说明：

|  |
| --- |
| 使用Dialog 集成：ZFR(DR-借项凭证)销售订单VA01和开具发票VF01 BAPI批量，用于节省会计零散操作。 |

界面：

|  |
| --- |
| 主操作界面    维护价格界面 |

数据表

|  |
| --- |
| ZTFI040 |

代码实现：ZRPFI062

|  |
| --- |
| *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** *\*& Report  ZRPFI062* *\*& 描述:本程序主要是为了创建ZFR SO 订单和一键化操作产生INV* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** *\*&   创建人:JOSEPH.L* *\*&   创建时间: 20190321* *\*&   目的:* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** *\* REPORT NAME :报表名称及格式* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** *"程序名称* REPORT ZRPFI062.  *\*线型池引用* TYPE-POOLS SLIS.  *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** *\*TABLES :声明报表中使用到的数据表* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** TABLES: VBAK,ZTFI040,BSID,KNA1.  *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** *\* DATA :定义报表中使用到的变量* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** *"主记录表结构* DATA: BEGIN OF IT\_LIST OCCURS 0,         SEL    TYPE CHAR1,         STATUS TYPE CHAR1,         LIGHT  LIKE ICONS-TEXT,         SSNUM  TYPE INT4,         AUART  LIKE VBAK-AUART,         ERDAT  LIKE VBAK-ERDAT,         KUNNR  LIKE KNA1-KUNNR,         NAME1  LIKE KNA1-NAME1,         WERKS  LIKE VBAP-WERKS,         MATNR  LIKE VBAP-MATNR,         ARKTX  LIKE VBAP-ARKTX,         NETWR  LIKE VBAP-NETWR,         VBELN  LIKE VBAP-VBELN,         BELNR  LIKE BSID-BELNR,         ZTXT1  TYPE TDLINE,         ZTXT2  TYPE TDLINE,       END OF IT\_LIST.  *"BAPI返回的消息记录表* DATA: BEGIN OF IT\_MSEG OCCURS 0,         SSNUM   TYPE INT4,         KUNNR   LIKE KNA1-KUNNR,         MATNR   LIKE VBAP-MATNR,         TYPE    LIKE BAPIRET2-TYPE,         MESSAGE LIKE BAPIRET2-MESSAGE,       END OF IT\_MSEG.  *"用于维护价格* DATA: BEGIN OF IT\_PRICE OCCURS 0,         SEL   TYPE CHAR1,         ZGUID LIKE ZTFI040-ZGUID,         MATNR LIKE ZTFI040-MATNR,         MAKTX LIKE ZTFI040-MAKTX,         PRICE LIKE ZTFI040-PRICE,         CUREC LIKE ZTFI040-CUREC,       END OF IT\_PRICE. *"用于更新维护的价格* DATA:IT\_ZFI40 LIKE ZTFI040 OCCURS 0 WITH HEADER LINE.  *"记录GUID* DATA:P\_GUID LIKE SYSUUID-X. *"確認POPU 變量* DATA: LV\_ANS TYPE C. *"定義提示信息* DATA:STR\_MESSAGE TYPE STRING VALUE ''. *"默认初始化行数* DATA:I\_INITAL TYPE INT4 VALUE 33. *"显示TableControl控件序号列* DATA:SSNUM    TYPE INT4 VALUE 0. *"读取客人信息* DATA:P\_KUNNR LIKE KNA1-KUNNR. *"读取物料信息* DATA:P\_MATNR LIKE MARA-MATNR. *"DATA CONTROL 行數* DATA: L\_ROW TYPE I. *"DATA CONTROL 字段* DATA: L\_FIELD TYPE CHAR30. *"TableControl控件：维护价格* CONTROLS:TABCONT01 TYPE TABLEVIEW USING SCREEN 1001. *"TableControl控件：主操作表* CONTROLS:TABCONT02 TYPE TABLEVIEW USING SCREEN 1002. *""TableControl控件行数* DATA:G\_TABCONT01\_LINES  LIKE SY-LOOPC. DATA:G\_TABCONT02\_LINES  LIKE SY-LOOPC.  *"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 'ZLONGTEXT',     *"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*  *"记录总数* DATA:I\_RECORD TYPE INT4. *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** *\* FIELD SYMBOLS/定义指针* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** FIELD-SYMBOLS : <IT\_LIST> LIKE IT\_LIST. FIELD-SYMBOLS : <IT\_PRICE> LIKE IT\_PRICE.  *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** *\*START-OF-SELECTION:程序运行所处理的代码* *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** START-OF-SELECTION.   *"登记操作记录*   CALL FUNCTION 'ZRFSY002'     EXPORTING       I\_UNAME = SY-UNAME       I\_TCODE = SY-TCODE       I\_PRONM = SY-REPID.   *"默认初始化程序*   PERFORM FRM\_INITAL.   *"打开默认主操作界面*   CALL SCREEN 1002.  *\*&---------------------------------------------------------------------\** *\*&      Module  STATUS\_1002  OUTPUT* *\*&---------------------------------------------------------------------\** MODULE STATUS\_1002 OUTPUT.   SET PF-STATUS 'STATUS\_1002'.   SET TITLEBAR 'TITLE'.    *"绑定控件行数*   DESCRIBE TABLE IT\_LIST LINES TABCONT02-LINES. ENDMODULE.  *\*&---------------------------------------------------------------------\** *\*&      Module  USER\_COMMAND\_1002  INPUT* *\*&---------------------------------------------------------------------\** MODULE USER\_COMMAND\_1002 INPUT.   CASE SY-UCOMM.     WHEN 'BACK' OR 'EXIT'.       LEAVE PROGRAM.     WHEN 'MAINTIAN'.       PERFORM FRM\_GET\_PRICE\_DATA.       CALL SCREEN 1001.     WHEN 'SHOWSO'.       CALL TRANSACTION 'ZSD01'.     WHEN 'SELALL'.       PERFORM FRM\_SELECT\_1002 USING 'ALL'.     WHEN 'SELNO'.       PERFORM FRM\_SELECT\_1002 USING 'NO'.     WHEN 'SELOP'.       PERFORM FRM\_SELECT\_1002 USING 'OP'.     WHEN 'CLEAR'.       PERFORM FRM\_CLARE\_DATA.     WHEN 'SALESINVOICE'.       DATA:P\_CHECK TYPE CHAR1 VALUE ''.       LOOP AT IT\_LIST WHERE SEL = 'X'.         P\_CHECK = 'X'.       ENDLOOP.       IF P\_CHECK = 'X'.         PERFORM FRM\_CREATE\_ORDER.         PERFORM FRM\_CREATE\_INVOICE.       ENDIF.     WHEN 'DBCLICK'.       GET CURSOR LINE L\_ROW.       L\_ROW = TABCONT02-TOP\_LINE + L\_ROW - 1.       GET CURSOR FIELD L\_FIELD.       IF ( L\_FIELD = 'IT\_LIST-VBELN').         READ TABLE IT\_LIST INDEX L\_ROW.         IF SY-SUBRC = 0 AND IT\_LIST-VBELN <> ''.           SET PARAMETER ID 'AUN' FIELD IT\_LIST-VBELN.           CALL TRANSACTION 'VA03' AND SKIP FIRST SCREEN.         ENDIF.       ENDIF.       IF ( L\_FIELD = 'IT\_LIST-BELNR').         READ TABLE IT\_LIST INDEX L\_ROW.         IF SY-SUBRC = 0 AND IT\_LIST-BELNR <> ''.           SET PARAMETER ID 'VF' FIELD IT\_LIST-BELNR.           CALL TRANSACTION 'VF03' AND SKIP FIRST SCREEN.         ENDIF.       ENDIF.     WHEN 'MESSAGE'.       GET CURSOR LINE L\_ROW.       L\_ROW = TABCONT02-TOP\_LINE + L\_ROW - 1.       GET CURSOR FIELD L\_FIELD.       IF ( L\_FIELD = 'IT\_LIST-LIGHT').         READ TABLE IT\_LIST INDEX L\_ROW.         IF SY-SUBRC = 0 AND IT\_LIST-STATUS = 'E'.           CALL SCREEN 1003 STARTING AT 30 10 ENDING   AT 92   25.         ENDIF.       ENDIF.   ENDCASE.   CLEAR SY-UCOMM. ENDMODULE.  *\*&---------------------------------------------------------------------\** *\*&      Form  FRM\_INITAL* *\*&---------------------------------------------------------------------\** FORM FRM\_INITAL.   CLEAR:IT\_LIST.   DO I\_INITAL TIMES.     SSNUM = SSNUM + 1.     IT\_LIST-STATUS = ''.     IT\_LIST-SSNUM = SSNUM.     IT\_LIST-AUART = 'ZFR'.     IT\_LIST-ERDAT = SY-DATUM.     APPEND IT\_LIST.     CLEAR  IT\_LIST.   ENDDO. ENDFORM.  *\*&---------------------------------------------------------------------\** *\*&      Form  FRM\_GET\_PRICE\_DATA* *\*&---------------------------------------------------------------------\** FORM FRM\_GET\_PRICE\_DATA.   SELECT \* INTO CORRESPONDING FIELDS OF TABLE IT\_PRICE FROM ZTFI040.   SORT IT\_PRICE BY MATNR  ASCENDING.   I\_RECORD = LINES( IT\_PRICE ).   IF I\_RECORD = 0.     IT\_PRICE-ZGUID = 'ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ'.     IT\_PRICE-MATNR = ''.     IT\_PRICE-MAKTX = ''.     IT\_PRICE-PRICE = ''.     IT\_PRICE-CUREC = 'USD'.     APPEND IT\_PRICE.     TABCONT01-LINES = 1.   ELSE.     TABCONT01-LINES = LINES( IT\_PRICE ).   ENDIF. ENDFORM.  *\*&---------------------------------------------------------------------\** *\*&      Form  FRM\_SELECT\_1002* *\*&---------------------------------------------------------------------\** FORM FRM\_SELECT\_1002 USING STROPTION TYPE CHAR10.   LOOP AT IT\_LIST ASSIGNING <IT\_LIST>.     IF STROPTION = 'ALL'.       <IT\_LIST>-SEL = 'X'.     ELSEIF STROPTION = 'NO'.       <IT\_LIST>-SEL = ''.     ELSEIF STROPTION = 'OP'.       IF <IT\_LIST>-SEL = 'X'.         <IT\_LIST>-SEL = ''.       ELSE.         <IT\_LIST>-SEL = 'X'.       ENDIF.     ENDIF.   ENDLOOP. ENDFORM.  *\*&---------------------------------------------------------------------\** *\*&      Form  FRM\_SELECT\_1001* *\*&---------------------------------------------------------------------\** FORM FRM\_SELECT\_1001 USING STROPTION TYPE CHAR10.   LOOP AT IT\_PRICE ASSIGNING <IT\_PRICE>.     IF STROPTION = 'ALL'.       <IT\_PRICE>-SEL = 'X'.     ELSEIF STROPTION = 'NO'.       <IT\_PRICE>-SEL = ''.     ELSEIF STROPTION = 'OP'.       IF <IT\_PRICE>-SEL = 'X'.         <IT\_PRICE>-SEL = ''.       ELSE.         <IT\_PRICE>-SEL = 'X'.       ENDIF.     ENDIF.   ENDLOOP. ENDFORM.  *"""""[STATMENT]*  *\*&---------------------------------------------------------------------\** *\*&      Form  FRM\_CLARE\_DATA* *\*&---------------------------------------------------------------------\** FORM FRM\_CLARE\_DATA.   LOOP AT IT\_LIST ASSIGNING <IT\_LIST> WHERE SEL = 'X'                                       AND VBELN = '' AND BELNR = ''.     <IT\_LIST>-SEL = ''.     <IT\_LIST>-KUNNR = ''.     <IT\_LIST>-NAME1 = ''.     <IT\_LIST>-WERKS = ''.     <IT\_LIST>-MATNR = ''.     <IT\_LIST>-ARKTX = ''.     <IT\_LIST>-NETWR = ''.     <IT\_LIST>-ZTXT1 = ''.     <IT\_LIST>-ZTXT2 = ''.   ENDLOOP. ENDFORM.  *\*&---------------------------------------------------------------------\** *\*&      Form  FRM\_CREATE\_ORDER* *\*&---------------------------------------------------------------------\** FORM FRM\_CREATE\_ORDER.    *"BAPI需要的记录表*   DATA: HEADER\_IN            TYPE BAPISDHD1,         HEADER\_INX           TYPE BAPISDHD1X,         P\_VBELN              LIKE VBAK-VBELN,         P\_ZTERM              LIKE KNB1-ZTERM,         IT\_ORDER\_TEXT        TYPE TABLE OF BAPISDTEXT WITH HEADER LINE,         LT\_RETURN            TYPE TABLE OF BAPIRET2   WITH HEADER LINE,         ITEMS\_IN             TYPE TABLE OF BAPISDITM  WITH HEADER LINE,         ITEMS\_INX            TYPE TABLE OF BAPISDITMX WITH HEADER LINE,         PARTNERS             TYPE TABLE OF BAPIPARNR  WITH HEADER LINE,         SCHEDULES\_IN         TYPE TABLE OF BAPISCHDL  WITH HEADER LINE,         ORDER\_CONDITIONS\_IN  TYPE BAPICOND   OCCURS 0 WITH HEADER LINE,         ORDER\_CONDITIONS\_INX TYPE BAPICONDX  OCCURS 0 WITH HEADER LINE,         LOGIC\_SWITCH         TYPE BAPISDLS   OCCURS 0 WITH HEADER LINE,         EXTENSIONIN          TYPE  BAPIPAREX  OCCURS 0 WITH HEADER LINE.    *"保存前检查数据*   DATA:P\_CHECK TYPE CHAR1 VALUE ''.   DATA:P\_SSNUM TYPE CHAR10 VALUE ''.    LOOP AT IT\_LIST WHERE SEL = 'X'.     P\_CHECK = ''.     IF IT\_LIST-KUNNR = '' OR IT\_LIST-WERKS = '' OR        IT\_LIST-MATNR = '' OR IT\_LIST-NETWR = ''.       P\_CHECK = 'X'.     ENDIF.     IF P\_CHECK = 'X'.       P\_SSNUM = IT\_LIST-SSNUM.       CONDENSE P\_SSNUM.       CONCATENATE '第【 ' P\_SSNUM ' 】号记录中有关键信息未填写完整，请先检查！' INTO STR\_MESSAGE.       MESSAGE STR\_MESSAGE  TYPE 'I' DISPLAY LIKE 'E'.       EXIT.     ENDIF.   ENDLOOP.    IF P\_CHECK = 'X'.     RETURN.   ENDIF.    *"循环执行BAPI创建SO*   LOOP AT IT\_LIST ASSIGNING <IT\_LIST> WHERE SEL = 'X'.     *"清空变量内表*     CLEAR: P\_VBELN,P\_ZTERM,     HEADER\_IN,HEADER\_INX,     ORDER\_CONDITIONS\_IN,ORDER\_CONDITIONS\_INX,     ITEMS\_IN[],ITEMS\_IN,     ITEMS\_INX[],ITEMS\_INX,     PARTNERS[],PARTNERS,     SCHEDULES\_IN[],SCHEDULES\_IN,     IT\_ORDER\_TEXT[],IT\_ORDER\_TEXT.      *"补充数据*     CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'       EXPORTING         INPUT  = <IT\_LIST>-KUNNR       IMPORTING         OUTPUT = <IT\_LIST>-KUNNR.      CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'       EXPORTING         INPUT  = <IT\_LIST>-MATNR       IMPORTING         OUTPUT = <IT\_LIST>-MATNR.      SELECT SINGLE ZTERM INTO P\_ZTERM FROM KNB1 WHERE KUNNR = <IT\_LIST>-KUNNR.      *"抬头项目*     HEADER\_IN-DOC\_TYPE    = <IT\_LIST>-AUART.     HEADER\_IN-SALES\_ORG   = '1000'.     HEADER\_IN-DISTR\_CHAN  = '10'.     HEADER\_IN-DIVISION    = '10'.     HEADER\_IN-SD\_DOC\_CAT  = 'C'.     HEADER\_IN-DOC\_DATE    = SY-DATUM.     HEADER\_IN-PURCH\_DATE  = SY-DATUM.     HEADER\_IN-PRICE\_DATE  = SY-DATUM.     HEADER\_IN-PYMT\_METH   = P\_ZTERM.      HEADER\_INX-DOC\_TYPE    = 'X'.     HEADER\_INX-SALES\_ORG   = 'X'.     HEADER\_INX-DISTR\_CHAN  = 'X'.     HEADER\_INX-DIVISION    = 'X'.     HEADER\_INX-SD\_DOC\_CAT  = 'X'.     HEADER\_INX-DOC\_DATE    = 'X'.     HEADER\_INX-PURCH\_DATE  = 'X'.     HEADER\_INX-PRICE\_DATE  = 'X'.     HEADER\_INX-PYMT\_METH   = 'X'.      *"行项目*     ITEMS\_IN-ITM\_NUMBER = '10'.     ITEMS\_IN-MATERIAL   = <IT\_LIST>-MATNR.     ITEMS\_IN-PLANT      = <IT\_LIST>-WERKS.     ITEMS\_IN-ITEM\_CATEG = 'ZTAX'.     APPEND ITEMS\_IN.     CLEAR ITEMS\_IN.      ITEMS\_INX-ITM\_NUMBER  = 'X'.     ITEMS\_INX-MATERIAL    = 'X'.     ITEMS\_INX-PLANT       = 'X'.     ITEMS\_IN-ITEM\_CATEG   = 'X'.     APPEND ITEMS\_INX.     CLEAR ITEMS\_INX.      *"计划行数量*     SCHEDULES\_IN-ITM\_NUMBER   = '10'.     SCHEDULES\_IN-REQ\_QTY      = '1'.     APPEND SCHEDULES\_IN.     CLEAR SCHEDULES\_IN.      *"价格条件*     ORDER\_CONDITIONS\_IN-ITM\_NUMBER = '10'.  *"itm\_number*     ORDER\_CONDITIONS\_IN-COND\_TYPE  = 'PR00'.            *"定价条件*     ORDER\_CONDITIONS\_IN-COND\_VALUE = <IT\_LIST>-NETWR.     ORDER\_CONDITIONS\_IN-CURRENCY   = 'USD'.    *"*     ORDER\_CONDITIONS\_IN-COND\_UNIT  = 'EA'.  *"条件单位*     APPEND ORDER\_CONDITIONS\_IN.      ORDER\_CONDITIONS\_INX-ITM\_NUMBER = '10'.     ORDER\_CONDITIONS\_INX-COND\_TYPE  = 'PR00'.     ORDER\_CONDITIONS\_INX-COND\_VALUE = 'X'.     ORDER\_CONDITIONS\_INX-CURRENCY   = 'X'.     ORDER\_CONDITIONS\_INX-COND\_UNIT  = 'X'.     ORDER\_CONDITIONS\_INX-COND\_P\_UNT = 'X'.     APPEND ORDER\_CONDITIONS\_INX.      *"送达方 售达方 开票方 付款方 发票样式*     PARTNERS-PARTN\_ROLE = 'WE'.     PARTNERS-PARTN\_NUMB = <IT\_LIST>-KUNNR.     APPEND PARTNERS.     CLEAR PARTNERS.      PARTNERS-PARTN\_ROLE = 'RE'.     PARTNERS-PARTN\_NUMB = <IT\_LIST>-KUNNR.     APPEND PARTNERS.     CLEAR PARTNERS.      PARTNERS-PARTN\_ROLE = 'RG'.     PARTNERS-PARTN\_NUMB = <IT\_LIST>-KUNNR.     APPEND PARTNERS.     CLEAR PARTNERS.      PARTNERS-PARTN\_ROLE = 'AG'.     PARTNERS-PARTN\_NUMB = <IT\_LIST>-KUNNR.     APPEND PARTNERS.     CLEAR PARTNERS.      PARTNERS-PARTN\_ROLE = 'Z2'.     PARTNERS-PARTN\_NUMB = '0000000701'.     APPEND PARTNERS.     CLEAR PARTNERS.      *"header text*     IT\_ORDER\_TEXT-ITM\_NUMBER = SPACE.     IT\_ORDER\_TEXT-TEXT\_ID    = '0002'.     IT\_ORDER\_TEXT-LANGU      = SY-LANGU.     IT\_ORDER\_TEXT-FORMAT\_COL = '\*'.     IT\_ORDER\_TEXT-TEXT\_LINE  = <IT\_LIST>-ZTXT1.     APPEND IT\_ORDER\_TEXT.     CLEAR IT\_ORDER\_TEXT.      *"itemer remark*     IT\_ORDER\_TEXT-ITM\_NUMBER = '10'.     IT\_ORDER\_TEXT-TEXT\_ID    = 'Z007'.     IT\_ORDER\_TEXT-LANGU      = SY-LANGU.     IT\_ORDER\_TEXT-FORMAT\_COL = '\*'.     IT\_ORDER\_TEXT-TEXT\_LINE  = <IT\_LIST>-ZTXT2.     APPEND IT\_ORDER\_TEXT.     CLEAR IT\_ORDER\_TEXT.      *"执行BAPI*     CALL FUNCTION 'BAPI\_SALESORDER\_CREATEFROMDAT2'       EXPORTING         ORDER\_HEADER\_IN      = HEADER\_IN         ORDER\_HEADER\_INX     = HEADER\_INX         LOGIC\_SWITCH         = LOGIC\_SWITCH       IMPORTING         SALESDOCUMENT        = P\_VBELN       TABLES         RETURN               = LT\_RETURN         ORDER\_ITEMS\_IN       = ITEMS\_IN         ORDER\_ITEMS\_INX      = ITEMS\_INX         ORDER\_PARTNERS       = PARTNERS         ORDER\_SCHEDULES\_IN   = SCHEDULES\_IN         ORDER\_CONDITIONS\_IN  = ORDER\_CONDITIONS\_IN         ORDER\_CONDITIONS\_INX = ORDER\_CONDITIONS\_INX         ORDER\_TEXT           = IT\_ORDER\_TEXT         EXTENSIONIN          = EXTENSIONIN.     READ TABLE LT\_RETURN WITH KEY TYPE = 'E'.     IF SY-SUBRC = 0.       CALL FUNCTION 'BAPI\_TRANSACTION\_ROLLBACK'.       <IT\_LIST>-STATUS = 'E'.       *"赋值错误消息给消息表IT\_MSAG*       IT\_MSEG-SSNUM   = <IT\_LIST>-SSNUM.       IT\_MSEG-KUNNR   = <IT\_LIST>-KUNNR.       IT\_MSEG-MATNR   = <IT\_LIST>-MATNR.       IT\_MSEG-TYPE    = LT\_RETURN-TYPE.       IT\_MSEG-MESSAGE = LT\_RETURN-MESSAGE.       APPEND IT\_MSEG.       CLEAR  IT\_MSEG.     ELSE.       CALL FUNCTION 'BAPI\_TRANSACTION\_COMMIT'         EXPORTING           WAIT = 'X'.       <IT\_LIST>-STATUS = 'S'.       <IT\_LIST>-VBELN  = P\_VBELN.     ENDIF.   ENDLOOP. ENDFORM.  *\*&---------------------------------------------------------------------\** *\*&      Form  FRM\_CREATE\_INVOICE* *\*&---------------------------------------------------------------------\** FORM FRM\_CREATE\_INVOICE.    DATA:LS\_CREATORDATAIN TYPE BAPICREATORDATA,        IT\_BILLING       TYPE TABLE OF BAPIVBRK WITH HEADER LINE,        IT\_RETURN        TYPE TABLE OF BAPIRETURN1 WITH HEADER LINE,        IT\_SUCCESS       TYPE TABLE OF BAPIVBRKSUCCESS WITH HEADER LINE.    *"保存前检查数据*   DATA:P\_CHECK TYPE CHAR1 VALUE ''.   DATA:P\_SSNUM TYPE CHAR10 VALUE ''.    LOOP AT IT\_LIST WHERE SEL = 'X'.     P\_CHECK = ''.     IF IT\_LIST-VBELN = ''.       P\_CHECK = 'X'.     ENDIF.     IF P\_CHECK = 'X'.       P\_SSNUM = IT\_LIST-SSNUM.       CONDENSE P\_SSNUM.       CONCATENATE '第【 ' P\_SSNUM ' 】号记录中未创建销售订单,请检查！' INTO STR\_MESSAGE.       MESSAGE STR\_MESSAGE  TYPE 'I' DISPLAY LIKE 'E'.       EXIT.     ENDIF.      P\_CHECK = ''.     IF IT\_LIST-BELNR <> ''.       P\_CHECK = 'X'.     ENDIF.     IF P\_CHECK = 'X'.       P\_SSNUM = IT\_LIST-SSNUM.       CONDENSE P\_SSNUM.       CONCATENATE '第【 ' P\_SSNUM ' 】号记录中已创建销售发票,请检查！' INTO STR\_MESSAGE.       MESSAGE STR\_MESSAGE  TYPE 'I' DISPLAY LIKE 'E'.       EXIT.     ENDIF.   ENDLOOP.    IF P\_CHECK = 'X'.     RETURN.   ENDIF.    LS\_CREATORDATAIN-CREATED\_BY = SY-UNAME.   LS\_CREATORDATAIN-CREATED\_ON = SY-DATUM.   *"循环执行BAPI创建SO*   LOOP AT IT\_LIST ASSIGNING <IT\_LIST> WHERE SEL = 'X'.     IT\_BILLING-REF\_DOC    = <IT\_LIST>-VBELN.     IT\_BILLING-BILL\_DATE  = <IT\_LIST>-ERDAT.     IT\_BILLING-REF\_DOC\_CA = 'C'.     APPEND IT\_BILLING.     CLEAR  IT\_BILLING.     CALL FUNCTION 'BAPI\_BILLINGDOC\_CREATEMULTIPLE'       EXPORTING         CREATORDATAIN = LS\_CREATORDATAIN       TABLES         BILLINGDATAIN = IT\_BILLING         RETURN        = IT\_RETURN         SUCCESS       = IT\_SUCCESS.     READ TABLE IT\_SUCCESS INDEX 1.     IF SY-SUBRC = 0.       CALL FUNCTION 'BAPI\_TRANSACTION\_COMMIT'         EXPORTING           WAIT = 'X'.       <IT\_LIST>-STATUS = 'S'.       <IT\_LIST>-BELNR  = IT\_SUCCESS-BILL\_DOC.     ELSE.       CALL FUNCTION 'BAPI\_TRANSACTION\_ROLLBACK'.       <IT\_LIST>-STATUS = 'E'.       *"赋值错误消息给消息表IT\_MSAG*       READ TABLE IT\_RETURN WITH KEY TYPE = 'E'.       IF SY-SUBRC = 0.         IT\_MSEG-SSNUM   = <IT\_LIST>-SSNUM.         IT\_MSEG-KUNNR   = <IT\_LIST>-KUNNR.         IT\_MSEG-MATNR   = <IT\_LIST>-MATNR.         IT\_MSEG-TYPE    = IT\_RETURN-TYPE.         IT\_MSEG-MESSAGE = IT\_RETURN-MESSAGE.         APPEND IT\_MSEG.         CLEAR  IT\_MSEG.       ENDIF.     ENDIF.   ENDLOOP. ENDFORM.  *\*&---------------------------------------------------------------------\** *\*&      Module  SET\_STATUS\_DATA  OUTPUT* *\*&---------------------------------------------------------------------\** MODULE SET\_STATUS\_DATA OUTPUT.   READ TABLE IT\_LIST  INDEX TABCONT02-CURRENT\_LINE.   IF SY-SUBRC = 0.     IF IT\_LIST-STATUS = 'S'.       CALL FUNCTION 'ICON\_CREATE'         EXPORTING           NAME                  = 'ICON\_GREEN\_LIGHT'           INFO                  = 'S'           ADD\_STDINF            = 'X'         IMPORTING           RESULT                = IT\_LIST-LIGHT         EXCEPTIONS           ICON\_NOT\_FOUND        = 1           OUTPUTFIELD\_TOO\_SHORT = 2           OTHERS                = 3.     ELSEIF IT\_LIST-STATUS = 'E'.       CALL FUNCTION 'ICON\_CREATE'         EXPORTING           NAME                  = 'ICON\_RED\_LIGHT'           INFO                  = 'E'           ADD\_STDINF            = 'X'         IMPORTING           RESULT                = IT\_LIST-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\_LIST-LIGHT         EXCEPTIONS           ICON\_NOT\_FOUND        = 1           OUTPUTFIELD\_TOO\_SHORT = 2           OTHERS                = 3.     ENDIF.   ENDIF. ENDMODULE.  *\*&---------------------------------------------------------------------\** *\*&      Module  TABCONT02\_GET\_LINES  OUTPUT* *\*&---------------------------------------------------------------------\** MODULE TABCONT02\_GET\_LINES OUTPUT.   G\_TABCONT02\_LINES = SY-LOOPC. ENDMODULE.  *\*&---------------------------------------------------------------------\** *\*&      Module  GET\_MASTER\_DATA  INPUT* *\*&---------------------------------------------------------------------\** MODULE GET\_MASTER\_DATA\_1002 INPUT.   *"读取：客人主数据*   IF IT\_LIST-KUNNR IS NOT INITIAL AND IT\_LIST-NAME1 IS INITIAL.     CLEAR P\_KUNNR.     CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'       EXPORTING         INPUT  = IT\_LIST-KUNNR       IMPORTING         OUTPUT = P\_KUNNR.      SELECT SINGLE NAME1 INTO IT\_LIST-NAME1 FROM KNA1 WHERE KUNNR = P\_KUNNR.     SELECT SINGLE VWERK INTO IT\_LIST-WERKS FROM KNVV WHERE KUNNR = P\_KUNNR AND VKORG = '1000' AND VTWEG = '10' AND SPART = '10'.   ENDIF.    *"读取：物料主数据*   IF IT\_LIST-MATNR IS NOT INITIAL.     CLEAR P\_MATNR.     CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'       EXPORTING         INPUT  = IT\_LIST-MATNR       IMPORTING         OUTPUT = P\_MATNR.     SELECT SINGLE MAKTX INTO IT\_LIST-ARKTX FROM MAKT WHERE MATNR = P\_MATNR AND SPRAS = 'E'.     IF IT\_LIST-NETWR = ''.       SELECT SINGLE PRICE INTO IT\_LIST-NETWR FROM ZTFI040 WHERE MATNR = P\_MATNR.     ENDIF.   ENDIF.    MOVE-CORRESPONDING TABCONT02 TO IT\_LIST.   MODIFY IT\_LIST INDEX TABCONT02-CURRENT\_LINE. ENDMODULE.  *\*&---------------------------------------------------------------------\** *\*&      Module  GET\_MASTER\_DATA  INPUT* *\*&---------------------------------------------------------------------\** MODULE GET\_MASTER\_DATA\_1001 INPUT.   *"读取：物料主数据*   READ TABLE IT\_PRICE INDEX TABCONT01-CURRENT\_LINE.   IF SY-SUBRC = 0.     IF IT\_PRICE-MATNR IS NOT INITIAL AND IT\_PRICE-MAKTX IS INITIAL.       CLEAR P\_MATNR.       CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'         EXPORTING           INPUT  = IT\_PRICE-MATNR         IMPORTING           OUTPUT = P\_MATNR.       SELECT SINGLE MAKTX INTO IT\_PRICE-MAKTX FROM MAKT WHERE MATNR = P\_MATNR AND SPRAS = 'E'.     ENDIF.   ENDIF.    MOVE-CORRESPONDING TABCONT01 TO IT\_PRICE.   MODIFY IT\_PRICE INDEX TABCONT01-CURRENT\_LINE. ENDMODULE.  *\*&---------------------------------------------------------------------\** *\*&      Module  STATUS\_1001  OUTPUT* *\*&---------------------------------------------------------------------\** MODULE STATUS\_1001 OUTPUT.   SET PF-STATUS 'STATUS\_1001'.   SET TITLEBAR 'TITLE'.    DESCRIBE TABLE IT\_PRICE LINES TABCONT01-LINES. ENDMODULE.  *\*&---------------------------------------------------------------------\** *\*&      Module  USER\_COMMAND\_1001  INPUT* *\*&---------------------------------------------------------------------\** MODULE USER\_COMMAND\_1001 INPUT.   CASE SY-UCOMM.     WHEN 'BACK' .       CLEAR SY-UCOMM.       CALL SCREEN 1002.     WHEN 'EXIT'.       LEAVE PROGRAM.     WHEN 'SELALL'.       PERFORM FRM\_SELECT\_1001 USING 'ALL'.     WHEN 'SELNO'.       PERFORM FRM\_SELECT\_1001 USING 'NO'.     WHEN 'SELOP'.       PERFORM FRM\_SELECT\_1001 USING 'OP'.     WHEN 'ADD'.       PERFORM FRM\_IT\_PRICE\_OPTION USING 'ADD'.     WHEN 'DELETE'.       PERFORM FRM\_IT\_PRICE\_OPTION USING 'DELETE'.     WHEN 'REFRESH'.       PERFORM FRM\_IT\_PRICE\_OPTION USING 'REFRESH'.     WHEN 'SAVE'.       PERFORM FRM\_SAVE\_PRICE\_DATA.     WHEN OTHERS.   ENDCASE.   CLEAR SY-UCOMM. ENDMODULE.  *\*&---------------------------------------------------------------------\** *\*&      Module  RES\_INPUTS\_1002  INPUT* *\*&---------------------------------------------------------------------\** MODULE RES\_INPUTS\_1002\_01 INPUT.   MOVE-CORRESPONDING TABCONT02 TO IT\_LIST.   MODIFY IT\_LIST INDEX TABCONT02-CURRENT\_LINE. ENDMODULE.  *\*&---------------------------------------------------------------------\** *\*&      Module  RES\_INPUTS  INPUT* *\*&---------------------------------------------------------------------\** MODULE RES\_INPUTS1 INPUT.   MOVE-CORRESPONDING TABCONT01 TO IT\_PRICE.   MODIFY IT\_PRICE INDEX TABCONT01-CURRENT\_LINE. ENDMODULE.  *\*&---------------------------------------------------------------------\** *\*&      Form  FRM\_SAVE\_PRICE\_DATA* *\*&---------------------------------------------------------------------\** FORM FRM\_IT\_PRICE\_OPTION   USING P\_OPTION TYPE CHAR10.    *"维护价钱操作*   IF P\_OPTION = 'ADD'.     CLEAR P\_GUID.     CALL FUNCTION 'SYSTEM\_UUID\_CREATE'       IMPORTING         UUID = P\_GUID.     IT\_PRICE-ZGUID = P\_GUID.     IT\_PRICE-MATNR = ''.     IT\_PRICE-MAKTX = ''.     IT\_PRICE-PRICE = ''.     IT\_PRICE-CUREC = 'USD'.     APPEND IT\_PRICE.   ELSEIF P\_OPTION = 'DELETE'.     STR\_MESSAGE = '是否確認刪除选中的记录？'.     CALL FUNCTION 'POPUP\_TO\_CONFIRM'       EXPORTING         TITLEBAR              = '確定'         TEXT\_QUESTION         = STR\_MESSAGE         TEXT\_BUTTON\_1         = '是,刪除'         ICON\_BUTTON\_1         = 'ICON\_OKAY'         TEXT\_BUTTON\_2         = '否,取消'         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'.       LOOP AT IT\_PRICE WHERE SEL = 'X'.         DELETE FROM ZTFI040 WHERE ZGUID = IT\_PRICE-ZGUID.       ENDLOOP.       *"刷新数据*       PERFORM FRM\_GET\_PRICE\_DATA.     ENDIF.   ELSEIF P\_OPTION = 'REFRESH'.     PERFORM FRM\_GET\_PRICE\_DATA.   ENDIF. ENDFORM.  *\*&---------------------------------------------------------------------\** *\*&      Form  FRM\_SAVE\_PRICE\_DATA* *\*&---------------------------------------------------------------------\** FORM FRM\_SAVE\_PRICE\_DATA.   *"目标表赋值*   LOOP AT IT\_PRICE WHERE ZGUID <> '' AND  MATNR <> '' AND MAKTX <> ''.     IT\_ZFI40-ZGUID = IT\_PRICE-ZGUID.     IT\_ZFI40-MATNR = IT\_PRICE-MATNR.     IT\_ZFI40-MAKTX = IT\_PRICE-MAKTX.     IT\_ZFI40-PRICE = IT\_PRICE-PRICE.     IT\_ZFI40-CUREC = IT\_PRICE-CUREC.     APPEND IT\_ZFI40.   ENDLOOP.    *"有数据的时候保存*   IF LINES( IT\_ZFI40 ) > 0.     MODIFY ZTFI040 FROM TABLE IT\_ZFI40.     STR\_MESSAGE = '数据已经成功保存！'.     MESSAGE STR\_MESSAGE  TYPE 'I' DISPLAY LIKE 'S'.     PERFORM FRM\_GET\_PRICE\_DATA.   ENDIF. ENDFORM.  *\*&---------------------------------------------------------------------\** *\*&      Module  STATUS\_1003  OUTPUT* *\*&---------------------------------------------------------------------\** MODULE STATUS\_1003 OUTPUT.   SET PF-STATUS 'STATUS\_1003'.   PERFORM FRM\_CREATE\_LONG\_TEXT USING 1. ENDMODULE.  *\*&---------------------------------------------------------------------\** *\*&      Module  USER\_COMMAND\_1003  INPUT* *\*&---------------------------------------------------------------------\** MODULE USER\_COMMAND\_1003 INPUT.   CASE SY-UCOMM.     WHEN 'EXIT'.       LEAVE TO SCREEN 0.   ENDCASE. 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.   DATA: LT\_TEXT TYPE STANDARD TABLE OF CHAR255,  *"中间处理行数255字符*         LS\_TEXT LIKE LINE OF LT\_TEXT.            *"文本内表*    *"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 ).    *"最大获取长文本数值255字符，若需要处理多行则执行控制255行数统计*   LOOP AT IT\_MSEG.     LS\_TEXT = IT\_MSEG-MESSAGE.     APPEND LS\_TEXT TO LT\_TEXT.   ENDLOOP.    *"绑定长文本数值*   GUI\_LONG\_TEXTEDITOR->SET\_TEXT\_AS\_R3TABLE( TABLE = LT\_TEXT ). ENDFORM. |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |
|  |
| PROCESS BEFORE OUTPUT.   MODULE STATUS\_1001.   LOOP AT IT\_PRICE WITH CONTROL TABCONT01.     MODULE ENABLE\_FIELDS.   ENDLOOP.  PROCESS AFTER INPUT.  LOOP AT IT\_PRICE.     CHAIN.       FIELD IT\_PRICE-SEL.       FIELD IT\_PRICE-MATNR.       FIELD IT\_PRICE-MAKTX.       FIELD IT\_PRICE-PRICE.       FIELD IT\_PRICE-CUREC.       MODULE RES\_INPUTS1 ON CHAIN-REQUEST.     ENDCHAIN.     MODULE GET\_MASTER\_DATA\_1001.   ENDLOOP.    MODULE USER\_COMMAND\_1001. |
|  |
| PROCESS BEFORE OUTPUT.   MODULE STATUS\_1002.   LOOP AT IT\_LIST WITH CONTROL TABCONT02.     MODULE SET\_STATUS\_DATA.     MODULE TABCONT02\_GET\_LINES.   ENDLOOP.  PROCESS AFTER INPUT.   LOOP AT IT\_LIST.     CHAIN.       FIELD IT\_LIST-SEL.       FIELD IT\_LIST-LIGHT.       FIELD IT\_LIST-KUNNR.       FIELD IT\_LIST-WERKS.       FIELD IT\_LIST-MATNR.       FIELD IT\_LIST-VBELN.       FIELD IT\_LIST-NETWR.       FIELD IT\_LIST-ZTXT1.       FIELD IT\_LIST-ZTXT2.       MODULE RES\_INPUTS\_1002\_01 ON CHAIN-REQUEST.       MODULE GET\_MASTER\_DATA\_1002.     ENDCHAIN.   ENDLOOP.    MODULE USER\_COMMAND\_1002. |
|  |
| PROCESS BEFORE OUTPUT.   MODULE STATUS\_1003.  PROCESS AFTER INPUT.   MODULE USER\_COMMAND\_1003. |

1001

|  |
| --- |
|  |
|  |

1002

|  |
| --- |
|  |
|  |
|  |

1003

|  |
| --- |
|  |