BOM 查询：含后台执行

```
*Code listing for: ZPP009
*Description: BOM清单查询
*---------------------------------------------------------------------

*&---------------------------------------------------------------------*
*& Report  ZPP009
*&
*&---------------------------------------------------------------------*
*&
*&
*&---------------------------------------------------------------------*

REPORT zpp009.


TYPE-POOLS: slis.
TABLES: mast, stpo, stpox, tc04.
************************************************************************
*                                                                      *
************************************************************************
SELECTION-SCREEN BEGIN OF BLOCK blk_004 WITH FRAME TITLE text-bk1. "blk_004 .
SELECT-OPTIONS s_matnr FOR mast-matnr.
PARAMETERS: s_sxpq LIKE sy-datum DEFAULT sy-datum.
PARAMETERS: s_sl LIKE stpo-menge ."OBLIGATORY. DEFAULT '1'."数量
PARAMETERS: s_stlan LIKE stpox-stlan OBLIGATORY DEFAULT '1' ."用途
SELECT-OPTIONS: s_werks FOR mast-werks. "工厂
*parameters: s_werks like mast-werks obligatory ."工厂
PARAMETERS: s_capid LIKE tc04-capid DEFAULT 'PP01'.
SELECTION-SCREEN END OF BLOCK blk_004.

SELECTION-SCREEN BEGIN OF BLOCK blk_003 WITH FRAME TITLE text-bk2. "blk_003 .
PARAMETERS: p1 RADIOBUTTON GROUP g1,
          p_list RADIOBUTTON GROUP g1,  "List方式
          p3 RADIOBUTTON GROUP g1 MODIF ID nac,
          p7 RADIOBUTTON GROUP g1 MODIF ID nac.
SELECTION-SCREEN END OF BLOCK blk_003.

SELECTION-SCREEN BEGIN OF BLOCK blk_002 WITH FRAME TITLE text-bk3. "blk_002 .
PARAMETERS: p2 RADIOBUTTON GROUP g2,
          p4 RADIOBUTTON GROUP g2.
SELECTION-SCREEN END OF BLOCK blk_002.

SELECTION-SCREEN BEGIN OF BLOCK blk_001 WITH FRAME TITLE text-bk4. "blk_001 .
PARAMETERS: p5 RADIOBUTTON GROUP g3 MODIF ID nac,
          p6 RADIOBUTTON GROUP g3 MODIF ID nac.
SELECTION-SCREEN END OF BLOCK blk_001.

SELECTION-SCREEN BEGIN OF BLOCK bk5 WITH FRAME TITLE text-bk5.
PARAMETERS: onlyerr AS CHECKBOX.  "只显示有错误的
SELECTION-SCREEN END OF BLOCK bk5.
************************************************************************
*                                                                      *
************************************************************************
TYPES BEGIN OF ts_cstmat.
INCLUDE TYPE cstmat AS head RENAMING WITH SUFFIX _h.
TYPES END OF ts_cstmat.
TYPES BEGIN OF ts_stpox.
INCLUDE TYPE stpox AS item RENAMING WITH SUFFIX _i.
TYPES END OF ts_stpox.

*...最终显示内表
TYPES BEGIN OF ts_dis.
       INCLUDE TYPE ts_cstmat.
       INCLUDE TYPE ts_stpox.
TYPES:  posit(12),  "  位置串号
       posit_sort TYPE string,  "辅助排序的位置串号
       ebort LIKE stpu-ebort,  "所有的位置号
       matne LIKE mara-matnr,  "上阶物料料号
       potx_str TYPE string,   "项目长文本
       bismt_i LIKE mara-bismt,  "组件旧物料号
       bismt_h LIKE mara-bismt,  "旧物料号
       eknam_i TYPE t024-eknam,  "采购组描述
       beskz_i LIKE marc-beskz,  "采购类型
       ir_bz_i TYPE flag,        "标准采购信息记录 存在
       ir_wx_i TYPE flag,        "外协采购信息记录 存在
       "组件分配的工序
       plnty TYPE plmz-plnty,    "任务清单类型
```

```
        plnnr TYPE plmz-plnnr,    "组
        plnal TYPE plmz-plnal,    "组计数器
        plnfl TYPE plmz-plnfl,    "序列
        plnkn TYPE plmz-plnkn,    "任务清单节点
        vornr TYPE plpo-vornr,    "工序

        color TYPE c LENGTH 4,    "行颜色
        err_x TYPE flag,          "错误标志

***       add by song
        LTEXT TYPE C LENGTH 40,
        RGEKZ TYPE RGEKZ,
        CJTXT TYPE C LENGTH 100.
TYPES END OF ts_dis.
TYPES:  tt_dis TYPE STANDARD TABLE OF ts_dis.
DATA: gt_dis TYPE tt_dis.

************************************************************************
*                                                                      *
************************************************************************
AT SELECTION-SCREEN OUTPUT.
  LOOP AT SCREEN.
    IF screen-group1 = 'NAC'.
      screen-active = 0.
      MODIFY SCREEN.
    ENDIF.
  ENDLOOP.

START-OF-SELECTION.
  PERFORM get_data.   "保持通用性？
  PERFORM pro_data.   "数据的定制处理

END-OF-SELECTION.
  PERFORM dis_data.


*&---------------------------------------------------------------------*
*&      Form  GET_DATA
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
FORM get_data .
*-  bom 范围
  DATA: lt_mast TYPE STANDARD TABLE OF mast.

  SELECT *
     INTO CORRESPONDING FIELDS OF TABLE lt_mast
     FROM mast AS a
       JOIN stko AS b ON b~stlnr = a~stlnr AND b~stlal = a~stlal
     WHERE a~matnr IN s_matnr
       AND a~werks IN s_werks
       AND b~lkenz = space.
  SORT lt_mast BY werks matnr stlan stlal.

*----------------------------------------------------------------------*
  DATA: ls_dis LIKE LINE OF gt_dis,
        ls_topmat_h TYPE ts_cstmat,
        ls_stpox_i TYPE ts_stpox.
  DATA: ld_datuv TYPE stko-datuv,
        multilevel TYPE csdata-xfeld,
        ld_new TYPE flag. "新版本、新Bom标示
  DATA: ls_topmat TYPE cstmat,
        lt_stpox TYPE STANDARD TABLE OF stpox,
        lt_matcat TYPE STANDARD TABLE OF cscmat,
        ls_matcat LIKE LINE OF lt_matcat.
  DATA: lv_tdname LIKE stxh-tdname.

  CHECK lt_mast[] IS NOT INITIAL.
  ld_datuv = s_sxpq.
  IF p2 = 'X'.
    multilevel = 'X'.
  ELSE.
    multilevel = space.
  ENDIF.

  LOOP AT lt_mast INTO mast.
    CLEAR ls_topmat. REFRESH lt_stpox. REFRESH lt_matcat.
    CALL FUNCTION 'CS_BOM_EXPL_MAT_V2'
```

```
    EXPORTING
      capid                = s_capid        "Application ID
      datuv                = ld_datuv       "Validity date
      emeng                  = s_sl         "Required quantity
      auskz                = space          "scrap
      mehrs                 = multilevel          "Multi-level explosion
      mtnrv                = mast-matnr       "Material
      stlal               = mast-stlal  "'01' "Alternative BOM
      stlan                = s_stlan        "BOM usage
      werks                = mast-werks       "Plant
      rndkz                = '1'        "Round off: ' '=always, '1'=never, '2'=only levels > 1
    IMPORTING
      topmat               = ls_topmat
    TABLES
      stb                 = lt_stpox
      matcat               = lt_matcat
    EXCEPTIONS
      alt_not_found        = 1
      call_invalid         = 2
      material_not_found    = 3
      missing_authorization = 4
      no_bom_found          = 5
      no_plant_data        = 6
      no_suitable_bom_found = 7
      conversion_error      = 8
      OTHERS               = 9. "CS_BOM_EXPL_MAT_V2
  CHECK sy-subrc = 0.

  ld_new = 'X'.
  LOOP AT lt_stpox INTO stpox WHERE loekz = space AND xloek = space.
    ls_topmat_h = ls_topmat.
    ls_stpox_i = stpox.

    "添加0层信息
    IF ld_new IS NOT INITIAL.
      CLEAR ls_dis.
      MOVE-CORRESPONDING ls_topmat_h TO ls_dis.
      ls_dis-posit = '0'.
      ls_dis-posnr_i = '0001'.
      ls_dis-idnrk_i = ls_dis-matnr_h.  "组件
      ls_dis-ojtxp_i = ls_dis-maktx_h.  "组件描述
      ls_dis-mngko_i = ls_dis-emmbm_h.  "组件数量
      ls_dis-meins_i = ls_dis-bmein_h.    "组件单位
      ls_dis-sanka_i = 'X'.  "成本核算标志
      APPEND ls_dis TO gt_dis.
    ENDIF.

    CLEAR ls_dis.
    MOVE-CORRESPONDING ls_topmat_h TO ls_dis.
    MOVE-CORRESPONDING ls_stpox_i TO ls_dis.

    PERFORM get_posit USING ls_topmat-matnr stpox-stufe ld_new CHANGING ls_dis-posit ls_dis-posit_sort.  "获得位置串号
    ld_new = space.

*   IF p5 = 'X'.
      PERFORM get_ebort USING stpox-stlnr stpox-stlkn stpox-stpoz
                    CHANGING ls_dis-cjtxt.
*   ENDIF.
    "上级物料号
    READ TABLE lt_matcat INTO ls_matcat INDEX stpox-ttidx.
    IF sy-subrc = 0.
      ls_dis-matne = ls_matcat-matnr.
    ENDIF.

    "项目长文本
    IF stpox-ltxsp IS NOT INITIAL.
      CONCATENATE sy-mandt stpox-stlty stpox-stlnr stpox-stlkn stpox-stpoz INTO lv_tdname.
      PERFORM read_text USING 'MPO' stpox-ltxsp lv_tdname 'BOM' CHANGING ls_dis-potx_str.
    ENDIF.
    IF ls_dis-potx_str IS INITIAL AND ( stpox-potx1 IS NOT INITIAL OR stpox-potx2 IS NOT INITIAL ).
      "如果长文本不存在，则长文本显示为 项目文本1+项目文本2
      CONCATENATE stpox-potx1 stpox-potx2 INTO ls_dis-potx_str SEPARATED BY space.
    ENDIF.
    "组件旧物料号
    SELECT SINGLE bismt FROM mara INTO ls_dis-bismt_i
       WHERE matnr = ls_dis-idnrk_i.
    "BOM物料旧物料号
    SELECT SINGLE bismt FROM mara INTO ls_dis-bismt_h
       WHERE matnr = ls_dis-matnr_h.

    APPEND ls_dis TO gt_dis.
```

```
    ENDLOOP.
  ENDLOOP.

  SORT gt_dis BY werks_h matnr_h stlal_h posit_sort.
  FREE lt_mast.
ENDFORM.                    " GET_DATA
*&---------------------------------------------------------------------*
*&      Form  DIS_DATA
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
FORM dis_data .
  DATA: lt_fieldcat TYPE slis_t_fieldcat_alv,  "alv显示所需参数
        ls_layout TYPE slis_layout_alv,    "ALV布局设置
        l_title TYPE lvc_title.      "ALV标题
*  data: lt_filter type  slis_t_filter_alv,  "筛选
*        ls_filter like line of lt_filter.
*  "只显示错误的
*  if onlyerr = 'X'.
*    clear ls_filter.
*    ls_filter-fieldname = 'ERR_X'.
*    ls_filter-sign0 = 'I'.
*    ls_filter-optio = 'EQ'.
*    ls_filter-valuf = 'X'.
*    append ls_filter to lt_filter.
*  endif.

  "布局与字段目录需与下面后台运行时一致
  PERFORM fieldcat_init TABLES lt_fieldcat.  "字段标题

  ls_layout-colwidth_optimize = 'X'.    "设置Grid的字段列宽度自动适应
  ls_layout-zebra = 'X'.        "设置Grid的行颜色变换显示
  ls_layout-info_fieldname = 'COLOR'.    "设置行颜色值的内表字段名

  DATA: lt_sort TYPE slis_t_sortinfo_alv WITH HEADER LINE.
  CLEAR lt_sort.
  lt_sort-fieldname = 'WERKS_H'. lt_sort-up = 'X'. lt_sort-down = space. lt_sort-subtot = ''. APPEND lt_sort. CLEAR lt_sort.
  lt_sort-fieldname = 'MATNR_H'. lt_sort-up = 'X'. lt_sort-down = space. lt_sort-subtot = ''. APPEND lt_sort. CLEAR lt_sort.
*  lt_sort-fieldname = 'STLAL_H'. lt_sort-up = 'X'. lt_sort-down = space. lt_sort-subtot = ''. APPEND lt_sort. CLEAR lt_sort.
*  lt_sort-fieldname = 'POSIT'. lt_sort-up = 'X'. lt_sort-down = space. lt_sort-subtot = ''. APPEND lt_sort. CLEAR lt_sort.

  CHECK lt_fieldcat[] IS NOT INITIAL.

  IF sy-batch IS INITIAL.
    DATA: ld_funname TYPE string. "函数名

    IF p1 = 'X'.  "alv
      ld_funname = 'REUSE_ALV_GRID_DISPLAY'.
    ELSEIF p_list = 'X'.  "list
      ld_funname = 'REUSE_ALV_LIST_DISPLAY'.
    ENDIF.
    CALL FUNCTION ld_funname  "'REUSE_ALV_GRID_DISPLAY'
      EXPORTING
        i_callback_program = sy-cprog    "回调的程序(本程序)
*       I_CALLBACK_PF_STATUS_SET    = 'SET_PF_STATUS' "设置gui状态条
*       i_callback_user_command     = 'USERCOMMAND' "事件调用的FORM
*       I_CALLBACK_HTML_TOP_OF_PAGE = 'ALV_TOP_OF_PAGE' "如果需要表头
        it_sort           = lt_sort[]       "如果有排序和分类汇总，需要该行
*       i_grid_title      = l_title
        i_save            = 'A'
        is_layout         = ls_layout
        it_fieldcat       = lt_fieldcat
*       it_filter         = lt_filter   "筛选
      TABLES
        t_outtab          = gt_dis       "显示的内表
      EXCEPTIONS
        program_error     = 1
        OTHERS            = 2.

  ELSE.
    "后台执行  后台作业 后台程序 后台生成xls
*    DATA: ls_layout_oo TYPE lvc_s_layo,
*          lt_fieldcatalog_oo TYPE lvc_t_fcat.
*
*    ls_layout_oo-cwidth_opt = 'X'.    "设置Grid的字段列宽度自动适应
*    ls_layout_oo-zebra = 'X'.        "设置Grid的行颜色变换显示
*    ls_layout_oo-info_fname = 'COLOR'.    "设置行颜色值的内表字段名
*
```

```
*    PERFORM pre_fieldcat_oo TABLES lt_fieldcatalog_oo USING '01'.

*    PERFORM download_background TABLES gt_dis lt_fieldcatalog_oo USING ls_layout_oo .

    PERFORM download_background_txt TABLES gt_dis.
  ENDIF.

ENDFORM.                    " DIS_DATA
*&---------------------------------------------------------------------*
*&      Form  usercommand
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*      -->P_UCOMM    text
*      -->P_SELF     text
*----------------------------------------------------------------------*
FORM usercommand USING p_ucomm TYPE sy-ucomm
                       p_self TYPE slis_selfield.
  DATA: lr_grid TYPE REF TO cl_gui_alv_grid.
  CALL FUNCTION 'GET_GLOBALS_FROM_SLVC_FULLSCR'
    IMPORTING
      e_grid = lr_grid.
  CALL METHOD lr_grid->check_changed_data.
  p_self-refresh = 'X'.    "刷新数据


  DATA: ls_dis LIKE LINE OF gt_dis.

  IF p_self-tabindex > 0.
    READ TABLE gt_dis INTO ls_dis INDEX p_self-tabindex.
    IF sy-subrc = 0.
      PERFORM dis_bomcom USING ls_dis.  "显示BOM组件
    ENDIF.
  ELSE.
    MESSAGE e028(00) WITH '请选择有效行'.
    STOP.
  ENDIF.

ENDFORM.                    "usercommand
*&---------------------------------------------------------------------*
*&      Form  FIELDCAT_INIT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*      -->P_LT_FIELDCAT  text
*----------------------------------------------------------------------*
FORM fieldcat_init  TABLES    ct_fieldcat TYPE slis_t_fieldcat_alv.
*                    USING    I_FLAG.
  CONSTANTS:
    k_fn_sim(100) VALUE
      'FIELDNAME*REPTEXT_DDIC*OUTPUTLEN*EMPHASIZE*KEY*NO_ZERO',
    k_fn_ref(100) VALUE
      'FIELDNAME*REF_TABNAME*REF_FIELDNAME*KEY*NO_ZERO*NO_OUT*DO_SUM*OUTPUTLEN*EMPHASIZE',
    k_fn_r02(100) VALUE
      'FIELDNAME*REF_TABNAME*REF_FIELDNAME*KEY*NO_ZERO*SELTEXT_S*SELTEXT_M*SELTEXT_L*REPTEXT_DDIC',
    k_fn_q(100) VALUE
      'FIELDNAME*REPTEXT_DDIC*NO_ZERO*QFIELDNAME*QUANTITY*OUTPUTLEN*EMPHASIZE',
    k_fn_c(100) VALUE
      'FIELDNAME*REPTEXT_DDIC*NO_ZERO*CFIELDNAME*CURRENCY*OUTPUTLEN*EMPHASIZE'.

  DATA: ls_fcat     TYPE slis_fieldcat_alv,
        lt_fcat     TYPE slis_t_fieldcat_alv WITH HEADER LINE,
        l_tabix     TYPE sy-tabix.
  FIELD-SYMBOLS: <ls_fcat> TYPE slis_fieldcat_alv.

  REFRESH ct_fieldcat.

*  refresh lt_fcat.
*  call function 'REUSE_ALV_FIELDCATALOG_MERGE'
*    exporting
*      i_program_name         = sy-repid
*      i_structure_name       = 'CSTMAT'
*    changing
*      ct_fieldcat            = lt_fcat[]
*    exceptions
*      inconsistent_interface = 1
*      program_error          = 2
*      others                 = 3.
*  loop at lt_fcat assigning <ls_fcat>.
*    concatenate <ls_fcat>-fieldname '_H' into <ls_fcat>-fieldname.
*  endloop.
```

```
*  append lines of lt_fcat to ct_fieldcat.
*
*  refresh lt_fcat.
*  call function 'REUSE_ALV_FIELDCATALOG_MERGE'
*    exporting
*      i_program_name         = sy-repid
*      i_structure_name       = 'STPOX'
*    changing
*      ct_fieldcat            = lt_fcat[]
*    exceptions
*      inconsistent_interface = 1
*      program_error          = 2
*      others                 = 3.
*  loop at lt_fcat assigning <ls_fcat>.
*    concatenate <ls_fcat>-fieldname '_I' into <ls_fcat>-fieldname.
*  endloop.
*  append lines of lt_fcat to ct_fieldcat.

  PERFORM make_fieldcat:
    TABLES ct_fieldcat USING k_fn_r02 'WERKS_H*CSTMAT*WERKS',
    TABLES ct_fieldcat USING k_fn_r02 'MATNR_H*CSTMAT*MATNR',
    TABLES ct_fieldcat USING k_fn_r02 'BISMT_H*MARA*BISMT',
    TABLES ct_fieldcat USING k_fn_r02 'MAKTX_H*CSTMAT*MAKTX',
    TABLES ct_fieldcat USING k_fn_r02 'POSIT*STPOX*POSIT* * *BOM层次*BOM层次*BOM层次*BOM层次',
    TABLES ct_fieldcat USING k_fn_r02 'STLAL_H*CSTMAT*STLAL* * *BOM版本*BOM版本*BOM版本*BOM版本',
    TABLES ct_fieldcat USING k_fn_r02 'STKTX_H*CSTMAT*STKTX',
    TABLES ct_fieldcat USING k_fn_r02 'EMMBM_H*CSTMAT*BMENG',
    TABLES ct_fieldcat USING k_fn_r02 'POSNR_I*STPOX*POSNR',
    TABLES ct_fieldcat USING k_fn_r02 'ALPGR_I*STPOX*ALPGR',
    TABLES ct_fieldcat USING k_fn_r02 'ALPRF_I*STPOX*ALPRF',
    TABLES ct_fieldcat USING k_fn_r02 'EWAHR_I*STPOX*EWAHR',
    TABLES ct_fieldcat USING k_fn_r02 'IDNRK_I*STPOX*IDNRK',
    TABLES ct_fieldcat USING k_fn_r02 'BISMT_I*MARA*BISMT* * *组件旧物料号*组件旧物料号*组件旧物料号*组件旧物料号',
    TABLES ct_fieldcat USING k_fn_r02 'OJTXP_I*STPOX*OJTXP* * *组件描述*组件描述*组件描述*组件描述',
    TABLES ct_fieldcat USING k_fn_r02 'POSTP_I*STPOX*POSTP',
    TABLES ct_fieldcat USING k_fn_r02 'MNGKO_I*STPOX*MENGE',
    TABLES ct_fieldcat USING k_fn_r02 'AUSCH_I*STPO*AUSCH',
    TABLES ct_fieldcat USING k_fn_r02 'MEINS_I*MARA*MEINS',
    TABLES ct_fieldcat USING k_fn_r02 'KZAUS_I*STPOX*KZAUS',
    TABLES ct_fieldcat USING k_fn_r02 'AUSDT_I*STPOX*AUSDT',
    TABLES ct_fieldcat USING k_fn_r02 'NFMAT_I*STPOX*NFMAT',
*    tables ct_fieldcat using k_fn_r02 'ZEINR*MARA*ZEINR',
    TABLES ct_fieldcat USING k_fn_r02 'POTX1_I*STPOX*POTX1',
    TABLES ct_fieldcat USING k_fn_r02 'POTX2_I*STPOX*POTX2',
    TABLES ct_fieldcat USING k_fn_sim 'POTX_STR*项目长文本*2000',
*    tables ct_fieldcat using k_fn_r02 'EBORT*STPU*EBORT* * *物料位号*物料位号*物料位号*物料位号',
    TABLES ct_fieldcat USING k_fn_r02 'MATNE*MARA*MATNR* * *上级物料*上级物料*上级物料*上级物料',
    TABLES ct_fieldcat USING k_fn_r02 'SANKA_I*STPOX*SANKA',
    TABLES ct_fieldcat USING k_fn_sim 'ERR_X*错误',
    TABLES ct_fieldcat USING k_fn_r02 'EKNAM_I*T024*EKNAM* * *组件采购组*组件采购组*组件采购组*组件采购组',
    TABLES ct_fieldcat USING k_fn_r02 'BESKZ_I*MARC*BESKZ* * *组件采购类型*组件采购类型*组件采购类型*组件采购类型',
    TABLES ct_fieldcat USING k_fn_r02 'SOBSL_I*MARC*SOBSL* * *组件特殊采购类型*组件特殊采购类型*组件特殊采购类型*组件特殊采购类型',
    TABLES ct_fieldcat USING k_fn_sim 'IR_BZ_I*标准采购信息记录',
    TABLES ct_fieldcat USING k_fn_sim 'IR_WX_I*外协采购信息记录',
    TABLES ct_fieldcat USING k_fn_r02 'PLNNR*PLMZ*PLNNR',
    TABLES ct_fieldcat USING k_fn_r02 'VORNR*PLPO*VORNR',
****          add by song 20140531
    TABLES ct_fieldcat USING k_fn_sim 'LTEXT*长描述',
    TABLES ct_fieldcat USING k_fn_sim 'CJTXT*插件',
    TABLES ct_fieldcat USING k_fn_sim 'RGEKZ*反冲'

         .

ENDFORM.                    " FIELDCAT_INIT
FORM pre_fieldcat_oo  TABLES   ct_fcat     TYPE lvc_t_fcat
                  USING    i_flag.
  CONSTANTS:
    k_fn_s01(100) VALUE
      'FIELDNAME*COLTEXT*KEY*NO_ZERO',
    k_fn_r01(100) VALUE
      'FIELDNAME*REF_TABLE*REF_FIELD*KEY*NO_ZERO*COLTEXT',
*    K_FN_REF(100) VALUE
*      'FIELDNAME*REF_TABNAME*REF_FIELDNAME*KEY*NO_ZERO*NO_OUT*DO_SUM*OUTPUTLEN*EMPHASIZE',
    k_fn_q(100) VALUE
      'FIELDNAME*COLTEXT*NO_ZERO*QFIELDNAME*QUANTITY*OUTPUTLEN*EMPHASIZE',
    k_fn_c(100) VALUE
      'FIELDNAME*COLTEXT*NO_ZERO*CFIELDNAME*CURRENCY*OUTPUTLEN*EMPHASIZE'.
*---------------------------------------------------------------------*
  DATA: ls_fcat     TYPE LINE OF lvc_t_fcat.
  FIELD-SYMBOLS: <ls_fcat> TYPE lvc_s_fcat.
```

```
  REFRESH ct_fcat.

  CASE i_flag.
    WHEN '01'.
      PERFORM make_fieldcat_oo:
        TABLES ct_fcat USING k_fn_r01 'WERKS_H*CSTMAT*WERKS',
        TABLES ct_fcat USING k_fn_r01 'MATNR_H*CSTMAT*MATNR',
        TABLES ct_fcat USING k_fn_r01 'BISMT_H*MARA*BISMT',
        TABLES ct_fcat USING k_fn_r01 'MAKTX_H*CSTMAT*MAKTX',
        TABLES ct_fcat USING k_fn_r01 'POSIT*STPOX*POSIT* * *BOM层次',
        TABLES ct_fcat USING k_fn_r01 'STLAL_H*CSTMAT*STLAL* * *BOM版本',
        TABLES ct_fcat USING k_fn_r01 'STKTX_H*CSTMAT*STKTX',
        TABLES ct_fcat USING k_fn_r01 'EMMBM_H*CSTMAT*BMENG',
        TABLES ct_fcat USING k_fn_r01 'POSNR_I*STPOX*POSNR',
        TABLES ct_fcat USING k_fn_r01 'ALPGR_I*STPOX*ALPGR',
        TABLES ct_fcat USING k_fn_r01 'ALPRF_I*STPOX*ALPRF',
        TABLES ct_fcat USING k_fn_r01 'EWAHR_I*STPOX*EWAHR',
        TABLES ct_fcat USING k_fn_r01 'IDNRK_I*STPOX*IDNRK',
        TABLES ct_fcat USING k_fn_r01 'BISMT_I*MARA*BISMT* * *组件旧物料号',
        TABLES ct_fcat USING k_fn_r01 'OJTXP_I*STPOX*OJTXP* * *组件描述',
        TABLES ct_fcat USING k_fn_r01 'POSTP_I*STPOX*POSTP',
        TABLES ct_fcat USING k_fn_r01 'MNGKO_I*STPOX*MENGE',
        TABLES ct_fcat USING k_fn_r01 'AUSCH_I*STPO*AUSCH',
        TABLES ct_fcat USING k_fn_r01 'MEINS_I*MARA*MEINS',
        TABLES ct_fcat USING k_fn_r01 'KZAUS_I*STPOX*KZAUS',
        TABLES ct_fcat USING k_fn_r01 'AUSDT_I*STPOX*AUSDT',
        TABLES ct_fcat USING k_fn_r01 'NFMAT_I*STPOX*NFMAT',
*       tables CT_FCAT using K_FN_R01 'ZEINR*MARA*ZEINR',
        TABLES ct_fcat USING k_fn_r01 'POTX1_I*STPOX*POTX1',
        TABLES ct_fcat USING k_fn_r01 'POTX2_I*STPOX*POTX2',
        TABLES ct_fcat USING k_fn_s01 'POTX_STR*项目长文本*2000',
*       tables CT_FCAT using k_fn_r02 'EBORT*STPU*EBORT* * *物料位号*物料位号*物料位号*物料位号',
        TABLES ct_fcat USING k_fn_r01 'MATNE*MARA*MATNR* * *上级物料',
        TABLES ct_fcat USING k_fn_r01 'SANKA_I*STPOX*SANKA',
        TABLES ct_fcat USING k_fn_s01 'ERR_X*错误',
        TABLES ct_fcat USING k_fn_r01 'EKNAM_I*T024*EKNAM* * *组件采购组',
        TABLES ct_fcat USING k_fn_r01 'BESKZ_I*MARC*BESKZ* * *组件采购类型',
        TABLES ct_fcat USING k_fn_r01 'SOBSL_I*MARC*SOBSL* * *组件特殊采购类型',
        TABLES ct_fcat USING k_fn_s01 'IR_BZ_I*标准采购信息记录',
        TABLES ct_fcat USING k_fn_s01 'IR_WX_I*外协采购信息记录',
        TABLES ct_fcat USING k_fn_r01 'PLNNR*PLMZ*PLNNR',
        TABLES ct_fcat USING k_fn_r01 'VORNR*PLPO*VORNR'.
  ENDCASE.
ENDFORM.
FORM make_fieldcat_oo   TABLES ct_fieldcat TYPE lvc_t_fcat
                        USING fieldnames values.

  DATA: t_fieldnames(20) OCCURS 0 WITH HEADER LINE,
        t_values(40) OCCURS 0 WITH HEADER LINE.
  FIELD-SYMBOLS: <f>.
  DATA: str(60).
  DATA: ld_tabix TYPE sy-tabix.
  DATA: ls_fieldcat TYPE lvc_s_fcat.

  SPLIT fieldnames AT '*' INTO TABLE t_fieldnames.
  SPLIT values     AT '*' INTO TABLE t_values.

*  CLEAR: ls_fieldcat.

  LOOP AT t_fieldnames.
    ld_tabix = sy-tabix.

    TRANSLATE t_fieldnames TO UPPER CASE.

    CLEAR: str.
    CONCATENATE 'LS_FIELDCAT-' t_fieldnames INTO str.
    ASSIGN (str) TO <f>.
    IF sy-subrc EQ 0.
      READ TABLE t_values INDEX ld_tabix.
      IF sy-subrc EQ 0.
        <f> = t_values.
      ENDIF.
      UNASSIGN <f>.
    ENDIF.
  ENDLOOP.

  APPEND ls_fieldcat TO ct_fieldcat.

ENDFORM.
*&---------------------------------------------------------------------*
*&      Form  MAKE_FIELDCAT
```

```
*&---------------------------------------------------------------------*
*      text
*----------------------------------------------------------------------*
*     -->CT_FIELDCAT  text
*     -->FIELDNAMES   text
*     -->VALUES       text
*----------------------------------------------------------------------*
FORM make_fieldcat  TABLES ct_fieldcat TYPE slis_t_fieldcat_alv
                    USING fieldnames values.

  DATA: t_fieldnames(20) OCCURS 0 WITH HEADER LINE,
        t_values(40) OCCURS 0 WITH HEADER LINE.
  FIELD-SYMBOLS: <f>.
  DATA: str(60).
  DATA: ld_tabix TYPE sy-tabix.
  DATA: ls_fieldcat TYPE slis_fieldcat_alv.

  SPLIT fieldnames AT '*' INTO TABLE t_fieldnames.
  SPLIT values     AT '*' INTO TABLE t_values.

*  CLEAR: ls_fieldcat.

  LOOP AT t_fieldnames.
    ld_tabix = sy-tabix.

    TRANSLATE t_fieldnames TO UPPER CASE.

    CLEAR: str.
    CONCATENATE 'LS_FIELDCAT-' t_fieldnames INTO str.
    ASSIGN (str) TO <f>.
    IF sy-subrc EQ 0.
      READ TABLE t_values INDEX ld_tabix.
      IF sy-subrc EQ 0.
        <f> = t_values.
      ENDIF.
      UNASSIGN <f>.
    ENDIF.
  ENDLOOP.

  APPEND ls_fieldcat TO ct_fieldcat.

ENDFORM.                    "MAKE_FIELDCAT

*&---------------------------------------------------------------------*
*&      Form  GET_POSIT
*&---------------------------------------------------------------------*
*      text
*----------------------------------------------------------------------*
*     -->P_ITAB2_STUFE  text
*     <--P_ITAB2_POSIT  text
*----------------------------------------------------------------------*
FORM get_posit  USING    ud_matnr
                         ud_stufe
                         ud_new
                CHANGING cd_posit
                         cd_posit_sort.

  STATICS:t1 TYPE string VALUE 0,
          t2 TYPE string VALUE 0,
          t3 TYPE string VALUE 0,
          t4 TYPE string VALUE 0,
          t5 TYPE string VALUE 0,
          t6 TYPE string VALUE 0,
          t7 TYPE string VALUE 0,
          t8 TYPE string VALUE 0.
  DATA: ld_t1n TYPE n LENGTH 4,
          ld_t2n TYPE n LENGTH 4,
          ld_t3n TYPE n LENGTH 4,
          ld_t4n TYPE n LENGTH 4,
          ld_t5n TYPE n LENGTH 4,
          ld_t6n TYPE n LENGTH 4,
          ld_t7n TYPE n LENGTH 4,
          ld_t8n TYPE n LENGTH 4.

  STATICS:temp LIKE mara-matnr.
*      data:temp like mara-matnr.

  IF temp = space.
    temp = ud_matnr.
  ENDIF.
```

```
 IF temp = ud_matnr AND ud_new = space.

   IF ud_stufe = 1.
     t1 = t1 + 1. t2 = 0. t3 = 0. t4 = 0.t5 = 0. t6 = 0. t7 = 0.t8 = 0.
     cd_posit = t1.
*      MODIFY ITAB2.
   ELSEIF ud_stufe = 2.
     t2 = t2 + 1.t3 = 0. t4 = 0. t5 = 0. t6 = 0. t7 = 0.t8 = 0.
     CONDENSE:t1,t2.
     CONCATENATE t1 '.' t2 INTO cd_posit.
*      MODIFY ITAB2 .
   ELSEIF ud_stufe = 3.
     t3 = t3 + 1. t4 = 0. t5 = 0. t6 = 0. t7 = 0.t8 = 0.
     CONDENSE:t1,t2,t3.
     CONCATENATE t1 t2 t3 INTO cd_posit SEPARATED BY '.' .
*      MODIFY ITAB2.
   ELSEIF ud_stufe = 4.
     t4 = t4 + 1.t5 = 0. t6 = 0. t7 = 0.t8 = 0.
     CONDENSE:t1,t2,t3,t4.
     CONCATENATE t1 t2 t3 t4 INTO cd_posit SEPARATED BY '.'.
*      MODIFY ITAB2.
   ELSEIF ud_stufe = 5.
     t5 = t5 + 1.t6 = 0. t7 = 0.t8 = 0.
     CONDENSE:t1,t2,t3,t4,t5.
     CONCATENATE t1 t2 t3 t4 t5 INTO cd_posit SEPARATED BY '.'.
*      MODIFY ITAB2.
   ELSEIF ud_stufe = 6.
     t6 = t6 + 1.t7 = 0.t8 = 0.
     CONDENSE:t1,t2,t3,t4,t5,t6.
     CONCATENATE t1 t2 t3 t4 t5 t6 INTO cd_posit SEPARATED BY '.'.
*      MODIFY ITAB2.
   ELSEIF ud_stufe = 7.
     t7 = t7 + 1.t8 = 0.
     CONDENSE:t1,t2,t3,t4,t5,t6,t7.
     CONCATENATE t1 t2 t3 t4 t5 t6 t7 INTO cd_posit SEPARATED BY '.'.
*      MODIFY ITAB2.
   ELSEIF ud_stufe = 8.
     t8 = t8 + 1.
     CONDENSE:t1,t2,t3,t4,t5,t6,t7,t8.
     CONCATENATE t1 t2 t3 t4 t5 t6 t7 t8 INTO cd_posit SEPARATED BY '.'.
*      MODIFY ITAB2.
   ENDIF.

 ELSE.
   t1 = 0.t2 = 0. t3 = 0. t4 = 0.t5 = 0. t6 = 0. t7 = 0.t8 = 0.

   CASE ud_stufe.
     WHEN 1.
       t1 = t1 + 1. t2 = 0. t3 = 0. t4 = 0.t5 = 0. t6 = 0. t7 = 0.t8 = 0.
       cd_posit = t1.
*        MODIFY ITAB2.
     WHEN 2.
       t2 = t2 + 1.t3 = 0. t4 = 0. t5 = 0. t6 = 0. t7 = 0.t8 = 0.
       CONDENSE:t1,t2.
       CONCATENATE t1 '.' t2 INTO cd_posit.
*        MODIFY ITAB2 .
     WHEN 3.
       t3 = t3 + 1. t4 = 0. t5 = 0. t6 = 0. t7 = 0.t8 = 0.
       CONDENSE:t1,t2,t3.
       CONCATENATE t1 t2 t3 INTO cd_posit SEPARATED BY '.' .
*        MODIFY ITAB2.
     WHEN 4.
       t4 = t4 + 1.t5 = 0. t6 = 0. t7 = 0.t8 = 0.
       CONDENSE:t1,t2,t3,t4.
       CONCATENATE t1 t2 t3 t4 INTO cd_posit SEPARATED BY '.'.
*        MODIFY ITAB2.
     WHEN 5.
       t5 = t5 + 1.t6 = 0. t7 = 0.t8 = 0.
       CONDENSE:t1,t2,t3,t4,t5.
       CONCATENATE t1 t2 t3 t4 t5 INTO cd_posit SEPARATED BY '.'.
*        MODIFY ITAB2.
     WHEN 6.
       t6 = t6 + 1.t7 = 0.t8 = 0.
       CONDENSE:t1,t2,t3,t4,t5,t6.
       CONCATENATE t1 t2 t3 t4 t5 t6 INTO cd_posit SEPARATED BY '.'.
*        MODIFY ITAB2.
     WHEN 7.
       t7 = t7 + 1.t8 = 0.
       CONDENSE:t1,t2,t3,t4,t5,t6,t7.
       CONCATENATE t1 t2 t3 t4 t5 t6 t7 INTO cd_posit SEPARATED BY '.'.
*        MODIFY ITAB2.
```

```
      WHEN 8.
        t8 = t8 + 1.
        CONDENSE:t1,t2,t3,t4,t5,t6,t7,t8.
        CONCATENATE t1 t2 t3 t4 t5 t6 t7 t8 INTO cd_posit SEPARATED BY '.'.
*         MODIFY ITAB2.
    ENDCASE.

    temp = ud_matnr.
  ENDIF.

  ld_t1n = t1. ld_t2n = t2. ld_t3n = t3. ld_t4n = t4. ld_t5n = t5. ld_t6n = t6. ld_t7n = t7. ld_t8n = t8.
  CONCATENATE ld_t1n ld_t2n ld_t3n ld_t4n ld_t5n ld_t6n ld_t7n ld_t8n INTO cd_posit_sort SEPARATED BY '.'.
ENDFORM.                    " GET_POSIT
*&---------------------------------------------------------------------*
*&      Form  GET_EBORT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*      <--P_LS_DIS_EBORT  text
*----------------------------------------------------------------------*
FORM get_ebort    USING ud_stlnr ud_stlkn ud_stpoz
                CHANGING cd_cjtxt.

  DATA: BEGIN  OF  lt_UPTXT  OCCURS  0,
         UPTXT LIKE stpu-UPTXT,
        END  OF lt_UPTXT.
  DATA: ld_UPTXT TYPE c LENGTH 100.

  SELECT stpu~UPTXT
    INTO TABLE lt_UPTXT
    FROM stpu
    WHERE stpu~stlnr = ud_stlnr
      AND stpu~stlkn = ud_stlkn
      AND stpu~stpoz = ud_stpoz.

  LOOP AT lt_UPTXT.
    IF ld_UPTXT = space.
      ld_UPTXT = lt_UPTXT-UPTXT.
    ELSE.
      CONCATENATE ld_UPTXT lt_UPTXT-UPTXT INTO ld_UPTXT SEPARATED BY ','.
    ENDIF.
  ENDLOOP.

  cd_cjtxt = ld_UPTXT.

ENDFORM.                    " GET_EBORT
*&---------------------------------------------------------------------*
*&      Form  READ_TEXT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*      -->PV_ID       text
*      -->PV_LANGUAGE  text
*      -->PV_NAME      text
*      -->PV_OBJECT    text
*      -->PV_TEXT      text
*----------------------------------------------------------------------*
FORM read_text  USING    pv_id TYPE thead-tdid
                         pv_language TYPE thead-tdspras
                         pv_name TYPE thead-tdname
                         pv_object TYPE thead-tdobject
                CHANGING pv_text.
  DATA: lv_string TYPE string,
        lt_lines LIKE STANDARD TABLE OF tline,
        ls_lines LIKE LINE OF lt_lines.

  CALL FUNCTION 'READ_TEXT'
    EXPORTING
      id                  = pv_id                   "'R001'
      language              = pv_language "sy-langu
      name                = pv_name "lv_tdname
      object              = pv_object "'VBBK'
    TABLES
      lines               = lt_lines
    EXCEPTIONS
      id                  = 1
      language              = 2
      name                = 3
      not_found             = 4
      object              = 5
      reference_check       = 6
```

```
      wrong_access_to_archive = 7
      OTHERS                  = 8.
  LOOP AT lt_lines INTO ls_lines.
    CONCATENATE lv_string ls_lines-tdline INTO lv_string.
  ENDLOOP.
  pv_text = lv_string.
  FREE lt_lines.
ENDFORM.                    "READ_TEXT
*&---------------------------------------------------------------------*
*&      Form  DIS_BOMCOM
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*      -->P_LS_DIS  text
*----------------------------------------------------------------------*
FORM dis_bomcom  USING    ps_dis TYPE ts_dis.
  DATA: BEGIN OF csin.
        INCLUDE STRUCTURE csin.
  DATA: END OF csin.

  csin-datuv = ps_dis-datuv_h.
  csin-datub = ps_dis-datub_h.
  csin-emeng = 0 .
*d csin-idnrk = stb-idnrk.                          "note 149116
*del CSIN-MATNR = STB-MATNR.                        "YHG137469
  csin-matnr = ps_dis-matne. "MATCAT-MATNR.                    "YHG137469
  csin-stlal = ps_dis-stlal_i.
  csin-stlan = ps_dis-stlan_i.
  csin-stlkn = ps_dis-stlkn_i.
*d csin-stlty = typ_mat.                            "HGA024434
  csin-stlty = ps_dis-stlty_i.                       "HGA024434

*   IF CSIN-STLTY EQ 'K'.                          "HGA024434
*     CSIN-VBELN = PM_VBELN.                       "HGA024434
*     CSIN-VBPOS = PM_VBPOS.                       "HGA024434
*   ENDIF.                                         "HGA024434
*
*   IF CSIN-STLTY EQ TYP_PRJ.                      "HGA046836
*     CSIN-pspnr = PM_pspnr.                       "HGA046836
*   ENDIF.                                         "HGA046836

  csin-trtyp = 'A'.
*d csin-werks = pm_werks.
  csin-werks = ps_dis-werks_i. "MATCAT-PRWRK.                  "HGA027225
  csin-cmode = '01'.
  csin-stuez = ps_dis-stpoz_i.                       "YHG061577

  CALL DIALOG 'CS_BOM_DISPLAY'
    EXPORTING
      csin.
ENDFORM.                    " DIS_BOMCOM
*&---------------------------------------------------------------------*
*&      Form  PRO_DATA
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
FORM pro_data .
  DATA: ld_tabix TYPE sy-tabix,
        ls_dis LIKE LINE OF gt_dis.
  DATA: lt_dis_for_sel TYPE tt_dis.
  DATA: lt_marc TYPE STANDARD TABLE OF marc,
        ls_marc LIKE LINE OF lt_marc,
        lt_t024 TYPE STANDARD TABLE OF t024 WITH HEADER LINE.
  DATA: BEGIN OF ls_ir, "采购信息记录
          matnr TYPE eina-matnr,
          werks TYPE eine-werks,
          esokz TYPE eine-esokz,
          infnr TYPE eina-infnr,
        END OF ls_ir,
        lt_ir LIKE STANDARD TABLE OF ls_ir WITH KEY matnr werks esokz.

  "组件的额外信息
  lt_dis_for_sel[] = gt_dis[].
  DELETE lt_dis_for_sel WHERE werks_i IS INITIAL OR idnrk_i IS INITIAL.
  SORT lt_dis_for_sel BY werks_i idnrk_i.
  DELETE ADJACENT DUPLICATES FROM lt_dis_for_sel COMPARING werks_i idnrk_i.
  IF lt_dis_for_sel[] IS NOT INITIAL.
    SELECT matnr werks "key
```

```
          ekgrp beskz sobsl RGEKZ
      FROM marc INTO CORRESPONDING FIELDS OF TABLE lt_marc
      FOR ALL ENTRIES IN lt_dis_for_sel
      WHERE matnr = lt_dis_for_sel-idnrk_i
        AND werks = lt_dis_for_sel-werks_i.
  "采购信息记录
  SELECT matnr werks eine~esokz eina~infnr
      INTO TABLE lt_ir
      FROM eina
        JOIN eine ON eine~infnr = eina~infnr
      FOR ALL ENTRIES IN lt_dis_for_sel
      WHERE matnr = lt_dis_for_sel-idnrk_i
        AND werks = lt_dis_for_sel-werks_i
        AND eina~loekz = ''
        AND eine~loekz = ''.
  SORT lt_ir BY matnr werks esokz infnr.
 ENDIF.
 "采购组描述
 SELECT * FROM t024 INTO TABLE lt_t024.

 "集中读取组件分配的工序
 DATA: BEGIN OF ls_plmz,
       stlty TYPE plmz-stlty,
       stlnr TYPE plmz-stlnr,
       stlal TYPE plmz-stlal,
       stlkn TYPE plmz-stlkn,
       plnty TYPE plmz-plnty,
       plnnr TYPE plmz-plnnr,
       zuonr TYPE plmz-zuonr,
       zaehl TYPE plmz-zaehl,
       plnal TYPE plmz-plnal,
       plnfl TYPE plmz-plnfl,
       plnkn TYPE plmz-plnkn,
       vornr TYPE plpo-vornr,
      END OF ls_plmz,
      lt_plmz LIKE STANDARD TABLE OF ls_plmz.
 lt_dis_for_sel[] = gt_dis[].
 SORT lt_dis_for_sel BY stlty_i stlnr_i stlal_i stvkn_i. "stlkn_i.
 DELETE ADJACENT DUPLICATES FROM lt_dis_for_sel COMPARING stlty_i stlnr_i stlal_i stlkn_i.
 IF lt_dis_for_sel[] IS NOT INITIAL.
   SELECT a~stlty a~stlnr a~stlal a~stlkn
       a~plnty a~plnnr a~zuonr a~zaehl a~plnal a~plnfl a~plnkn
       c~vornr
      INTO CORRESPONDING FIELDS OF TABLE lt_plmz
      FROM plmz AS a
       JOIN plas AS b ON b~plnty = a~plnty AND b~plnnr = a~plnnr AND b~plnal = a~plnal
                 AND b~plnfl = a~plnfl AND b~plnkn = a~plnkn
       JOIN plpo AS c ON c~plnty = b~plnty AND c~plnnr = b~plnnr AND c~plnkn = b~plnkn
                 AND c~zaehl = b~zaehl
      FOR ALL ENTRIES IN lt_dis_for_sel
      WHERE a~stlty = lt_dis_for_sel-stlty_i
       AND a~stlnr = lt_dis_for_sel-stlnr_i
       AND a~stlal = lt_dis_for_sel-stlal_i
       AND a~stlkn = lt_dis_for_sel-stvkn_i.
   SORT lt_plmz BY stlty stlnr stlal stlkn.
 ENDIF.

*-
 LOOP AT gt_dis INTO ls_dis.
   ld_tabix = sy-tabix.
   "检查 成本核算相关字段
   IF ls_dis-idnrk_i+10(1) = '5'.
     IF ls_dis-sanka_i <> ''.
       ls_dis-err_x = 'X'.
     ENDIF.
   ELSE.
     IF ls_dis-sanka_i = ''.
       ls_dis-err_x = 'X'.
     ENDIF.
   ENDIF.
   "行颜色
   IF ls_dis-err_x = 'X'.
     ls_dis-color = 'C610'.
   ENDIF.

   "组件工厂视图
   READ TABLE lt_marc INTO ls_marc
      WITH KEY matnr = ls_dis-idnrk_i werks = ls_dis-werks_i.
   IF sy-subrc = 0.

     ls_dis-RGEKZ = ls_marc-RGEKZ." add by song 20140531
```

```
      ls_dis-ekgrp_i = ls_marc-ekgrp.
      READ TABLE lt_t024 WITH KEY ekgrp = ls_dis-ekgrp_i.
      IF sy-subrc = 0.
        ls_dis-eknam_i = lt_t024-eknam.
      ENDIF.
      ls_dis-beskz_i = ls_marc-beskz.
      ls_dis-sobsl_i = ls_marc-sobsl.
    ENDIF.

    "采购信息记录
    READ TABLE lt_ir TRANSPORTING NO FIELDS BINARY SEARCH
       WITH KEY matnr = ls_dis-idnrk_i werks = ls_dis-werks_i esokz = '0'.
    IF sy-subrc = 0.
      ls_dis-ir_bz_i = 'X'. "标准采购信息记录存在
    ENDIF.
    READ TABLE lt_ir TRANSPORTING NO FIELDS BINARY SEARCH
       WITH KEY matnr = ls_dis-idnrk_i werks = ls_dis-werks_i esokz = '3'.
    IF sy-subrc = 0.
      ls_dis-ir_wx_i = 'X'. "外协采购信息记录存在
    ENDIF.

    "组件分配的工序
    READ TABLE lt_plmz INTO ls_plmz BINARY SEARCH
       WITH KEY stlty = ls_dis-stlty_i stlnr = ls_dis-stlnr_i
            stlal = ls_dis-stlal_i stlkn = ls_dis-stvkn_i.
    IF sy-subrc = 0.
      ls_dis-plnty = ls_plmz-plnty.
      ls_dis-plnnr = ls_plmz-plnnr.
      ls_dis-plnal = ls_plmz-plnal.
      ls_dis-plnfl = ls_plmz-plnfl.
      ls_dis-plnkn = ls_plmz-plnkn.
      ls_dis-vornr = ls_plmz-vornr.
    ENDIF.


****    add by song 20140431
**   material long text
    DATA:l_id TYPE THEAD-TDID,
         l_name TYPE THEAD-TDNAME,
         l_object TYPE THEAD-TDOBJECT,
         lt_lines TYPE STANDARD TABLE OF TLINE,
         ls_lines TYPE TLINE.
    l_id = 'GRUN'.
    l_name = ls_dis-matnr_h.
    l_object = 'MATERIAL'.
    CALL FUNCTION 'READ_TEXT'
      EXPORTING
*       CLIENT                  = SY-MANDT
        id                      = l_id
        language                = sy-langu
        NAME                    = l_name
        OBJECT                  = l_object
*       ARCHIVE_HANDLE          = 0
*       LOCAL_CAT               = ' '
*     IMPORTING
*       HEADER                  =
*       OLD_LINE_COUNTER        =
      TABLES
        lines                   = lt_lines
      EXCEPTIONS
        ID                      = 1
        LANGUAGE                = 2
        NAME                    = 3
        NOT_FOUND               = 4
        OBJECT                  = 5
        REFERENCE_CHECK         = 6
        WRONG_ACCESS_TO_ARCHIVE = 7
        OTHERS                  = 8
            .
    IF lt_lines is NOT INITIAL.
      LOOP AT lt_lines INTO ls_lines.
        CONCATENATE ls_dis-ltext ls_lines-TDLINE INTO ls_dis-ltext.
      ENDLOOP.
    ENDIF.

    MODIFY gt_dis FROM ls_dis INDEX ld_tabix.
    CLEAR: ls_dis.
  ENDLOOP.

  "只显示错误的
  IF onlyerr = 'X'.
```

```
    DELETE gt_dis WHERE err_x IS INITIAL.
  ENDIF.
ENDFORM.                    " PRO_DATA

*Text elements
*----------------------------------------------------------
* BK1 查询条件
* BK2 输出格式
* BK3 层次选择
* BK4 位号选择
* BK5 显示过滤


*Selection texts
*----------------------------------------------------------
* ONLYERR        只显示成本核算设置错误
* P1        ALV输出
* P2        展开到底层
* P3        SMARTFORMS输出
* P4        单层展开
* P5        显示位号
* P6        不显示位号
* P7        导出到EXCEL
* P_LIST        List输出
* S_CAPID D       .
* S_MATNR D       .
* S_SL        展开数量
* S_STLAN D       .
* S_SXPQ        展开日期
* S_WERKS D       .


*Messages
*----------------------------------------------------------
*
* Message class: 00
*028   位字符串,只有\'0'或'1'被允许


*&---------------------------------------------------------------------*
*&      Form  DOWNLOAD_BACKGROUND
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
FORM download_background  TABLES it_data TYPE STANDARD TABLE
                            it_fieldcatalog TYPE lvc_t_fcat
                          USING is_layout TYPE lvc_s_layo.
**-  获取ALV的布局及字段目录
*  DATA: lo_alv TYPE REF TO cl_gui_alv_grid,
*        is_layout TYPE lvc_s_layo,
*        it_fieldcatalog TYPE lvc_t_fcat.
*
*  CALL FUNCTION 'GET_GLOBALS_FROM_SLVC_FULLSCR'
*    IMPORTING
*      e_grid = lo_alv.
*  CHECK lo_alv IS NOT INITIAL.
*  "布局
*  CALL METHOD lo_alv->get_frontend_layout
*    IMPORTING
*      es_layout = is_layout.
*  "字段目录
*  CALL METHOD lo_alv->get_frontend_fieldcatalog
*    IMPORTING
*      et_fieldcatalog = it_fieldcatalog.

*-  输出到服务器
  DATA: lo_result_data TYPE REF TO cl_salv_ex_result_data_table.
  DATA: lo_data TYPE REF TO data.
  DATA: l_flavour TYPE string.
  DATA: ls_xml_choice TYPE if_salv_bs_xml=>s_type_xml_choice.
  DATA: lt_xml_choice TYPE if_salv_bs_xml=>t_type_xml_choice.
  DATA: xml           TYPE xstring.
  FIELD-SYMBOLS: <lt_data> TYPE ANY TABLE.

  CREATE DATA lo_data LIKE it_data[].
  ASSIGN lo_data->* TO <lt_data>.
  <lt_data>[] = it_data[].
```

```
  lo_result_data = cl_salv_ex_util=>factory_result_data_table(
     r_data                      = lo_data
     s_layout                    = is_layout
     t_fieldcatalog              = it_fieldcatalog[] ).

  l_flavour = if_salv_bs_c_tt=>c_tt_xml_flavour_export.

  CLEAR ls_xml_choice.
  ls_xml_choice-text              = 'BOM'.   "'XXL-Export'
  ls_xml_choice-xslt_name         = ''.
  ls_xml_choice-key               = 29.
  ls_xml_choice-frontend          = cl_alv_bds=>mc_mhtml_frontend.
  ls_xml_choice-default_file_name = 'export.mhtml'. " 'export.xml'.        "#EC notext
  ls_xml_choice-initial_directory = 'C:'.
  ls_xml_choice-xml_type          = if_salv_bs_xml=>c_type_mhtml. "c_type_xxl.
  APPEND ls_xml_choice TO lt_xml_choice.

  CALL METHOD cl_salv_bs_tt_util=>if_salv_bs_tt_util~transform
    EXPORTING
      xml_type      = ls_xml_choice-xml_type
      xml_version   = if_salv_bs_xml=>version_25
      r_result_data = lo_result_data
      xml_flavour   = l_flavour
      gui_type      = if_salv_bs_xml=>c_gui_type_gui      "Y6DK066330
    IMPORTING
      xml           = xml.

  "后台执行时，保存到服务器指定路径
  CONSTANTS: ld_file_pre TYPE rlgrap-filename VALUE 'bom_'.
  DATA: ld_file TYPE rlgrap-filename.
  CONCATENATE ld_file_pre sy-datum '-' sy-uzeit '.xls' INTO ld_file.
  OPEN DATASET ld_file FOR OUTPUT IN BINARY MODE.
  IF sy-subrc = 0.
    TRANSFER xml TO ld_file.
    IF sy-subrc <> 0.
      WRITE: / 'Error'.
    ELSE.
      WRITE: / '成功输出文件', ld_file.
    ENDIF.
    CLOSE DATASET ld_file.
  ELSE.
    WRITE: / 'Open error'.
  ENDIF.

ENDFORM.                    " DOWNLOAD_BACKGROUND
*&---------------------------------------------------------------------*
*&      Form  DOWNLOAD_BACKGROUND_TXT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*      -->P_GT_DIS  text
*----------------------------------------------------------------------*
FORM download_background_txt  TABLES   it_dis TYPE tt_dis.
  DATA: ld_string TYPE string,
        ld_string_all TYPE string,
        ld_xstring TYPE xstring,
        ld_line_length TYPE i.
*  DATA: lt_fldname TYPE STANDARD TABLE OF string,
*        ls_fldname TYPE string.
  DATA: BEGIN OF ls_fldname,
          fld TYPE string,
          des TYPE string,
        END OF ls_fldname,
        lt_fldname LIKE STANDARD TABLE OF ls_fldname.
  DATA: ld_tabix TYPE sy-tabix,
        ls_dis LIKE LINE OF it_dis,
        ld_c1024 TYPE c LENGTH 1024.
  DATA: BEGIN OF ls_output,
          line TYPE string,
        END OF ls_output,
        lt_output LIKE STANDARD TABLE OF ls_output.
  FIELD-SYMBOLS: <fld> TYPE any.

  DEFINE append_fld.
    ls_fldname-fld = &1.
    ls_fldname-des = &2.
    append ls_fldname to lt_fldname.
  END-OF-DEFINITION.

  append_fld 'WERKS_H' '工厂'.
  append_fld 'MATNR_H' '物料号'.
```

```
  append_fld 'BISMT_H' '旧物料号'.
  append_fld 'MAKTX_H' '物料描述'.
  append_fld 'POSIT' 'BOM层次'.

  append_fld 'STLAL_H' 'BOM版本'.
  append_fld 'STKTX_H' '可选文本'.
  append_fld 'EMMBM_H' '基本数量'.
  append_fld 'POSNR_I' '项目'.
  append_fld 'ALPGR_I' '替代项目组'.

  append_fld 'ALPRF_I' '替代优先级'.
  append_fld 'EWAHR_I' '使用可能性'.
  append_fld 'IDNRK_I' '组件'.
  append_fld 'BISMT_I' '组件旧物料号'.
  append_fld 'OJTXP_I' '组件描述'.

  append_fld 'POSTP_I' '项目类别'.
  append_fld 'MNGKO_I' '数量'.
  append_fld 'AUSCH_I' '组件废品率'.
  append_fld 'MEINS_I' '单位'.
  append_fld 'KZAUS_I' '中断'.

  append_fld 'AUSDT_I' '中断日期'.
  append_fld 'NFMAT_I' '后续物料'.
  append_fld 'POTX1_I' '项目文本1'.
  append_fld 'POTX2_I' '项目文本2'.
  append_fld 'POTX_STR' '项目长文本'.

  append_fld 'MATNE' '上级物料'.
  append_fld 'SANKA_I' '成本核算'.
  append_fld 'ERR_X' '错误'.
  append_fld 'EKNAM_I' '组件采购组'.
  append_fld 'BESKZ_I' '组件采购类型'.

  append_fld 'SOBSL_I' '组件特殊采购类型'.
  append_fld 'IR_BZ_I' '标准采购信息记录'.
  append_fld 'IR_WX_I' '外协采购信息记录'.
  append_fld 'PLNNR' '工艺路线组号'.
  append_fld 'VORNR' '工序'.

  "字段描述
  LOOP AT lt_fldname INTO ls_fldname.
    ld_tabix = sy-tabix.
    ASSIGN COMPONENT ls_fldname-fld OF STRUCTURE ls_dis TO <fld>.
    CHECK sy-subrc = 0.
    ld_string = ls_fldname-des.
    IF ld_tabix = 1.
      ls_output-line = ld_string.
    ELSE.
      CONCATENATE ls_output-line cl_abap_char_utilities=>horizontal_tab ld_string INTO ls_output-line.
    ENDIF.
  ENDLOOP.
  CONCATENATE ld_string_all cl_abap_char_utilities=>cr_lf ls_output-line INTO ld_string_all.

  LOOP AT it_dis INTO ls_dis.
    CLEAR ls_output.
    LOOP AT lt_fldname INTO ls_fldname.
      ld_tabix = sy-tabix.
      ASSIGN COMPONENT ls_fldname-fld OF STRUCTURE ls_dis TO <fld>.
      CHECK sy-subrc = 0.
      WRITE <fld> TO ld_c1024.
      ld_string = ld_c1024.
      IF ld_tabix = 1.
        ls_output-line = ld_string.
      ELSE.
        CONCATENATE ls_output-line cl_abap_char_utilities=>horizontal_tab ld_string INTO ls_output-line.
      ENDIF.
    ENDLOOP.
    CONCATENATE ld_string_all cl_abap_char_utilities=>cr_lf ls_output-line INTO ld_string_all.
*   APPEND ls_output TO lt_output.
  ENDLOOP.

*  CALL FUNCTION 'SOTR_SERV_TABLE_TO_STRING'
*    EXPORTING
*      line_length = ld_line_length
*      langu       = sy-langu
*    IMPORTING
*      text        = ld_string
*    TABLES
*      text_tab    = it_dis[].
```

```
  CALL FUNCTION 'SCMS_STRING_TO_XSTRING'
    EXPORTING
      text   = ld_string_all
    IMPORTING
      buffer = ld_xstring
    EXCEPTIONS
      failed = 1
      OTHERS = 2.

  "后台执行时，保存到服务器指定路径
  CONSTANTS ld_file_pre TYPE rlgrap-filename VALUE 'bom_'.
  DATA: ld_file TYPE rlgrap-filename.
  CONCATENATE ld_file_pre sy-datum '-' sy-uzeit '.txt' INTO ld_file.
  OPEN DATASET ld_file FOR OUTPUT IN BINARY MODE.
  IF sy-subrc = 0.
    TRANSFER ld_xstring TO ld_file.
*    LOOP AT lt_output INTO ls_output.
*      TRANSFER ls_output-line TO ld_file.
*    ENDLOOP.
    IF sy-subrc <> 0.
      WRITE: / 'Error'.
    ELSE.
      WRITE: / '成功输出文件', ld_file.
    ENDIF.
    CLOSE DATASET ld_file.
  ELSE.
    WRITE: / 'Open error'.
  ENDIF.
ENDFORM.                    " DOWNLOAD_BACKGROUND_TXT
```