ZTPP_MD03 Single level – Multiple Item Functionality – An Enhancement to MRP

Applies to:

SAP 6.0 ECC SP4 EH3. For more information, visit the Manufacturing homepage.

Summary

In standard SAP we can either run MRP for Multilevel – Single item (MD02) or Single level-Single item (MD03). However, it is not possible to run MRP for Single level – Multiple items. This functionality is required in some industries due changes in master data after MRP run. My article provides complete solution to such scenario.

Author: Krishna Prasad Nanduri

Company: Intelligroup

Created on: 14 September 2010

Author Bio

Mr. Krishna Prasad Nanduri has around fifteen years of work experience which includes around five years of experience in SAP specialized in different modules and ten years of domain experience as Operations Manager. He successfully completed full life cycle implementation, roll outs and Support projects in the areas of PP, MM & QM modules. He has implementation and roll outs exposure in Transmission devices & belts and Petroleum Verticals. His core strengths include establishing Business processes with in built controls and standardization and integration of Supply chain across group companies. He was involved in SAP logistics implementation, accessing requirements, mapping business processes, implementing SAP systems, and conducting all phases of testing using ASAP methodology. He gained hands on experience in requirements gathering and analysis to project planning, solution deployment, and support. His PP expertise includes Discrete, Repetitive and Process industries skills and development of Project based production where the production takes place with respect to project orders (MTO), Creation of New Production Order types, and Assigning Order types to Inventory Management and handling different Lot Sizing procedures. He possess good MM experience in configuration of SAP R/3 processes to support Business functions such as Sub-contracting, Third party, Consignment, Stock transport order, vendor Evaluation, Logistics Invoice Verification, Release procedure and pricing procedures. He is experienced in Creation of New Material types, Project based stocks maintenance, Creation of New Movement types, SLED based Inventory Management, Quality based inventory management and Transfer order of Warehouse management.

Table of Contents

MRP run for Single level – Multiple items 3
Layout and Extract 3
ABAP Program logic: 4
ABAP Program 5
Related Content 16
Disclaimer and Liability Notice 17

MRP run for Single level – Multiple items

Whenever some changes took place in master data of some materials after MRP run, it is necessary to run MRP again for those materials at single level. This is most common, in case of Make-to-order scenario as the customer may change his requirements after placing the order. This scenario can also be experienced in Projects as some materials may undergo master data changes during the project execution. Such situation leads to re-run of MRP at single level for multiple items. To achieve this we need to identify the materials and re-run the MRP for every material that underwent master data change.

For those users, if such list of materials is huge, identifying and running MRP for every item becomes a tedious job. Keeping the difficulties in re-running the MRP for every material, we have developed a functionality which can upload a list of materials that needs to undergo MRP run for second time at single level-multiple items.

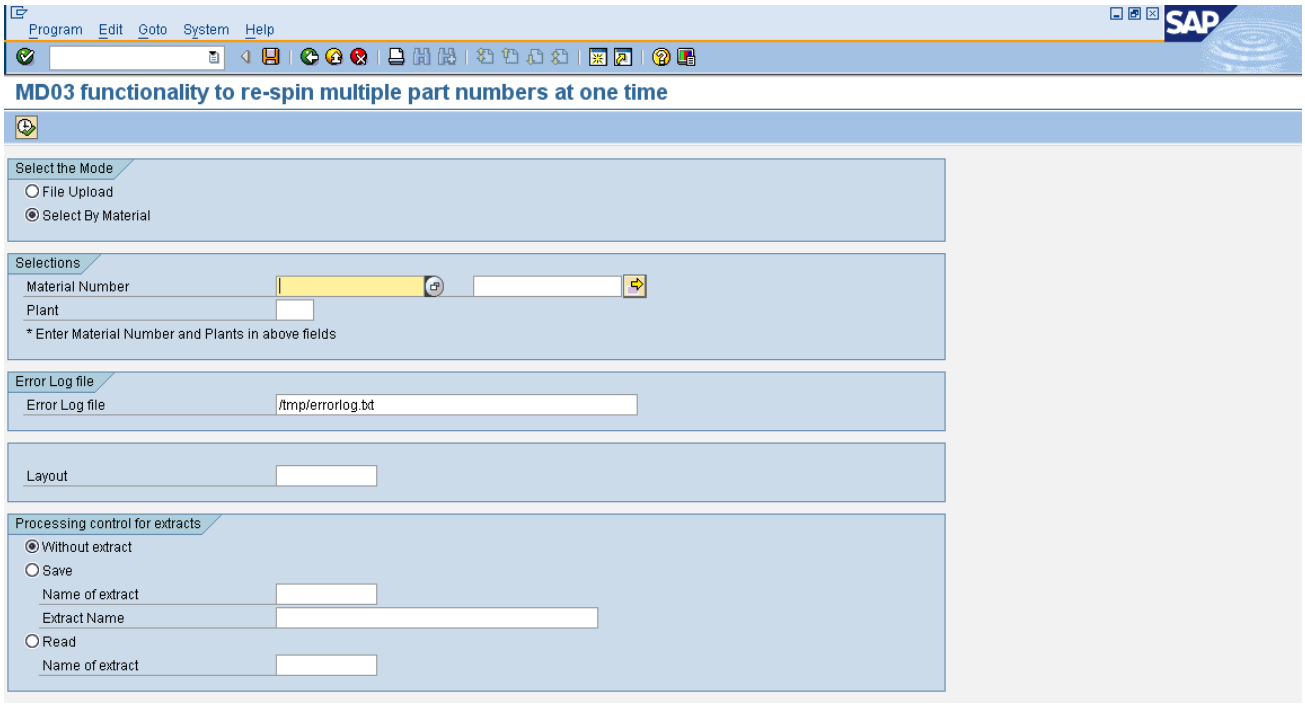
The list of materials can be uploaded in two different ways

- From a file Upload
- Selection by material

Layout and Extract

The results of the MRP re-run can be displayed in a layout pre-defined so as to facilitate the users to see what changes were taken place before and after MRP run. It is also possible to store the planning file as an extract and can be stored and viewed any time.

The error log will be generated in case the MRP run fails for some materials.



Note: Screen shot showing the details of Single level – Multiple item MRP run.

ABAP Program logic:

The SAP standard program for MD03 which is used for single item, single level is basically used. The program is copied to Z program.

Input Screen:

The input data screen is enhanced to enter the list of materials that needs to undergo MRP run.

The input screen also provides facility to upload the list of material either from a file or a range of materials from mara table and one can save the extract.

The Input data screen is also equipped with error logging facility and the same can be save in case the MRP run fails for certain materials.

Main Program logic:

After validating the data of the input screen the program calls the BDC of MD03 and reads the material as usual. To perform multi spin, a loop is created to until the MRP run for each material is completed.

Of course, the complete logic can be seen in the program that is attached here.

Out Put Screen :

The out put screen layout is defined in the program to see the list of materials for every material that underwent MRP run. It is also possible to change the layout as per user requirements.

ABAP Program

The following code is developed for the above mentioned functionality.

```
************************************************************************

*  Program ID        :  ZINPP_MD03_MUTIPLE_PART_NUM                     *
*  Program Title     :  MD03 functionality to re-spin multiple part     *
*                       numbers at one time                             *
*  Created By        :  RamanjiReddy Gatla                              *
*  Creation Date     :  10.24.2009                                      *
*  CRF#/Requ#        :  KRDK979769/Task : 190025
*  Description       :  Currently in MD03 we could run only one part at*
*                       a time which inhibits planning of certain parts*
*                       in between MRP runs.It is essential for KT to  *
*                       have a functionality developed to re-spin      *
*                       multiple parts in MD03 at one entry, else its
*                       time lost in having to input one part number   *
*                       and wait for the transaction to complete in    *
*                       order to start the next part number.           *
*
*  Additional Information:                                             *
************************************************************************
*                      Modification History                            *
************************************************************************

REPORT  zinpp_md03_mutiple_part_num  NO STANDARD PAGE HEADING
        MESSAGE-ID zcus.
*---------------------------------------------------------------------*
*  T A B L E S
*---------------------------------------------------------------------*
TABLES : mara,   "General Material Data
         marc,   "Plant Data for Material
         t001w,  "Plants
         mdkp.

************************************************************************
*               T Y P E     D E C L A R A T I O N S                   *
************************************************************************
TYPES : BEGIN OF ty_mara,
        matnr TYPE mara-matnr,
        werks TYPE marc-werks,
        END OF ty_mara.
************************************************************************
*               I n t e r n a l    T a b l e s                        *
************************************************************************
*Internal table for Material
DATA: it_matnr TYPE TABLE OF ty_mara.

*work Area of above internal table
DATA: wa_matnr LIKE LINE OF it_matnr.

*Internal table for BDC data
DATA :it_bdcdata       LIKE bdcdata    OCCURS 10 WITH HEADER LINE.

*Internal tabel of hold the messages
DATA: it_msgtab        LIKE bdcmsgcoll OCCURS 10 WITH HEADER LINE.
```

```
*Internal table for Error records
DATA : BEGIN OF it_error OCCURS 0,
        matnr TYPE mara-matnr,
        werks TYPE marc-werks,
        msg(50) TYPE c,
      END OF it_error.

*Internal table for Success records
DATA : BEGIN OF it_success OCCURS 0,
        matnr TYPE mara-matnr,
        werks TYPE marc-werks,
        msg(50) TYPE c,
      END OF it_success.

* Work area for Success records
DATA : wa_success LIKE LINE OF it_success.
* Work Area for Error records
DATA : wa_error LIKE LINE OF it_error.

DATA : BEGIN OF it_alv_data OCCURS 0,
       msg_ind TYPE c,
       matnr   LIKE mara-matnr,
       werks   LIKE marc-werks,
       msg(50) TYPE c,
       END OF it_alv_data.

*********************************************************************
*        C o n s t a n t s     D e c l a r a t i o n                *
*********************************************************************
CONSTANTS:
        c_check(1) TYPE c VALUE 'X',
        c_tab  TYPE c VALUE cl_abap_char_utilities=>horizontal_tab,
         c_xflag_flg      TYPE c VALUE 'X'.


*********************************************************************
*       V a r i a b l e s     D e c l a r a t i o n                 *
*********************************************************************
DATA      w_alv_table_name LIKE dd02d-strname VALUE 'it_alv_data'.
DATA      w_msg(80)        TYPE c.

DATA :w_error LIKE sy-tfill,
      w_success LIKE sy-tfill.

*********************************************************************
*  S e l e c t i o n     S c r e e n                                  *
*********************************************************************

SELECTION-SCREEN: BEGIN OF BLOCK b1 WITH FRAME TITLE text-t02.
PARAMETERS : p_upload RADIOBUTTON GROUP r1 USER-COMMAND u1 ,
             p_sel  RADIOBUTTON GROUP r1 DEFAULT 'X'.
SELECTION-SCREEN: END OF BLOCK b1.

SELECTION-SCREEN : BEGIN OF BLOCK b2 WITH FRAME TITLE text-t03.
PARAMETERS : p_file TYPE localfile MODIF ID zup.
```

```
SELECTION-SCREEN : END OF BLOCK b2.

SELECTION-SCREEN: BEGIN OF BLOCK b3 WITH FRAME TITLE text-t01.
SELECT-OPTIONS : s_matnr FOR  mara-matnr  MODIF ID zse.
PARAMETERS     : p_werks TYPE marc-werks  MODIF ID zse.
SELECTION-SCREEN : COMMENT /1(50) text-002 MODIF ID zse.
SELECTION-SCREEN : END OF BLOCK b3.

SELECTION-SCREEN : BEGIN OF BLOCK b4 WITH FRAME TITLE text-t06.
PARAMETERS: p_err LIKE  rlgrap-filename MODIF ID zap
                              OBLIGATORY DEFAULT '/tmp/errorlog.txt'.
SELECTION-SCREEN : END OF BLOCK b4.


************************************************************************
*     S t a n d a r d      I n c l u d e                               *
************************************************************************
* Include that Contains ALV Related Forms
INCLUDE zalv_reuse_forms_2.
************************************************************************
*     I n i t i a l i z a t i o n                                      *
************************************************************************
INITIALIZATION.
* Initializing Variant
  PERFORM initialize_variant.
  PERFORM extract_initialize.

************************************************************************
*              A T  S E L E C T I O N   S C R E E N                    *
************************************************************************
AT SELECTION-SCREEN ON VALUE-REQUEST FOR p_zex1.
  PERFORM f4_for_save_extract.

AT SELECTION-SCREEN ON VALUE-REQUEST FOR p_zex2.
  PERFORM f4_for_existing_extract.

AT SELECTION-SCREEN ON VALUE-REQUEST FOR p_vari.
  PERFORM f4_for_variant.

AT SELECTION-SCREEN ON VALUE-REQUEST FOR p_file.
  PERFORM f4_filename USING p_file.

AT SELECTION-SCREEN.
  PERFORM pai_of_selection_screen.
  PERFORM user_action.
  PERFORM check_extracts_count.
  PERFORM validation_selection_screen.

AT SELECTION-SCREEN OUTPUT.
  PERFORM mode_select.

************************************************************************
*     S t a r t    O f    S e l e c t i o n                            *
************************************************************************
START-OF-SELECTION.
```

```
  IF p_upload = 'X'.

* Read data from Excel file.
    PERFORM upload_excel_data.
* call BDC of MD03.
    IF NOT it_matnr[] IS INITIAL.
      PERFORM bdc_md03.
    ENDIF.
  ELSE.
* Get data from tables
    PERFORM get_data.
* call BDC of MD03.
    IF NOT it_matnr[] IS INITIAL.
      PERFORM bdc_md03.
    ENDIF.

  ENDIF.
  PERFORM error_log.
  PERFORM display_data.

*&---------------------------------------------------------------------*
*    E n d    O f    S e l e c t i o n                                 *
*&---------------------------------------------------------------------*
* Display the report.
  PERFORM alv_common_stuff.

*&---------------------------------------------------------------------*
*&      Form  VALIDATE_SELECTION_SCREEN
*&---------------------------------------------------------------------*
*&---------------------------------------------------------------------*
*&      Form  F4_FILENAME
*&---------------------------------------------------------------------*

FORM f4_filename  USING    p_file.

  CALL FUNCTION 'F4_FILENAME'
    EXPORTING
      program_name  = syst-cprog
      dynpro_number = syst-dynnr
    IMPORTING
      file_name     = p_file.

ENDFORM.                    " F4_FILENAME
*&---------------------------------------------------------------------*
*&      Form  MODE_SELECT
*&---------------------------------------------------------------------*
FORM mode_select .

  LOOP AT SCREEN.
    CASE screen-group1.
      WHEN 'ZUP'.
        IF p_upload  = 'X'.
          screen-active = '1'.
        ELSE.
          screen-active = '0'.
        ENDIF.
```

```
      WHEN 'ZSE'.
        IF p_sel = 'X'.
          screen-active = '1'.
        ELSE.
          screen-active = '0'.
        ENDIF.
    ENDCASE.
    MODIFY SCREEN.
  ENDLOOP.
ENDFORM.                    " MODE_SELECT

*&---------------------------------------------------------------------*
*&      Form  VALIDATION_SELECTION_SCREEN
*&---------------------------------------------------------------------*
FORM validation_selection_screen .
  IF NOT s_matnr[] IS  INITIAL.
    SELECT SINGLE * FROM mara
           WHERE matnr IN s_matnr.
    IF sy-subrc NE 0.
      MESSAGE e000 WITH text-t04.
    ENDIF.
  ENDIF.

  IF NOT p_werks IS INITIAL.
    SELECT SINGLE * FROM t001w
         WHERE werks EQ p_werks.
    IF sy-subrc NE 0.
      MESSAGE e000 WITH text-t05 .
    ENDIF.
  ENDIF.
ENDFORM.                    " VALIDATION_SELECTION_SCREEN
*&---------------------------------------------------------------------*
*&      Form  UPLOAD_EXCEL_DATA
*&---------------------------------------------------------------------*
*       read excel data
*----------------------------------------------------------------------*
FORM upload_excel_data .
  DATA  : lit_temp    LIKE STANDARD TABLE OF alsmex_tabline.
  DATA  : lit_temps    LIKE alsmex_tabline.
  DATA  : lw_col      TYPE i.
  DATA: lwa_infile TYPE localfile.
  FIELD-SYMBOLS : <fs> TYPE ANY.
  lwa_infile = p_file.

  CALL FUNCTION 'ALSM_EXCEL_TO_INTERNAL_TABLE'
    EXPORTING
      filename                = p_file
      i_begin_col             = 1
      i_begin_row             = 2
      i_end_col               = 255
      i_end_row               = 65535
    TABLES
      intern                  = lit_temp
    EXCEPTIONS
      inconsistent_parameters = 1
      upload_ole              = 2
```

```
      OTHERS                  = 3.                            "#EC *
  IF sy-subrc <> 0.
  ENDIF.
  SORT lit_temp  BY row col.

  LOOP AT lit_temp INTO lit_temps.
    MOVE lit_temps-col TO lw_col.
    ASSIGN COMPONENT lw_col OF STRUCTURE wa_matnr TO <fs>.
    IF sy-subrc EQ 0.
      MOVE lit_temps-value TO <fs>.
    ENDIF.
    AT END OF row.
      APPEND wa_matnr TO it_matnr.
      CLEAR wa_matnr.
    ENDAT.
  ENDLOOP.
ENDFORM.                    " UPLOAD_EXCEL_DATA
*&---------------------------------------------------------------------*
*&      Form  BDC_MD03
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
FORM bdc_md03 .

  DATA : lw_date(10) TYPE c.
  WRITE sy-datum TO lw_date.

  LOOP AT it_matnr INTO wa_matnr.

    PERFORM bdc_dynpro      USING 'SAPMM61X' '0200'.
    PERFORM bdc_field       USING 'BDC_CURSOR'
                                  'RM61X-MATNR'.
    PERFORM bdc_field       USING 'BDC_OKCODE'
                                  '/00'.
    PERFORM bdc_field       USING 'RM61X-MATNR'
                                  wa_matnr-matnr.
    PERFORM bdc_field       USING 'RM61X-WERKS'
                                  wa_matnr-werks.
    PERFORM bdc_field       USING 'RM61X-VERSL'
                                  'NETCH'.
    PERFORM bdc_field       USING 'RM61X-BANER'
                                  '2'.
    PERFORM bdc_field       USING 'RM61X-LIFKZ'
                                  '3'.
    PERFORM bdc_field       USING 'RM61X-DISER'
                                  '1'.
    PERFORM bdc_field       USING 'RM61X-PLMOD'
                                  '1'.
    PERFORM bdc_field       USING 'RM61X-TRMPL'
                                  '1'.
    PERFORM bdc_field       USING 'RM61X-DISPD'
                                  lw_date.
    PERFORM call_transaction.
```

```
  ENDLOOP.
ENDFORM.                                                    " BDC_MD03
*&---------------------------------------------------------------------*
*       BDC Field
*---------------------------------------------------------------------*
FORM bdc_field  USING   pr_fnam LIKE bdcdata-fnam
                        pr_fval TYPE any.

  CLEAR it_bdcdata.
  it_bdcdata-fnam = pr_fnam.
  it_bdcdata-fval = pr_fval.
  APPEND it_bdcdata.

ENDFORM.                    " BDC_FIELD

*&---------------------------------------------------------------------*
*&      Form  BDC_DYNPRO
*&---------------------------------------------------------------------*
*       BDC Dynpro
*---------------------------------------------------------------------*
FORM bdc_dynpro  USING   p_program LIKE bdcdata-program
                         p_dynpro  LIKE bdcdata-dynpro.

  CLEAR it_bdcdata.
  it_bdcdata-program  = p_program.
  it_bdcdata-dynpro   = p_dynpro.
  it_bdcdata-dynbegin = 'X'.
  APPEND it_bdcdata.

ENDFORM.                    " BDC_DYNPRO
*&---------------------------------------------------------------------*
*&      Form  CALL_TRANSACTION
*&---------------------------------------------------------------------*
FORM call_transaction .

  CALL TRANSACTION 'MD03' USING it_bdcdata
                          MODE   'N'
                          UPDATE 'S'
                          MESSAGES INTO it_msgtab.
  IF sy-subrc = 0.
    CLEAR it_success.
    CONCATENATE 'MRP carried out for'(001)
                wa_matnr-matnr '  ' wa_matnr-werks
                        INTO it_success-msg.

    MOVE-CORRESPONDING wa_matnr TO it_success.
    APPEND it_success.
    CLEAR wa_matnr.
  ELSE.
    LOOP AT it_msgtab.   " WHERE msgtyp = 'E'.

      CALL FUNCTION 'FORMAT_MESSAGE'
       EXPORTING
         id              = it_msgtab-msgid
         lang            = sy-langu
         no              = it_msgtab-msgnr
```

```
          v1              = it_msgtab-msgv1
          v2              = it_msgtab-msgv2
          v3              = it_msgtab-msgv3
          v4              = it_msgtab-msgv4
        IMPORTING
          msg             = w_msg
*         EXCEPTIONS
*           NOT_FOUND       = 1
*           OTHERS          = 2
                  .
      IF sy-subrc <> 0.
* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
*         WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
      ENDIF.
      wa_error = wa_matnr.

      wa_error-msg = w_msg.
      APPEND wa_error TO it_error.
      CLEAR wa_error.
    ENDLOOP.

  ENDIF.
  CLEAR: it_bdcdata,it_msgtab,wa_matnr.
  REFRESH: it_bdcdata,it_msgtab.

ENDFORM.                    " CALL_TRANSACTION
*&---------------------------------------------------------------------*
*&      Form  GET_DATA
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
FORM get_data .
  CLEAR : it_matnr[],it_matnr.
  PERFORM get_material_data.
ENDFORM.                    " GET_DATA

*&---------------------------------------------------------------------*
*&      Form  ERROR_LOG
*&---------------------------------------------------------------------*
FORM error_log .
  DATA : lw_fname LIKE rlgrap-filename,
         lw_string TYPE string.

  lw_fname = p_err.
  TRANSLATE lw_fname TO LOWER CASE.
  OPEN DATASET lw_fname FOR OUTPUT IN TEXT MODE ENCODING DEFAULT.
  IF sy-subrc NE 0.
    MESSAGE e000 WITH 'Error opening file'(004) space space space.
  ELSE.
    LOOP AT it_error.
      CONCATENATE it_error-matnr
                  it_error-werks
                  it_error-msg
                  INTO lw_string SEPARATED BY c_tab.
      TRANSFER lw_string TO lw_fname.
    ENDLOOP.
```

```
    CLOSE DATASET lw_fname.
  ENDIF.

ENDFORM.                    " ERROR_LOG

*
*&---------------------------------------------------------------------*
*&      Form  GET_MATERIAL_DATA
*&---------------------------------------------------------------------*
FORM get_material_data .

  SELECT matnr werks
         FROM marc
         INTO TABLE it_matnr
        WHERE matnr IN s_matnr
       AND    werks EQ p_werks.
  IF sy-subrc EQ 0.
    SORT it_matnr BY matnr werks.
  ENDIF.

ENDFORM.                    " GET_MATERIAL_DATA
*&---------------------------------------------------------------------*
*&      Form  ALV_COMMON_STUFF
*&---------------------------------------------------------------------*

FORM alv_common_stuff .
  DATA: lw_line           TYPE slis_listheader.
  w_msg = 'Preparing ALV Display...'(010).
*  w_alv_table_name = 'it_alv_data'.
  PERFORM issue_progress_msg         USING w_msg.
  PERFORM initialize_alv_variables USING w_alv_table_name  space
                                         'COLOR'           space space
                                         c_xflag_flg.
*  W_STRUCTURE = 'IT_ALV_DATA'.
  PERFORM build_events             USING c_xflag_flg c_xflag_flg.
  DESCRIBE TABLE it_alv_data LINES w_total_records.
  PERFORM build_comments_top.
*Populating Top of Page Event
  MOVE 'S'   TO lw_line-typ.
  WRITE 'Success Records' TO lw_line-key.
  WRITE w_success TO lw_line-info.
  APPEND lw_line TO  it_list_top_of_page.

  WRITE 'Error Records' TO lw_line-key.
  WRITE w_error TO lw_line-info.
  APPEND lw_line TO  it_list_top_of_page.
  PERFORM field_cat_build.
  PERFORM modify_catalog.
  PERFORM call_alv_fm_user_status.
ENDFORM.                    " ALV_COMMON_STUFF
*&---------------------------------------------------------------------*
*&      Form  MODIFY_CATALOG
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
```

```
*  <--  p2        text
*---------------------------------------------------------------------*
FORM modify_catalog .
  DATA lw_fieldcat TYPE slis_fieldcat_alv.
  LOOP AT it_fieldcat INTO lw_fieldcat.
    CASE lw_fieldcat-fieldname.

      WHEN 'MSG_IND'.
        lw_fieldcat-seltext_s  = 'Msg Type'(011).
        lw_fieldcat-seltext_m  = 'Message Type'(012).
        lw_fieldcat-seltext_l  = 'Message Type'(012).
        lw_fieldcat-reptext_ddic = 'Message Type'(012).
        lw_fieldcat-ddictxt    = 'L'.
        lw_fieldcat-key       =  space.

      WHEN 'MATNR'.
        lw_fieldcat-seltext_s  = 'Mat num'(013).
        lw_fieldcat-seltext_m  = 'Material Number'(014).
        lw_fieldcat-seltext_l  = 'Material Number'(014).
        lw_fieldcat-reptext_ddic = 'Material Number'(014).
        lw_fieldcat-ddictxt    = 'L'.
        lw_fieldcat-key       =  space.

      WHEN 'WERKS'.
        lw_fieldcat-seltext_s  = 'Plant'(015).
        lw_fieldcat-seltext_m  = 'Plant'(015).
        lw_fieldcat-seltext_l  = 'Plant'(015).
        lw_fieldcat-reptext_ddic = 'Plant'(015).
        lw_fieldcat-ddictxt    = 'L'.
        lw_fieldcat-key       =  space.

      WHEN 'MSG'.
        lw_fieldcat-seltext_s  = 'Message'(016).
        lw_fieldcat-seltext_m  = 'Message'(016).
        lw_fieldcat-seltext_l  = 'Message'(016).
        lw_fieldcat-reptext_ddic = 'Message'(016).
        lw_fieldcat-ddictxt    = 'L'.
        lw_fieldcat-key       =  space.
    ENDCASE.
    MODIFY it_fieldcat FROM lw_fieldcat.
  ENDLOOP.

ENDFORM.                    " MODIFY_CATALOG
*&---------------------------------------------------------------------*
*       EXIT routine for user_command
*----------------------------------------------------------------------*
*      -->R_UCOMM      Checks for sy-ucomm for &IC1
*      -->RS_SELFIELD  structure to store selected row
*----------------------------------------------------------------------*
FORM user_command USING p_ucomm     LIKE sy-ucomm
                        p_selfield TYPE slis_selfield.      "#EC *
ENDFORM.                    "user_command
*&---------------------------------------------------------------------*
*&      Form  DISPLAY_DATA
*&---------------------------------------------------------------------*
FORM display_data .
```

```
  CLEAR : it_alv_data,it_alv_data[].
  DESCRIBE TABLE it_error LINES w_error.
  DESCRIBE TABLE it_success LINES w_success.
*Error records
  LOOP AT it_error INTO wa_error.
    it_alv_data-msg_ind = 'E'.
    it_alv_data-matnr   = wa_error-matnr.
    it_alv_data-werks   = wa_error-werks.
    it_alv_data-msg     = wa_error-msg.
    APPEND it_alv_data.
    CLEAR:  it_alv_data, wa_error.
  ENDLOOP.

*Success records

  LOOP AT it_success INTO wa_success.
    it_alv_data-msg_ind = 'S'.
    it_alv_data-matnr   = wa_success-matnr.
    it_alv_data-werks   = wa_success-werks.
    it_alv_data-msg     = wa_success-msg.
    APPEND it_alv_data.
    CLEAR:  it_alv_data, wa_success.
  ENDLOOP.

ENDFORM.                    " DISPLAY_DATA
```

Related Content

MRP Run Process

MRP Issue

MRP area

For more information, visit the Manufacturing homepage.

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.