         EXPORTING           I\_DNNUM     = IT\_CWDN-DN\_NUM         IMPORTING           O\_STATUS    = LV\_STATUS           O\_MESSAGE   = LV\_MESSAGE         TABLES           IT\_DN\_CATCH = IT\_DN\_CATCH.     ENDIF.   ENDLOOP.    WAIT UP TO 3 SECONDS.    *"判断是否已经更新抄磅信息-DN 关键信息修改历史*   LOOP AT IT\_DATA ASSIGNING <IT\_DATA> WHERE HAVECWT = 'YES'.     CLEAR LV\_BRGEW.     SELECT SINGLE FNAME INTO LV\_BRGEW FROM CDPOS WHERE OBJECTCLAS = 'LIEFERUNG' AND OBJECTID = <IT\_DATA>-DN\_NUM       AND TABNAME ='LIPS' AND FNAME = 'BRGEW' AND CHNGIND = 'U'.     CLEAR LV\_NTGEW.     SELECT SINGLE FNAME INTO LV\_NTGEW FROM CDPOS WHERE OBJECTCLAS = 'LIEFERUNG' AND OBJECTID = <IT\_DATA>-DN\_NUM       AND TABNAME ='LIPS' AND FNAME = 'NTGEW' AND CHNGIND = 'U'.     CLEAR LV\_UMVKN.     SELECT SINGLE FNAME INTO LV\_UMVKN FROM CDPOS WHERE OBJECTCLAS = 'LIEFERUNG' AND OBJECTID = <IT\_DATA>-DN\_NUM       AND TABNAME ='LIPS' AND FNAME = 'UMVKN' AND CHNGIND = 'U'.     CLEAR LV\_UMVKZ.     SELECT SINGLE FNAME INTO LV\_UMVKZ FROM CDPOS WHERE OBJECTCLAS = 'LIEFERUNG' AND OBJECTID = <IT\_DATA>-DN\_NUM       AND TABNAME ='LIPS' AND FNAME = 'UMVKZ' AND CHNGIND = 'U'.     CLEAR LV\_VOLUM.     SELECT SINGLE FNAME INTO LV\_VOLUM FROM CDPOS WHERE OBJECTCLAS = 'LIEFERUNG' AND OBJECTID = <IT\_DATA>-DN\_NUM       AND TABNAME ='LIPS' AND FNAME = 'VOLUM' AND CHNGIND = 'U'.      <IT\_DATA>-UPDATCWT = ''.     IF LV\_BRGEW <> '' AND LV\_NTGEW <> '' AND LV\_UMVKZ <> ''.*" AND LV\_VOLUM <> ''. "AND LV\_UMVKN <> ''*       <IT\_DATA>-UPDATCWT = 'YES'.     ENDIF.   ENDLOOP. ENDFORM.  *\*&---------------------------------------------------------------------\** *\*&      Module  STATUS\_1002  OUTPUT* *\*&---------------------------------------------------------------------\** *\*       text* *\*----------------------------------------------------------------------\** MODULE STATUS\_1002 OUTPUT.   SET PF-STATUS 'STATUS\_1002'.   PERFORM FRM\_CREATE\_LONG\_TEXT USING 1. ENDMODULE.  *\*&---------------------------------------------------------------------------------\** *\*&  Form FRM\_CREATE\_LONG\_TEXT* *\*&---------------------------------------------------------------------------------\** *\*&  Parameter:P\_STATUS用户控制控件状态 1表示只读，0表示可编辑* *\*&---------------------------------------------------------------------------------\** FORM FRM\_CREATE\_LONG\_TEXT USING P\_STATUS TYPE I.    *"判断GUI控件释放已经初始化*   IF GUI\_LONG\_TEXTEDITOR IS INITIAL.     *"创建长文本对象*     CREATE OBJECT GTM\_COMPONENT\_LTEXT       EXPORTING         CONTAINER\_NAME              = GUI\_COMPONENT\_LTEXT       EXCEPTIONS         CNTL\_ERROR                  = 1         CNTL\_SYSTEM\_ERROR           = 2         CREATE\_ERROR                = 3         LIFETIME\_ERROR              = 4         LIFETIME\_DYNPRO\_DYNPRO\_LINK = 5.      *"创建控件相关属性和值*     CREATE OBJECT GUI\_LONG\_TEXTEDITOR       EXPORTING         PARENT                     = GTM\_COMPONENT\_LTEXT         WORDWRAP\_MODE              = CL\_GUI\_TEXTEDIT=>WORDWRAP\_AT\_WINDOWBORDER         WORDWRAP\_TO\_LINEBREAK\_MODE = CL\_GUI\_TEXTEDIT=>TRUE.   ENDIF.    *"控制长文本释放只读:1表示只读，0表示可编辑*   GUI\_LONG\_TEXTEDITOR->SET\_READONLY\_MODE( P\_STATUS ).   *"控制是否显示工具栏：TRUE表示显示；FALSE表示隐藏*   GUI\_LONG\_TEXTEDITOR->SET\_TOOLBAR\_MODE( TOOLBAR\_MODE = CL\_GUI\_TEXTEDIT=>FALSE ).   *"控制是否显示状态栏：TRUE表示显示；FALSE表示隐藏*   GUI\_LONG\_TEXTEDITOR->SET\_STATUSBAR\_MODE( STATUSBAR\_MODE = CL\_GUI\_TEXTEDIT=>FALSE ).    *"绑定长文本数值*   GUI\_LONG\_TEXTEDITOR->SET\_TEXT\_AS\_R3TABLE( TABLE = LT\_TEXT ). ENDFORM.  *\*&---------------------------------------------------------------------------------\** *\*&  Form FRM\_CREATE\_LONG\_TEXT* *\*&---------------------------------------------------------------------------------\** FORM FRM\_GET\_DN\_CATCH\_WEIGHT USING P\_DNNUM LIKE LIKP-VBELN.   DATA:IT\_DN\_CATCH LIKE LIPS OCCURS 0 WITH HEADER LINE.    CLEAR LT\_TEXT.   REFRESH LT\_TEXT.    *"查询抄磅物料信息*   SELECT \* INTO CORRESPONDING FIELDS OF TABLE IT\_DN\_CATCH     FROM LIPS     WHERE VBELN = P\_DNNUM AND MATNR LIKE '%CW%'. *"."*   SORT IT\_DN\_CATCH BY VBELN ASCENDING.    DATA:STR\_LFIMG TYPE CHAR20 VALUE ''.   DATA:INT\_LFIMG TYPE INT4.   DATA:STR\_NTGEW TYPE CHAR20 VALUE ''.    *"每一交货单位重量*   DATA:FLT\_PERUNIT LIKE LIPS-LGMNG.   DATA:STR\_PERUNIT TYPE CHAR20 VALUE ''.    DATA:LV\_TEXT\_NAME TYPE THEAD-TDNAME.   DATA:IT\_LINES LIKE TABLE OF TLINE WITH HEADER LINE .   LOOP AT IT\_DN\_CATCH.     *"串写物料信息 & 数量 & 总磅数*     CLEAR LS\_TEXT.     CLEAR STR\_LFIMG.     CLEAR STR\_NTGEW.      *"物料编码*     CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_OUTPUT'       EXPORTING         INPUT  = IT\_DN\_CATCH-MATNR       IMPORTING         OUTPUT = IT\_DN\_CATCH-MATNR.      *"数量 转换成INT过滤小数点*     INT\_LFIMG = IT\_DN\_CATCH-LFIMG.     STR\_LFIMG = INT\_LFIMG.     CONDENSE STR\_LFIMG.      *"总净重*     STR\_NTGEW = IT\_DN\_CATCH-NTGEW.     CONDENSE STR\_NTGEW.      *"单位输出转换-销售单位*     CALL FUNCTION 'CONVERSION\_EXIT\_CUNIT\_OUTPUT'       EXPORTING         INPUT  = IT\_DN\_CATCH-VRKME       IMPORTING         OUTPUT = IT\_DN\_CATCH-VRKME.     *"单位输出转换-基础单位*     CALL FUNCTION 'CONVERSION\_EXIT\_CUNIT\_OUTPUT'       EXPORTING         INPUT  = IT\_DN\_CATCH-MEINS       IMPORTING         OUTPUT = IT\_DN\_CATCH-MEINS.      *"每一交货单位重量*     FLT\_PERUNIT = IT\_DN\_CATCH-LGMNG / IT\_DN\_CATCH-LFIMG.     STR\_PERUNIT = FLT\_PERUNIT.     CONDENSE STR\_PERUNIT.      CONCATENATE IT\_DN\_CATCH-POSNR '- 【' IT\_DN\_CATCH-MATNR '】 ' IT\_DN\_CATCH-ARKTX                 '【' STR\_PERUNIT  IT\_DN\_CATCH-MEINS '/' IT\_DN\_CATCH-VRKME '】'                 INTO LS\_TEXT SEPARATED BY SPACE.     APPEND LS\_TEXT TO LT\_TEXT.      CLEAR LS\_TEXT.     CONCATENATE '【数量：' STR\_LFIMG IT\_DN\_CATCH-VRKME '，总净重：' STR\_NTGEW ' LB】' INTO LS\_TEXT SEPARATED BY SPACE.     APPEND LS\_TEXT TO LT\_TEXT.      *"文本中记录的抄磅信息-明细*     CLEAR LS\_TEXT.     CONCATENATE IT\_DN\_CATCH-VBELN IT\_DN\_CATCH-POSNR INTO LV\_TEXT\_NAME.     REFRESH IT\_LINES.     CALL FUNCTION 'READ\_TEXT'       EXPORTING         CLIENT                  = SY-MANDT         ID                      = '0002'         LANGUAGE                = SY-LANGU         NAME                    = LV\_TEXT\_NAME         OBJECT                  = 'VBBP'       TABLES         LINES                   = IT\_LINES       EXCEPTIONS         ID                      = 1         LANGUAGE                = 2         NAME                    = 3         NOT\_FOUND               = 4         OBJECT                  = 5         REFERENCE\_CHECK         = 6         WRONG\_ACCESS\_TO\_ARCHIVE = 7         OTHERS                  = 8.     LOOP AT IT\_LINES.       CLEAR LS\_TEXT.       LS\_TEXT = IT\_LINES-TDLINE.       APPEND LS\_TEXT TO LT\_TEXT.     ENDLOOP.      *"空行分隔*     CLEAR LS\_TEXT.     LS\_TEXT = '  '.     APPEND LS\_TEXT TO LT\_TEXT.   ENDLOOP. ENDFORM.  *\*&---------------------------------------------------------------------\** *\*&      Module  USER\_COMMAND\_1002  INPUT* *\*&---------------------------------------------------------------------\** *\*       text* *\*----------------------------------------------------------------------\** MODULE USER\_COMMAND\_1002 INPUT.   *"判斷事件對象*   CASE SY-UCOMM.     WHEN 'CLOSE'.       CLEAR GV\_FILE.       LEAVE TO SCREEN 0.   ENDCASE.   CLEAR SY-UCOMM. ENDMODULE.  *\*&---------------------------------------------------------------------\** *\*&      FRM\_PRINT\_INVOICE 批量打印发票* *\*&---------------------------------------------------------------------\** FORM FRM\_PRINT\_INVOICE.   RANGES: R\_VBELN FOR VBRK-VBELN,           R\_FKART FOR VBRK-FKART,           R\_FKDAT FOR VBRK-FKDAT.    *"清空打印条件*   CLEAR R\_VBELN.REFRESH R\_VBELN.   CLEAR R\_FKART.REFRESH R\_FKART.   CLEAR R\_FKDAT.REFRESH R\_FKDAT.   *"按单号关键打印*   LOOP AT IT\_DATA WHERE SEL = 'X' AND IV\_NUM IS NOT INITIAL .     R\_VBELN-SIGN   = 'I'.     R\_VBELN-OPTION = 'EQ'.     R\_VBELN-LOW    = IT\_DATA-IV\_NUM.     APPEND R\_VBELN.   ENDLOOP.    IF R\_VBELN[] IS NOT INITIAL.     SUBMIT YSD\_SALES\_INVOICE\_MASSPRINT WITH SO\_VBELN IN R\_VBELN                     WITH SO\_FKART IN R\_FKART                     WITH SO\_FKDAT IN R\_FKDAT                     *"WITH C\_PVIEW EQ 'X'  "打印预览*                     WITH C\_SFLAG EQ 'X' *"合并打印*                     *"WITH C\_SPRINT EQ 'X' "单张打印*                     AND RETURN.   ELSE.     MESSAGE '请先选中需要打印的发票号码！' TYPE 'I' DISPLAY LIKE 'E'.   ENDIF. ENDFORM.  *\*&---------------------------------------------------------------------\** *\*&      FRM\_SAVE\_LAST\_STOP 设置最后一站* *\*&---------------------------------------------------------------------\** FORM FRM\_SAVE\_LAST\_STOP.   DATA:P\_FLAG TYPE CHAR1 VALUE ''.   *"判断是否有勾选*   READ TABLE IT\_DATA WITH KEY SEL = 'X'.   IF SY-SUBRC <> 0.     MESSAGE '您未勾选最后一站（Last Stop），请检查！' TYPE 'I' DISPLAY LIKE 'E'.     P\_FLAG = 'X'.   ENDIF.    *"判断勾选是否一个*   IT\_TEMP[] = IT\_DATA[].   DELETE IT\_TEMP WHERE SEL <> 'X'.   IF LINES( IT\_TEMP ) > 1.     MESSAGE '您勾选多于一个最后一站（Last Stop），请检查！' TYPE 'I' DISPLAY LIKE 'E'.     P\_FLAG = 'X'.   ENDIF.    IF P\_FLAG = 'X'.     RETURN.   ENDIF.    *"判断是否已经排司机*   IT\_TEMP[] = IT\_DATA[].   DELETE IT\_TEMP WHERE SJZ1\_NAM = ''.   IF LINES( IT\_TEMP ) = 0.     MESSAGE '您的单据中仍没有维护司机信息，请检查！' TYPE 'I' DISPLAY LIKE 'E'.     P\_FLAG = 'X'.   ENDIF.    IF P\_FLAG = 'X'.     RETURN.   ENDIF.    *"续写数据到目标数据表*   CLEAR IT\_ZWM41.REFRESH IT\_ZWM41.   IF P\_FLAG = ''.     LOOP AT IT\_DATA ASSIGNING <IT\_DATA> WHERE SEL = 'X'.       IT\_ZWM41-MANDT = SY-MANDT.       IT\_ZWM41-SDATE = <IT\_DATA>-DN\_DATE.       SELECT SINGLE KUNNR INTO IT\_ZWM41-Z1KNR FROM KNA1 WHERE SORTL = <IT\_DATA>-SJZ1\_NAM.       IT\_ZWM41-SORTL = <IT\_DATA>-SJZ1\_NAM.       IF  <IT\_DATA>-SJZ3\_NAM <> ''.         IT\_ZWM41-LSTOP = <IT\_DATA>-SJZ3\_NAM.       ENDIF.       IT\_ZWM41-CRTOR = SY-UNAME.       IT\_ZWM41-CRDAT = SY-DATUM.       IT\_ZWM41-CRTME = SY-UZEIT.       APPEND IT\_ZWM41.     ENDLOOP.   ENDIF.    *"有续写数据*   IF LINES( IT\_ZWM41 ) > 0.     MODIFY ZTWM041 FROM TABLE IT\_ZWM41.     MESSAGE '最后一张已经成功保存！' TYPE 'I' DISPLAY LIKE 'I'.   ENDIF.    *"刷新数据*   LOOP AT IT\_DATA ASSIGNING <IT\_DATA>.     *"关联是否最后一张*     <IT\_DATA>-LAST = ''.     SELECT SINGLE SORTL INTO <IT\_DATA>-LAST FROM ZTWM041 WHERE SDATE = <IT\_DATA>-DN\_DATE AND SORTL = <IT\_DATA>-SJZ1\_NAM AND LSTOP = <IT\_DATA>-SJZ3\_NAM.     IF <IT\_DATA>-LAST <> ''.       <IT\_DATA>-LAST = 'Last'.     ENDIF.   ENDLOOP. ENDFORM.  *\*&---------------------------------------------------------------------\** *\*&      FRM\_PUSH\_WEB\_API\_YAMIMEAL 推送Yamimeal WEB API* *\*&---------------------------------------------------------------------\** FORM FRM\_PUSH\_WEB\_API\_YAMIMEAL.    *"WEB API &JSON涉及变量参数*   DATA: LV\_SERVICE1 TYPE STRING.   DATA: LV\_SERVICE2 TYPE STRING.   DATA: LV\_JSON     TYPE STRING.   DATA: LV\_RESULT   TYPE STRING.   DATA: LV\_STATUS   TYPE STRING.    DATA: LT\_LIPS\_CW     LIKE LIPS OCCURS 0 WITH HEADER LINE.   DATA: BEGIN OF LS\_JSON,           SO\_ID      TYPE VBRP-VGBEL,           ITEM\_NO    TYPE VBRP-POSNR,           MATNR      TYPE VBRP-MATNR,           INVOICE\_ID TYPE ACCIT-AWREF,           DRIVER     TYPE KNA1-SORTL,           QUANTITY   TYPE FKIMG,           SITE(45)   TYPE C,           TOTAL\_LB   TYPE VBRP-NTGEW,           LB\_LIST    TYPE ZSLB\_LIST1,         END OF LS\_JSON.   DATA: LT\_JSON      LIKE TABLE OF LS\_JSON.   DATA: LV\_TDNAME   TYPE STXH-TDNAME.   DATA: LT\_LINE     TYPE TABLE OF TLINE.   DATA: LS\_LB\_LIST   TYPE ZSLB\_LIST.   DATA: LT\_LB\_LIST   TYPE TABLE OF ZSLB\_LIST.    DATA: LO\_JSON\_SER  TYPE REF TO CL\_TREX\_JSON\_SERIALIZER.   DATA: LC\_POST\_API TYPE REF TO ZCL\_SD\_001.    DATA: LV\_SONUM    LIKE VBAK-VBELN.    *"只针对选中且是ZPM的订单类型*   LOOP AT IT\_DATA WHERE SEL = 'X' AND SO\_TYPE = 'ZPM'.      *"取表第一条数据*     SELECT SINGLE TEXT01 INTO LV\_SERVICE1 FROM ZTSY003 WHERE TEXT02 = '01'.     *"获取抄磅货品:SJFOOD VINCENT要求抛所有物料。*     SELECT \* INTO CORRESPONDING FIELDS OF TABLE LT\_LIPS\_CW       FROM LIPS WHERE VBELN = IT\_DATA-DN\_NUM. *" AND MATNR LIKE '%CW%'.*      LOOP AT LT\_LIPS\_CW.       LS\_JSON-SO\_ID      = LT\_LIPS\_CW-VGBEL.       LS\_JSON-ITEM\_NO    = LT\_LIPS\_CW-VGPOS.       LS\_JSON-MATNR      = LT\_LIPS\_CW-MATNR.       LS\_JSON-DRIVER     = IT\_DATA-SJZ1\_NAM.       LS\_JSON-QUANTITY   = LT\_LIPS\_CW-LFIMG.       LS\_JSON-SITE       = IT\_DATA-SJZ3\_NAM.       LS\_JSON-TOTAL\_LB   = LT\_LIPS\_CW-NTGEW.        CLEAR LV\_TDNAME.       LV\_TDNAME = LT\_LIPS\_CW-VBELN && LT\_LIPS\_CW-POSNR.        CLEAR: LT\_LINE[].       CALL FUNCTION 'READ\_TEXT'         EXPORTING           OBJECT          = 'VBBP'           NAME            = LV\_TDNAME           ID              = '0002'           LANGUAGE        = 'E'         TABLES           LINES           = LT\_LINE         EXCEPTIONS           OBJECT          = 1           ID              = 2           LANGUAGE        = 3           NAME            = 4           NOT\_FOUND       = 5           REFERENCE\_CHECK = 6.       CLEAR LT\_LB\_LIST[].       LOOP AT LT\_LINE INTO DATA(LS\_LINE).         REPLACE ALL OCCURRENCES OF 'Catch Wgt:' IN LS\_LINE-TDLINE WITH ''.         LS\_LB\_LIST-SORT = SY-TABIX.         LS\_LB\_LIST-TDLINE = LS\_LINE-TDLINE.         APPEND LS\_LB\_LIST TO LT\_LB\_LIST.       ENDLOOP.       APPEND LINES OF LT\_LB\_LIST TO LS\_JSON-LB\_LIST.       APPEND LS\_JSON TO LT\_JSON.       CLEAR LS\_JSON.     ENDLOOP.      *"内表->JSON*     CREATE OBJECT LO\_JSON\_SER       EXPORTING         DATA = LT\_JSON[].     CALL METHOD LO\_JSON\_SER->SERIALIZE.     CALL METHOD LO\_JSON\_SER->GET\_DATA       RECEIVING         RVAL = LV\_JSON.      REPLACE ALL OCCURRENCES OF 'so\_id'      IN LV\_JSON WITH '"so\_id"'.     REPLACE ALL OCCURRENCES OF 'item\_no'    IN LV\_JSON WITH '"item\_no"'.     REPLACE ALL OCCURRENCES OF 'matnr'      IN LV\_JSON WITH '"matnr"'.     REPLACE ALL OCCURRENCES OF 'invoice\_id' IN LV\_JSON WITH '"invoice\_id"'.     REPLACE ALL OCCURRENCES OF 'driver'     IN LV\_JSON WITH '"driver"'.     REPLACE ALL OCCURRENCES OF 'quantity'   IN LV\_JSON WITH '"Quantity"'.     REPLACE ALL OCCURRENCES OF 'site'       IN LV\_JSON WITH '"Site"'.     REPLACE ALL OCCURRENCES OF 'total\_lb'   IN LV\_JSON WITH '"Total\_lb"'.     REPLACE ALL OCCURRENCES OF 'lb\_list'    IN LV\_JSON WITH '"lb\_list"'.     REPLACE ALL OCCURRENCES OF 'sort'       IN LV\_JSON WITH '"sort"'.     REPLACE ALL OCCURRENCES OF 'tdline'     IN LV\_JSON WITH '"tdline"'.      CHECK LV\_JSON IS NOT INITIAL.     CREATE OBJECT LC\_POST\_API.     CALL METHOD LC\_POST\_API->POST\_DATA       EXPORTING         IV\_JSON    = LV\_JSON         IV\_SERVICE = LV\_SERVICE1       IMPORTING         EV\_RESULT  = LV\_RESULT         EV\_STATUS  = LV\_STATUS.     IF SY-SUBRC EQ 0.       *"获取最新的折扣价格*     ENDIF.      *"Start 如果是ZPM订单成功开具发票后需要更新外平台的订单状态\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**     CLEAR LV\_SERVICE2.     CLEAR LV\_SONUM.     LV\_SONUM = IT\_DATA-SO\_NUM.     IF LV\_SONUM <>''.       SELECT SINGLE TEXT01 INTO LV\_SERVICE2 FROM ZTSY003 WHERE TEXT02 = '05'.       LV\_SERVICE2 = LV\_SERVICE2 && LV\_SONUM.       CREATE OBJECT LC\_POST\_API.       CALL METHOD LC\_POST\_API->POST\_DATA         EXPORTING           IV\_JSON    = LV\_JSON           IV\_SERVICE = LV\_SERVICE2         IMPORTING           EV\_RESULT  = LV\_RESULT           EV\_STATUS  = LV\_STATUS.     ENDIF.     *"End 如果是ZPM订单成功开具发票后需要更新外平台的订单状态\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**   ENDLOOP. ENDFORM.  *\*&---------------------------------------------------------------------\** *\*&      Form  FRM\_REFRESH\_DATA* *\*&---------------------------------------------------------------------\** FORM FRM\_REFRESH\_DATA.    *"修改数据*   LOOP AT IT\_DATA ASSIGNING <IT\_DATA>.      *"判断是否已经PGI*     <IT\_DATA>-DNPGI = ''.     SELECT SINGLE WBSTK INTO <IT\_DATA>-DNPGI FROM VBUK WHERE VBELN = <IT\_DATA>-DN\_NUM.     IF <IT\_DATA>-DNPGI = 'C'.       <IT\_DATA>-DNPGI = 'PGI'.     ELSE.       <IT\_DATA>-DNPGI = ''.     ENDIF.      *"判断是否已经Confirm 1st TO*     <IT\_DATA>-1STTO = ''.     <IT\_DATA>-1STCFM = ''.     IF <IT\_DATA>-DN\_NUM <> '' AND <IT\_DATA>-DN\_TYPE = 'LF'.       SELECT SINGLE TANUM KQUIT INTO (<IT\_DATA>-1STTO,<IT\_DATA>-1STCFM) FROM LTAK         WHERE LGNUM = <IT\_DATA>-DN\_HOUSE  AND REFNR = <IT\_DATA>-DN\_GROUP AND VBELN = ''.       IF <IT\_DATA>-1STCFM = 'X'.         <IT\_DATA>-1STCFM = 'Confirm'.       ENDIF.     ENDIF.     IF <IT\_DATA>-DN\_TYPE = 'LR'.       <IT\_DATA>-1STCFM = 'No Need'.     ENDIF.      *"判断是否已经Confirm 2nd TO*     <IT\_DATA>-2NDTO = ''.     IF <IT\_DATA>-DN\_NUM <> '' AND <IT\_DATA>-DN\_TYPE = 'LF'.       SELECT SINGLE VBELN INTO <IT\_DATA>-2NDTO FROM VBFA WHERE VBELV = <IT\_DATA>-DN\_NUM AND VBTYP\_N = 'Q'.       IF <IT\_DATA>-2NDTO <> ''.         SELECT SINGLE KQUIT INTO <IT\_DATA>-2NDCFM FROM LTAK WHERE LGNUM = <IT\_DATA>-DN\_HOUSE  AND TANUM = <IT\_DATA>-2NDTO.         IF <IT\_DATA>-2NDCFM = 'X'.           <IT\_DATA>-2NDCFM = 'Confirm'.         ENDIF.       ENDIF.     ENDIF.     IF <IT\_DATA>-DN\_TYPE = 'LR'.       <IT\_DATA>-2NDCFM = 'No Need'.     ENDIF.      *"关联是否有发票号*     IF <IT\_DATA>-DN\_NUM <> ''.       SELECT SINGLE VBELN INTO <IT\_DATA>-IV\_NUM FROM VBFA WHERE VBELV = <IT\_DATA>-DN\_NUM AND VBTYP\_N = 'M'.     ENDIF.      *"退货订单发票号*     IF <IT\_DATA>-DN\_TYPE = 'LR'.       SELECT SINGLE VBELN INTO <IT\_DATA>-IV\_NUM FROM VBFA WHERE VBELV = <IT\_DATA>-SO\_NUM AND VBTYP\_N = 'O'.     ENDIF.      *"关联是否最后一张*     <IT\_DATA>-LAST = ''.     SELECT SINGLE SORTL INTO <IT\_DATA>-LAST FROM ZTWM041 WHERE SDATE = <IT\_DATA>-DN\_DATE AND SORTL = <IT\_DATA>-SJZ1\_NAM AND LSTOP = <IT\_DATA>-SJZ3\_NAM.     IF <IT\_DATA>-LAST <> ''.       <IT\_DATA>-LAST = 'Last'.     ENDIF.   ENDLOOP.    *"判断是否已经更新抄磅信息-DN 关键信息修改历史*   LOOP AT IT\_DATA ASSIGNING <IT\_DATA> WHERE HAVECWT = 'YES'.     CLEAR LV\_BRGEW.     SELECT SINGLE FNAME INTO LV\_BRGEW FROM CDPOS WHERE OBJECTCLAS = 'LIEFERUNG' AND OBJECTID = <IT\_DATA>-DN\_NUM       AND TABNAME ='LIPS' AND FNAME = 'BRGEW' AND CHNGIND = 'U'.     CLEAR LV\_NTGEW.     SELECT SINGLE FNAME INTO LV\_NTGEW FROM CDPOS WHERE OBJECTCLAS = 'LIEFERUNG' AND OBJECTID = <IT\_DATA>-DN\_NUM       AND TABNAME ='LIPS' AND FNAME = 'NTGEW' AND CHNGIND = 'U'.     CLEAR LV\_UMVKN.     SELECT SINGLE FNAME INTO LV\_UMVKN FROM CDPOS WHERE OBJECTCLAS = 'LIEFERUNG' AND OBJECTID = <IT\_DATA>-DN\_NUM       AND TABNAME ='LIPS' AND FNAME = 'UMVKN' AND CHNGIND = 'U'.     CLEAR LV\_UMVKZ.     SELECT SINGLE FNAME INTO LV\_UMVKZ FROM CDPOS WHERE OBJECTCLAS = 'LIEFERUNG' AND OBJECTID = <IT\_DATA>-DN\_NUM       AND TABNAME ='LIPS' AND FNAME = 'UMVKZ' AND CHNGIND = 'U'.     CLEAR LV\_VOLUM.     SELECT SINGLE FNAME INTO LV\_VOLUM FROM CDPOS WHERE OBJECTCLAS = 'LIEFERUNG' AND OBJECTID = <IT\_DATA>-DN\_NUM       AND TABNAME ='LIPS' AND FNAME = 'VOLUM' AND CHNGIND = 'U'.      <IT\_DATA>-UPDATCWT = ''.     IF LV\_BRGEW <> '' AND LV\_NTGEW <> '' AND LV\_UMVKZ <> '' .*"AND LV\_VOLUM <> ''. "AND LV\_UMVKN <> ''*       <IT\_DATA>-UPDATCWT = 'YES'.     ENDIF.   ENDLOOP.    *"默認可控制行数*   TABCONT01-LINES = LINES( IT\_DATA ).   LV\_RECORD = LINES( IT\_DATA ). ENDFORM. |



ZRFWM049 BarCode:Batch Confrim TOs From Delivery Order One Dirver

|  |
| --- |
|  |
| *"BDC相关内表* DATA:GT\_BDCDATA    LIKE BDCDATA OCCURS 0 WITH HEADER LINE,      GT\_BDCMSGCOLL LIKE BDCMSGCOLL OCCURS 0 WITH HEADER LINE,      GT\_MSG        LIKE BAPIRET2 OCCURS 0 WITH HEADER LINE.  FUNCTION ZRFWM049. *\*"----------------------------------------------------------------------* *\*"\*"Local Interface:* *\*"  IMPORTING* *\*"     VALUE(I\_PLANT) LIKE  LIPS-WERKS OPTIONAL* *\*"  EXPORTING* *\*"     VALUE(O\_STATUS) TYPE  CHAR1* *\*"     VALUE(O\_MESSAGE) TYPE  CHAR100* *\*"  TABLES* *\*"      IT\_DN\_LIST STRUCTURE  ZSWM035 OPTIONAL* *\*"----------------------------------------------------------------------* *\*  逻辑：* *\* 1.按DN单查找GROUP，按GROUP检查1st TO是否已经你确认，未确认的话进行确认。* *\* 2.按GROUP检查是否创建 2nd TO，未创建TO的话创建TO LT42* *\* 3.检查2nd TO是否已经确认，未确认的话进行确认* *\*"----------------------------------------------------------------------*   IF I\_PLANT = ''.     O\_STATUS = 'E'.     O\_MESSAGE = 'Plant is Required!'.     RETURN.   ENDIF.    IF LINES( IT\_DN\_LIST ) = 0.     O\_STATUS = 'E'.     O\_MESSAGE = 'DN Order is Required!'.     RETURN.   ENDIF.    *"货仓编码*   DATA: LV\_LGNUM LIKE LTAK-LGNUM.   DATA: LV\_TANUM LIKE LTAK-TANUM VALUE ''.   DATA: LV\_DN\_WEIGHT LIKE LIKP-BTGEW VALUE 0.    *"查找对应的GROUP*   DATA: IT\_VBSS LIKE VBSS OCCURS 0 WITH HEADER LINE.   DATA: IT\_GROUP\_DN LIKE VBSS OCCURS 0 WITH HEADER LINE.   *"1st TO第一步TO*   DATA: IT\_LTAK\_1ST LIKE LTAK OCCURS 0 WITH HEADER LINE.   *"2nd TO 第一步TO*   DATA: IT\_LTAK\_2ND LIKE LTAK OCCURS 0 WITH HEADER LINE.   *"BAPI结构表*   DATA: LT\_LTAP\_CONF LIKE TABLE OF LTAP\_CONF WITH HEADER LINE.    *"----------------------------------------------------------------------*   FIELD-SYMBOLS: <IT\_DN\_LIST> LIKE IT\_DN\_LIST.   *"----------------------------------------------------------------------*    *"获取对应的WM货仓号*   CALL FUNCTION 'ZRFSY005'     EXPORTING       I\_WERKS = I\_PLANT     IMPORTING       O\_LGNUM = LV\_LGNUM.    *"处理查询数据*   SORT IT\_DN\_LIST BY DN\_NUM ASCENDING.   DELETE ADJACENT DUPLICATES FROM IT\_DN\_LIST COMPARING DN\_NUM.   LOOP AT IT\_DN\_LIST ASSIGNING <IT\_DN\_LIST>.     *"补0*     CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'       EXPORTING         INPUT  = <IT\_DN\_LIST>-DN\_NUM       IMPORTING         OUTPUT = <IT\_DN\_LIST>-DN\_NUM.      *"如果DN订单的重量为0，则理解成系服务订单本质不需要TO，所以默认TO全部成功。*     CLEAR LV\_DN\_WEIGHT.     LV\_DN\_WEIGHT = 0.     SELECT SINGLE BTGEW INTO LV\_DN\_WEIGHT       FROM LIKP WHERE VBELN = <IT\_DN\_LIST>-DN\_NUM.     IF LV\_DN\_WEIGHT = 0.       <IT\_DN\_LIST>-TO1\_CONFIRM = 'S'.       <IT\_DN\_LIST>-TO2\_CONFIRM = 'S'.       <IT\_DN\_LIST>-MESSAGE = 'Service Order Not need TO(s)'.     ENDIF.      *"默认所有TO都能够被确认*     <IT\_DN\_LIST>-TO1\_CONFIRM = 'S'.   ENDLOOP.   *"处理查询数据*   SORT IT\_DN\_LIST BY DN\_NUM ASCENDING.   DELETE ADJACENT DUPLICATES FROM IT\_DN\_LIST COMPARING DN\_NUM.    *"查询未确认的: 1st - TOs*   IF LINES( IT\_DN\_LIST ) > 0.     *"查找DN对应的GROUP*     SELECT \* INTO CORRESPONDING FIELDS OF TABLE IT\_VBSS       FROM VBSS       FOR ALL ENTRIES IN IT\_DN\_LIST       WHERE VBELN = IT\_DN\_LIST-DN\_NUM.     SORT IT\_VBSS BY SAMMG ASCENDING.     DELETE ADJACENT DUPLICATES FROM IT\_VBSS COMPARING SAMMG.      *"查找第一步TO*     IF LINES( IT\_VBSS ) > 0.       SELECT \* INTO CORRESPONDING FIELDS OF TABLE IT\_LTAK\_1ST         FROM LTAK         FOR ALL ENTRIES IN IT\_VBSS         WHERE LGNUM = LV\_LGNUM AND KQUIT = '' AND VBELN = ''         AND REFNR = IT\_VBSS-SAMMG.     ENDIF.   ENDIF.    *"循环确认第一步TO*   LOOP AT IT\_LTAK\_1ST WHERE KQUIT = ''.     CLEAR LT\_LTAP\_CONF.REFRESH LT\_LTAP\_CONF.     CALL FUNCTION 'L\_TO\_CONFIRM'       EXPORTING         I\_LGNUM     = IT\_LTAK\_1ST-LGNUM  *"仓库号*         I\_TANUM     = IT\_LTAK\_1ST-TANUM  *"TO号码*         I\_SQUIT     = 'X'         I\_ENAME     = SY-UNAME       TABLES         T\_LTAP\_CONF = LT\_LTAP\_CONF[].     IF SY-SUBRC EQ 0.       CALL FUNCTION 'BAPI\_TRANSACTION\_COMMIT'         EXPORTING           WAIT = 'X'.     ELSE.       *"失败则回滚数据*       CALL FUNCTION 'BAPI\_TRANSACTION\_ROLLBACK'.     ENDIF.   ENDLOOP.    *"按DN检查是否已经有第二步TO，如果无则创建*   IF LINES( IT\_DN\_LIST ) > 0.     SELECT \* INTO CORRESPONDING FIELDS OF TABLE  IT\_LTAK\_2ND       FROM LTAK       FOR ALL ENTRIES IN IT\_DN\_LIST       WHERE LGNUM = LV\_LGNUM AND BETYP = 'L'       AND VBELN = IT\_DN\_LIST-DN\_NUM.      *"如果第二步TO不存在，则创建 LT42*     IF LINES( IT\_LTAK\_2ND ) = 0.       *"按理只有一个司机-GROUP*       LOOP AT IT\_VBSS.         *"查找当前司机GROUP下包括的DN订单*         CLEAR IT\_GROUP\_DN. REFRESH IT\_GROUP\_DN.         SELECT \* INTO CORRESPONDING FIELDS OF TABLE IT\_GROUP\_DN           FROM VBSS WHERE SAMMG = IT\_VBSS-SAMMG.          *"按DN订单逐一创建第二步TO*         LOOP AT IT\_GROUP\_DN.           CLEAR:GT\_BDCDATA,GT\_BDCMSGCOLL,GT\_MSG.           REFRESH:GT\_BDCDATA,GT\_BDCMSGCOLL,GT\_MSG.           CLEAR LV\_TANUM.           *"按DN查找是否已经有第二步TO，如果无则创建*           SELECT SINGLE TANUM INTO LV\_TANUM FROM LTAK             WHERE BWART = '601'              AND BWLVS = '601'              AND LGNUM = LV\_LGNUM            AND REFNR = IT\_VBSS-SAMMG              AND VBELN = IT\_GROUP\_DN-VBELN.           IF LV\_TANUM IS INITIAL.             PERFORM BDC\_DYNPRO    USING 'SAPML05S' '0110'.             PERFORM BDC\_FIELD     USING 'BDC\_CURSOR' 'RL05S-REFNR'.             PERFORM BDC\_FIELD     USING 'BDC\_OKCODE' 'STRT'.             PERFORM BDC\_FIELD     USING 'RL05S-LGNUM' LV\_LGNUM.             PERFORM BDC\_FIELD     USING 'RL05S-REFNR' IT\_VBSS-SAMMG.             PERFORM BDC\_FIELD     USING 'RL05S-DUNKL' 'D'.              PERFORM BDC\_DYNPRO    USING 'SAPML03T' '0152'.             PERFORM BDC\_FIELD     USING 'BDC\_OKCODE' '=BU'.             PERFORM BDC\_FIELD     USING 'BDC\_CURSOR' 'LVBLK-L2SKR(01)'.             *"创建第二步TO*             REFRESH GT\_BDCMSGCOLL.             CALL TRANSACTION 'LT42' USING GT\_BDCDATA                   MODE  'N'                   *"UPDATE 'S'*                   MESSAGES INTO GT\_BDCMSGCOLL.             READ TABLE  GT\_BDCMSGCOLL WITH KEY MSGTYP = 'S'.             IF SY-SUBRC = 0.               *"WAIT UP TO 1 SECONDS.*             ENDIF.           ENDIF.         ENDLOOP.       ENDLOOP.        *"创建第二步TO后再查询*       SELECT \* INTO CORRESPONDING FIELDS OF TABLE  IT\_LTAK\_2ND         FROM LTAK         FOR ALL ENTRIES IN IT\_DN\_LIST         WHERE LGNUM = LV\_LGNUM AND BETYP = 'L'         AND VBELN = IT\_DN\_LIST-DN\_NUM.     ENDIF.   ENDIF.    LOOP AT IT\_DN\_LIST ASSIGNING <IT\_DN\_LIST> WHERE TO2\_CONFIRM = ''.     READ TABLE IT\_LTAK\_2ND WITH  KEY VBELN = <IT\_DN\_LIST>-DN\_NUM.     IF SY-SUBRC <> 0.       <IT\_DN\_LIST>-TO2\_CONFIRM = 'E'.       <IT\_DN\_LIST>-MESSAGE = '2nd TO Not Found!'.     ENDIF.   ENDLOOP.    LOOP AT IT\_LTAK\_2ND WHERE KQUIT = ''.     CLEAR LT\_LTAP\_CONF. REFRESH LT\_LTAP\_CONF.     CALL FUNCTION 'L\_TO\_CONFIRM'       EXPORTING         I\_LGNUM     = IT\_LTAK\_2ND-LGNUM  *"仓库号*         I\_TANUM     = IT\_LTAK\_2ND-TANUM  *"TO号码*         I\_SQUIT     = 'X'         I\_ENAME     = SY-UNAME       TABLES         T\_LTAP\_CONF = LT\_LTAP\_CONF[].     IF SY-SUBRC EQ 0.       CALL FUNCTION 'BAPI\_TRANSACTION\_COMMIT'         EXPORTING           WAIT = 'X'.     ELSE.       *"失败则回滚数据*       CALL FUNCTION 'BAPI\_TRANSACTION\_ROLLBACK'.     ENDIF.   ENDLOOP.    WAIT UP TO 1 SECONDS.    *"如果执行完BAPI后查询仍有未确认的TO就提示接口端*   DATA:LV\_TO\_NUM LIKE LTAK-TANUM.   DATA:LV\_TO\_CFM LIKE LTAK-KQUIT.   LOOP AT IT\_DN\_LIST ASSIGNING <IT\_DN\_LIST> WHERE TO2\_CONFIRM = ''.     CLEAR LV\_TO\_NUM.CLEAR LV\_TO\_CFM.     SELECT SINGLE TANUM KQUIT INTO (LV\_TO\_NUM,LV\_TO\_CFM )       FROM LTAK WHERE LGNUM = LV\_LGNUM AND BETYP = 'L' AND BWART = '601'       AND VBELN = <IT\_DN\_LIST>-DN\_NUM.     IF LV\_TO\_NUM IS NOT INITIAL.       IF LV\_TO\_CFM = 'X'.         <IT\_DN\_LIST>-TO2\_CONFIRM = 'S'.         <IT\_DN\_LIST>-MESSAGE = '2nd TOs be Confirmed'.       ELSE.         <IT\_DN\_LIST>-TO2\_CONFIRM = 'E'.         <IT\_DN\_LIST>-MESSAGE = '2nd TOs Not be Confirmed'.       ENDIF.     ELSE.       <IT\_DN\_LIST>-TO2\_CONFIRM = 'E'.       <IT\_DN\_LIST>-MESSAGE = '2nd TOs Not Found'.     ENDIF.   ENDLOOP.    O\_STATUS = 'S'.   O\_MESSAGE = 'Option Finished'. ENDFUNCTION.  *\*&---------------------------------------------------------------------\** *\*&      FORM  BDC\_DYNPRO* *\*&---------------------------------------------------------------------\** FORM BDC\_DYNPRO  USING PROG SCR.   CLEAR GT\_BDCDATA.   GT\_BDCDATA-PROGRAM = PROG.   GT\_BDCDATA-DYNPRO  = SCR.   GT\_BDCDATA-DYNBEGIN = 'X'.   APPEND GT\_BDCDATA. ENDFORM.                    *" BDC\_DYNPRO*  *\*&---------------------------------------------------------------------\** *\*&      FORM  BDC\_FIELD* *\*&---------------------------------------------------------------------\** FORM BDC\_FIELD  USING    FNAM FVAL.   CLEAR GT\_BDCDATA.   GT\_BDCDATA-FNAM = FNAM.   GT\_BDCDATA-FVAL  = FVAL.   APPEND GT\_BDCDATA. ENDFORM. |

ZRFWM050 BarCode:Delivery Order Catch Weight

|  |
| --- |
|  |
| FUNCTION ZRFWM050. *\*"----------------------------------------------------------------------* *\*"\*"Local Interface:* *\*"  IMPORTING* *\*"     VALUE(I\_DNNUM) LIKE  LIKP-VBELN OPTIONAL* *\*"  EXPORTING* *\*"     VALUE(O\_STATUS) TYPE  CHAR1* *\*"     VALUE(O\_MESSAGE) TYPE  CHAR100* *\*"  TABLES* *\*"      IT\_DN\_CATCH STRUCTURE  ZSWM036 OPTIONAL* *\*"----------------------------------------------------------------------*    IF I\_DNNUM = ''.     O\_STATUS = 'E'.     O\_MESSAGE = 'DN is Required!'.     RETURN.   ENDIF.    IF LINES( IT\_DN\_CATCH ) = 0.     O\_STATUS = 'E'.     O\_MESSAGE = 'DN Catch Weight is Required!'.     RETURN.   ENDIF.    *"修改DN必要参数*   DATA:LS\_HEADER\_DATA    LIKE BAPIOBDLVHDRCHG,                *"更改外向交货拣配数据表头等级*        LS\_HEADER\_CONTROL LIKE BAPIOBDLVHDRCTRLCHG,            *"外向交货标题级别控制数据*        LS\_TECH\_CONTROL   TYPE BAPIDLVCONTROL,        LV\_DELIVERY       LIKE BAPIOBDLVHDRCHG-DELIV\_NUMB,     *"交货单号*        LT\_ITEM\_DATA      LIKE TABLE OF BAPIOBDLVITEMCHG WITH HEADER LINE,      *"更改外向交货拣配数据项目等级*        LS\_ITEM\_DATA      LIKE LINE OF  LT\_ITEM\_DATA,        LT\_ITEM\_CONTROL   LIKE TABLE OF BAPIOBDLVITEMCTRLCHG WITH HEADER LINE,  *"外向交货项目级别控制数据*        LS\_ITEM\_CONTROL   LIKE LINE OF LT\_ITEM\_CONTROL,        LT\_RETURN         LIKE TABLE OF BAPIRET2   WITH HEADER LINE, *"返回参数*        LS\_RETURN         LIKE LINE OF LT\_RETURN.    *"临时处理内表*   DATA:IT\_TEMP LIKE IT\_DN\_CATCH OCCURS 0 WITH HEADER LINE.   DATA:ZTWM13\_HDR LIKE ZTWM013\_HDR OCCURS 0 WITH HEADER LINE.   DATA:ZTWM13\_DTL LIKE ZTWM013\_DTL OCCURS 0 WITH HEADER LINE.    TYPES: BEGIN OF TY\_LINES\_132,            LINE TYPE TDLINE,          END OF TY\_LINES\_132.    *"货仓编码*   DATA:LV\_LGNUM LIKE LIPS-LGNUM.   *"销售订单号*   DATA:LV\_SALES\_ORDER LIKE LIPS-VGBEL.   *"DN GROUD*   DATA:LV\_DN\_GROUP LIKE VBSS-SAMMG.    *"转换比例*   DATA:LV\_X\_NUM LIKE LIPS-UMVKN VALUE 100. *"抄磅数据放大1000倍*   DATA:LV\_DLV\_MAT LIKE LIPS-MATNR.          *"交货物料*   DATA:LV\_DLV\_QTY LIKE LIPS-LFIMG.          *"交货数量*   DATA:LV\_DLV\_QTY\_FLO LIKE LIPS-LFIMG.      *"交货数量*   DATA:LV\_DLV\_UNT LIKE LIPS-VRKME.          *"交货数量-单位*   *"抄磅总磅数*   DATA:LV\_TOTAL\_WGT LIKE LIPS-LGMNG VALUE 0.   *"Text抄磅信息*   DATA:LT\_LINES\_132    TYPE STANDARD TABLE OF TY\_LINES\_132,        LS\_LINES\_132    LIKE LINE OF LT\_LINES\_132,        LS\_HEAD         TYPE THEAD,        LT\_TLINE        TYPE STANDARD TABLE OF TLINE,        LS\_TLINE        LIKE LINE OF LT\_TLINE,        LV\_STRING(2000) TYPE C,        LV\_WEIGHT       TYPE CHAR13,        LV\_WEIGHT\_N     TYPE GSGEW.   *"----------------------------------------------------------------------*   FIELD-SYMBOLS: <IT\_DN\_CATCH> LIKE IT\_DN\_CATCH.   *"----------------------------------------------------------------------*   *"处理单号*   CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'     EXPORTING       INPUT  = I\_DNNUM     IMPORTING       OUTPUT = I\_DNNUM.    *"处理查询数据*   SORT IT\_DN\_CATCH BY VBELN POSNR ASCENDING.   LOOP AT IT\_DN\_CATCH ASSIGNING <IT\_DN\_CATCH>.     CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'       EXPORTING         INPUT  = <IT\_DN\_CATCH>-VBELN       IMPORTING         OUTPUT = <IT\_DN\_CATCH>-VBELN.      CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'       EXPORTING         INPUT  = <IT\_DN\_CATCH>-POSNR       IMPORTING         OUTPUT = <IT\_DN\_CATCH>-POSNR.   ENDLOOP.   *"确保一次只处理一张DN订单数据*   DELETE IT\_DN\_CATCH WHERE VBELN <> I\_DNNUM.    *"提前准备数据:货仓编码、DN-Group、销售订单号*   SELECT SINGLE LGNUM VGBEL INTO ( LV\_LGNUM,LV\_SALES\_ORDER ) FROM LIPS WHERE VBELN = I\_DNNUM.   SELECT SINGLE SAMMG INTO LV\_DN\_GROUP FROM VBSS WHERE VBELN = I\_DNNUM.    *"处理查询数据*   SORT IT\_DN\_CATCH BY VBELN POSNR CATCH ASCENDING.   IT\_TEMP[] = IT\_DN\_CATCH[].   DELETE ADJACENT DUPLICATES FROM IT\_TEMP COMPARING VBELN POSNR.    *"S:登记数据入两步执自定义表：ZTWM013\_HDR & ZTWM013\_DTL\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**   CLEAR ZTWM13\_HDR.REFRESH ZTWM13\_HDR.   ZTWM13\_HDR-REFNR = LV\_DN\_GROUP.   ZTWM13\_HDR-VBELN = I\_DNNUM.   ZTWM13\_HDR-LGNUM = LV\_LGNUM.   ZTWM13\_HDR-SALES\_ORDER = LV\_SALES\_ORDER.   ZTWM13\_HDR-ERNAM = SY-UNAME.   ZTWM13\_HDR-ERZET = SY-UZEIT.   ZTWM13\_HDR-ERDAT = SY-DATUM.   APPEND ZTWM13\_HDR.   IF LINES( ZTWM13\_HDR ) > 0.     MODIFY ZTWM013\_HDR FROM @ZTWM13\_HDR.   ENDIF.    LOOP AT IT\_TEMP.     CLEAR ZTWM13\_DTL.REFRESH ZTWM13\_DTL.     CLEAR LV\_DLV\_QTY.CLEAR LV\_DLV\_QTY\_FLO.CLEAR LV\_DLV\_UNT.     SELECT SINGLE MATNR LFIMG LFIMG\_FLO VRKME INTO ( LV\_DLV\_MAT,LV\_DLV\_QTY,LV\_DLV\_QTY\_FLO,LV\_DLV\_UNT ) FROM LIPS       WHERE VBELN = IT\_TEMP-VBELN AND POSNR = IT\_TEMP-POSNR.      ZTWM13\_DTL-LGNUM = LV\_LGNUM.     CLEAR ZTWM13\_DTL-TANUM.     CLEAR ZTWM13\_DTL-TAPOS.     SELECT SINGLE TANUM TAPOS INTO ( ZTWM13\_DTL-TANUM, ZTWM13\_DTL-TAPOS )       FROM LTAP WHERE LGNUM = LV\_LGNUM       AND VLPLA = LV\_DN\_GROUP       AND MATNR = LV\_DLV\_MAT.     ZTWM13\_DTL-VBELN = IT\_TEMP-VBELN.     ZTWM13\_DTL-POSNR = IT\_TEMP-POSNR.     ZTWM13\_DTL-MATNR = LV\_DLV\_MAT.     ZTWM13\_DTL-EXTWG = 'CW'.     ZTWM13\_DTL-UMVKN = LV\_X\_NUM.     *"计算总磅数*     CLEAR LV\_TOTAL\_WGT.     LOOP AT IT\_DN\_CATCH WHERE VBELN = IT\_TEMP-VBELN AND POSNR = IT\_TEMP-POSNR.       LV\_TOTAL\_WGT = LV\_TOTAL\_WGT + IT\_DN\_CATCH-CATCH.     ENDLOOP.     ZTWM13\_DTL-UMVKZ = LV\_TOTAL\_WGT \* LV\_X\_NUM.     ZTWM13\_DTL-NTGEW = LV\_TOTAL\_WGT / LV\_DLV\_QTY.     ZTWM13\_DTL-QUANTITY = LV\_DLV\_QTY.     ZTWM13\_DTL-WEIGHT = LV\_TOTAL\_WGT.      ZTWM13\_DTL-ERNAM = SY-UNAME.     ZTWM13\_DTL-ERZET = SY-UZEIT.     ZTWM13\_DTL-ERDAT = SY-DATUM.     APPEND ZTWM13\_DTL.     IF LINES( ZTWM13\_DTL ) > 0.       MODIFY ZTWM013\_DTL FROM @ZTWM13\_DTL.     ENDIF.      *"保存DN单上ITEM NOTES 抄磅信息*     LS\_HEAD-TDID      = '0002'.     LS\_HEAD-TDSPRAS   = SY-LANGU.     LS\_HEAD-TDOBJECT  = 'VBBP'.     CONCATENATE IT\_TEMP-VBELN IT\_TEMP-POSNR INTO LS\_HEAD-TDNAME.      LV\_STRING = 'Catch Wgt:'.     LOOP AT IT\_DN\_CATCH WHERE VBELN = IT\_TEMP-VBELN AND POSNR = IT\_TEMP-POSNR.       CLEAR LV\_WEIGHT.       TRY.           LV\_WEIGHT = IT\_DN\_CATCH-CATCH.         CATCH CX\_SY\_CONVERSION\_ERROR.           *"O\_MESSAGE = E011(ZRF).*       ENDTRY.       CONCATENATE LV\_STRING LV\_WEIGHT INTO LV\_STRING SEPARATED BY SPACE.       CONDENSE LV\_STRING.     ENDLOOP.      CLEAR  LT\_LINES\_132.     REFRESH  LT\_LINES\_132.     CALL FUNCTION 'RKD\_WORD\_WRAP'       EXPORTING         TEXTLINE            = LV\_STRING         DELIMITER           = SPACE         OUTPUTLEN           = '132'       TABLES         OUT\_LINES           = LT\_LINES\_132       EXCEPTIONS         OUTPUTLEN\_TOO\_LARGE = 1         OTHERS              = 2.      CLEAR LT\_TLINE.     REFRESH LT\_TLINE.     LOOP AT LT\_LINES\_132 INTO LS\_LINES\_132.       IF SY-TABIX EQ 1.         LS\_TLINE-TDFORMAT = '\*'.       ENDIF.       LS\_TLINE-TDLINE   = LS\_LINES\_132-LINE.       APPEND LS\_TLINE TO LT\_TLINE.       CLEAR LS\_TLINE.     ENDLOOP.      IF LT\_TLINE[] IS NOT INITIAL.       CALL FUNCTION 'SAVE\_TEXT'         EXPORTING           CLIENT          = SY-MANDT           HEADER          = LS\_HEAD           SAVEMODE\_DIRECT = 'X '         TABLES           LINES           = LT\_TLINE         EXCEPTIONS           ID              = 1           LANGUAGE        = 2           NAME            = 3           OBJECT          = 4           OTHERS          = 5.     ENDIF.   ENDLOOP.   *"E:登记数据入两步执自定义表：ZTWM013\_HDR & ZTWM013\_DTL\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**     *"头信息表信息交货单号*   LV\_DELIVERY = I\_DNNUM.  *"交货单号*   LS\_HEADER\_DATA-DELIV\_NUMB     = I\_DNNUM. *"交货单号*   LS\_TECH\_CONTROL-UPD\_IND       = 'U'.   LS\_TECH\_CONTROL-RECV\_WHS\_NO   = LV\_LGNUM.   LS\_HEADER\_CONTROL-DELIV\_NUMB  = I\_DNNUM. *"交货单号*    *"抄磅信息*   REFRESH: LT\_ITEM\_DATA, LT\_ITEM\_CONTROL.   LOOP AT IT\_TEMP WHERE VBELN = I\_DNNUM.     *"抄磅信息*     CLEAR: LT\_ITEM\_DATA, LT\_ITEM\_CONTROL.     LS\_ITEM\_DATA-DELIV\_NUMB         = IT\_TEMP-VBELN.     LS\_ITEM\_DATA-DELIV\_ITEM         = IT\_TEMP-POSNR.      CLEAR LV\_DLV\_QTY.CLEAR LV\_DLV\_QTY\_FLO.CLEAR LV\_DLV\_UNT.     SELECT SINGLE MATNR LFIMG LFIMG\_FLO VRKME INTO ( LV\_DLV\_MAT,LV\_DLV\_QTY,LV\_DLV\_QTY\_FLO,LV\_DLV\_UNT )       FROM LIPS       WHERE VBELN = IT\_TEMP-VBELN AND POSNR = IT\_TEMP-POSNR.     LS\_ITEM\_DATA-MATERIAL           = LV\_DLV\_MAT.      *"计算总磅数*     CLEAR LV\_TOTAL\_WGT.     LOOP AT IT\_DN\_CATCH WHERE VBELN = IT\_TEMP-VBELN AND POSNR = IT\_TEMP-POSNR.       LV\_TOTAL\_WGT = LV\_TOTAL\_WGT + IT\_DN\_CATCH-CATCH.     ENDLOOP.      LS\_ITEM\_DATA-DLV\_QTY            = LV\_DLV\_QTY.     LS\_ITEM\_DATA-FACT\_UNIT\_NOM      = LV\_TOTAL\_WGT / LV\_DLV\_QTY  \* LV\_X\_NUM .     LS\_ITEM\_DATA-FACT\_UNIT\_DENOM    = LV\_X\_NUM.     LS\_ITEM\_DATA-DLV\_QTY\_IMUNIT     = LV\_DLV\_QTY \* LS\_ITEM\_DATA-FACT\_UNIT\_NOM  / LS\_ITEM\_DATA-FACT\_UNIT\_DENOM.     LS\_ITEM\_DATA-CONV\_FACT          = LS\_ITEM\_DATA-FACT\_UNIT\_NOM  / LS\_ITEM\_DATA-FACT\_UNIT\_DENOM.     LS\_ITEM\_DATA-SALES\_UNIT         = LV\_DLV\_UNT.     SELECT SINGLE ISOCODE INTO LS\_ITEM\_DATA-SALES\_UNIT\_ISO FROM T006 WHERE MSEHI = LS\_ITEM\_DATA-SALES\_UNIT.     LS\_ITEM\_DATA-BASE\_UOM           = 'LB'.     SELECT SINGLE ISOCODE INTO LS\_ITEM\_DATA-BASE\_UOM\_ISO FROM T006 WHERE MSEHI = LS\_ITEM\_DATA-BASE\_UOM.     LS\_ITEM\_DATA-DEL\_QTY\_FLO        = LV\_DLV\_QTY\_FLO.     *"LS\_ITEM\_DATA-DLV\_QTY\_ST\_FLO     = LV\_DLV\_QTY\_FLO.*     APPEND LS\_ITEM\_DATA TO LT\_ITEM\_DATA.      *"控制参数*     LS\_ITEM\_CONTROL-DELIV\_NUMB = IT\_TEMP-VBELN.     LS\_ITEM\_CONTROL-DELIV\_ITEM = IT\_TEMP-POSNR.     LS\_ITEM\_CONTROL-CHG\_DELQTY = ABAP\_TRUE.     APPEND LS\_ITEM\_CONTROL TO LT\_ITEM\_CONTROL.   ENDLOOP.    REFRESH: LT\_RETURN.   SET UPDATE TASK LOCAL.    *"修改DN订单*   CALL FUNCTION 'BAPI\_OUTB\_DELIVERY\_CHANGE'     EXPORTING       HEADER\_DATA    = LS\_HEADER\_DATA       HEADER\_CONTROL = LS\_HEADER\_CONTROL       DELIVERY       = LV\_DELIVERY       TECHN\_CONTROL  = LS\_TECH\_CONTROL     TABLES       ITEM\_DATA      = LT\_ITEM\_DATA       ITEM\_CONTROL   = LT\_ITEM\_CONTROL       RETURN         = LT\_RETURN.   READ TABLE LT\_RETURN WITH KEY TYPE = 'E' TRANSPORTING NO FIELDS.   IF SY-SUBRC NE 0.     COMMIT WORK AND WAIT.     *"Commit changes*     CALL FUNCTION 'BAPI\_TRANSACTION\_COMMIT'       EXPORTING         WAIT = 'X'.     O\_STATUS = 'S'.     O\_MESSAGE = 'Successfully'.   ELSE.     CALL FUNCTION 'BAPI\_TRANSACTION\_ROLLBACK'.      CLEAR O\_MESSAGE.     LOOP AT LT\_RETURN INTO LS\_RETURN WHERE TYPE = 'E'.       MESSAGE ID LS\_RETURN-ID             TYPE LS\_RETURN-TYPE           NUMBER LS\_RETURN-NUMBER             WITH LS\_RETURN-MESSAGE\_V1                  LS\_RETURN-MESSAGE\_V2                  LS\_RETURN-MESSAGE\_V3                  LS\_RETURN-MESSAGE\_V4             INTO LS\_RETURN-MESSAGE.       CONCATENATE LS\_RETURN-MESSAGE O\_MESSAGE INTO O\_MESSAGE.     ENDLOOP.     IF O\_MESSAGE IS INITIAL.       O\_MESSAGE = TEXT-E11.     ENDIF.     O\_STATUS = 'E'.   ENDIF.  ENDFUNCTION. |

ZRFWM051 BarCode:Delivery Order Post Goods Issue

|  |
| --- |
|  |
| FUNCTION ZRFWM051. *\*"----------------------------------------------------------------------* *\*"\*"Local Interface:* *\*"  EXPORTING* *\*"     VALUE(O\_STATUS) TYPE  CHAR1* *\*"     VALUE(O\_MESSAGE) TYPE  CHAR100* *\*"  TABLES* *\*"      IT\_DN\_LIST STRUCTURE  ZSWM038 OPTIONAL* *\*"----------------------------------------------------------------------*   *"检查数据*   IF LINES( IT\_DN\_LIST ) = 0.     O\_STATUS = 'E'.     O\_MESSAGE = 'DN Order is Required!'.     RETURN.   ENDIF.    *"判断DN是否已经PGI*   DATA:LV\_DNPGI LIKE VBUK-WBSTK.   *"定义BAPI使用到的变量*   DATA:LS\_VBKOK     TYPE VBKOK,        LT\_PROT      TYPE STANDARD TABLE OF PROTT,        LS\_PROT      TYPE PROTT,        LV\_ANY\_ERROR TYPE XFELD,        LV\_GI\_ERROR  TYPE XFELD.    *"----------------------------------------------------------------------*   FIELD-SYMBOLS: <IT\_DN\_LIST> LIKE IT\_DN\_LIST.   *"----------------------------------------------------------------------*    *"处理查询数据*   SORT IT\_DN\_LIST BY DN\_NUM ASCENDING.   DELETE ADJACENT DUPLICATES FROM IT\_DN\_LIST COMPARING DN\_NUM.   LOOP AT IT\_DN\_LIST ASSIGNING <IT\_DN\_LIST>.     *"补0*     CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'       EXPORTING         INPUT  = <IT\_DN\_LIST>-DN\_NUM       IMPORTING         OUTPUT = <IT\_DN\_LIST>-DN\_NUM.      *"默认所有DN都能够被PGI*     <IT\_DN\_LIST>-DN\_PGI = 'S'.     <IT\_DN\_LIST>-MESSAGE = 'Post Goods Issue Successfully!'.   ENDLOOP.   *"处理查询数据*   SORT IT\_DN\_LIST BY DN\_NUM ASCENDING.   DELETE ADJACENT DUPLICATES FROM IT\_DN\_LIST COMPARING DN\_NUM.    *"PGI处理数据*   LOOP AT IT\_DN\_LIST ASSIGNING <IT\_DN\_LIST>.      *"如果已经PGI的话不用使用BAPI操作直接登记成功*     CLEAR LV\_DNPGI.     SELECT SINGLE WBSTK INTO LV\_DNPGI FROM VBUK WHERE VBELN = <IT\_DN\_LIST>-DN\_NUM.     IF LV\_DNPGI = 'C'.       <IT\_DN\_LIST>-DN\_PGI = 'S'.       <IT\_DN\_LIST>-MESSAGE = 'Successfully'.     ELSE.       CLEAR LS\_VBKOK.       LS\_VBKOK-VBELN\_VL = <IT\_DN\_LIST>-DN\_NUM.       LS\_VBKOK-WABUC    = ABAP\_TRUE.       REFRESH: LT\_PROT.       CLEAR: LV\_ANY\_ERROR, LV\_GI\_ERROR.       *"PGI*       CALL FUNCTION 'WS\_DELIVERY\_UPDATE'         EXPORTING           VBKOK\_WA                  = LS\_VBKOK           COMMIT                    = 'X'           DELIVERY                  = <IT\_DN\_LIST>-DN\_NUM           UPDATE\_PICKING            = 'X'           NICHT\_SPERREN             = 'X'           IF\_DATABASE\_UPDATE        = '1'         IMPORTING           EF\_ERROR\_ANY\_0            = LV\_ANY\_ERROR           EF\_ERROR\_IN\_GOODS\_ISSUE\_0 = LV\_GI\_ERROR         TABLES           PROT                      = LT\_PROT.       IF LV\_ANY\_ERROR IS INITIAL.         *"Commit work*         CALL FUNCTION 'BAPI\_TRANSACTION\_COMMIT'           EXPORTING             WAIT = 'X'.         <IT\_DN\_LIST>-DN\_PGI = 'S'.         <IT\_DN\_LIST>-MESSAGE = 'Successfully'.          *"Update PGI date*         UPDATE ZTWM013\_HDR SET WADAT\_IST      = SY-DATUM                                STATUS         = 'P'                                STATUS\_TEXT    = 'PGI'                            WHERE VBELN = <IT\_DN\_LIST>-DN\_NUM.         *"Update PGI date*         UPDATE ZTWM013\_DTL SET WADAT\_IST      = SY-DATUM                                STATUS         = 'P'                                STATUS\_TEXT    = 'PGI'                            WHERE VBELN = <IT\_DN\_LIST>-DN\_NUM.       ELSE.         *"失败则回滚数据*         CALL FUNCTION 'BAPI\_TRANSACTION\_ROLLBACK'.         <IT\_DN\_LIST>-DN\_PGI = 'E'.         *"读取错误信息*         READ TABLE LT\_PROT INTO LS\_PROT WITH KEY POSNR = '000000'.         IF SY-SUBRC EQ 0.           CONDENSE LS\_PROT-MSGV1 NO-GAPS.           CONDENSE LS\_PROT-MSGV2 NO-GAPS.           CONDENSE LS\_PROT-MSGV3 NO-GAPS.           CONDENSE LS\_PROT-MSGV4 NO-GAPS.            CLEAR <IT\_DN\_LIST>-MESSAGE.           MESSAGE ID LS\_PROT-MSGID TYPE LS\_PROT-MSGTY NUMBER LS\_PROT-MSGNO             WITH LS\_PROT-MSGV1 LS\_PROT-MSGV2 LS\_PROT-MSGV3 LS\_PROT-MSGV4             INTO <IT\_DN\_LIST>-MESSAGE.         ELSE.            LOOP AT LT\_PROT INTO LS\_PROT WHERE MSGTY = 'E'.             MESSAGE ID LS\_PROT-MSGID TYPE LS\_PROT-MSGTY NUMBER LS\_PROT-MSGNO               WITH LS\_PROT-MSGV1 LS\_PROT-MSGV2 LS\_PROT-MSGV3 LS\_PROT-MSGV4               INTO <IT\_DN\_LIST>-MESSAGE.           ENDLOOP.            IF <IT\_DN\_LIST>-MESSAGE IS INITIAL.             <IT\_DN\_LIST>-MESSAGE = TEXT-E10.           ENDIF.         ENDIF.       ENDIF.     ENDIF.   ENDLOOP.    *"读取是否有错误*   READ TABLE IT\_DN\_LIST WITH KEY DN\_PGI = 'E'.   IF SY-SUBRC = 0.     O\_STATUS = 'E'.     O\_MESSAGE = 'Exist Error Record'.   ELSE.     O\_STATUS = 'S'.     O\_MESSAGE = 'All Successfully'.   ENDIF.  ENDFUNCTION. |

ZRFWM052 BarCode:Delivery Order Created Invoice

|  |
| --- |
|  |
| FUNCTION ZRFWM052. *\*"----------------------------------------------------------------------* *\*"\*"Local Interface:* *\*"  EXPORTING* *\*"     VALUE(O\_STATUS) TYPE  CHAR1* *\*"     VALUE(O\_MESSAGE) TYPE  CHAR100* *\*"  TABLES* *\*"      IT\_DN\_LIST STRUCTURE  ZSWM039 OPTIONAL* *\*"----------------------------------------------------------------------*   *"检查数据*   IF LINES( IT\_DN\_LIST ) = 0.     O\_STATUS = 'E'.     O\_MESSAGE = 'DN is Required!'.     RETURN.   ENDIF.    *"判断DN是否已经开发票*   DATA:LV\_DNINV LIKE VBUK-FKSTK.    DATA:LS\_DATAIN     TYPE BAPICREATORDATA.   DATA:LT\_BILLINGDATAIN TYPE TABLE OF BAPIVBRK,        LS\_BILLINGDATAIN TYPE BAPIVBRK,        LT\_SUCCESS       TYPE TABLE OF BAPIVBRKSUCCESS,        LS\_SUCCESS       TYPE BAPIVBRKSUCCESS,        LT\_RETURN        TYPE TABLE OF BAPIRET1,        LS\_RETURN        TYPE BAPIRET1.    *"----------------------------------------------------------------------*   FIELD-SYMBOLS: <IT\_DN\_LIST> LIKE IT\_DN\_LIST.   *"----------------------------------------------------------------------*    *"处理查询数据*   SORT IT\_DN\_LIST BY DN\_NUM ASCENDING.   DELETE ADJACENT DUPLICATES FROM IT\_DN\_LIST COMPARING DN\_NUM.   LOOP AT IT\_DN\_LIST ASSIGNING <IT\_DN\_LIST>.     *"补0*     CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'       EXPORTING         INPUT  = <IT\_DN\_LIST>-DN\_NUM       IMPORTING         OUTPUT = <IT\_DN\_LIST>-DN\_NUM.      *"默认所有DN都能够开具发票*     <IT\_DN\_LIST>-INVOICE = ''.     <IT\_DN\_LIST>-DN\_INV = 'S'.     <IT\_DN\_LIST>-MESSAGE = 'Invoice Successfully!'.   ENDLOOP.   *"处理查询数据*   SORT IT\_DN\_LIST BY DN\_NUM ASCENDING.   DELETE ADJACENT DUPLICATES FROM IT\_DN\_LIST COMPARING DN\_NUM.    *"处理数据*   LOOP AT IT\_DN\_LIST ASSIGNING <IT\_DN\_LIST>.      *"如果已经PGI的话不用使用BAPI操作直接登记成功*     CLEAR LV\_DNINV.     SELECT SINGLE FKSTK INTO LV\_DNINV FROM VBUK WHERE VBELN = <IT\_DN\_LIST>-DN\_NUM.     IF LV\_DNINV = 'C'.       SELECT SINGLE VBELN INTO <IT\_DN\_LIST>-INVOICE FROM VBFA         WHERE VBTYP\_N = 'M' AND VBELV = <IT\_DN\_LIST>-DN\_NUM.       <IT\_DN\_LIST>-DN\_INV = 'S'.       <IT\_DN\_LIST>-MESSAGE = 'Successfully'.     ELSE.        CLEAR LS\_DATAIN.       LS\_DATAIN-CREATED\_BY = SY-DATUM.       LS\_DATAIN-CREATED\_ON = SY-DATUM.        CLEAR LS\_BILLINGDATAIN.       LS\_BILLINGDATAIN-REF\_DOC    = <IT\_DN\_LIST>-DN\_NUM.       LS\_BILLINGDATAIN-REF\_DOC\_CA = 'J'.       LS\_BILLINGDATAIN-BILL\_DATE  = SY-DATUM.       APPEND LS\_BILLINGDATAIN TO LT\_BILLINGDATAIN.        SET UPDATE TASK LOCAL.        *"执行BAPI*       CALL FUNCTION 'BAPI\_BILLINGDOC\_CREATEMULTIPLE'         EXPORTING           CREATORDATAIN = LS\_DATAIN           TESTRUN       = ''         TABLES           BILLINGDATAIN = LT\_BILLINGDATAIN           RETURN        = LT\_RETURN           SUCCESS       = LT\_SUCCESS.       IF LT\_SUCCESS IS NOT INITIAL.         CALL FUNCTION 'BAPI\_TRANSACTION\_COMMIT'           EXPORTING             WAIT = 'X'.          READ TABLE LT\_SUCCESS INTO LS\_SUCCESS INDEX 1.         IF SY-SUBRC EQ 0.           <IT\_DN\_LIST>-INVOICE = LS\_SUCCESS-BILL\_DOC.         ENDIF.         <IT\_DN\_LIST>-DN\_INV = 'S'.         <IT\_DN\_LIST>-MESSAGE = 'Successfully'.          *"更新两步执出货自定义表*         READ TABLE LT\_SUCCESS INTO LS\_SUCCESS INDEX 1.         IF SY-SUBRC EQ 0.           UPDATE ZTWM013\_HDR SET BILL\_DOC    = <IT\_DN\_LIST>-INVOICE                                  STATUS\_TEXT = ''                                  FKDAT       = SY-DATUM                                  STATUS      = 'C'                              WHERE VBELN     = <IT\_DN\_LIST>-DN\_NUM.            UPDATE ZTWM013\_DTL SET STATUS\_TEXT = ''                                  STATUS      = 'C'                              WHERE VBELN = <IT\_DN\_LIST>-DN\_NUM.         ENDIF.        ELSE.         CALL FUNCTION 'BAPI\_TRANSACTION\_ROLLBACK'.         *"更新两步执出货自定义表*         <IT\_DN\_LIST>-INVOICE = ''.         <IT\_DN\_LIST>-DN\_INV = 'E'.         <IT\_DN\_LIST>-MESSAGE = 'Error,请与管理员联系！'.          UPDATE ZTWM013\_HDR SET STATUS      = 'E'                                STATUS\_TEXT = <IT\_DN\_LIST>-MESSAGE                                ERNAM       = SY-UNAME                                ERZET       = SY-UZEIT                                ERDAT       = SY-DATUM                    WHERE VBELN = <IT\_DN\_LIST>-DN\_NUM.       ENDIF.     ENDIF.   ENDLOOP.     *"读取是否有错误*   READ TABLE IT\_DN\_LIST WITH KEY DN\_INV = 'E'.   IF SY-SUBRC = 0.     O\_STATUS = 'E'.     O\_MESSAGE = 'Exist Error Record'.   ELSE.     O\_STATUS = 'S'.     O\_MESSAGE = 'All Successfully'.   ENDIF. ENDFUNCTION. |