

REPORT ZSD00301

MESSAGE-ID Z3.

* TABLES *

TABLES: LIKP, " Delivery Note Header

LIPS, " Delivery Note Item

LIPSD, " Dynamic part of delivery item

* MCHA, " Batch Table

MCH1, " Batch Table 0004

VBUP, " Sales Doc. Item

MKPF, " Material Doc. Header

MSEG, " Material Segment

T9CON, " Mapping storage location " + BENEVIAN

MVKE. " Material Master: Sales Data "+005 BENEVIAN

* PARAMETERS *

SELECTION-SCREEN: SKIP.

SELECTION-SCREEN BEGIN OF BLOCK PART1

WITH FRAME TITLE TEXT-001.

PARAMETERS: P_VBELN LIKE LIKP-VBELN OBLIGATORY, " Delivery Note No.

P_LIFEX LIKE LIKP-LIFEX OBLIGATORY, " Trailer No.

P_DDATE LIKE SY-DATUM DEFAULT SY-DATUM.

SELECTION-SCREEN END OF BLOCK PART1.

SELECTION-SCREEN : SKIP.

* DATA *

* Internal Table populated if a duty movement from duty suspended to

* duty paid is required.

DATA: BEGIN OF DUTY OCCURS 100,

VBELN LIKE LIKP-VBELN,

POSNR LIKE LIPS-POSNR,

MATNR LIKE LIPS-MATNR,

LFIMG(10) TYPE C,

WERKS LIKE LIPS-WERKS,

LGORT LIKE LIPS-LGORT,

CHARG LIKE LIPS-CHARG,

BATCHEXISTS TYPE C,

LGORT_P LIKE LIPS-LGORT, " + BENEVIAN

NOT_01_GROUP(1) TYPE C. " +005 BENEVIAN

DATA: END OF DUTY.

* Internal Table for delivery items that are batch split

DATA: BEGIN OF SUB OCCURS 100,

UECHA LIKE LIPS-UECHA,

POSNR LIKE LIPS-POSNR,

MATNR LIKE LIPS-MATNR,

LFIMG(10) TYPE C,

WERKS LIKE LIPS-WERKS,

LGORT LIKE LIPS-LGORT,

CHARG LIKE LIPS-CHARG,

BATCHEXISTS TYPE C,

LGORT_P LIKE LIPS-LGORT, " + BENEVIAN
NOT_01_GROUP(1) TYPE C, " +005 BENEVIAN
MIX(1) TYPE C. " +006 BENEVIAN

DATA: END OF SUB.

DATA: BEGIN OF TSUB OCCURS 100,

UECHA LIKE LIPS-UECHA,
POSNR LIKE LIPS-POSNR,
MATNR LIKE LIPS-MATNR,
LFIMG(10) TYPE C,
WERKS LIKE LIPS-WERKS,
LGORT LIKE LIPS-LGORT,
CHARG LIKE LIPS-CHARG,
BATCHEXISTS TYPE C,
LGORT_P LIKE LIPS-LGORT. " + BENEVIAN

DATA: END OF TSUB.

*data: begin of tmkpf occurs 2.

* include structure mkpf.

*data: end of tmkpf.

* BDC Tables

DATA: BEGIN OF BDC_TABLE OCCURS 500.

INCLUDE STRUCTURE BDCDATA.

DATA: END OF BDC_TABLE.

DATA: BEGIN OF BDC_TABLE2 OCCURS 500.

INCLUDE STRUCTURE BDCDATA.

DATA: END OF BDC_TABLE2.

DATA: BEGIN OF BDC_TABLE3 OCCURS 500.

INCLUDE STRUCTURE BDCDATA.

DATA: END OF BDC_TABLE3.

* Message Table

DATA: BEGIN OF MESSTAB OCCURS 500.

INCLUDE STRUCTURE BDCMSGCOLL.

DATA: END OF MESSTAB.

* Data Fields

DATA: D_CHARG LIKE LIPS-CHARG,

D_PIKMG(10) TYPE C,

D_BATCHEXISTS,

D_INDEX,

D_NAME(20) TYPE C,

D_NAME2(20) TYPE C,

D_NOT_P,

D_SBATC,

D_LOCKED,

D_DDATE(10) TYPE C,

D_DUTYLINES(3) TYPE N,

D_DONE,

D_VBELNPOSNR LIKE MKPF-XBLNR,

D_STAT LIKE SY-SUBRC.

DATA: SUB_LINES(3) TYPE N. "FOB 17/09/96

DATA: POS_POSNR LIKE LIPS-POSNR.

DATA: T_LFIMG(10) TYPE C. "FOB 10/03/97

DATA: SPLIT(1) TYPE C. "FOB 11/03/97

DATA: L_YEAR(4) TYPE C.

* At Selection Screen

AT SELECTION-SCREEN.

IF P_DDATE <> SY-DATUM.

MESSAGE W025 WITH TEXT-002.

ENDIF.

SELECT SINGLE * FROM LIKP WHERE VBELN = P_VBELN.

IF SY-SUBRC NE 0.

MESSAGE E022.

ENDIF.

IF LIKP-LIFEX NE P_LIFEX.

MESSAGE E023.

ENDIF.

SELECT * FROM LIPS WHERE VBELN = LIKP-VBELN.

SELECT SINGLE * FROM VBUP WHERE VBELN = LIPS-VBELN AND

POSNR = LIPS-POSNR.

IF VBUP-WBSTA = 'C'.

MESSAGE E031.

ENDIF.

* temporary fix to stop duplicate duty postings fob 180397

* l_year = sy-datum(4).

* select * from mkpf

* where mjahr = l_year and

* vgart = 'WA' and

* blart = 'WA' and

```
*          blaum = 'PR'    and
*          tcode = 'MB1B'  and
*          xblnr = likp-vbeln.
* message e101 with likp-vbeln.
* endselect.
```

```
CALL FUNCTION 'ENQUEUE_EMMARCE'
```

```
EXPORTING
```

```
    MATNR      = LIPS-MATNR
```

```
    WERKS      = LIPS-WERKS
```

```
EXCEPTIONS
```

```
    FOREIGN_LOCK  = 2
```

```
    SYSTEM_FAILURE = 3.
```

```
CASE SY-SUBRC.
```

```
    WHEN 2.
```

```
        MESSAGE E029 WITH LIPS-MATNR LIPS-WERKS.
```

```
    WHEN 3.
```

```
        MESSAGE E028.
```

```
ENDCASE.
```

```
CALL FUNCTION 'DEQUEUE_EMMARCE'
```

```
EXPORTING
```

```
    MATNR = LIPS-MATNR
```

```
    WERKS = LIPS-WERKS.
```

```
* exceptions
```

```
*    others  = 1.
```

```
CALL FUNCTION 'ZENQUEUE_EVVBLKE'
```

```
EXPORTING
```

```

        VBELN = LIKP-VBELN

IMPORTING

        PA_STAT = D_STAT.

IF D_STAT EQ 0.

    CALL FUNCTION 'DEQUEUE_EVVBLKE'

    EXPORTING

        VBELN = LIKP-VBELN.

IF SY-SUBRC EQ 0.

    D_LOCKED = 'N'.

ENDIF.

ELSE.

    D_LOCKED = 'Y'.

    MESSAGE E029 WITH 'Delivery'(005) LIKP-VBELN.

ENDIF.

```

```

ENDSELECT.

```

```

*****

```

```

* START OF SELECTION                                *

```

```

*****

```

```

START-OF-SELECTION.

```

```

SELECT SINGLE * FROM LIKP WHERE VBELN = P_VBELN.

```

```

IF LIKP-LFART NE 'ZSTD' AND LIKP-LFART NE 'NL'.

```

```

    MESSAGE E044.

```

```

ENDIF.

```

SELECT * FROM LIPS WHERE VBELN = LIKP-VBELN AND

VTWEG = 'DP' AND

(WERKS = 'PARK' OR WERKS = 'RUNC' OR

WERKS = 'STRE' OR "+ BENEVIAN

WERKS = 'EXT1' OR WERKS = 'EXT2') AND

(MTART = 'BEER' OR MTART = 'BULK').

* Get Acct. assignment group;

* (For Stretton only allow movements for material of the Acct

* assignment group 01.) "+005 BENEVIAN

CLEAR MVKE. "+005 BENEVIAN

CLEAR DUTY-NOT_01_GROUP. "+005 BENEVIAN

IF LIPS-WERKS = 'STRE'. "+005 BENEVIAN

SELECT SINGLE KTGRM INTO MVKE-KTGRM FROM MVKE "+005 BENEVIAN

WHERE MATNR = LIPS-MATNR "+005 BENEVIAN

AND VKORG = LIKP-VKORG "+005 BENEVIAN

AND VTWEG = LIPS-VTWEG. "+005 BENEVIAN

* If Acct. assignment group = 01 set the flag duty-not_01_group = 'X'

IF MVKE-KTGRM NE '01'. "+006 BENEVIAN

DUTY-NOT_01_GROUP = 'X'. "+006 BENEVIAN

ENDIF. "+006 BENEVIAN

ENDIF. "+005 BENEVIAN

*

* Get Storage Location for Duty status = P (Paid)

SELECT SINGLE RESULT INTO T9CON-RESULT FROM T9CON "+ BENEVIAN

WHERE KEY1 = 'SL' "+ BENEVIAN

AND KEY2 = 'ZSD00300' "+ BENEVIAN

AND KEY3 = LIPS-LGORT. "+ BENEVIAN

IF SY-SUBRC = 0. " If Duty Suspended "+ BENEVIAN

IF LIPS-WERKS = 'STRE' AND MVKE-KTGRM NE '01'. "+005 BENEVIAN

* Storage location does not change "+005 BENEVIAN


```

        DUTY-LGORT_P = LIPS-LGORT.          " +005 BENEVIAN
* -006  duty-not_01_group = 'X'.          " +005 BENEVIAN
        ELSE.                              " +005 BENEVIAN
        DUTY-LGORT_P = T9CON-RESULT.        " + BENEVIAN
        ENDIF.                             " +005 BENEVIAN
ELSE.   "Else storage location does not change + BENEVIAN
        DUTY-LGORT_P = LIPS-LGORT.          " + BENEVIAN
        ENDIF.
* If item is a split batch item with sub items then SUB table is
* populated with sub items.
        IF LIPS-UECHA NE ' '.
        MOVE-CORRESPONDING LIPS TO SUB.
        SUB-LGORT_P = DUTY-LGORT_P.          " + BENEVIAN
        SUB-NOT_01_GROUP = DUTY-NOT_01_GROUP.  " +005 BENEVIAN
        APPEND SUB.
        CLEAR SUB.
        ELSE.
        MOVE-CORRESPONDING LIKP TO DUTY.
        MOVE-CORRESPONDING LIPS TO DUTY.
        APPEND DUTY.
        CLEAR DUTY.
        ENDIF.
        ENDSELECT.

END-OF-SELECTION.

```

```

*****

```

```

* PROCESS                                *

```

```

*****

```

MOVE P_DDATE+6(2) TO D_DDATE+0(2).

MOVE '.' TO D_DDATE+2(1).

MOVE P_DDATE+4(2) TO D_DDATE+3(2).

MOVE '.' TO D_DDATE+5(1).

MOVE P_DDATE+0(4) TO D_DDATE+6(4).

* +005 BENEVIAN Add 'P' in the batch field at header level (duty-charge)

* for all item from duty paid with split batch

* +006 BENEVIAN Add 'P' only when all the batches are Duty Paid, i.e.

* when the storage location at header level is not blank.

* Set sub-mix to X if the split has DS & DP batches

LOOP AT DUTY WHERE CHARG = ' '.

IF DUTY-LGORT = ' '. " +006 BENEVIAN

LOOP AT SUB WHERE Uecha = DUTY-POSNR. " +006 BENEVIAN

SUB-MIX = 'X'. " +006 BENEVIAN

MODIFY SUB. " +006 BENEVIAN

ENDLOOP. " +006 BENEVIAN

ELSE. " +006 BENEVIAN

LOOP AT SUB WHERE Uecha = DUTY-POSNR.

IF SUB-CHARG(1) = 'P'.

DUTY-CHARG = 'P'.

EXIT.

ENDIF.

ENDLOOP.

MODIFY DUTY.

ENDIF. " +006 BENEVIAN

ENDLOOP.

* End + 005 BENEVIAN

LOOP AT DUTY.

PERFORM CALL_ENQUEUE2.

IF D_LOCKED = 'N'.

PERFORM CALL_DEQUEUE2.

IF DUTY-CHARG(1) NE 'P'

AND DUTY-NOT_01_GROUP NE 'X'. " +005 BENEVIAN

MOVE 'Y' TO D_NOT_P.

* On non 'P' batches do a goods movement.

* If Stretton plant do good movement only if the material belongs to the

* 01 Acct. group +005 BENEVIAN

* fix to stop duplicate duty postings

L_YEAR = SY-DATUM(4).

IF DUTY-CHARG <> ' '.

* batch NOT split

CONCATENATE DUTY-VBELN DUTY-POSNR INTO D_VBELNPOSNR.

D_DONE = 'N'.

SELECT * FROM MKPF

WHERE MJAHR = L_YEAR AND

VGART = 'WA' AND

BLART = 'WA' AND

BLAUM = 'PR' AND

TCODE = 'MB1B' AND

XBLNR = D_VBELNPOSNR.

D_DONE = 'Y'.

ENDSELECT.

IF D_DONE <> 'Y'.

PERFORM FILL_MB11_GOODSMVMT.

ENDIF.

ELSE.

* batch splits

IF DUTY-NOT_01_GROUP NE 'X'. " +006 BENEVIAN

LOOP AT SUB WHERE UECHA = DUTY-POSNR.

IF SUB-CHARG(1) NE 'P'. " +006 BENEVIAN

MOVE 'Y' TO D_NOT_P. " +006 BENEVIAN

ELSE. " +006 BENEVIAN

CONTINUE. " +006 BENEVIAN

ENDIF. " +006 BENEVIAN

CONCATENATE DUTY-VBELN SUB-POSNR INTO D_VBELNPOSNR.

D_DONE = 'N'.

SELECT * FROM MKPF

WHERE MJAHR = L_YEAR AND

VGART = 'WA' AND

BLART = 'WA' AND

BLAUM = 'PR' AND

TCODE = 'MB1B' AND

XBLNR = D_VBELNPOSNR.

D_DONE = 'Y'.

ENDSELECT.

IF D_DONE <> 'Y'.

PERFORM FILL_MB11_GOODSMVMT.

ENDIF.

ENDLOOP.

ENDIF. " +006 BENEVIAN

ENDIF.

ENDIF.

ELSE.

SKIP.

WRITE TEXT-003.

SKIP.

ENDIF.

ENDLOOP.

* Get the last item number used for the split batches

MOVE SPACE TO SUB. " +006 BENEVIAN

DESCRIBE TABLE SUB LINES SUB_LINES. " +006 BENEVIAN

READ TABLE SUB INDEX SUB_LINES. " +006 BENEVIAN

MOVE SUB-POSNR TO POS_POSNR. " +006 BENEVIAN

MOVE SPACE TO SUB. " +006 BENEVIAN

IF D_LOCKED = 'N'.

IF D_NOT_P = 'Y'.

* On non 'P' batches that have had a goods movement done in previous

* loop, do an update to amend the status.

PERFORM FILL_VL02_DUTYPAID.

ENDIF.

* Perform Goods Issue once all batches have been processed.

* Picking screen of VL02 - SAPMV50A 0220

*changes made for picking of batch splits by FOB 17/09/96

* loop at sub.

* check sub-not_01_group ne 'X'. " +005 BENEVIAN

* check sub-charg(1) ne 'P'. " +005 BENEVIAN

* move-corresponding sub to tsub.

* append tsub.

* endloop.

* - 006 MOVE SPACE TO SUB.

* - 006 DESCRIBE TABLE SUB LINES SUB_LINES.

* - 006 READ TABLE SUB INDEX SUB_LINES.

* - 006 MOVE SUB-POSNR TO POS_POSNR.

* - 006 MOVE SPACE TO SUB.

LOOP AT SUB.

CHECK SUB-NOT_01_GROUP NE 'X'. " +005 BENEVIAN

* check sub-charg(1) ne 'P'. " +005 BENEVIAN

IF SUB-CHARG(1) = 'P'. " +006 BENEVIAN

CHECK SUB-MIX(1) NE ' '. " +006 BENEVIAN

ENDIF. " +006 BENEVIAN

MOVE SPACE TO T_LFIMG.

AT NEW UECHA.

*{ REPLACE GDTK900105 1

*\ PERFORM FILL_DYNPRO3 USING 'SAPMV50A' '0220' 'X'.

*\ PERFORM FILL_FIELDS3 USING 'BDC_OKCODE' 'UALL'.

*\ PERFORM FILL_DYNPRO3 USING 'SAPMV50A' '0220' 'X'.

*\ PERFORM FILL_FIELDS3 USING 'BDC_OKCODE' 'POPO'.

*\ PERFORM FILL_DYNPRO3 USING 'SAPMV50A' '0111' 'X'.

*\ PERFORM FILL_FIELDS3 USING 'RV50A-POSNR' SUB-UECHA.

*\ PERFORM FILL_DYNPRO3 USING 'SAPMV50A' '0220' 'X'.

*\ PERFORM FILL_FIELDS3 USING 'RV50A-LIPS_SELKZ(1)' 'X'.

*\ PERFORM FILL_FIELDS3 USING 'BDC_OKCODE' 'UCHA'.

** OAKLEYDA 23/03/98 - extra coding no longer required.

** loop at tsub where uecha = sub-uecha. "FOB

** t_lfimg = t_lfimg + tsub-lfimg. "FOB

** endloop. "FOB

*\ PERFORM FILL_DYNPRO3 USING 'SAPMV50A' '0220' 'X'.

** perform fill_fields3 using 'LIPSD-G_LFIMG(1)' t_lfimg."FOB

* UPGRADE to 4.7

perform fill_dynpro3 using 'SAPMV50A' '1000' 'X'.

```

perform fill_fields3 using 'BDC_OKCODE' 'UALL'.
perform fill_dynpro3 using 'SAPMV50A' '1000' 'X'.
perform fill_fields3 using 'BDC_OKCODE' '=POPO_T'.
perform fill_dynpro3 using 'SAPMV50A' '0111' 'X'.
perform fill_fields3 using 'RV50A-POSNR' sub-uecha.
perform fill_dynpro3 using 'SAPMV50A' '1000' 'X'.
perform fill_fields3 using 'RV50A-LIPS_SELKZ(1)' 'X'.
perform fill_fields3 using 'BDC_OKCODE' '=CHPL_T01'.
perform fill_dynpro3 using 'SAPMV50A' '1000' 'X'.

* UPGRADE to 4.7

*} REPLACE

ENDAT.

*{ REPLACE      GDTK900105                      2

*\   PERFORM FILL_FIELDS3 USING 'BDC_OKCODE' 'POPO'.

* UPGRADE to 4.7

    perform fill_fields3 using 'BDC_OKCODE' '=POPO_T'.

* UPGRADE to 4.7

*} REPLACE

PERFORM FILL_DYNPRO3 USING 'SAPMV50A' '0111' 'X'.

IF SUB-CHARG(1) NE 'P'.                      " +006 BENEVIAN

    POS_POSNR = POS_POSNR + 1.

    PERFORM FILL_FIELDS3 USING 'RV50A-POSNR' POS_POSNR.

ELSE.                      " +006 BENEVIAN

    PERFORM FILL_FIELDS3 USING 'RV50A-POSNR' SUB-POSNR. " +006 BENEVIAN

ENDIF.                      " +006 BENEVIAN

*{ REPLACE      GDTK900105                      3

*\   PERFORM FILL_DYNPRO3 USING 'SAPMV50A' '0220' 'X'.

* UPGRADE to 4.7

    perform fill_dynpro3 using 'SAPMV50A' '1000' 'X'.

* UPGRADE to 4.7

*} REPLACE

```

```
PERFORM FILL_FIELDS3 USING 'RV50A-LIPS_SELKZ(1)' 'X'.  
PERFORM FILL_FIELDS3 USING 'LIPSD-PIKMG(1)' SUB-LFIMG.  
ENDLOOP.
```

```
IF D_NOT_P NE 'Y'.
```

```
{ REPLACE      GDTK900105                4  
*\  PERFORM FILL_DYNPRO3 USING 'SAPMV50A' '0101' 'X'.  
*\  
*\  PERFORM FILL_FIELDS3 USING 'LIKP-VBELN' LIKP-VBELN.  
*\  PERFORM FILL_FIELDS3 USING 'BDC_OKCODE' '/8'.  
*\  
*\  PERFORM FILL_DYNPRO3 USING 'SAPMV50A' '0220' 'X'.  
*\  PERFORM FILL_FIELDS3 USING 'BDC_OKCODE' 'WADI'.  
*\  PERFORM FILL_DYNPRO3 USING 'SAPMV50A' '0301' 'X'.  
*\  PERFORM FILL_FIELDS3 USING 'likp-WADAT_IST' D_DDATE.  
*\  PERFORM FILL_FIELDS3 USING 'BDC_OKCODE' '/11'.  
*\  PERFORM FILL_DYNPRO3 USING 'SAPMV50A' '0220' 'X'.  
*\  PERFORM FILL_FIELDS3 USING 'likp-WADAT' D_DDATE.  
*\  PERFORM FILL_FIELDS3 USING 'BDC_OKCODE' '/8'.  
*\ ELSE.  
*\  PERFORM FILL_DYNPRO3 USING 'SAPMV50A' '0220' 'X'.  
*\  PERFORM FILL_FIELDS3 USING 'BDC_OKCODE' 'WADI'.  
*\  PERFORM FILL_DYNPRO3 USING 'SAPMV50A' '0301' 'X'.  
*\  PERFORM FILL_FIELDS3 USING 'likp-WADAT_IST' D_DDATE.  
*\  PERFORM FILL_FIELDS3 USING 'BDC_OKCODE' '/11'.  
*\  PERFORM FILL_DYNPRO3 USING 'SAPMV50A' '0220' 'X'.  
*\  PERFORM FILL_FIELDS3 USING 'likp-WADAT' D_DDATE.  
*\  PERFORM FILL_FIELDS3 USING 'BDC_OKCODE' '/8'.  
* UPGRADE to 4.7  
  perform fill_dynpro3 using 'SAPMV50A' '4004' 'X'.  
  perform fill_fields3 using 'LIKP-VBELN' likp-vbeln.
```



```
perform fill_fields3 using 'BDC_OKCODE' '/00'.
perform fill_dynpro3 using 'SAPMV50A' '4004' 'X'.
perform fill_fields3 using 'BDC_OKCODE' '=T\02'.
```

```
perform fill_dynpro3 using 'SAPMV50A' '1000' 'X'.
perform fill_fields3 using 'BDC_OKCODE' 'WADI'.
perform fill_dynpro3 using 'SAPMV50A' '0301' 'X'.
perform fill_fields3 using 'likp-WADAT_IST' d_ddate.
perform fill_fields3 using 'BDC_OKCODE' '/00'.
perform fill_dynpro3 using 'SAPMV50A' '1000' 'X'.
perform fill_fields3 using 'likp-WADAT' d_ddate.
perform fill_fields3 using 'BDC_OKCODE' '/8'.
```

else.

```
perform fill_dynpro3 using 'SAPMV50A' '1000' 'X'.
perform fill_fields3 using 'BDC_OKCODE' 'WADI'.
perform fill_dynpro3 using 'SAPMV50A' '0301' 'X'.
perform fill_fields3 using 'likp-WADAT_IST' d_ddate.
perform fill_fields3 using 'BDC_OKCODE' '/00'.
perform fill_dynpro3 using 'SAPMV50A' '1000' 'X'.
perform fill_fields3 using 'likp-WADAT' d_ddate.
perform fill_fields3 using 'BDC_OKCODE' '/8'.
```

* UPGRADE to 4.7

*} REPLACE

ENDIF.

LOOP AT DUTY.

PERFORM CALL_ENQUEUE2.

IF D_LOCKED = 'N'.

PERFORM CALL_DEQUEUE2.

ELSE.

EXIT.

ENDIF.
ENDLOOP.

IF D_LOCKED = 'N'.
PERFORM CALL_ENQUEUE_VBELN.
IF D_LOCKED = 'N'.
PERFORM CALL_DEQUEUE_VBELN.

*{ REPLACE GDTK900105 5
*\ CALL TRANSACTION 'VL02' USING BDC_TABLE3
* UPGRADE to 4.7
CALL TRANSACTION 'VL02N' USING BDC_TABLE3
* UPGRADE to 4.7
*} REPLACE
MESSAGES INTO MESSTAB
MODE 'E' UPDATE 'S'.

CLEAR BDC_TABLE3.
REFRESH BDC_TABLE3.
ELSE.
SKIP.
WRITE TEXT-004.
SKIP.
ENDIF.
ELSE.
SKIP.
WRITE TEXT-003.
SKIP.
ENDIF.
ELSE.
SKIP.

WRITE TEXT-003.

SKIP.

ENDIF.

* FORMS *

* Form FILL_mb11_GOODSMVMT *

* Fill in fields for transaction mb11 *

* Transaction mb11 = Goods Transfer from duty suspended storage *

* location to duty paid storage location. *

FORM FILL_MB11_GOODSMVMT.

IF DUTY-CHARG NE ' '.

* i.e. there are no sub items - it is not a batch split.

* Batch split processing follows after the next else.

D_CHARG = DUTY-CHARG.

IF DUTY-CHARG(1) = 'S'.

D_CHARG(1) = 'P'.

ENDIF.

PERFORM FILL_MSC1_CREATEBATCH.

* First screen of MB11

PERFORM FILL_MB11_FIRST.

* Second screen of mb11 - SAPMM07M 0421 - filled with DUTY tab:-

- * receiving plant,
- * receiving storage location,
- * receiving material,
- * material,
- * quantity.

perform fill_dynpro using 'SAPMM07M' '0410' 'X'.

perform fill_fields using 'MSEG-MATNR' duty-matnr.

perform fill_fields using 'MSEG-ERFMG' duty-lfimg.

perform fill_fields using 'MSEG-WERKS' duty-werks.

perform fill_fields using 'MSEG-LGORT' duty-lgort.

perform fill_fields using 'MSEG-CHARG' duty-charg.

perform fill_fields using 'MSEG-UMMAT' duty-matnr.

perform fill_fields using 'MSEG-UMWRK' duty-werks.

* - BENEVIAN perform fill_fields using 'MSEG-UMLGO' '0005'.

PERFORM FILL_FIELDS USING 'MSEG-UMLGO' DUTY-LGORT_P. " + BENEVIAN

D_CHARG = DUTY-CHARG.

D_CHARG(1) = 'P'.

perform fill_fields using 'MSEG-UMCHA' d_charg.

perform fill_dynpro using 'SAPLKACB' '0002' 'X'.

perform fill_dynpro using 'SAPMM07M' '0410' 'X'.

perform fill_fields using 'BDC_OKCODE' '/11'.

perform fill_dynpro using 'SAPLKACB' '0002' 'X'.

perform fill_dynpro using 'SAPMM07M' '0000' 'X'.

ELSE.

* It is a batch split and there are sub items.

DATA: L_ZEILE(4) TYPE N.

* loop at sub where uecha = duty-posnr.

D_CHARG = SUB-CHARG.

IF SUB-CHARG(1) = 'S'.

D_CHARG(1) = 'P'.

ENDIF.

* 'S' batches are moved to a corresponding 'P' batch which is created

* if it does not already exist.

* If it is a 'P' batch already the rest of this form is ignored.

PERFORM FILL_MSC1_CREATEBATCH.

*First screen of MB11

PERFORM FILL_MB11_FIRST.

* Second screen of mb11 - SAPMM07M 0421 - filled with SUB tab:-

- * receiving plant,
- * receiving storage location,
- * receiving material,
- * material,
- * quantity.

perform fill_dynpro using 'SAPMM07M' '0410' 'X'.

perform fill_fields using 'MSEG-MATNR' sub-matnr.

perform fill_fields using 'MSEG-WERKS' sub-werks.

perform fill_fields using 'MSEG-ERFMG' sub-lfimg.

perform fill_fields using 'MSEG-LGORT' sub-lgort.

perform fill_fields using 'MSEG-CHARG' sub-charg.

perform fill_fields using 'MSEG-UMMAT' sub-matnr.

perform fill_fields using 'MSEG-UMWRK' sub-werks.

* - BENEVIAN perform fill_fields using 'MSEG-UMLGO' '0005'.

PERFORM FILL_FIELDS USING 'MSEG-UMLGO' SUB-LGORT_P. " + BENEVIAN

SUB-CHARG(1) = 'P'.

perform fill_fields using 'MSEG-UMCHA' sub-charg.

perform fill_dynpro using 'SAPLKACB' '0002' 'X'.

perform fill_dynpro using 'SAPMM07M' '0410' 'X'.

perform fill_fields using 'BDC_OKCODE' '/11'.

perform fill_dynpro using 'SAPLKACB' '0002' 'X'.

perform fill_dynpro using 'SAPMM07M' '0000' 'X'.

* endloop.

ENDIF.

perform call_enqueue.

IF D_LOCKED = 'N'.

perform call_dequeue.

CALL TRANSACTION 'MB1B' USING BDC_TABLE

MESSAGES INTO MESSTAB

MODE 'E' UPDATE 'S'.

ENDIF.

CLEAR BDC_TABLE.

REFRESH BDC_TABLE.

ENDFORM. " FILL_mb11_GOODSMVMT

* Form FILL_mb11_FIRST

* Form to fill in first screen of transaction mb11

FORM FILL_MB11_FIRST.

* First screen of mb11 - SAPMM07M 0400 - filled with:-

- * movement type,
- * plant,
- * storage location,
- * material slip.

perform fill_dynpro using 'SAPMM07M' '0400' 'X'.

perform fill_fields using 'MKPF-BUDAT' d_ddate.

perform fill_fields using 'RM07M-BWARTWA' '901'.

perform fill_fields using 'RM07M-WERKS' duty-werks.

perform fill_fields using 'RM07M-LGORT' duty-lgort.

PERFORM FILL_FIELDS USING 'RM07M-MTSNR' D_VBELNPOSNR.

perform fill_fields using 'BDC_OKCODE' '/8'.

ENDFORM. " FILL_mb11_FIRST

* Form FILL_VL02_DUTYPAID

* Form to update transaction VL02 with change of status from

* duty suspended to duty paid. *

FORM FILL_VL02_DUTYPAID.

* First screen of VL02 - SAPMV50A 0101 - filled with:-

* delivery note number,

* and select picking.

*{ REPLACE GDTK900105 1

*\ PERFORM FILL_DYNPRO3 USING 'SAPMV50A' '0101' 'X'.

*\

*\ PERFORM FILL_FIELDS3 USING 'likp-VBELN' DUTY-VBELN.

*\ PERFORM FILL_FIELDS3 USING 'BDC_OKCODE' '/8'.

* UPGRADE to 4.7 version

perform fill_dynpro3 using 'SAPMV50A' '4004' 'X'.

perform fill_fields3 using 'likp-VBELN' duty-vbeln.

perform fill_fields3 using 'BDC_OKCODE' '/00'.

perform fill_dynpro3 using 'SAPMV50A' '1000' 'X'.

perform fill_fields3 using 'BDC_OKCODE' '=T\02'.

* UPGRADE to 4.7 version

*} REPLACE

LOOP AT DUTY.

* Picking screen of VL02 - SAPMV50A 0220 - filled with:-

* position the cursor on item to be changed.


```

*{ REPLACE      GDTK900105                2
*\  PERFORM FILL_DYNPRO3 USING 'SAPMV50A' '0220' 'X'.
*\  PERFORM FILL_FIELDS3 USING 'BDC_OKCODE' 'POPO'.
* UPGRADE to 4.7 version
  perform fill_dynpro3 using 'SAPMV50A' '1000' 'X'.
  perform fill_fields3 using 'BDC_OKCODE' '=POPO_T'.
* UPGRADE to 4.7 version
*} REPLACE

```

```

* Position screen - SAPMV50A 0111 - filled with:-
*
      item number.

```

```

PERFORM FILL_DYNPRO3 USING 'SAPMV50A' '0111' 'X'.

```

```

PERFORM FILL_FIELDS3 USING 'RV50A-POSNR' DUTY-POSNR.

```

```

IF DUTY-CHARG = ' '.

```

```

  MOVE SPACE TO SUB.                "fob 110397

```

```

  LOOP AT SUB WHERE UECHA = DUTY-POSNR. "fob 110397 allow 0 pick

```

```

    MOVE 'Y' TO SPLIT.                "fob 110397

```

```

  ENDLOOP.                            "fob 110397

```

```

  IF SY-SUBRC NE 0.                  "fob 110397

```

```

    MOVE 'N' TO SPLIT.                "fob 110397

```

```

  ENDIF.                             "fob 110397

```

```

  IF SPLIT = 'Y'.                    "fob 110397

```

```

    PERFORM BATCH_SPLIT.

```

```

  ENDIF.

```

```

ELSE.

```

```

  IF DUTY-CHARG(1) NE 'P' AND

```

```

    DUTY-LFIMG NE 0.                  "fob 110397

```

```

    PERFORM BATCH_NOT_SPLIT.

```

```

ENDIF.

ENDIF.

ENDLOOP.

ENDFORM.          " FILL_VL02_DUTYPAID

```

```

*****

```

```

*   Form BATCH_SPLIT
*-----*
*   Form to process a batch split item.      *

```

```

*****

```

```

FORM BATCH_SPLIT.

```

```

* First screen of VL02 - SAPMV50A 0220 - filled with:-

```

```

*           selected item,
*           storage location,
*           item -> batch split (CHSP)

```

```

*{ REPLACE      GDTK900105              1
*\ PERFORM FILL_DYNPRO3 USING 'SAPMV50A' '0220' 'X'.
* UPGRADE to 4.7 version
   perform fill_dynpro3 using 'SAPMV50A' '1000' 'X'.
* UPGRADE to 4.7 version
*} REPLACE

```

```

* if lips-lgort(1) = ' '. "change by FOB 16/09/96
* - BENEVIAN perform fill_fields3 using 'LIPS-LGORT(1)' '0005'.
   PERFORM FILL_FIELDS3 USING 'LIPS-LGORT(1)' DUTY-LGORT_P. " + BENEVIAN
* endif.
   PERFORM FILL_FIELDS3 USING 'RV50A-LIPS_SELKZ(1)' 'X'.

```

```

*{ REPLACE      GDTK900105              2

```

*\ PERFORM FILL_FIELDS3 USING 'BDC_OKCODE' 'CHSP'.

* UPGRADE to 4.7 version

perform fill_fields3 using 'BDC_OKCODE' '=CHSP_T'.

* UPGRADE to 4.7 version

*} REPLACE

* Batch split screen of VL02 - SAPMV50A 0250 - filled with:-

* move zero to sub item amts prior to deletion,

* select all sub items,

* delete all sub items,

* re-input sub items with new locations and batch no's

*{ REPLACE GDTK900105 3

*\ PERFORM FILL_DYNPRO3 USING 'SAPMV50A' '0250' 'X'.

* UPGRADE to 4.7 version

perform fill_dynpro3 using 'SAPMV50A' '3000' 'X'.

* UPGRADE to 4.7 version

*} REPLACE

MOVE 0 TO D_INDEX.

LOOP AT SUB WHERE UECHA = DUTY-POSNR.

ADD 1 TO D_INDEX.

* + 006 BENEVIAN Move zero only to DS split batch item quantities

CHECK SUB-CHARG(1) = 'S'. " +006 BENEVIAN

CONCATENATE 'RV50A-LIPS_SELKZ(' D_INDEX INTO D_NAME. " +006

CONCATENATE D_NAME ')' INTO D_NAME2. " +006

PERFORM FILL_FIELDS3 USING D_NAME2 'X'. " +006

CONCATENATE 'LIPS-LFIMG(' D_INDEX INTO D_NAME.

CONCATENATE D_NAME ')' INTO D_NAME2.

* Moves zero to all split batch item quantities prior to deleting them.

PERFORM FILL_FIELDS3 USING D_NAME2 '0'.

ENDLOOP.

* Select all batch split items.

* -006 BENEVIAN perform fill_fields3 using 'BDC_OKCODE' 'MKAL'.

* -006 BENEVIAN perform fill_dynpro3 using 'SAPMV50A' '0250' 'X'.

* Delete all batch split items.

*{ REPLACE GDTK900105 4

*\ PERFORM FILL_FIELDS3 USING 'BDC_OKCODE' 'POLO'.

*\

*\ PERFORM FILL_DYNPRO3 USING 'SAPMV50A' '0250' 'X'.

* UPGRADE to 4.7

perform fill_fields3 using 'BDC_OKCODE' '=POLO_T'.

perform fill_dynpro3 using 'SAPMV50A' '3000' 'X'.

* UPGRADE to 4.7

*} REPLACE

* perform fill_dynpro3 using 'SAPLSPO1' '0100' 'X'.

* set cursor field 'SPOp_OPTION1'.

* perform fill_dynpro3 using 'SAPMM50A' '0250' 'X'.

MOVE 0 TO D_INDEX.

SORT SUB BY UECHA CHARG. " +006

LOOP AT SUB WHERE UECHA = DUTY-POSNR.

ADD 1 TO D_INDEX.

* Re-input all batch split items as stored on SUB with all 'S' batch nos

* being replaced by 'P' batch numbers

IF SUB-CHARG(1) = 'S'.

SUB-CHARG(1) = 'P'.

* - BENEVIAN sub-lgort = '0005'.

SUB-LGORT = SUB-LGORT_P. " + BENEVIAN

ELSE.

CONTINUE. " +006 BENEVIAN

ENDIF.

CONCATENATE 'LIPS-CHARG(' D_INDEX INTO D_NAME.

CONCATENATE D_NAME ')' INTO D_NAME2.

PERFORM FILL_FIELDS3 USING D_NAME2 SUB-CHARG.

CONCATENATE 'LIPS-LGORT(' D_INDEX INTO D_NAME.

CONCATENATE D_NAME ')' INTO D_NAME2.

PERFORM FILL_FIELDS3 USING D_NAME2 SUB-LGORT.

CONCATENATE 'LIPS-LFIMG(' D_INDEX INTO D_NAME.

CONCATENATE D_NAME ')' INTO D_NAME2.

PERFORM FILL_FIELDS3 USING D_NAME2 SUB-LFIMG.

ENDLOOP.

*{ REPLACE GDTK900105 5

*\ PERFORM FILL_FIELDS3 USING 'BDC_OKCODE' '/3'.

* UPGRADE to 4.7

```

perform fill_fields3 using 'BDC_OKCODE' '=BACK_T'.

* UPGRADE to 4.7

*} REPLACE

* endif.

ENDFORM.                " BATCH_SPLIT

*****

*   Form BATCH_NOT_SPLIT
*-----*
*   Form to process ordinary batch items.      *
*****

FORM BATCH_NOT_SPLIT.

* Picking screen of VL02 - SAPMV50A 0220 - filled with:-
*
*       selected item,
*
*       new storage location,
*
*       new batch number,
*
*       quantity

*{ REPLACE      GDTK900105                      1
*\ PERFORM FILL_DYNPRO3 USING 'SAPMV50A' '0220' 'X'.

* UPGRADE to 4.7

perform fill_dynpro3 using 'SAPMV50A' '1000' 'X'.

* UPGRADE to 4.7

*} REPLACE

PERFORM FILL_FIELDS3 USING 'LIPS-CHARG(1)' ''.

PERFORM FILL_FIELDS3 USING 'LIPSD-PIKMG(1)' ''.

*{ REPLACE      GDTK900105                      2
*\ PERFORM FILL_DYNPRO USING 'SAPMV50A' '0220' 'X'.

```

* UPGRADE to 4.7

perform fill_fields3 using 'BDC_OKCODE' '=POPO_T'.

* UPGRADE to 4.7

*} REPLACE

PERFORM FILL_FIELDS3 USING 'BDC_OKCODE' 'POPO'.

* Position screen - SAPMV50A 0111 - filled with:-

* item number.

PERFORM FILL_DYNPRO3 USING 'SAPMV50A' '0111' 'X'.

PERFORM FILL_FIELDS3 USING 'RV50A-POSNR' DUTY-POSNR.

*{ REPLACE GDTK900105 3

*\ PERFORM FILL_DYNPRO3 USING 'SAPMV50A' '0220' 'X'.

* UPGRADE to 4.7

perform fill_dynpro3 using 'SAPMV50A' '1000' 'X'.

* UPGRADE to 4.7

*} REPLACE

* - benevian perform fill_fields3 using 'LIPS-LGORT(1)' '0005'.

PERFORM FILL_FIELDS3 USING 'LIPS-LGORT(1)' DUTY-LGORT_P. " + BENEVIAN

DUTY-CHARG(1) = 'P'.

PERFORM FILL_FIELDS3 USING 'LIPS-CHARG(1)' DUTY-CHARG.

PERFORM FILL_FIELDS3 USING 'LIPSD-PIKMG(1)' DUTY-LFIMG.

ENDFORM. " BATCH_NOT_SPLIT

* Form FILL_MSC1_CREATEBATCH

* Form to create a new DUTY PAID batch where it does not *
* already exist. *

FORM FILL_MSC1_CREATEBATCH.

*{ INSERT GDTK900105 1

* UPGRADE to 4.7

* Data for BAPI

data: material like bapibatchkey-material,
batch like bapibatchkey-batch,
plant like bapibatchkey-plant,
batchattributes like bapibatchatt,
batchcontrolfields like bapibatchctrl,
batchstoragelocation like bapibatchstoloc-stge_loc,
internalnumbercom like bapibncom,
extension1 like bapibncomz.

* UPGRADE to 4.7

*} INSERT

* Check whether 'P' batch created from 'S' batch exists already.

* SELECT SINGLE * FROM MCHA WHERE MATNR = DUTY-MATNR AND
SELECT SINGLE * FROM MCH1
WHERE MATNR = DUTY-MATNR
AND CHARG = D_CHARG.

IF SY-SUBRC <> 0.

*{ REPLACE GDTK900105 2

** First screen of MSC1 - SAPMM03S 0105 - filled with:-

** material,


```

*\*          plant,
*\*          storage location,
*\*          new 'P' batch.
*\
*\  PERFORM FILL_DYNPRO2 USING 'SAPMM03S' '0105' 'X'.
*\
*\  PERFORM FILL_FIELDS2 USING 'RM03S-MATNR' DUTY-MATNR.
*\  PERFORM FILL_FIELDS2 USING 'RM03S-WERKS' DUTY-WERKS.
*\* - BENEVIAN  perform fill_fields2 using 'RM03S-LGORT' '0005'.
*\  PERFORM FILL_FIELDS2 USING 'RM03S-LGORT' DUTY-LGORT_P. " + BENEVIAN
*\  PERFORM FILL_FIELDS2 USING 'RM03S-CHARG' D_CHARG.
*\  PERFORM FILL_FIELDS2 USING 'RM03S-REF_MATNR' ' '.
*\  PERFORM FILL_FIELDS2 USING 'RM03S-REF_WERKS' ' '.
*\  PERFORM FILL_FIELDS2 USING 'RM03S-REF_CHARG' ' '.
*\
*\* Second screen of MSC1 - SAPMM03S 0200 - filled with:-
*\*          'DUTY PAID'
*\
*\  PERFORM FILL_DYNPRO2 USING 'SAPMM03S' '0200' 'X'.
*\  PERFORM FILL_FIELDS2 USING 'MCHA-BWTAR' 'DUTY PAID'.
*\
*\  PERFORM FILL_FIELDS2 USING 'BDC_OKCODE' '/11'.
*\
*\  perform call_enqueue.
*\  IF D_LOCKED NE 'Y'.
*\  perform call_dequeue.
*\  CALL TRANSACTION 'MSC1' USING BDC_TABLE2
*\          MESSAGES INTO MESSTAB
*\          MODE 'E' UPDATE 'S'.
*\*transaction
*\

```

```

*\   CLEAR BDC_TABLE2.
*\   REFRESH BDC_TABLE2.
*\
*\   ENDIF.
* UPGRADE to 4.7
*   Prepare BAPI parameters for batch creation
      move duty-matnr  to material.
      move duty-werks  to plant.
      move duty-lgort_p to batchstoragelocation.
      move d_charg     to batch.
      move 'DUTY PAID' to batchattributes-val_type.

*   Create batch
      call function 'BAPI_BATCH_CREATE'
        exporting
          material      = material
          batch         = batch
          plant         = plant
          batchattributes = batchattributes
          batchstoragelocation = batchstoragelocation.
* UPGRADE to 4.7
*}  REPLACE

      ENDIF.

      ENDFORM.              " FILL_MSC1_CREATBATCH

```

```

*****

```

```

*   Form FILL_DYNPRO

```

* Form to call a screen for transaction MB11 *

FORM FILL_DYNPRO USING PROG

SCREEN

BEGIN.

MOVE PROG TO BDC_TABLE-PROGRAM.

MOVE SCREEN TO BDC_TABLE-DYNPRO.

MOVE BEGIN TO BDC_TABLE-DYNBEGIN.

APPEND BDC_TABLE.

CLEAR BDC_TABLE.

ENDFORM. " FILL_DYNPRO

* Form FILL_DYNPRO2

* Form to call a screen for transaction MSC1 *

FORM FILL_DYNPRO2 USING PROG

SCREEN

BEGIN.

MOVE PROG TO BDC_TABLE2-PROGRAM.

MOVE SCREEN TO BDC_TABLE2-DYNPRO.

MOVE BEGIN TO BDC_TABLE2-DYNBEGIN.

APPEND BDC_TABLE2.

CLEAR BDC_TABLE2.

ENDFORM. " FILL_DYNPRO

* Form FILL_DYNPRO3

* Form to call a screen for transaction VL02 *

FORM FILL_DYNPRO3 USING PROG

SCREEN

BEGIN.

MOVE PROG TO BDC_TABLE3-PROGRAM.

MOVE SCREEN TO BDC_TABLE3-DYNPRO.

MOVE BEGIN TO BDC_TABLE3-DYNBEGIN.

APPEND BDC_TABLE3.

CLEAR BDC_TABLE3.

ENDFORM. " FILL_DYNPRO

* Form FILL_FIELDS

* Form to fill in screen fields for transaction MB11 *

FORM FILL_FIELDS USING FLDNME

FLDVAL.

MOVE FLDNME TO BDC_TABLE-FNAM.

MOVE FLDVAL TO BDC_TABLE-FVAL.

APPEND BDC_TABLE.

CLEAR BDC_TABLE.

ENDFORM. " FILL_FIELDS

* Form FILL_FIELDS2

* Form to fill in screen fields for transaction MSC1 *

FORM FILL_FIELDS2 USING FLDNME

FLDVAL.

MOVE FLDNME TO BDC_TABLE2-FNAM.

MOVE FLDVAL TO BDC_TABLE2-FVAL.

APPEND BDC_TABLE2.

CLEAR BDC_TABLE2.

ENDFORM. " FILL_FIELDS

* Form FILL_FIELDS3

* Form to fill in screen fields for transaction VL02 *

FORM FILL_FIELDS3 USING FLDNME

FLDVAL.

MOVE FLDNME TO BDC_TABLE3-FNAM.

MOVE FLDVAL TO BDC_TABLE3-FVAL.

APPEND BDC_TABLE3.

CLEAR BDC_TABLE3.

ENDFORM. " FILL_FIELDS

* Form call_ENQUEUE

* Function to lock plant data. *

FORM CALL_ENQUEUE.

MOVE 'N' TO D_LOCKED.

CALL FUNCTION 'ENQUEUE_EMMARCE'

EXPORTING

MATNR = LIPS-MATNR

WERKS = LIPS-WERKS

EXCEPTIONS

FOREIGN_LOCK = 2

SYSTEM_FAILURE = 3.

CASE SY-SUBRC.

WHEN 2.

* message e029 with lips-werks lips-matnr.

MOVE 'Y' TO D_LOCKED.

WHEN 3.

MESSAGE E028.

ENDCASE.

endform. " call_ENQUEUE

* Form call_DEQUEUE

* Calls function to unlock plant data. *

FORM CALL_DEQUEUE.

CALL FUNCTION 'DEQUEUE_EMMARCE'

EXPORTING

MATNR = LIPS-MATNR

WERKS = LIPS-WERKS.

* exceptions

* others = 1.

endform. " call_DEQUEUE

&-----

*& Form CALL_ENQUEUE_VBELN

&-----

* text *

* --> p1 text

* <-- p2 text

FORM CALL_ENQUEUE_VBELN.

CALL FUNCTION 'ZENQUEUE_EVVBLKE'

EXPORTING

VBELN = LIKP-VBELN

IMPORTING

PA_STAT = D_STAT.

IF D_STAT <> 0.

D_LOCKED = 'Y'.

* message e029 with 'Delivery' likp-vbeln.

ENDIF.

ENDFORM. " CALL_ENQUEUE_VBELN

&-----

*& Form CALL_DEQUEUE_VBELN

&-----

* text *

* --> p1 text

* <-- p2 text

FORM CALL_DEQUEUE_VBELN.

CALL FUNCTION 'DEQUEUE_EVVBLKE'

EXPORTING

VBELN = LIKP-VBELN.

ENDFORM. " CALL_DEQUEUE_VBELN

&-----

*& Form CALL_ENQUEUE2

&-----

* text *

```
* --> p1    text
* <-- p2    text
*-----*
```

FORM CALL_ENQUEUE2.

MOVE 'N' TO D_LOCKED.

CALL FUNCTION 'ENQUEUE_EMMARCE'

EXPORTING

MATNR = DUTY-MATNR

WERKS = DUTY-WERKS

EXCEPTIONS

FOREIGN_LOCK = 2

SYSTEM_FAILURE = 3.

CASE SY-SUBRC.

WHEN 2.

* message e029 with lips-werks lips-matnr.

MOVE 'Y' TO D_LOCKED.

WHEN 3.

MESSAGE E028.

ENDCASE.

ENDFORM. " CALL_ENQUEUE2

```
*&-----*
```

```
*& Form CALL_DEQUEUE2
```

```
*&-----*
```

```
* text                      *
```

```
*-----*
```

```
* --> p1    text
```

```
* <-- p2    text
```

```
*-----*
```

```
FORM CALL_DEQUEUE2.
```

```
CALL FUNCTION 'DEQUEUE_EMMARCE'
```

```
EXPORTING
```

```
    MATNR = DUTY-MATNR
```

```
    WERKS = DUTY-WERKS.
```

```
* exceptions
```

```
*    others  = 1.
```

```
ENDFORM.          " CALL_DEQUEUE2
```