SD Subsequent Function Requirement Routines

Technical Solution Guide

The Smith Consulting Group, Inc.
PO Box 703
Flanders, NJ 07836-0703
973-713-5846
SAPtechsolutions.com

SD Subsequent Function Requirement Routines

Subsequent Function Requirement Routines

Subsequent Function Requirement Routines are used to control the subsequent processing of sales and delivery documents within the Sales and Distribution Module. The purpose of these routines is to prevent specific functions from occurring if certain conditions exist. These routines do not prevent subsequent documents from being created and they should not be confused with Copy Control. For example, these routines will not prevent an invoice from being created from a delivery, but they can prevent the picking of an existing delivery.

Subsequent Function Code

The system uses Subsequent Function Codes to determine which routines to call. Table TVFO contains all subsequent functions defined in the system. The following are the current functions.

Function	Description
01	Requirements from sales document (availability)
02	Purchase requisition from sales document
03	Requirements from sales document (procurement)
04	Create delivery due index
05	Production order/plan order from sales document
06	Release/Block-Assembly Order From Sales Document
11	Picking from delivery
12	Packing from delivery
13	Goods issue from delivery

Table TVFO

Table TVFO contains the routine mapping for each subsequent function code. For each function there are two routines specified. The System Routine (SyFORM routine) is the standard R/3 routine and should not be modified. The User Routine (Routine number) is the user defined requirement.

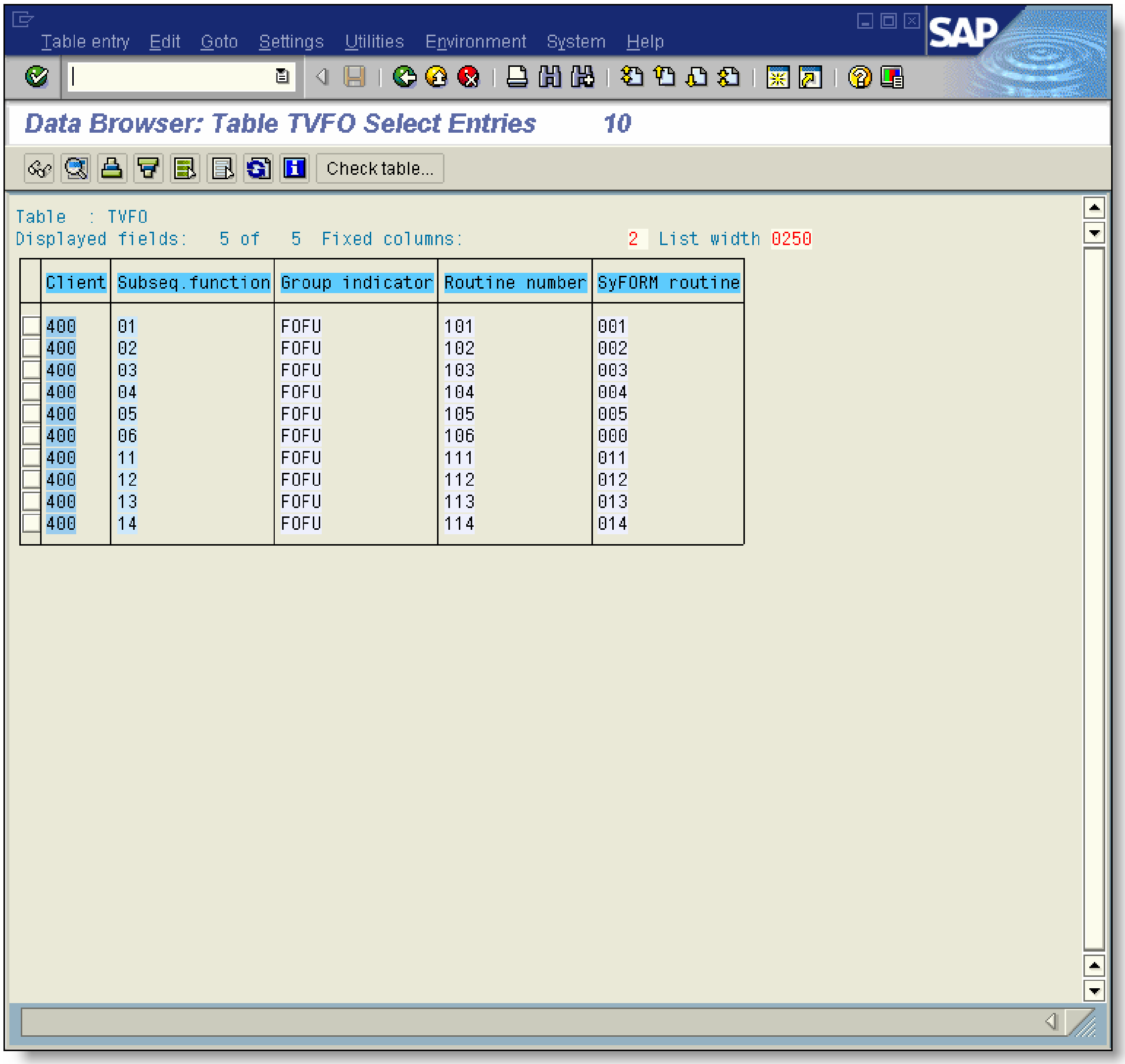


Table entry Edit Goto Settings Utilities Environment System Help
Check table...

Data Browser: Table TVFO Select Entries 10

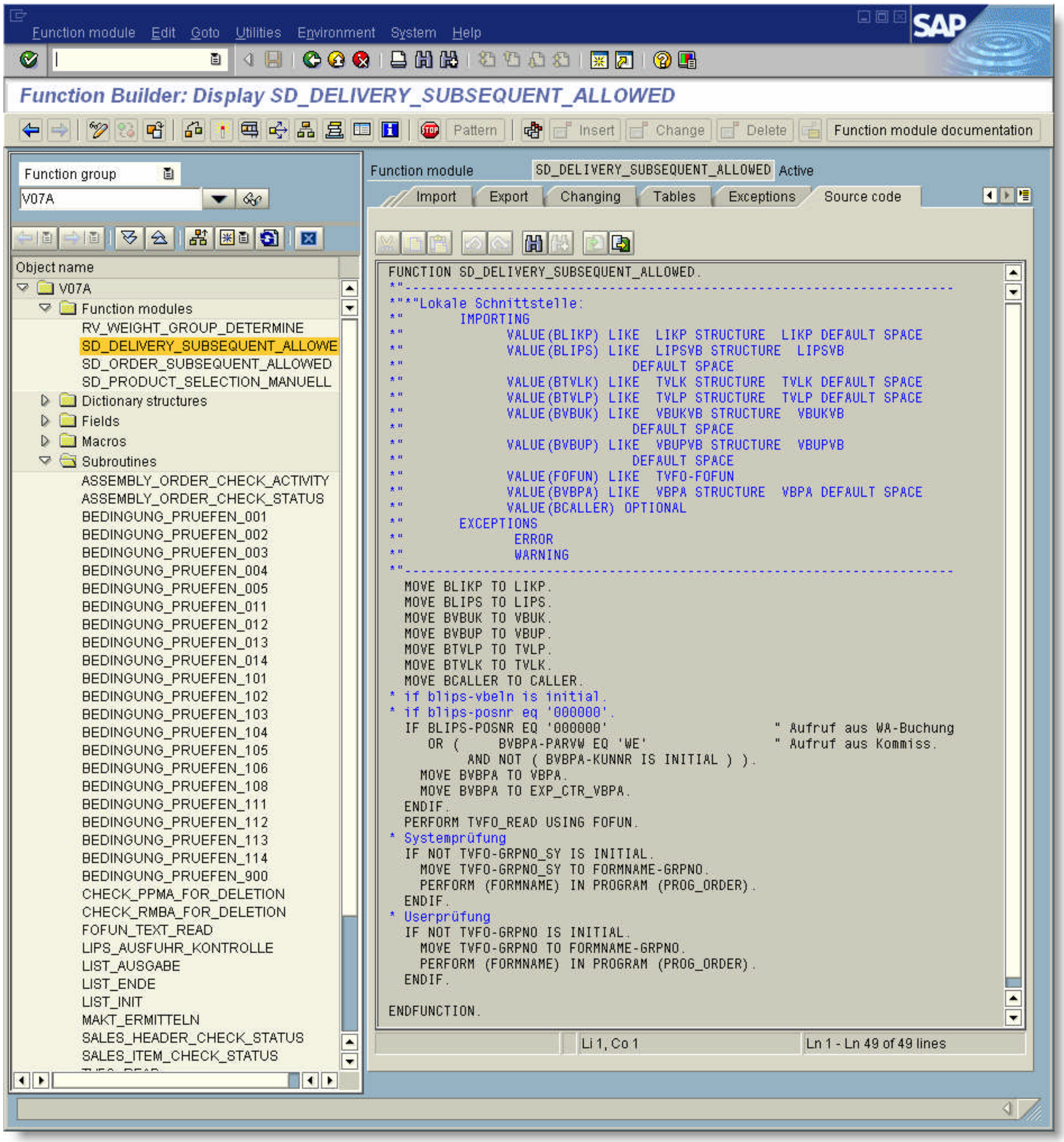
```
Table  : TVFO
Displayed fields:   5 of   5  Fixed columns:               2  List width 0250
```

Client	Subseq.function	Group indicator	Routine number	SyFORM routine
400	01	FOFU	101	001
400	02	FOFU	102	002
400	03	FOFU	103	003
400	04	FOFU	104	004
400	05	FOFU	105	005
400	06	FOFU	106	000
400	11	FOFU	111	011
400	12	FOFU	112	012
400	13	FOFU	113	013
400	14	FOFU	114	014

In the example above, function code 11 is the picking function code. Routine 011 is the standard system routine and routine 111 is the user routine.

Function Group V07A

Function group V07A contains the processing function modules and all of the requirement routines. Each routine is contained in a separate include. Both the include file and routine name contain the routine number.



Function module SD_DELIVERY_SUBSEQUENT_ALLOWED Active

```
FUNCTION SD_DELIVERY_SUBSEQUENT_ALLOWED.
*"----------------------------------------------------------------------
*"*"Lokale Schnittstelle:
*"       IMPORTING
*"             VALUE(BLIKP) LIKE  LIKP STRUCTURE  LIKP DEFAULT SPACE
*"             VALUE(BLIPS) LIKE  LIPSVB STRUCTURE  LIPSVB
*"                             DEFAULT SPACE
*"             VALUE(BTVLK) LIKE  TVLK STRUCTURE  TVLK DEFAULT SPACE
*"             VALUE(BTVLP) LIKE  TVLP STRUCTURE  TVLP DEFAULT SPACE
*"             VALUE(BVBUK) LIKE  VBUKVB STRUCTURE  VBUKVB
*"                             DEFAULT SPACE
*"             VALUE(BVBUP) LIKE  VBUPVB STRUCTURE  VBUPVB
*"                             DEFAULT SPACE
*"             VALUE(FOFUN) LIKE  TVFO-FOFUN
*"             VALUE(BVBPA) LIKE  VBPA STRUCTURE  VBPA DEFAULT SPACE
*"             VALUE(BCALLER) OPTIONAL
*"       EXCEPTIONS
*"              ERROR
*"              WARNING
*"----------------------------------------------------------------------
  MOVE BLIKP TO LIKP.
  MOVE BLIPS TO LIPS.
  MOVE BVBUK TO VBUK.
  MOVE BVBUP TO VBUP.
  MOVE BTVLP TO TVLP.
  MOVE BTVLK TO TVLK.
  MOVE BCALLER TO CALLER.
* if blips-vbeln is initial.
* if blips-posnr eq '000000'.
  IF BLIPS-POSNR EQ '000000'                       " Aufruf aus WA-Buchung
     OR (     BVBPA-PARVW EQ 'WE'                  " Aufruf aus Kommiss.
          AND NOT ( BVBPA-KUNNR IS INITIAL ) ).
    MOVE BVBPA TO VBPA.
    MOVE BVBPA TO EXP_CTR_VBPA.
  ENDIF.
  PERFORM TVFO_READ USING FOFUN.
* Systemprüfung
  IF NOT TVFO-GRPNO_SY IS INITIAL.
    MOVE TVFO-GRPNO_SY TO FORMNAME-GRPNO.
    PERFORM (FORMNAME) IN PROGRAM (PROG_ORDER).
  ENDIF.
* Userprüfung
  IF NOT TVFO-GRPNO IS INITIAL.
    MOVE TVFO-GRPNO TO FORMNAME-GRPNO.
    PERFORM (FORMNAME) IN PROGRAM (PROG_ORDER).
  ENDIF.

ENDFUNCTION.
```

Processing Function Modules

There are two function modules in this function group that process requirements. One function is for sales documents, and the other is for delivery documents. These function modules are called from various points in the Order and Delivery programs. The calling programs pass the appropriate function code to the function module. The logic reads table TVFO with the passed function code to determine which routines to call. Each function executes the non-modifiable system routine first, and then executes the user defined routine.

SD_ORDER_SUBSEQUENT_ALLOWED

Passes all relevant sales document structures and the function code.

```
FUNCTION SD_ORDER_SUBSEQUENT_ALLOWED
      IMPORTING
            BTVAP         LIKE   TVAP
            BTVEP         LIKE   TVEP
            BVBAK         LIKE   VBAK
            BVBAP         LIKE   VBAP
            BVBEP         LIKE   VBEP
            BVBEPVB       LIKE   VBEPVB
            BVBUK         LIKE   VBUK
            BVBUP         LIKE   VBUP
            FOFUN         LIKE   TVFO-FOFUN
            I_SIMUL_MODE                DEFAULT SPACE
            I_CHECK_CYCLE_FOFUN_04      DEFAULT SPACE
      EXPORTING
            NO_CREATE     LIKE   TPTEST-STATUS
            NO_CHANGE     LIKE   TPTEST-STATUS
      EXCEPTIONS
            ERROR
            WARNING
            ERROR_BUT_DO_NOT_DELETE.
```

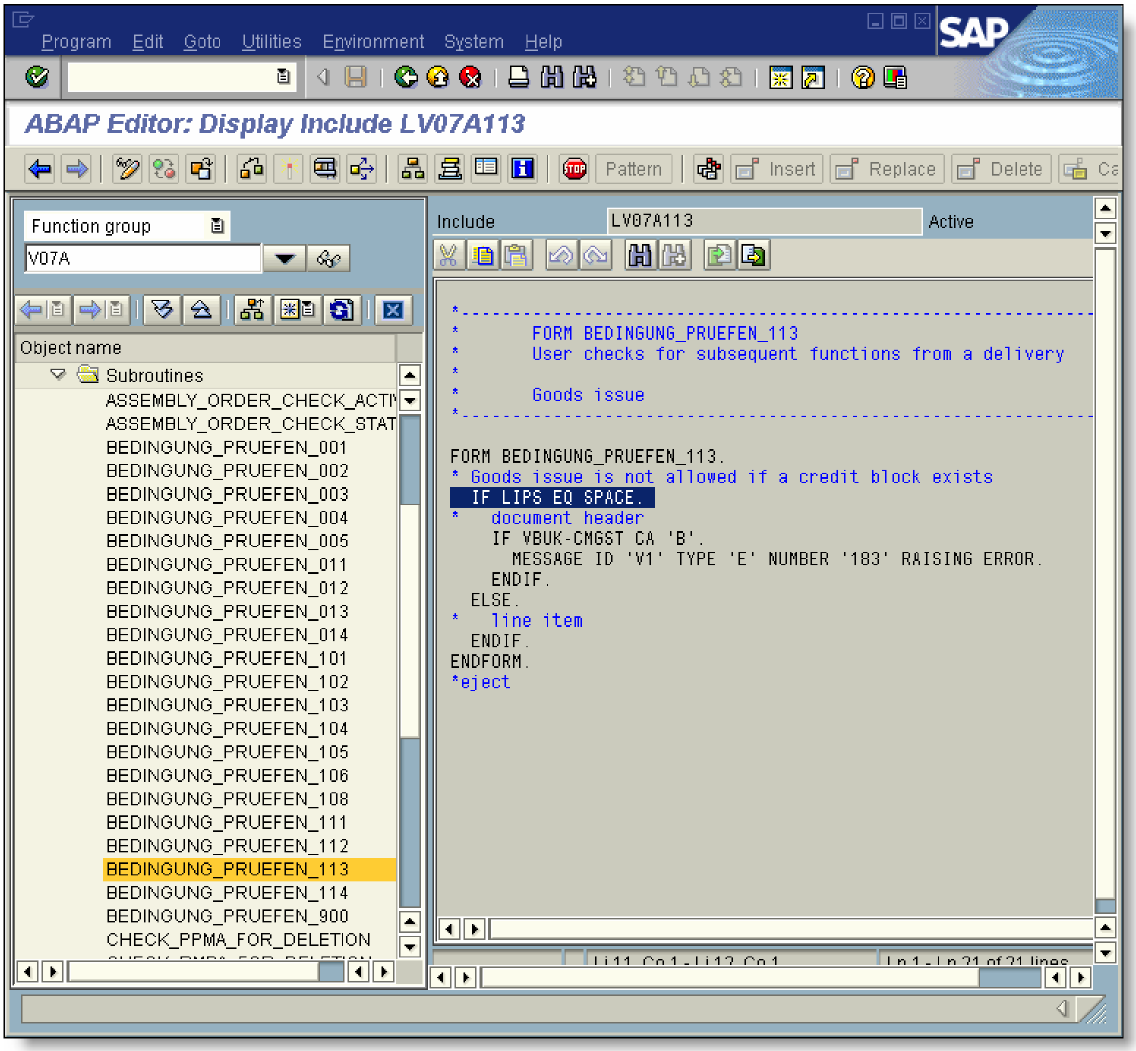
SD_DELIVERY_SUBSEQUENT_ALLOWED

Passes all relevant delivery document structures and the function code.

```
FUNCTION SD_DELIVERY_SUBSEQUENT_ALLOWED
        IMPORTING
                BLIKP           LIKE    LIKP
                BLIPS           LIKE    LIPSVB
                BTVLK           LIKE    TVLK
                BTVLP           LIKE    TVLP
                BVBUK           LIKE    VBUKVB
                BVBUP           LIKE    VBUPVB
                FOFUN           LIKE    TVFO-FOFUN
                BVBPA           LIKE    VBPA
                BCALLER         OPTIONAL
        EXCEPTIONS
                ERROR
                WARNING.
```

Multiple Calls

These functions are called multiple times from the calling program during the processing of a document. Each time the functions are called, the data contained in the structures may be different. In the case of delivery processing, the function is first called with only the header data populated. During subsequent calls, the header data is not passed, but item data is present.



ABAP Editor: Display Include LV07A113

Function group: V07A

Object name

- Subroutines
  - ASSEMBLY_ORDER_CHECK_ACTI
  - ASSEMBLY_ORDER_CHECK_STAT
  - BEDINGUNG_PRUEFEN_001
  - BEDINGUNG_PRUEFEN_002
  - BEDINGUNG_PRUEFEN_003
  - BEDINGUNG_PRUEFEN_004
  - BEDINGUNG_PRUEFEN_005
  - BEDINGUNG_PRUEFEN_011
  - BEDINGUNG_PRUEFEN_012
  - BEDINGUNG_PRUEFEN_013
  - BEDINGUNG_PRUEFEN_014
  - BEDINGUNG_PRUEFEN_101
  - BEDINGUNG_PRUEFEN_102
  - BEDINGUNG_PRUEFEN_103
  - BEDINGUNG_PRUEFEN_104
  - BEDINGUNG_PRUEFEN_105
  - BEDINGUNG_PRUEFEN_106
  - BEDINGUNG_PRUEFEN_108
  - BEDINGUNG_PRUEFEN_111
  - BEDINGUNG_PRUEFEN_112
  - BEDINGUNG_PRUEFEN_113
  - BEDINGUNG_PRUEFEN_114
  - BEDINGUNG_PRUEFEN_900
  - CHECK_PPMA_FOR_DELETION

Include: LV07A113 Active

```
*---------------------------------------------------------------
*       FORM BEDINGUNG_PRUEFEN_113
*       User checks for subsequent functions from a delivery
*
*       Goods issue
*---------------------------------------------------------------

FORM BEDINGUNG_PRUEFEN_113.
* Goods issue is not allowed if a credit block exists
  IF LIPS EQ SPACE.
*   document header
    IF VBUK-CMGST CA 'B'.
      MESSAGE ID 'V1' TYPE 'E' NUMBER '183' RAISING ERROR.
    ENDIF.
  ELSE.
*   line item
  ENDIF.
ENDFORM.
*eject
```

Li 11 Co 1 - Li 12 Co 1   Ln 1 - Ln 21 of 21 lines

The above example is the standard logic from the goods issue routine. The logic checks a field in VBUK only when LIPS is initial. The ELSE side of the IF statement is for checking item fields. This is because when LIPS is initial, the header data is being passed.

We suggest setting break points in the functions and routines to understand exactly how they work before implementing your own logic.

Custom Goods Issue Requirement Example

The following example demonstrates how to implement a custom goods issue requirement routine. The basic process is the same for the other available functions.

Business Requirement

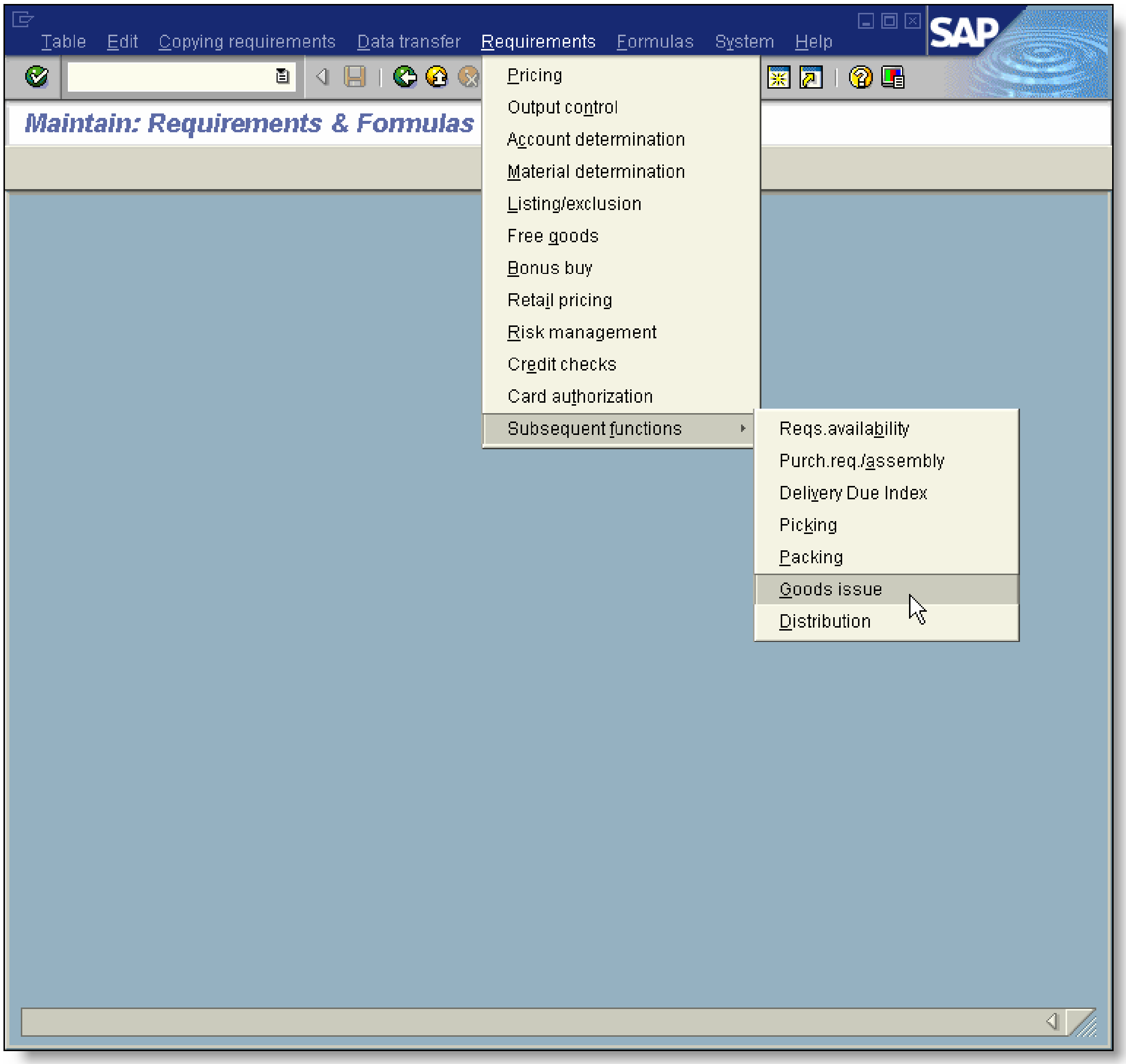
For a specific plant, inventory balances are maintained in a custom 'Z' table in addition to the standard R/3 inventory. This was done as part of a larger project to track the quantities of materials imported into Mexico for re-sale. Without elaborating on the design of the entire project, the purpose of this portion is to ensure that inventory exists in the 'Z' table during the goods issue process. If there is insufficient inventory in the table, the goods issue will not be posted.

Solution

To solve this problem, a custom goods issue requirement routine will be implemented. The logic in this routine will check the inventory balance in the 'Z' table and prevent the goods issue if the quantity in the table is less than the delivery quantity.

Step 1 Clone the standard routine

Using transaction VOFM, navigate to the subsequent function requirement routine for goods issue. Alternatively, you can go directly to the good issue requirement using transaction OVB7.



SD Subsequent Function Requirement Routines

There is only one routine allowed for each function even though the screen implies that multiple routines can be used. On this screen, press the 'Req. maintenance' button at the bottom of the screen.

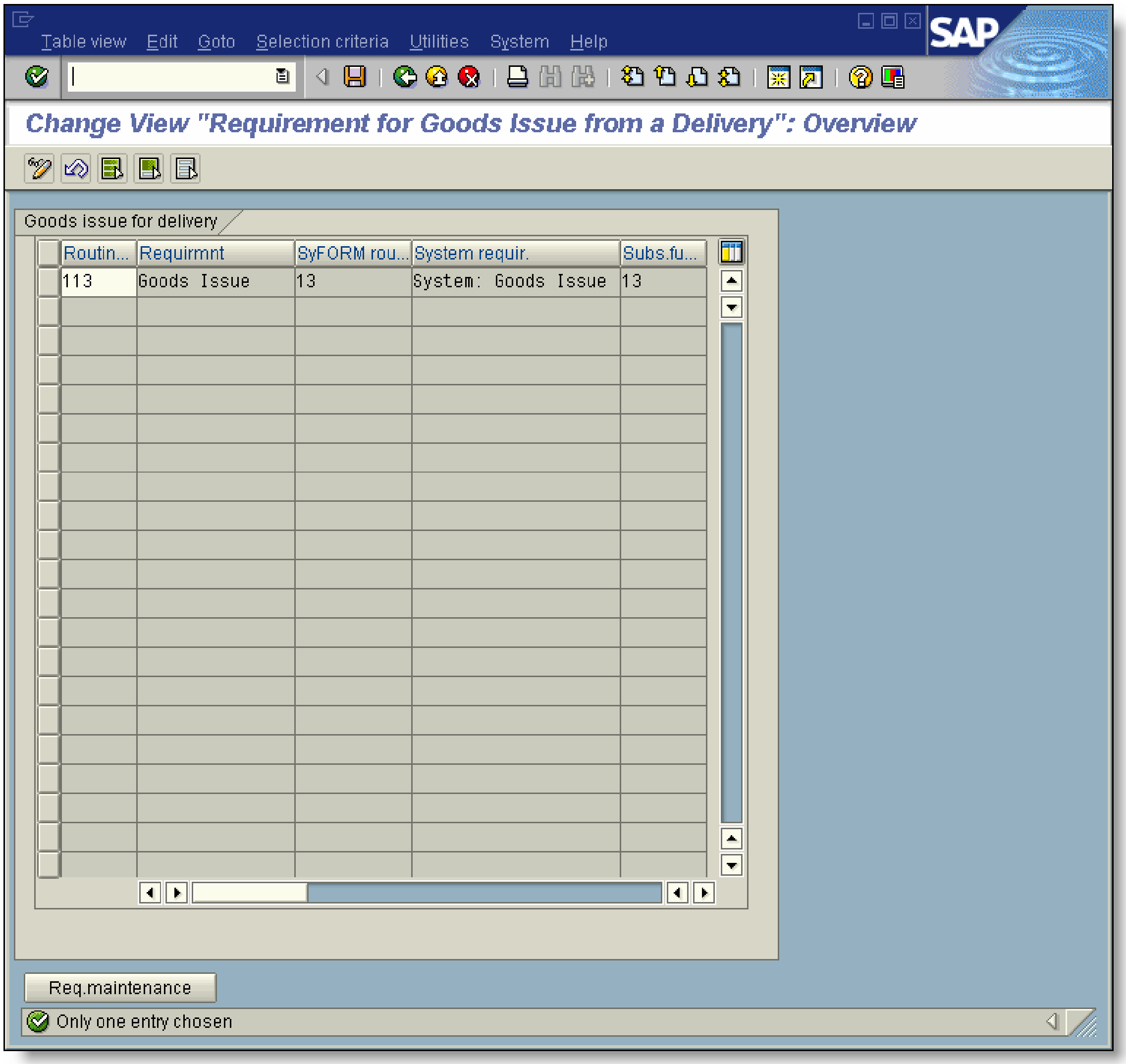


Table view Edit Goto Selection criteria Utilities System Help

Change View "Requirement for Goods Issue from a Delivery": Overview

Goods issue for delivery

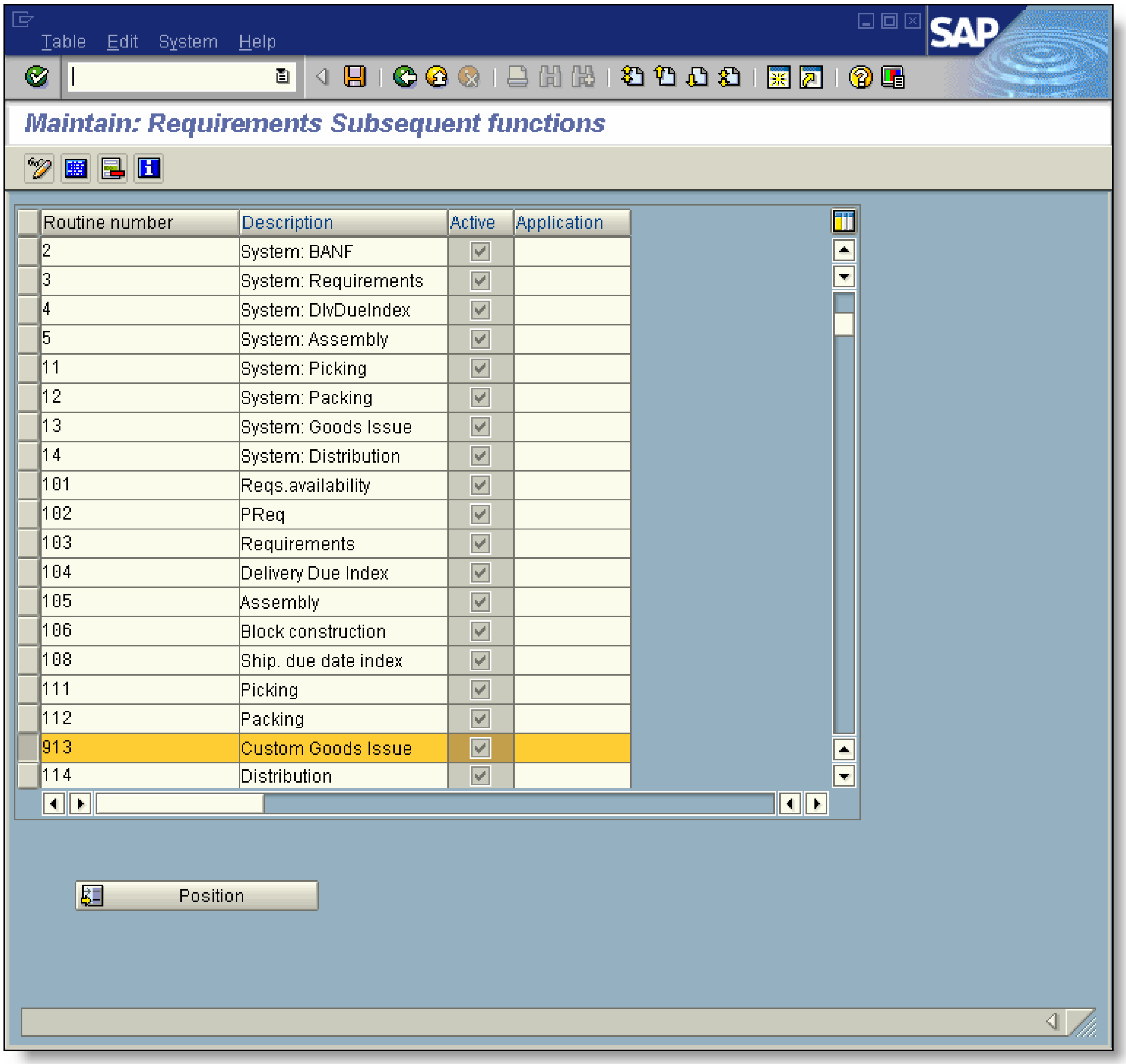
Routin...	Requirmnt	SyFORM rou...	System requir.	Subs.fu...
113	Goods Issue	13	System: Goods Issue	13

Req.maintenance

Only one entry chosen

SD Subsequent Function Requirement Routines

To clone the standard routine (113), type over the existing routine number and press enter. The system will prompt you for an object key. Obtain and specify the object key, then press ENTER.



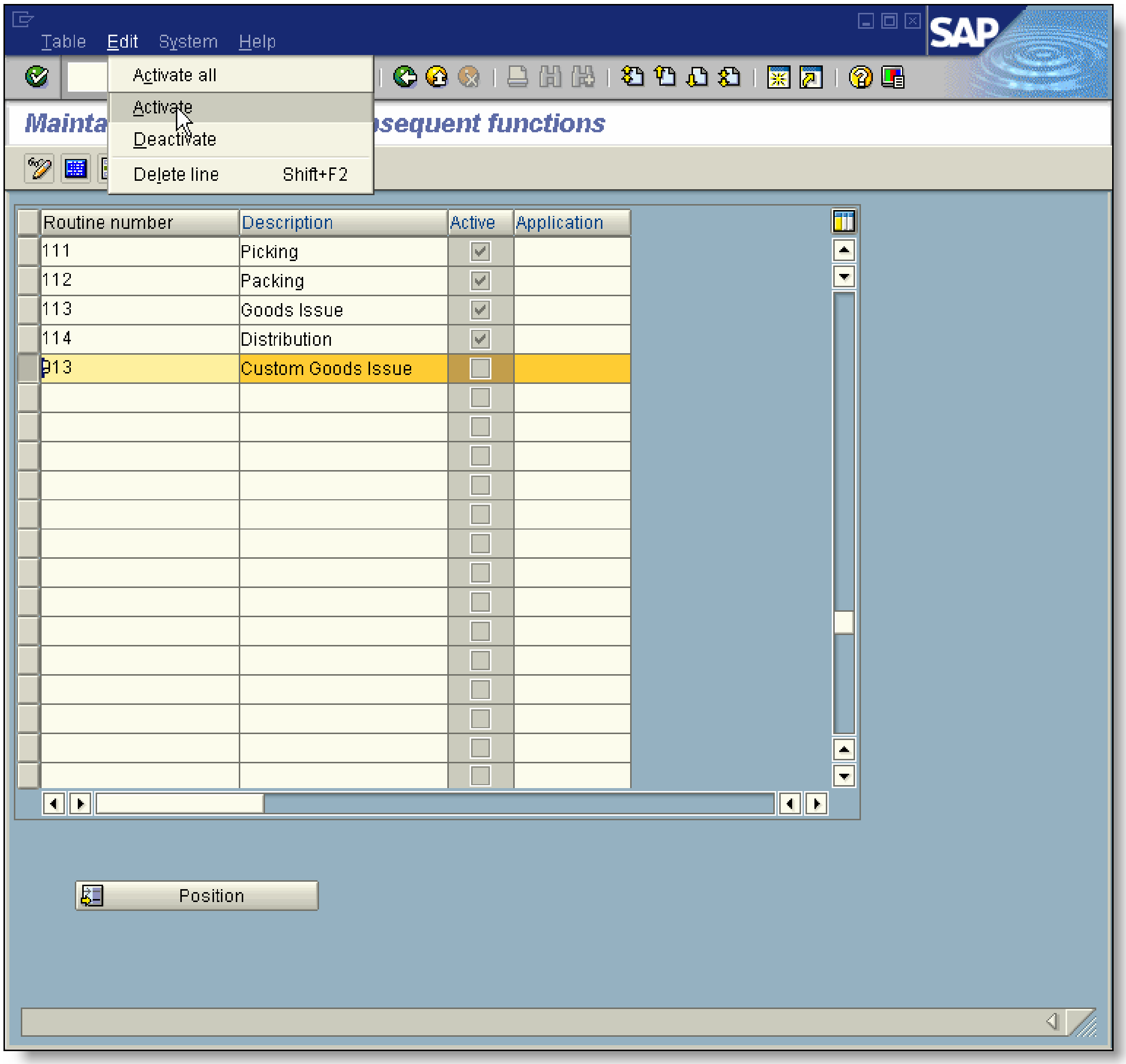
Maintain: Requirements Subsequent functions

Routine number	Description	Active	Application
2	System: BANF	☑	
3	System: Requirements	☑	
4	System: DlvDueIndex	☑	
5	System: Assembly	☑	
11	System: Picking	☑	
12	System: Packing	☑	
13	System: Goods Issue	☑	
14	System: Distribution	☑	
101	Reqs.availability	☑	
102	PReq	☑	
103	Requirements	☑	
104	Delivery Due Index	☑	
105	Assembly	☑	
106	Block construction	☑	
108	Ship. due date index	☑	
111	Picking	☑	
112	Packing	☑	
913	Custom Goods Issue	☑	
114	Distribution	☑	

In this example, we typed routine 913 over 113. It is a common practice to simply replace the first digit of these routines with a ‘9’. Using this standard makes it obvious which routine was used for cloning.

Step 2 *Activate the new routine*

VOFM routines need to be activated before they can be used.



Routine number	Description	Active	Application
111	Picking	☑	
112	Packing	☑	
113	Goods Issue	☑	
114	Distribution	☑	
913	Custom Goods Issue	□	

Step 3 Assign the new routine

Only one routine can be active in the system. There is no other configuration required.

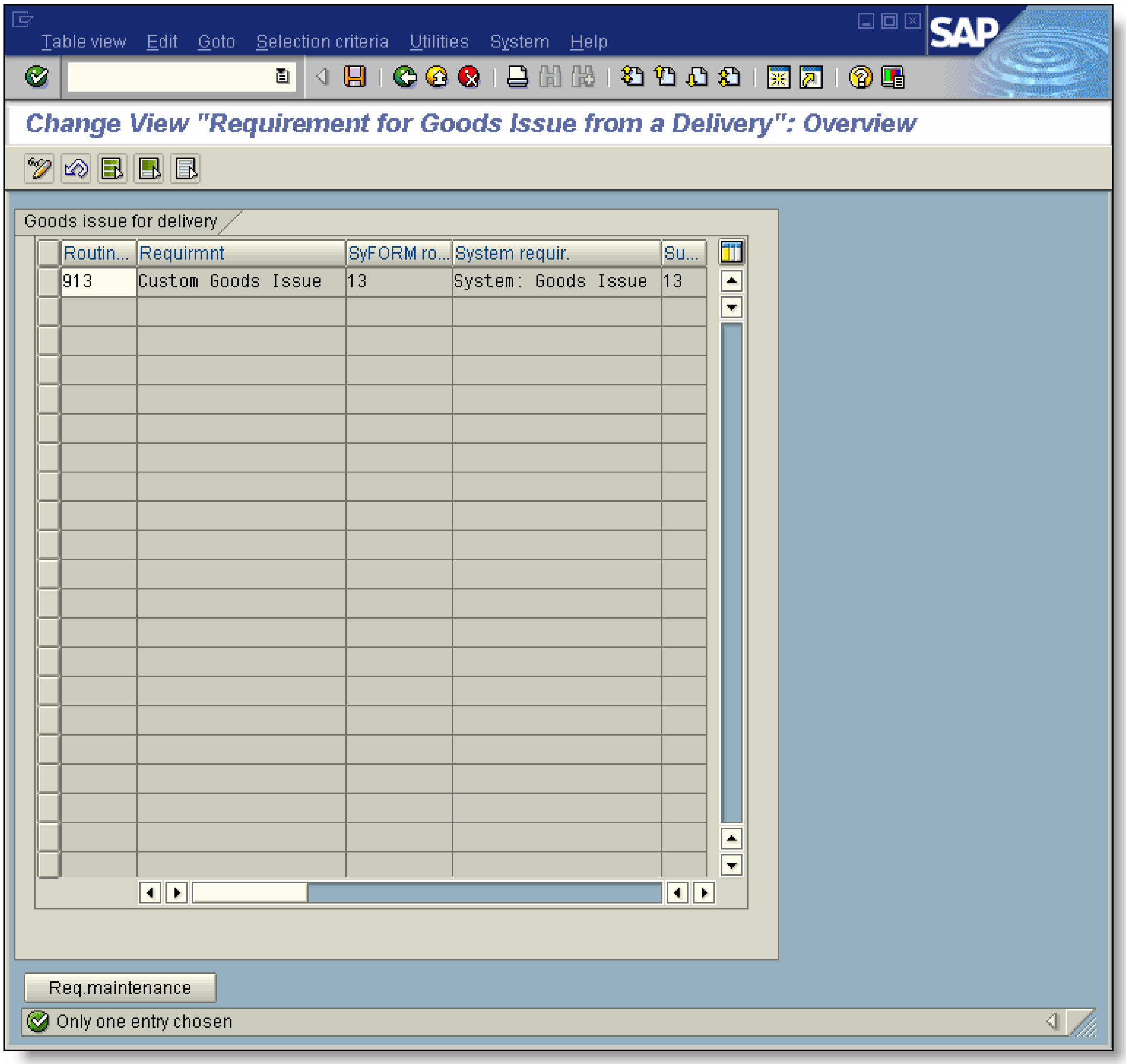


Table view Edit Goto Selection criteria Utilities System Help

Change View "Requirement for Goods Issue from a Delivery": Overview

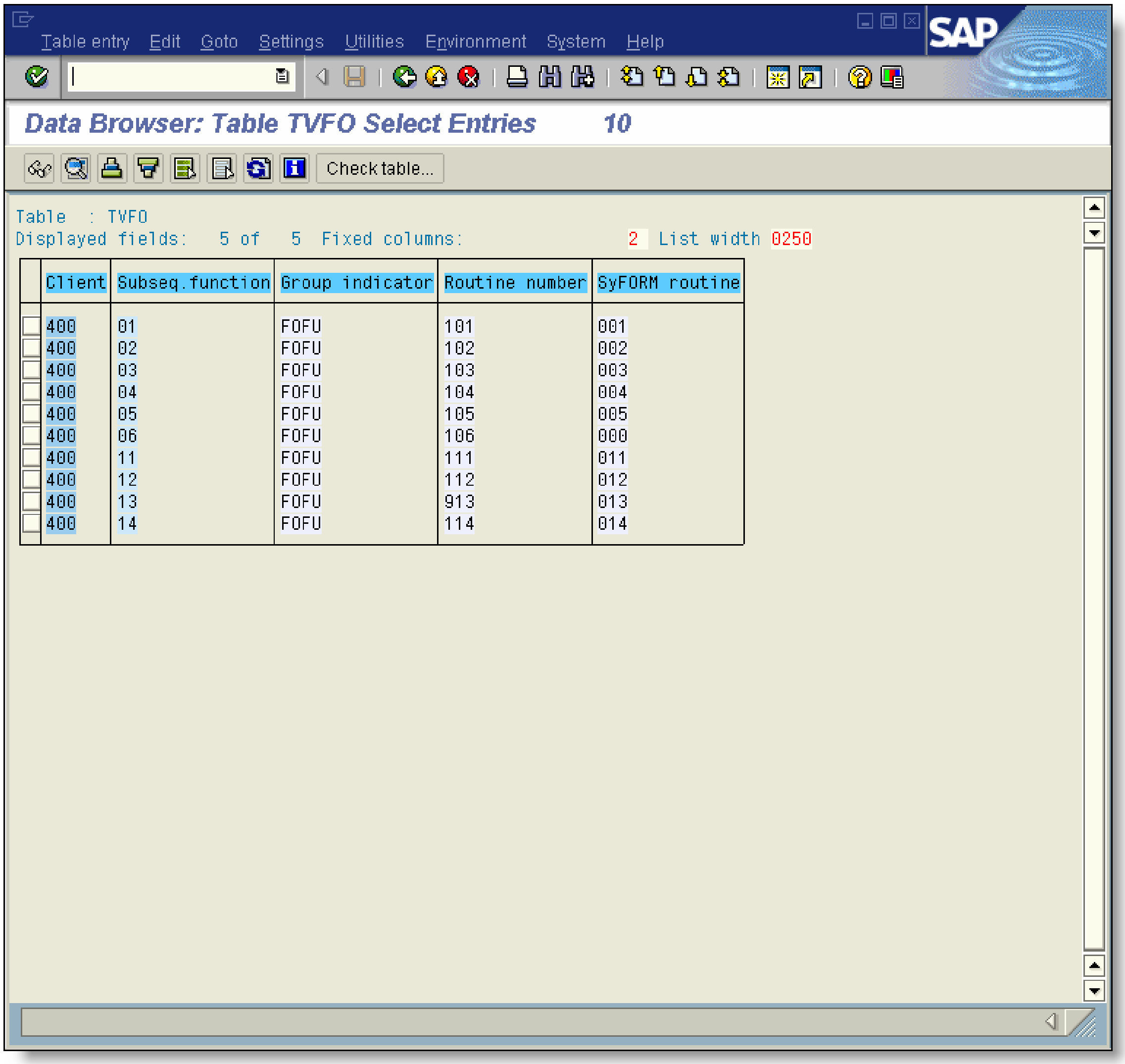
Goods issue for delivery

Routin...	Requirmnt	SyFORM ro...	System requir.	Su...
913	Custom Goods Issue	13	System: Goods Issue	13

Only one entry chosen

SD Subsequent Function Requirement Routines

Examining the TVFO table, we now see that subsequent function 13 points to routine number 913.



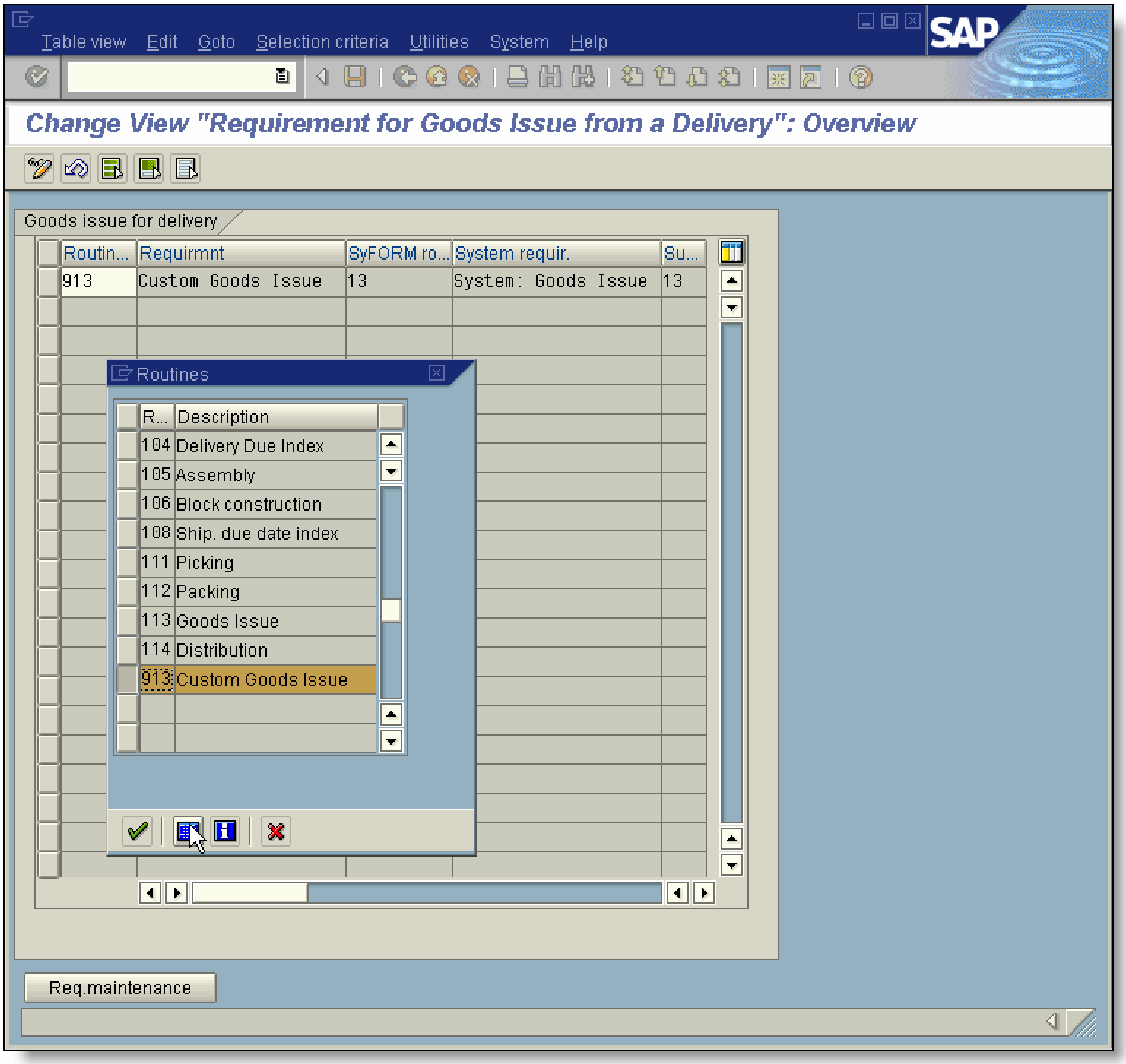
Data Browser: Table TVFO Select Entries 10

```
Table   : TVFO
Displayed fields:   5 of   5  Fixed columns:             2  List width 0250
```

	Client	Subseq.function	Group indicator	Routine number	SyFORM routine
□	400	01	FOFU	101	001
□	400	02	FOFU	102	002
□	400	03	FOFU	103	003
□	400	04	FOFU	104	004
□	400	05	FOFU	105	005
□	400	06	FOFU	106	000
□	400	11	FOFU	111	011
□	400	12	FOFU	112	012
□	400	13	FOFU	913	013
□	400	14	FOFU	114	014

Step 4 Code the routine

To access the source code, pull down the list of valid routines (F4), select the new routine and press the source code icon at the bottom of the dialog box.



Change View "Requirement for Goods Issue from a Delivery": Overview

Goods issue for delivery

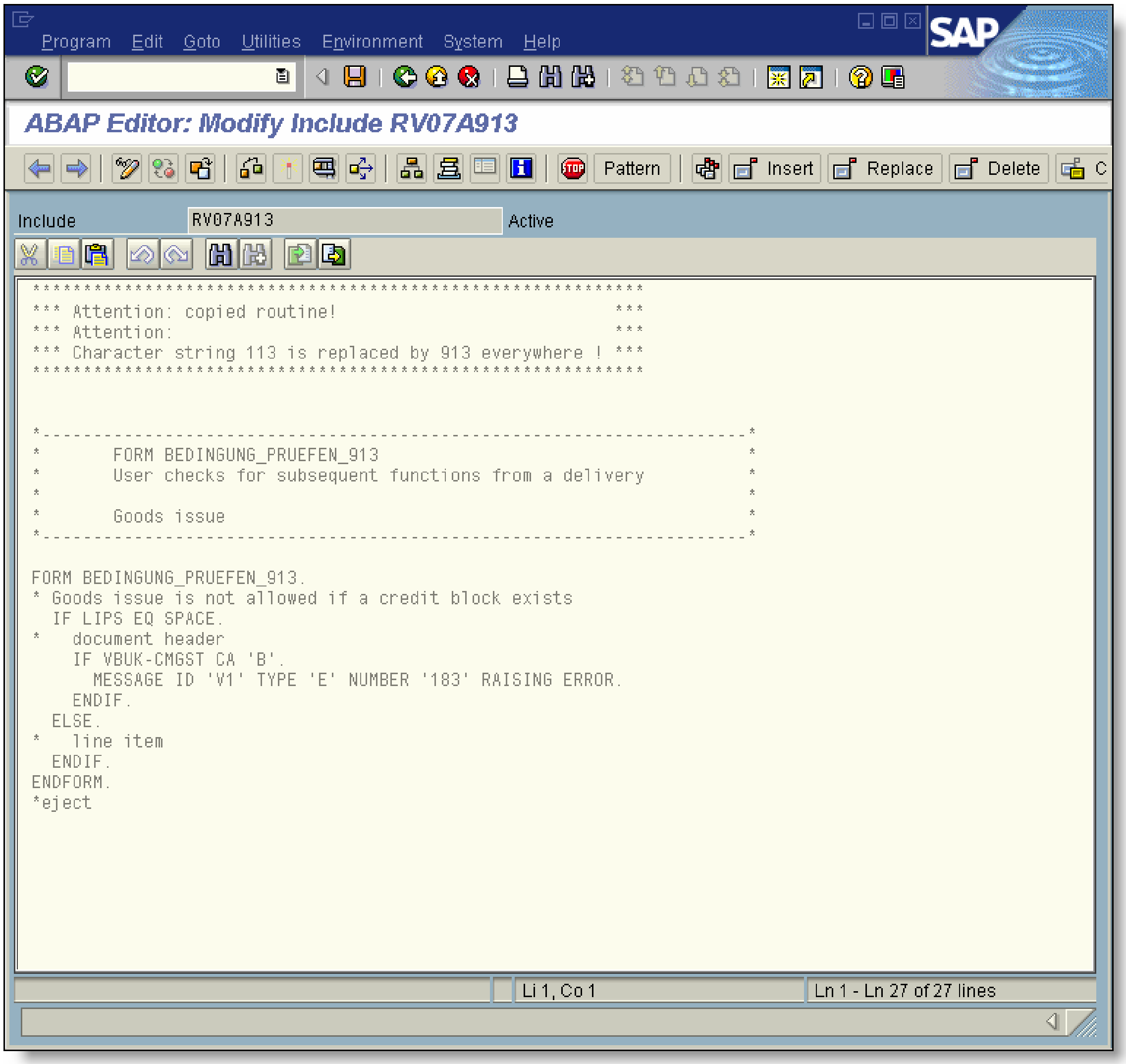
Routin...	Requirmnt	SyFORM ro...	System requir.	Su...
913	Custom Goods Issue	13	System: Goods Issue	13

Routines

R...	Description
104	Delivery Due Index
105	Assembly
106	Block construction
108	Ship. due date index
111	Picking
112	Packing
113	Goods Issue
114	Distribution
913	Custom Goods Issue

SD Subsequent Function Requirement Routines

When a routine is cloned, the system automatically inserts comments at the top of the include denoting that it was copied from another routine. Since the include generated (RV07A913) is within the R/3 name range, the system protects the code as if we were installing a source code modification.



ABAP Editor: Modify Include RV07A913

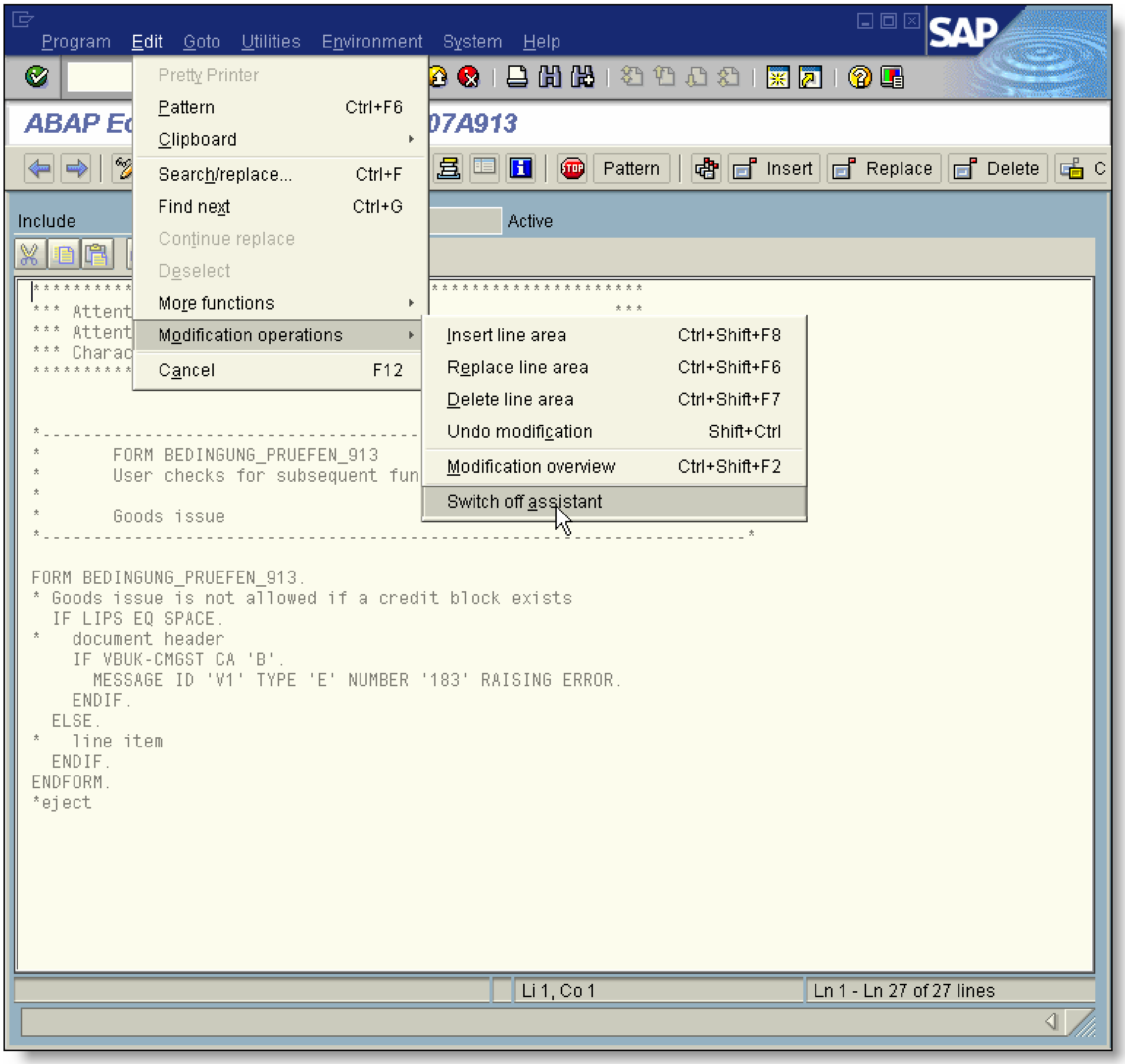
Include RV07A913 Active

```
*************************************************************
*** Attention: copied routine!                            ***
*** Attention:                                            ***
*** Character string 113 is replaced by 913 everywhere ! ***
*************************************************************


*---------------------------------------------------------------------*
*       FORM BEDINGUNG_PRUEFEN_913                                    *
*       User checks for subsequent functions from a delivery          *
*                                                                     *
*       Goods issue                                                   *
*---------------------------------------------------------------------*

FORM BEDINGUNG_PRUEFEN_913.
* Goods issue is not allowed if a credit block exists
  IF LIPS EQ SPACE.
*   document header
    IF VBUK-CMGST CA 'B'.
      MESSAGE ID 'V1' TYPE 'E' NUMBER '183' RAISING ERROR.
    ENDIF.
  ELSE.
*   line item
  ENDIF.
ENDFORM.
*eject
```

Li 1, Co 1 Ln 1 - Ln 27 of 27 lines

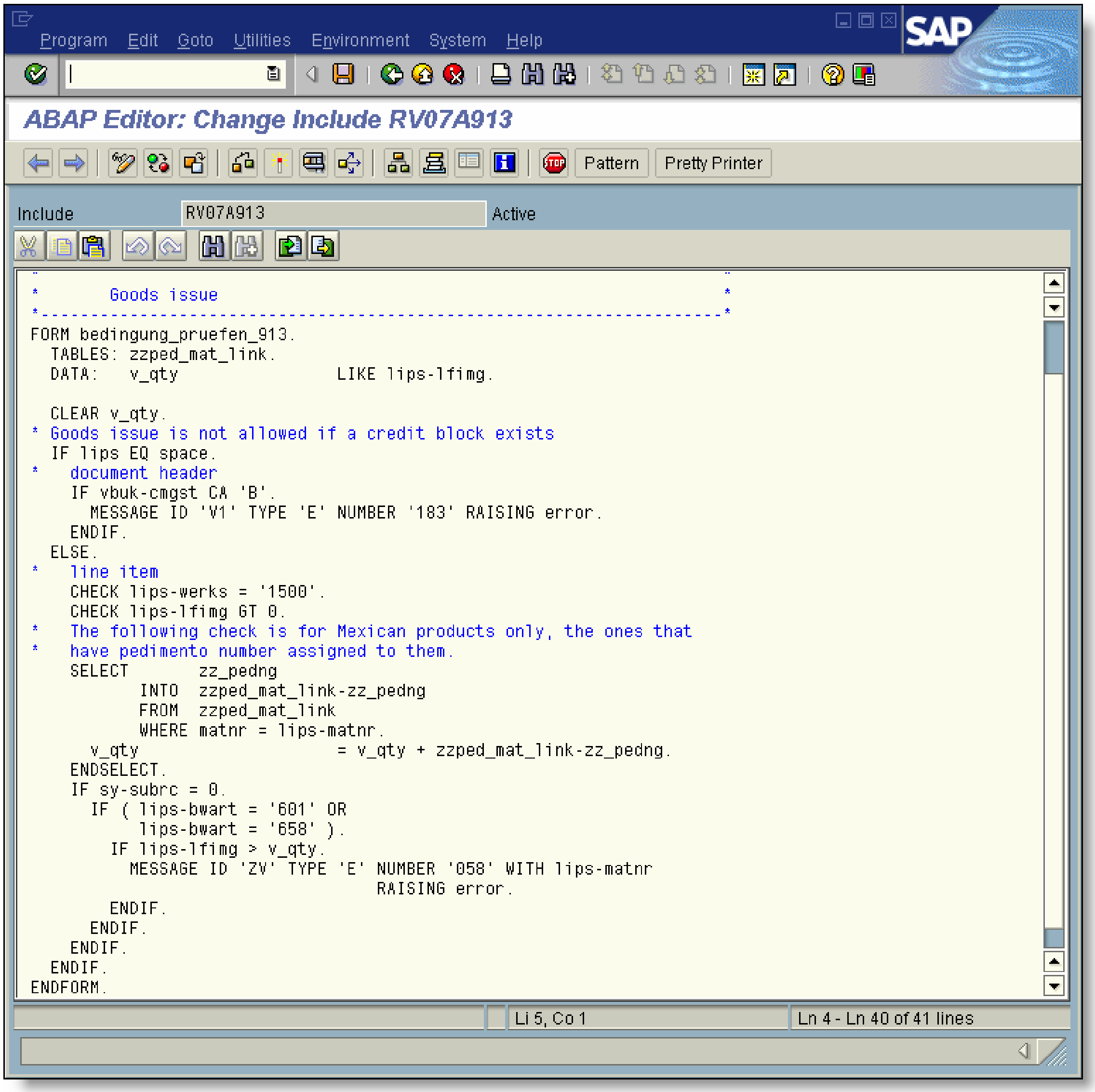
SD Subsequent Function Requirement Routines

For custom routines, it is acceptable to turn off the modification assistant as long as the routine number begins with a ‘9’.



```
**********                    *********************
*** Attent                                      ***
*** Attent
*** Charac
**********


*-----------------------------------
*       FORM BEDINGUNG_PRUEFEN_913
*       User checks for subsequent fun
*
*       Goods issue
*-----------------------------------------------------------------------*

FORM BEDINGUNG_PRUEFEN_913.
* Goods issue is not allowed if a credit block exists
  IF LIPS EQ SPACE.
*   document header
    IF VBUK-CMGST CA 'B'.
      MESSAGE ID 'V1' TYPE 'E' NUMBER '183' RAISING ERROR.
    ENDIF.
  ELSE.
*   line item
  ENDIF.
ENDFORM.
*eject
```

Our custom logic is added to the routine. The standard logic that was already present is kept.



ABAP Editor: Change Include RV07A913

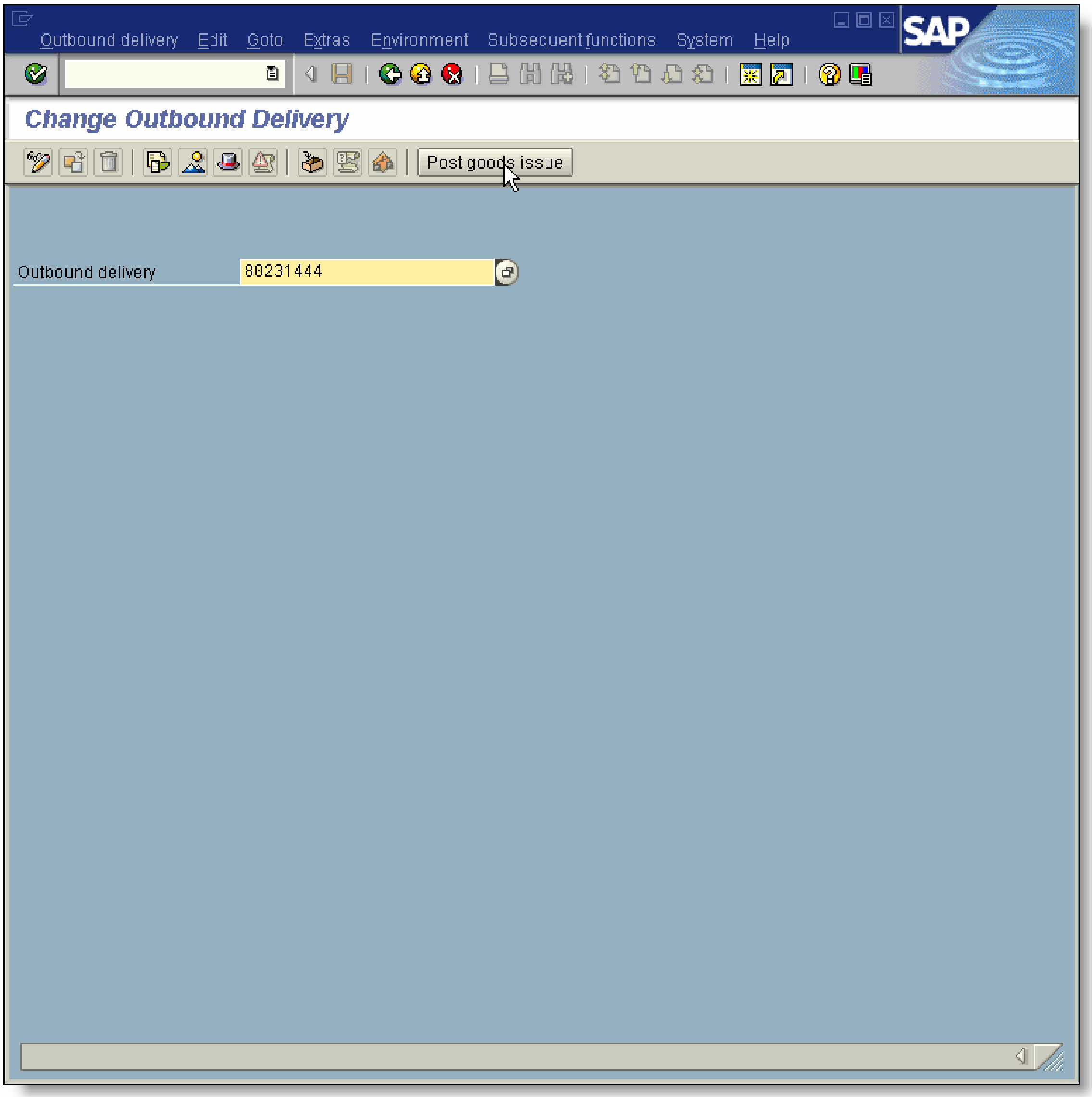
Include RV07A913 Active

```
*       Goods issue                                                   *
*---------------------------------------------------------------------*
FORM bedingung_pruefen_913.
  TABLES: zzped_mat_link.
  DATA:   v_qty                LIKE lips-lfimg.

  CLEAR v_qty.
* Goods issue is not allowed if a credit block exists
  IF lips EQ space.
*   document header
    IF vbuk-cmgst CA 'B'.
      MESSAGE ID 'V1' TYPE 'E' NUMBER '183' RAISING error.
    ENDIF.
  ELSE.
*   line item
    CHECK lips-werks = '1500'.
    CHECK lips-lfimg GT 0.
*   The following check is for Mexican products only, the ones that
*   have pedimento number assigned to them.
    SELECT       zz_pedng
           INTO  zzped_mat_link-zz_pedng
           FROM  zzped_mat_link
           WHERE matnr = lips-matnr.
      v_qty                 = v_qty + zzped_mat_link-zz_pedng.
    ENDSELECT.
    IF sy-subrc = 0.
      IF ( lips-bwart = '601' OR
           lips-bwart = '658' ).
        IF lips-lfimg > v_qty.
          MESSAGE ID 'ZV' TYPE 'E' NUMBER '058' WITH lips-matnr
                                   RAISING error.
        ENDIF.
      ENDIF.
    ENDIF.
  ENDIF.
ENDFORM.
```

Li 5, Co 1 Ln 4 - Ln 40 of 41 lines

Step 5 Test the new routine

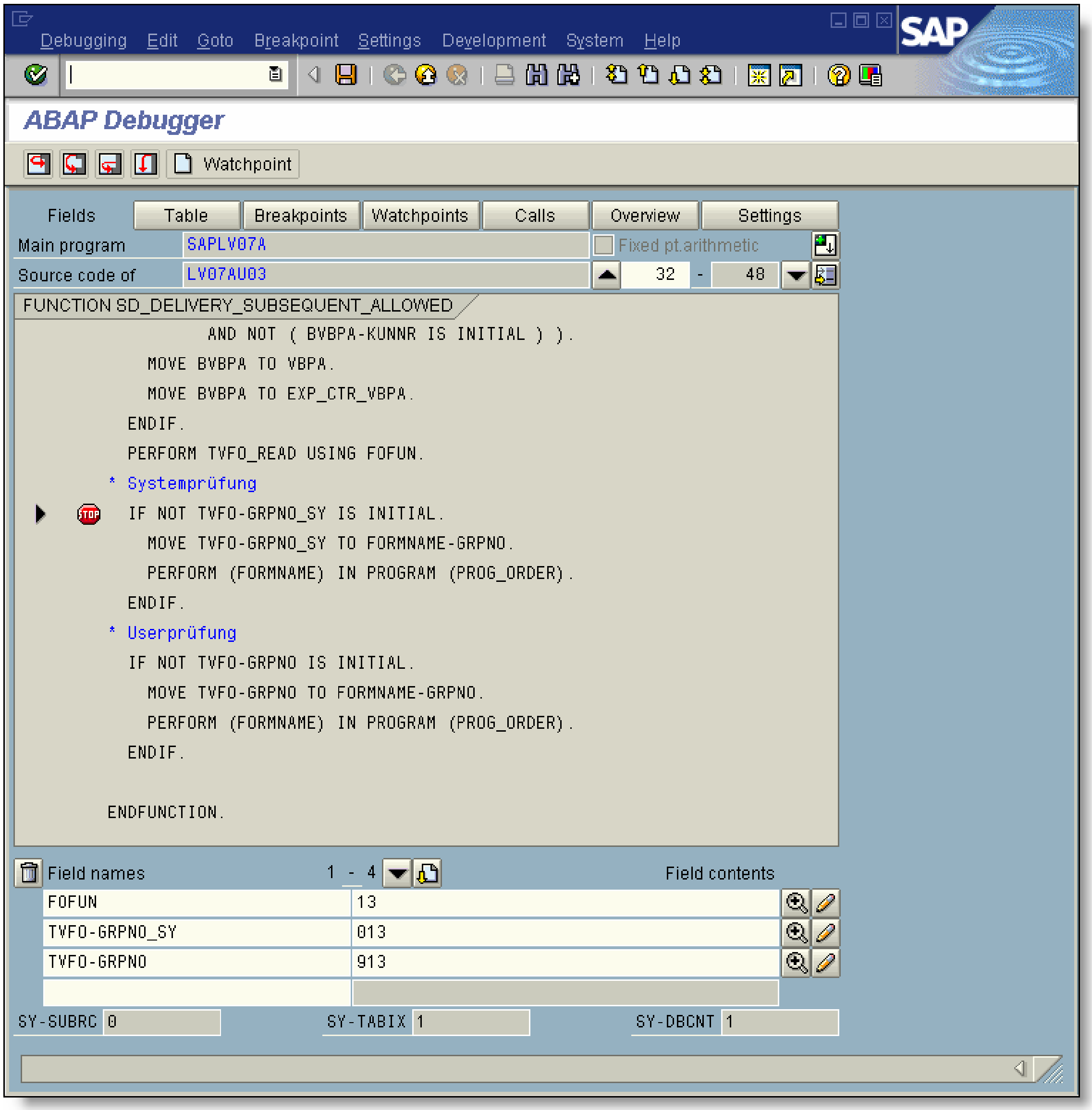
We have selected a delivery and pressed the Post goods issue button.



Outbound delivery 80231444

SD Subsequent Function Requirement Routines

A break point was set in the SD_DELIVERY_SUBSEQUENT_ALLOWED function module.



Main program SAPLV07A

Source code of LV07AU03 32 - 48

FUNCTION SD_DELIVERY_SUBSEQUENT_ALLOWED

```
     AND NOT ( BVBPA-KUNNR IS INITIAL ) ).
  MOVE BVBPA TO VBPA.
  MOVE BVBPA TO EXP_CTR_VBPA.
 ENDIF.
 PERFORM TVFO_READ USING FOFUN.
* Systemprüfung
 IF NOT TVFO-GRPNO_SY IS INITIAL.
  MOVE TVFO-GRPNO_SY TO FORMNAME-GRPNO.
  PERFORM (FORMNAME) IN PROGRAM (PROG_ORDER).
 ENDIF.
* Userprüfung
 IF NOT TVFO-GRPNO IS INITIAL.
  MOVE TVFO-GRPNO TO FORMNAME-GRPNO.
  PERFORM (FORMNAME) IN PROGRAM (PROG_ORDER).
 ENDIF.

ENDFUNCTION.
```

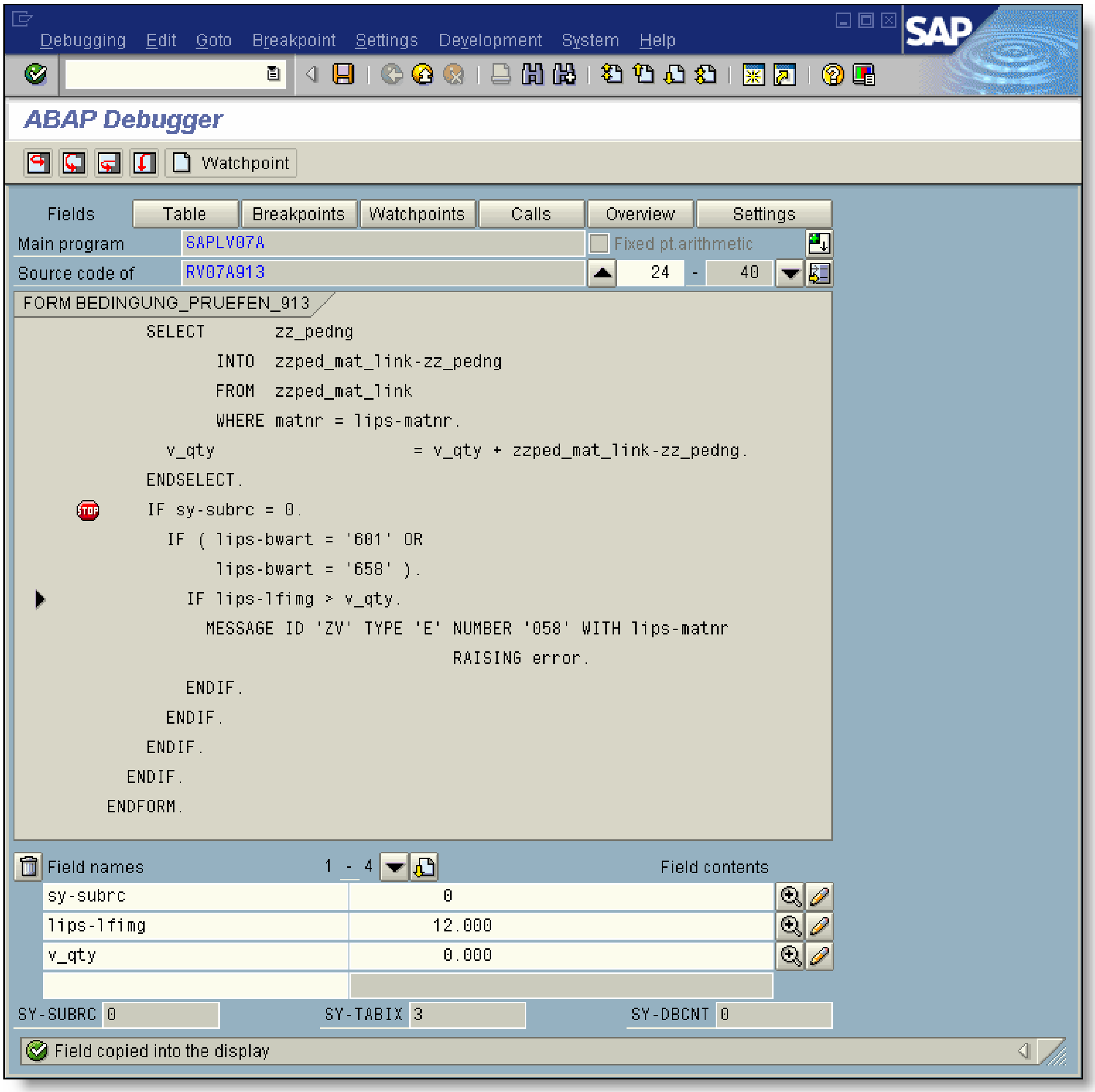
Field names 1 - 4	Field contents
FOFUN	13
TVFO-GRPNO_SY	013
TVFO-GRPNO	913

SY-SUBRC 0 SY-TABIX 1 SY-DBCNT 1

We can see that the function was called with a function code of ‘13’. The logic read table TVFO and determined the system and user defined routine numbers. The logic will build the form names with these numbers and execute the system form first and then the user form.

SD Subsequent Function Requirement Routines

The program is now in our custom routine. The inventory in our ‘Z’ table is not enough to cover the goods issue. The logic will issue an error using the MESSAGE statement. This will be trapped and the message will be displayed on the error log.



Main program SAPLV07A

☐ Fixed pt.arithmetic

Source code of RV07A913   24 - 40

FORM BEDINGUNG_PRUEFEN_913

```
    SELECT       zz_pedng
           INTO  zzped_mat_link-zz_pedng
           FROM  zzped_mat_link
           WHERE matnr = lips-matnr.
      v_qty                    = v_qty + zzped_mat_link-zz_pedng.
    ENDSELECT.
    IF sy-subrc = 0.
      IF ( lips-bwart = '601' OR
           lips-bwart = '658' ).
        IF lips-lfimg > v_qty.
          MESSAGE ID 'ZV' TYPE 'E' NUMBER '058' WITH lips-matnr
                                   RAISING error.
        ENDIF.
      ENDIF.
    ENDIF.
  ENDIF.
ENDFORM.
```

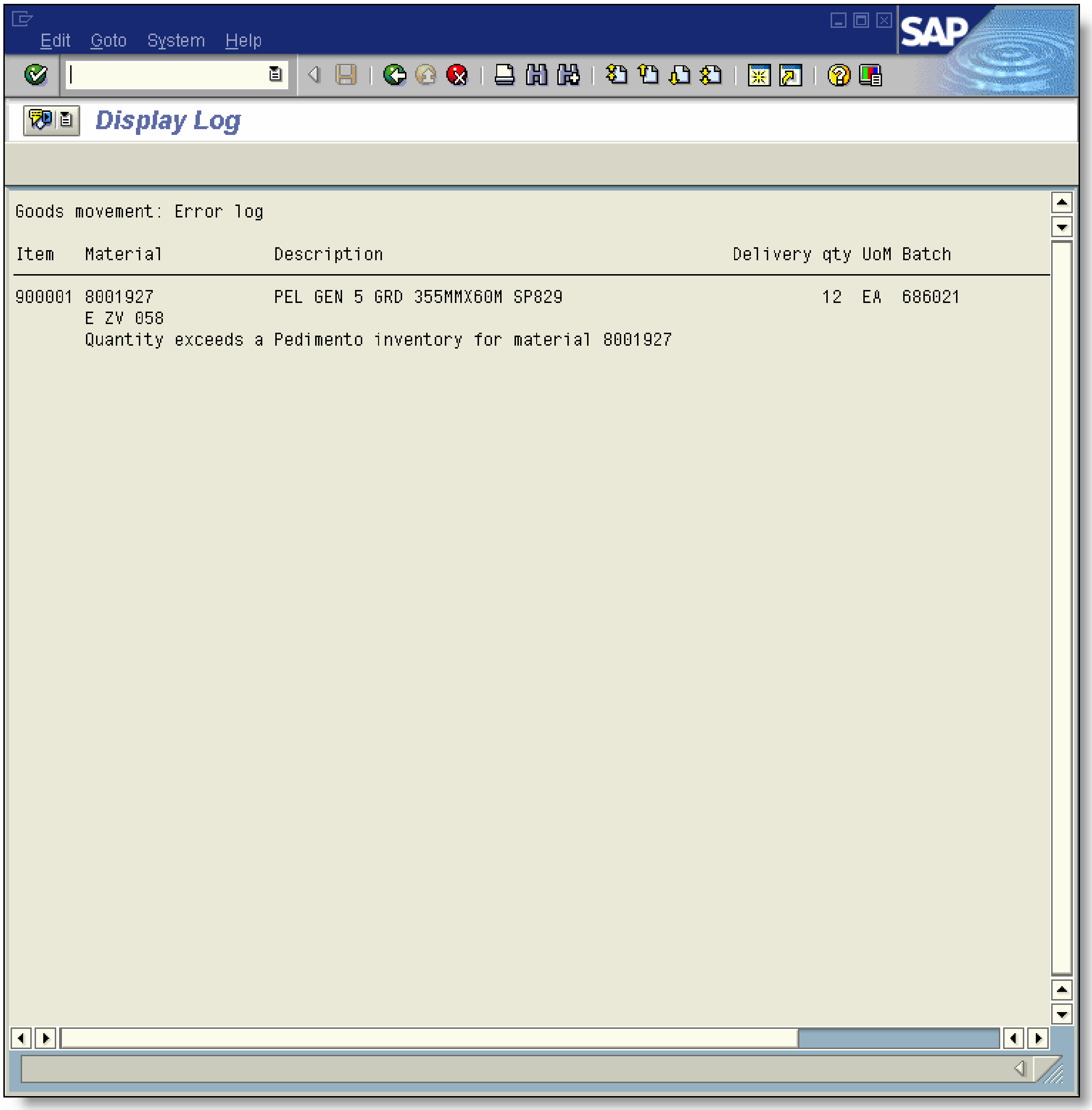
Field names 1 - 4   Field contents

```
sy-subrc        0
lips-lfimg      12.000
v_qty           0.000
```

SY-SUBRC 0   SY-TABIX 3   SY-DBCNT 0

Field copied into the display

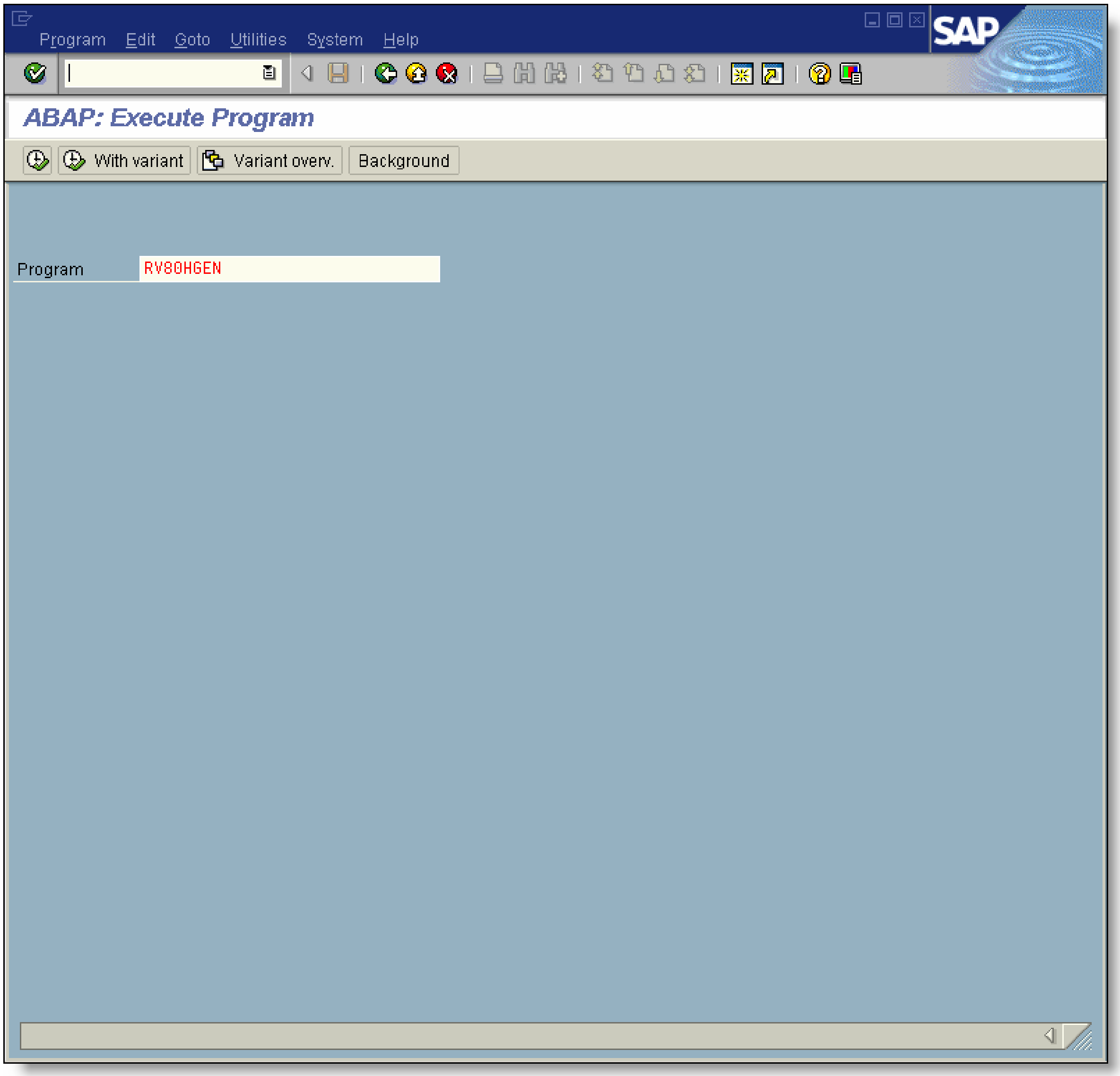
The error generated in the routine displays on the goods movement log. The goods issue did not post.



```
Goods movement: Error log

Item    Material           Description                                    Delivery qty UoM Batch

900001  8001927            PEL GEN 5 GRD 355MMX60M SP829                           12   EA  686021
        E ZV 058
        Quantity exceeds a Pedimento inventory for material 8001927
```

Step 6 Running the VOFM regeneration

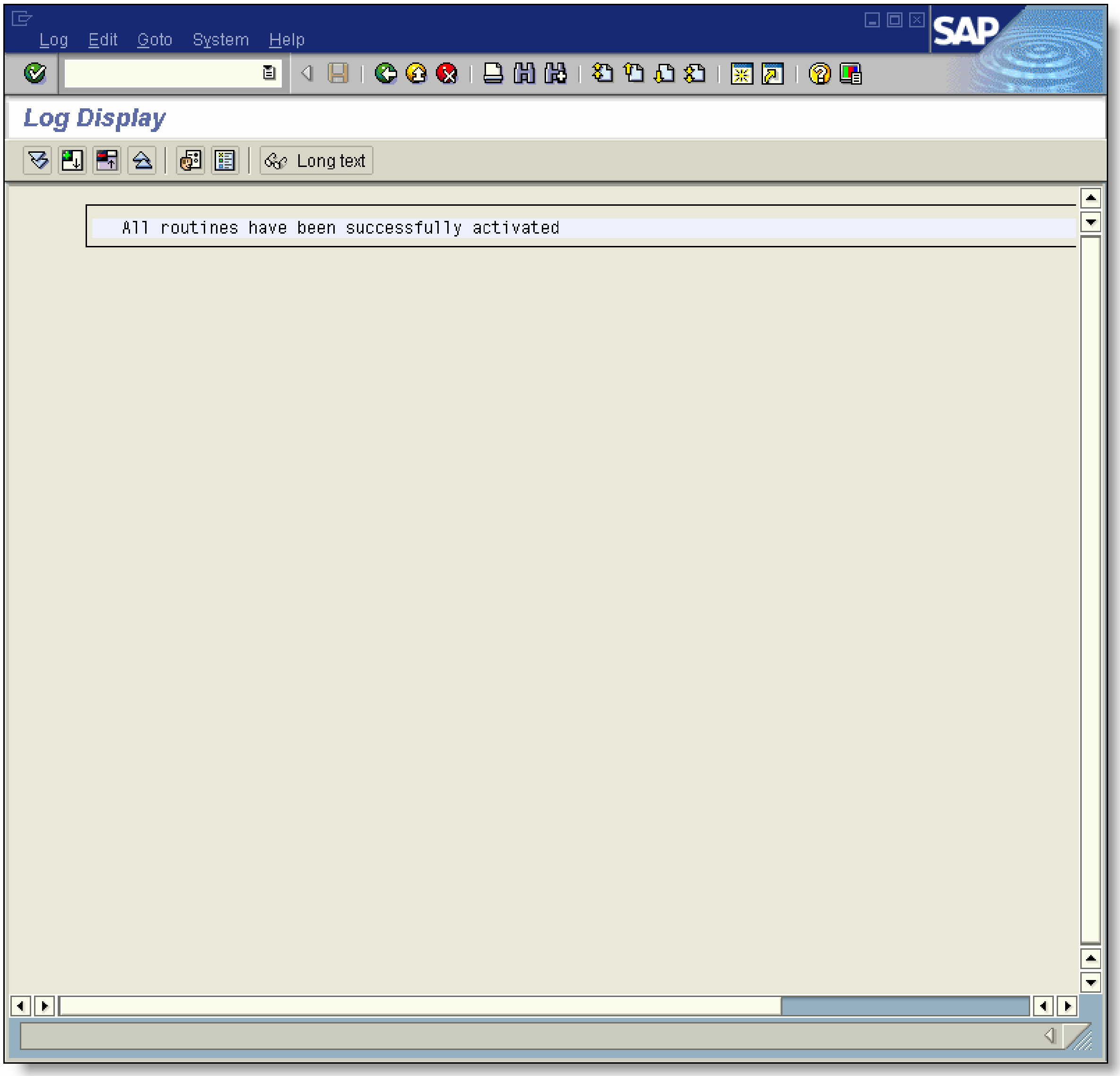
Whenever a VOFM routine is created, a special generation program (RV80HGEN) should be run in each system that the routine is moved to.



Program RV80HGEN

Note: This is a critical step that should not be overlooked. If this step is not executed, short dumps can result.

The RV80HGEN completed.



```
All routines have been successfully activated
```