Inside the VOFM Program

Technical Solution Guide

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

Inside the VOFM Program

Introduction

While many SAP specialists have used and created VOFM routines in R/3, few understand how these routines are technically implemented within the various programs and function groups in which they exist.

The purpose of this paper is to provide a technical explanation of how the VOFM transaction works. It is assumed that the reader has a working knowledge of the VOFM transaction and ABAP programming. The screen shots and data in this document were taken from an R/3 4.6C system.

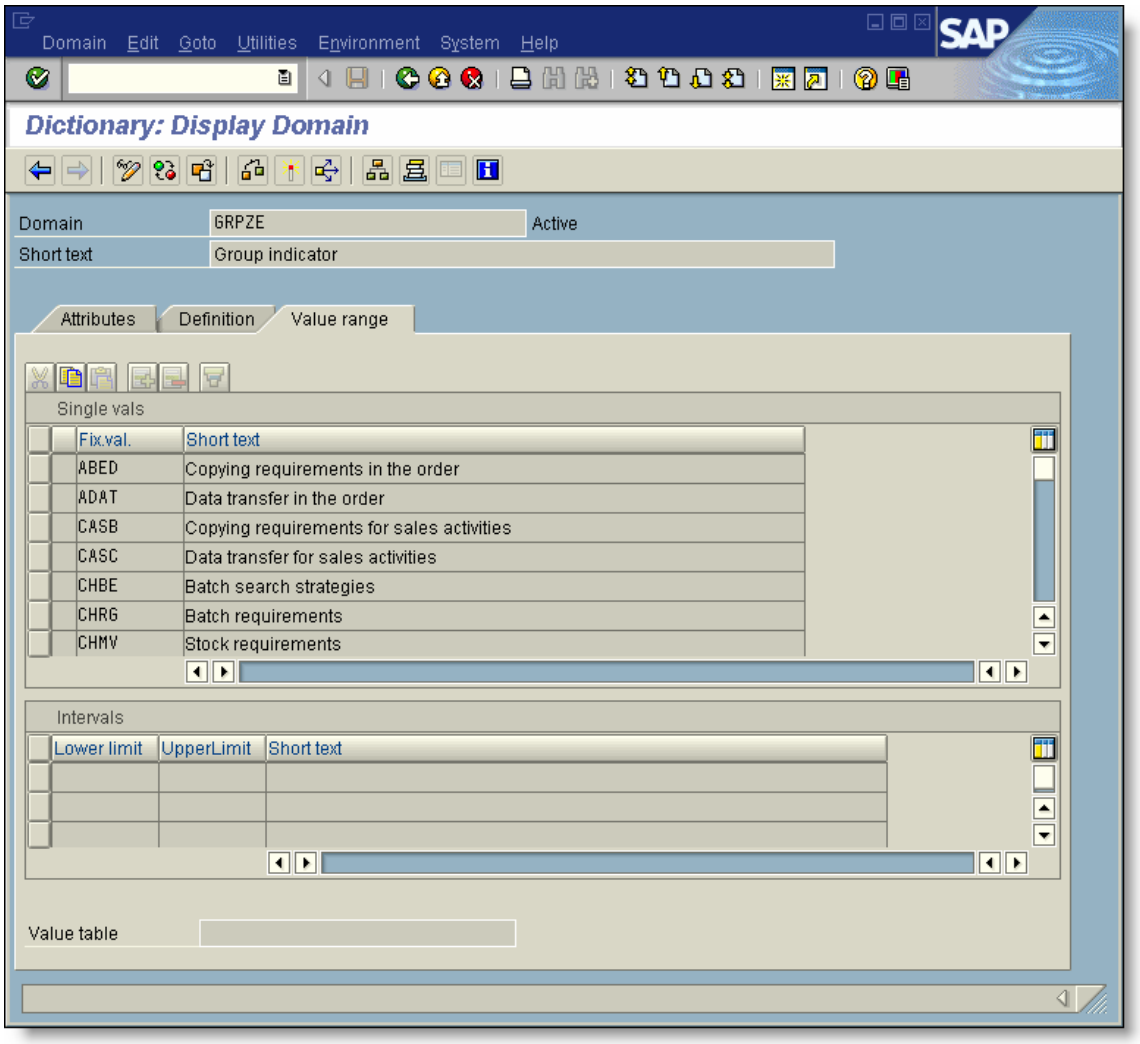
General structure of a VOFM routine

VOFM routines are ABAP FORM routines and they exist in a wide variety of programs and function groups. The routines are organized by a group code and there are many group codes available in the system. Each routine is contained in a unique ABAP include file. These include files are then 'included' in a parent include file. Finally, the parent include file is 'included' in the main program or function group. While this design may seem complicated, it allows the generation, activation and deactivation of these routines to occur without modifying the main program. When a routine is created, a new include is created and the associated parent include is updated. Activation and deactivation modify the parent include by adding or removing the include file that contains the routine.

Group Indicator Code

VOFM routines are organized using the Group Indicator Code. The VOFM program uses the group indicator to determine the program include and subroutine names when it dynamically creates these objects. The assignment of these names is hard coded in the program. The valid group indicator codes are defined in data base domain GRPZE using a value range.

Domain GRPZE



Group Indicator Codes contained in the VOFM Program

The following table contains the group indicator codes that are hard coded in the VOFM program. This list may be different depending on the release and does not correspond exactly to the domain value list.

Group	Description
ABED	Copying requirements in the order
ADAT	Data transfer in the order
CASB	Copying requirements for sales activities
CASC	Data transfer for sales activities
CHBE	Batch search strategies
CHMV	Stock requirements
CHRG	Batch requirements
EXKO	Export requirements
FBED	Copying requirements in the billing document
FDAT	Data transfer in the billing document
FOFU	Subsequent functions
LBED	Copying requirements in the delivery
LDAT	Data transfer in the delivery
LST1	Info blocks for RV reporting
LST1	Info blocks for RV reporting
MCA1	Work item requirements (MCA = Activity)
MCA2	Work item formulas (MCA = Activity)
MCB1	Stock control requirements
MCB2	Stock control formulas
MCE1	PURCHIS Requirements
MCE2	PURCHIS formulas
MCF1	SFIS requirements
MCF2	SFIS formulas
MCI1	PM requirements
MCI2	PM formulas
MCL1	WS requirements in LIS
MCL2	WS requirements in LIS
MCQ1	QMIS requirements
MCQ2	QMIS formulas
MCT1	TIS requirements
MCT2	TIS formulas
MCU1	UIS requirements
MCU2	UIS forms (Utility Information System)
MCV1	SIS requirements
MCV2	SIS formulas
MCW1	RIS requirements

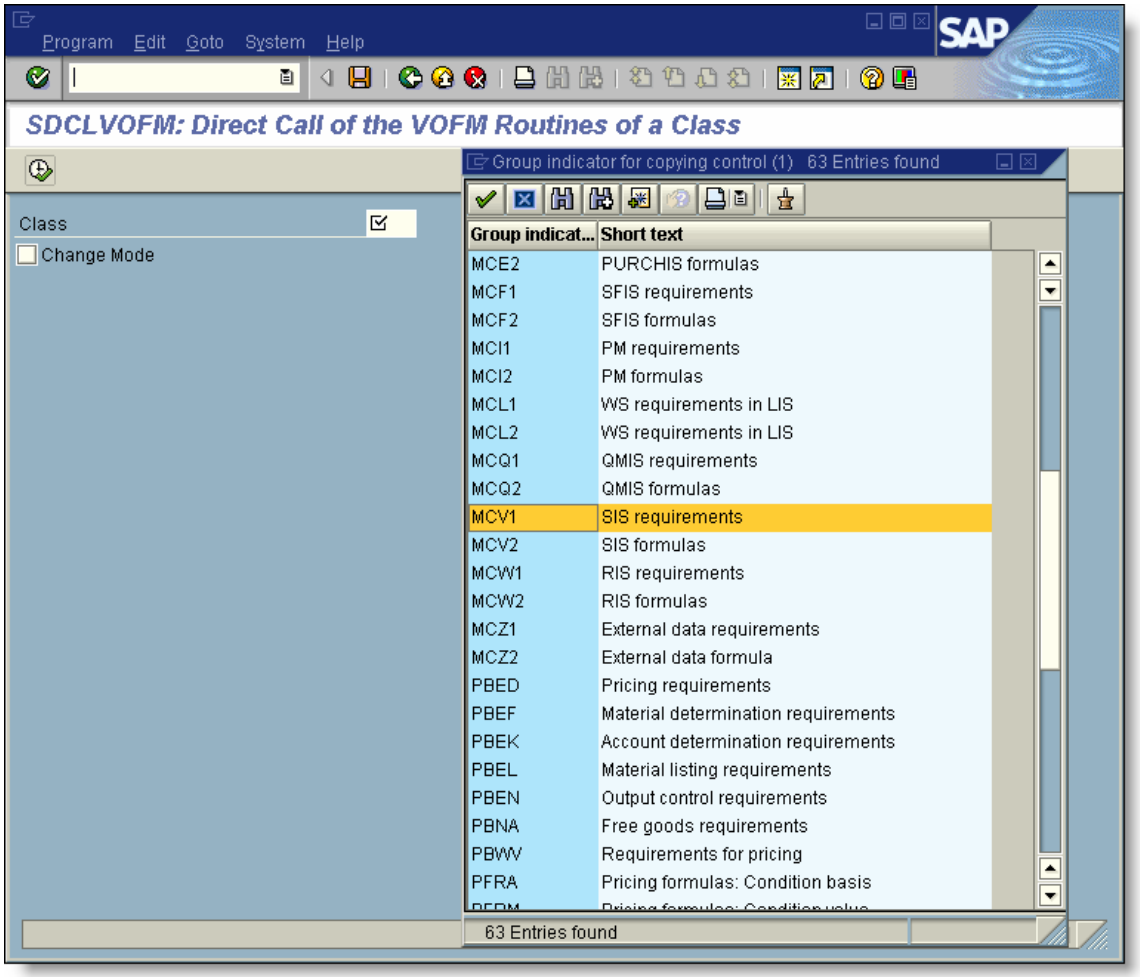
Group	Description
MCW2	RIS formulas
MCZ1	External data requirements
MCZ2	External data formula
PBED	Pricing requirements
PBEF	Material determination requirements
PBEK	Account determination requirements
PBEL	Material listing requirements
PBEN	Output control requirements
PBNA	Free goods requirements
PBWV	Requirements for pricing
PFRA	Pricing formulas: Condition basis
PFRM	Pricing formulas: Condition value
PFRS	Pricing formulas: Scale basis
PNAT	
PRUN	
PSTK	Structure of scale key for pricing
REAK	Archiving for orders
REKA	Archiving for sales activities
RELK	Archiving for deliveries
RERK	Archiving for billing documents
RISK	Risk management (form of payment guarantee)
TBED	Copying requirements for texts
TDAT	Data transfer involving texts
TNAM	Text names for word processing
TRAU	Data transfer transport
TXNM	Text names for copying modules (out of date)
VCAU	Authorization requirements for payment cards
VFCL	Multi-dimensional scales
VKMP	Credit check requirements
VSEL	Data transfer for shipping units

Routine Maintenance

Since VOFM routines are named after the transaction, most people access the routines using the VOFM transaction. Unfortunately, VOFM only provides access to a subset of the routines available. An alternative is transaction VOFN, which provides access to all routines in all groups.

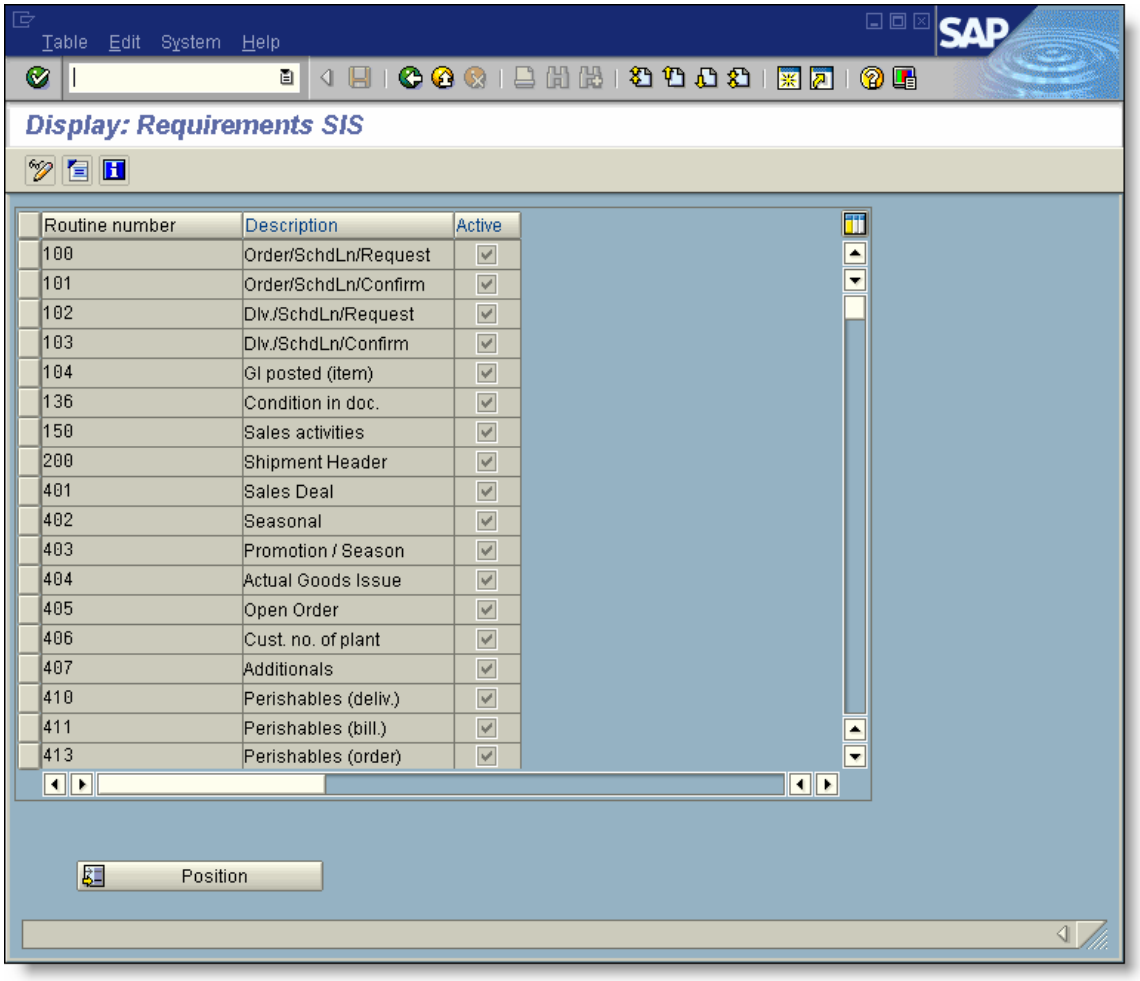
Transaction VOFN

VOFN is executed by group indicator (the screen refers to it as Class).



Inside the VOFM Program

Once the group indicator is selected, the detail screen looks and functions like the VOFM detail screen. It is actually the same program. The following example displays all of the routines in the MCV1 class (SIS Requirement Routines).

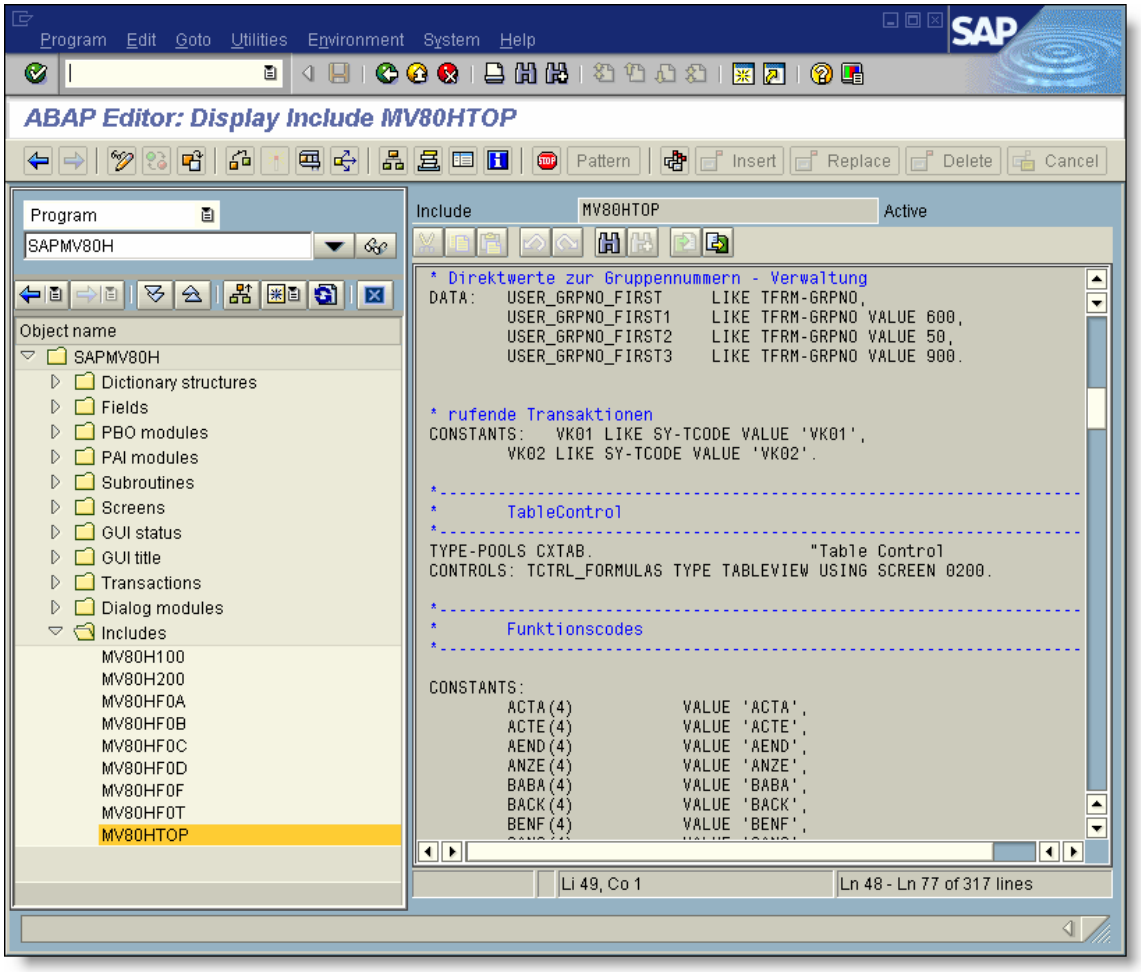


Building Custom Routines

When implementing custom VOFM routines, a question frequently asked is what routine numbers should be used. The answer is that it depends on the group indicator. The table in a subsequent section denotes the starting customer range for each group indicator (Customer Include column). The selection of an appropriate routine number is important because it determines the include file naming convention as well as the development class used when creating the routine. The customer routine ranges are hard coded in the program. The following screen shots detail how it works.

Customer Routine Ranges

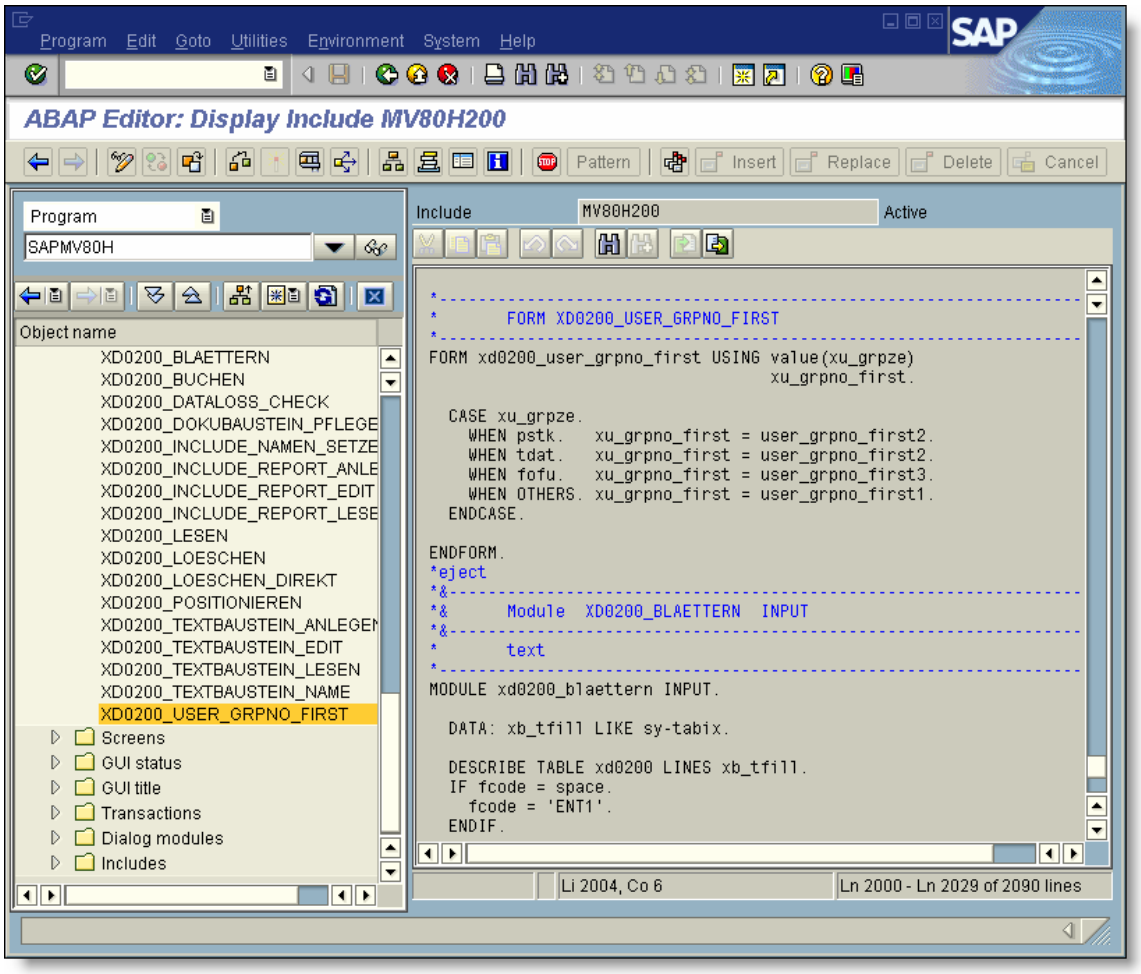
The following screen shot is from the global data section of the program. There are three possible starting customer ranges defined (USER_GRPNO_FIRST1, USER_GRPNO_FIRST2 and USER_GRPNO_FIRST3).



Customer Range Determination by Group Indicator

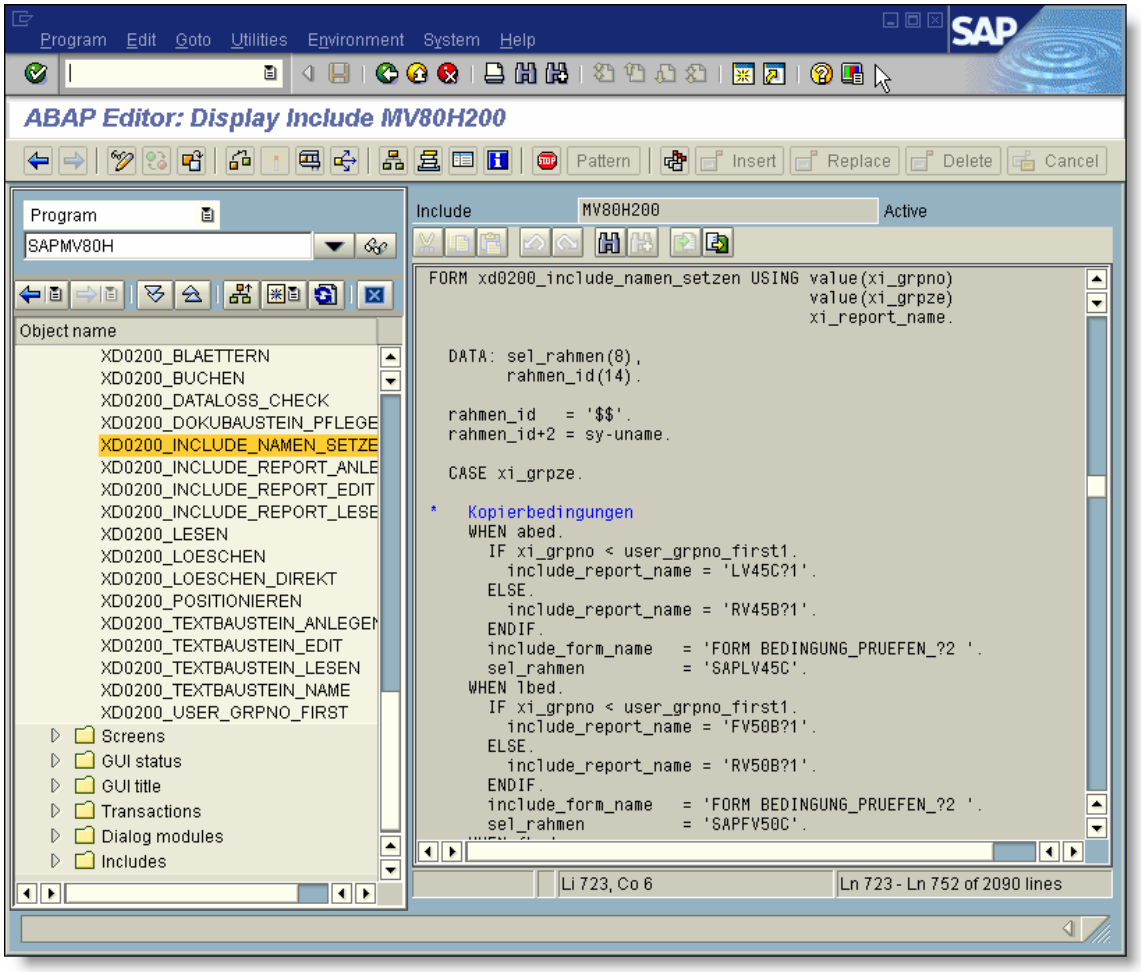
The group indicator is used by the logic that determines the starting number in the customer range. Essentially, the rules can be summarized as follows:

- For groups PSTK (Pricing Formula Group Structure) and TDAT (Data Transfer – Text), the customer range starts at 50.
- For group FOFU (Subsequent Functions) the customer range starts at 900.
- For all others, the customer range starts at 600.



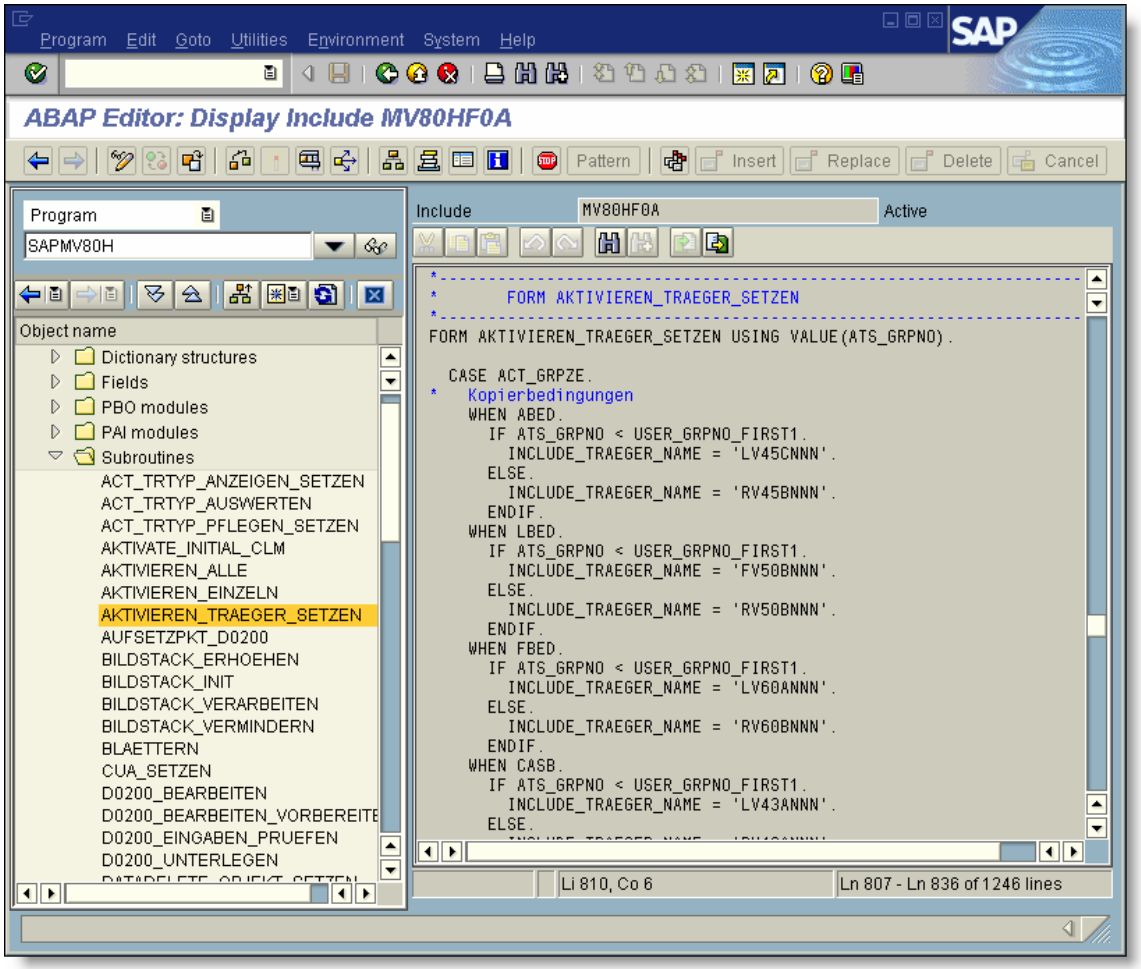
Include Name Determination by Group Indicator

The group indicator is also used to determine the include file name for routines. In the following example, Sales Order Copy Requirement routines (ABED) in the customer range will be named RV45Cxxx and routines in the SAP range will be named LV45Cxxx. In either case, the include will be created within function group SAPLV45C.



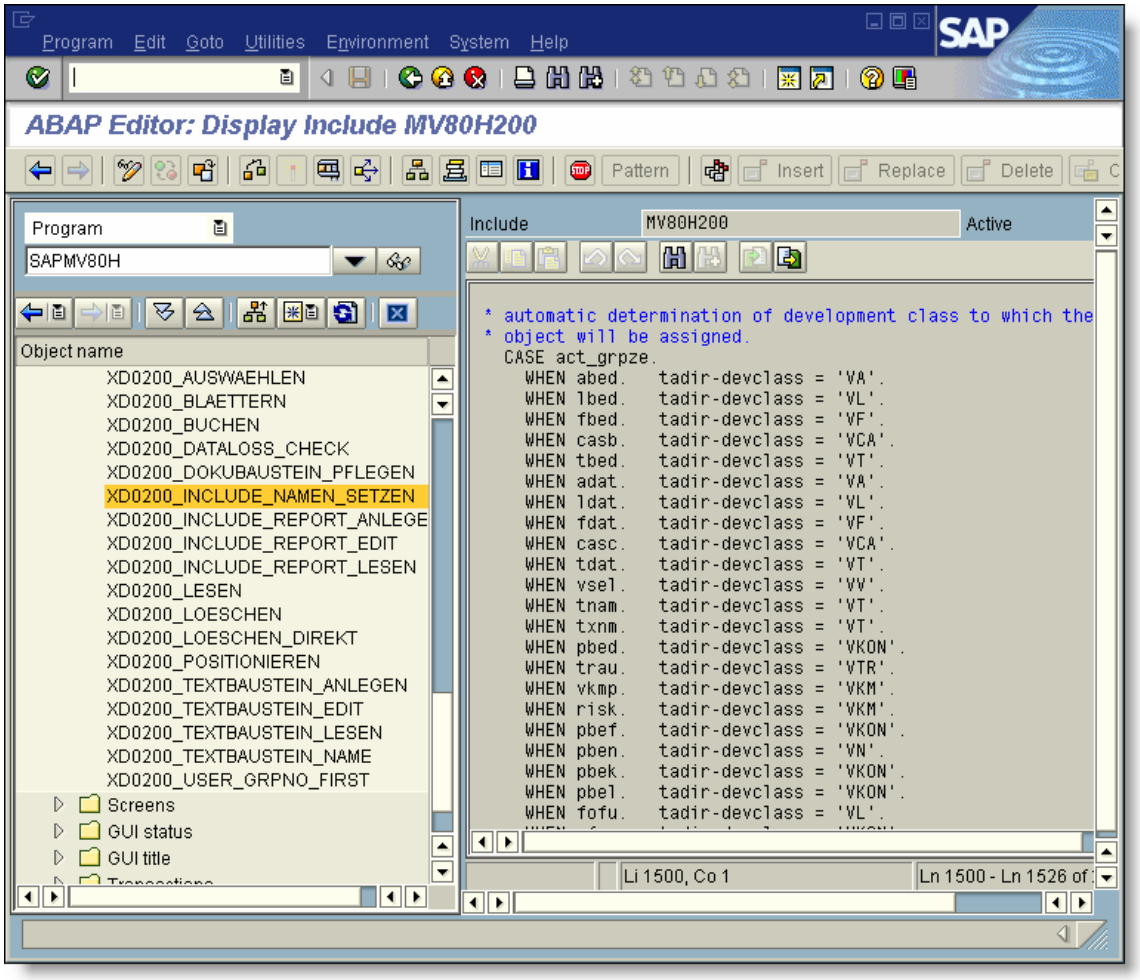
Parent Include Name Determination by Group Indicator

The group indicator is also used to determine the parent include file name for routines. In the following example, Sales Order Copy Requirement routines (ABED) in the customer range will be included in RV45BNNN and routines in the SAP range will be included in LV45CNNN.



Development Class Determination by Group Indicator

The group indicator is also used to determine the development class assigned to include files. In the following example, Sales Order Copy Requirement routines (ABED) will be assigned to development class VA. The same development class is assigned to both standard and customer created includes.



Source Include and Form Names Determined in the VOFM Program

The following table contains the naming convention used by the VOFM program to dynamically generate the source includes and form routines by Group Indicator. This list may be different depending on the release and does not correspond exactly to the domain value list.

Group	Dev Class	Main Program	Standard Include (Parent)	Standard Include	Customer Include (Parent)	Customer Include	Form Name
ABED	VA	SAPLV45C	LV45CNNN	LV45Cxxx	RV45BNNN	RV45B6xx	BEDINGUNG_PRUEFEN_xxx
ADAT	VA	SAPFV45C	FV45CNNN	FV45Cxxx	RV45CNNN	RV45C6xx	DATEN_KOPIEREN_xxx
BBYR	WBBY	SAPLV61N	FBBYNNNN	FBBYNxxx	RBBYNNNN	RBBYN6xx	BBYREQ_xxx
CASB	VCA	SAPLV43A	LV43ANNN	LV43Axxx	RV43ANNN	RV43A6xx	BEDINGUNG_PRUEFEN_xxx
CASC	VCA	SAPLV43A	FV44ANNN	FV44Axxx	RV44ANNN	RV44A6xx	DATEN_KOPIEREN_xxx
CHBE	VB	SAPL080M	L080MNNN	L080Mxxx	R080MNNN	R080M6xx	BED_SUCH_STRATEGIE_xxx
CHMV	MDBF	SAPLMDBF	LMDBFNNN	LMDBFxxx	RMDBFNNN	RMDBF6xx	BFMVS_xxx
CHRG	VB	SAPLV01F	LV01FNNN	LV01Fxxx	RV01FNNN	RV01F6xx	CHMVS_xxx
EXKO	VA	SAPLV52E	LV52ENNN	LV52Exxx	RV52ENNN	RV52E6xx	AUSFUHR_BED_PRUEFEN_xxx
FBED	VF	SAPLV60A	LV60ANNN	LV60Axxx	RV60BNNN	RV60B6xx	BEDINGUNG_PRUEFEN_xxx
FDAT	VF	SAPFV60C	FV60CNNN	FV60Cxxx	RV60CNNN	RV60C6xx	DATEN_KOPIEREN_xxx
FOFU	VL	SAPLV07A	LV07ANNN	LV07Axxx	RV07ACCC	RV07A9xx	BEDINGUNG_PRUEFEN_xxx
LBED	VL	SAPFV50C	FV50BNNN	FV50Bxxx	RV50BNNN	RV50B6xx	BEDINGUNG_PRUEFEN_xxx
LDAT	VL	SAPFV50C	FV50CNNN	FV50Cxxx	RV50CNNN	RV50C6xx	DATEN_KOPIEREN_xxx
LST1	VREP	SAPRV77S	RV77SNNN	RV77Sxxx	RV77UNNN	N/A	LST1_xxx
LST1	VREP	SAPRV77U	RV77SNNN	N/A	RV77UNNN	RV77U6xx	LST1_xxx
MCA1	MCWF	SAPFMCAF	SAPFMCAF	FMCA1xxx	RMCAUNNN	FMCA16xx	MCA1_xxx
MCA2	MCWF	SAPFMCAF	SAPFMCAF	FMCA2xxx	RMCAUNNN	FMCA26xx	MCA2_xxx
MCB1	MCS	SAPFMCBF	FMCB1NNN	FMCB1xxx	RMCB1NNN	FMCB16xx	MCB1_xxx
MCB2	MCS	SAPFMCBF	FMCB2NNN	FMCB2xxx	RMCB2NNN	FMCB26xx	MCB2_xxx
MCE1	MCS	SAPFMCEF	FMCE1NNN	FMCE1xxx	RMCE1NNN	FMCE16xx	MCE1_xxx

Group	Dev Class	Main Program	Standard Include (Parent)	Standard Include	Customer Include (Parent)	Customer Include	Form Name
MCE2	MCS	SAPFMCEF	FMCE2NNN	FMCE2xxx	RMCE2NNN	FMCE26xx	MCE2_xxx
MCF1	MCS	SAPFMCFF	FMCF1NNN	FMCF1xxx	RMCF1NNN	FMCF16xx	MCF1_xxx
MCF2	MCS	SAPFMCFF	FMCF2NNN	FMCF2xxx	RMCF2NNN	FMCF26xx	MCF2_xxx
MCI1	MCS	SAPFMCIF	SAPFMCIF	FMCI1xxx	RMCIUNNN	FMCI16xx	MCI1_xxx
MCI2	MCS	SAPFMCIF	SAPFMCIF	FMCI2xxx	RMCIUNNN	FMCI26xx	MCI2_xxx
MCL1	VL	SAPFMCLF	SAPFMCLF	FMCL1xxx	RMCLUNNN	FMCL16xx	MCL1_xxx
MCL2	VL	SAPFMCLF	SAPFMCLF	FMCL2xxx	RMCLUNNN	FMCL26xx	MCL2_xxx
MCQ1	MCS	SAPFMCQF	SAPFMCQF	FMCQ1xxx	RMCQUNNN	FMCQ16xx	MCQ1_xxx
MCQ2	MCS	SAPFMCQF	SAPFMCQF	FMCQ2xxx	RMCQUNNN	FMCQ26xx	MCQ2_xxx
MCT1	MCS	SAPFMCTF	FMCT1NNN	FMCT1xxx	RMCT1NNN	FMCT16xx	MCT1_xxx
MCT2	MCS	SAPFMCTF	FMCT2NNN	FMCT2xxx	RMCT2NNN	FMCT26xx	MCT2_xxx
MCU1	MCS	SAPFMCUF	FMCU1NNN	FMCU1xxx	RMCU1NNN	FMCU16xx	MCU1_xxx
MCU2	MCS	SAPFMCUF	FMCU2NNN	FMCU2xxx	RMCU2NNN	FMCU26xx	MCU2_xxx
MCV1	MCS	SAPFMCVF	FMCV1NNN	FMCV1xxx	RMCV1NNN	FMCV16xx	MCV1_xxx
MCV2	MCS	SAPFMCVF	FMCV2NNN	FMCV2xxx	RMCV2NNN	FMCV26xx	MCV2_xxx
MCW1	MCS	SAPFMCWF	FMCW1NNN	FMCW1xxx	RMCW1NNN	FMCW16xx	MCW1_xxx
MCW2	MCS	SAPFMCWF	FMCW2NNN	FMCW2xxx	RMCW2NNN	FMCW26xx	MCW2_xxx
MCZ1	MCS	SAPFMCZF	SAPFMCZF	FMCZ1xxx	RMCZUNNN	FMCZ16xx	MCZ1_xxx
MCZ2	MCS	SAPFMCZF	SAPFMCZF	FMCZ2xxx	RMCZUNNN	FMCZ26xx	MCZ2_xxx
PBED	VKON	SAPLV61A	LV61ANNN	LV61Axxx	RV61ANNN	RV61A6xx	KOBED_xxx, KOBEV_xxx
PBEF	VKON	SAPLV61D	LV61DNNN	LV61Dxxx	RV61DNNN	RV61D6xx	KOBED_xxx, KOBEV_xxx
PBEK	VKON	SAPLV61C	LV61CNNN	LV61Cxxx	RV61CNNN	RV61C6xx	KOBED_xxx, KOBEV_xxx
PBEL	VKON	SAPLV61G	LV61GNNN	LV61Gxxx	RV61GNNN	RV61G6xx	KOBED_xxx, KOBEV_xxx
PBEN	VN	SAPLV61B	LV61BNNN	LV61Bxxx	RV61BNNN	RV61B6xx	KOBED_xxx, KOBEV_xxx
PBNA	VKON	SAPLV61N	FV62NNNN	FV62Nxxx	RV62NNNN	RV62N6xx	KOBED_xxx, KOBEV_xxx

Group	Dev Class	Main Program	Standard Include (Parent)	Standard Include	Customer Include (Parent)	Customer Include	Form Name
PBWV	WVKP	SAPLWVK1	LWVK1NNN	LWVK1xxx	RWVK1NNN	RWVK16xx	WVBED_xxx
PFRA	VKON	SAPLV61A	FV63ANNN	FV63Axxx	RV63ANNN	RV63A6xx	FRM_KOND_BASIS_xxx
PFRM	VKON	SAPLV61A	FV64ANNN	FV64Axxx	RV64ANNN	RV64A6xx	FRM_KONDI_WERT_xxx
PFRS	VKON	SAPLV61A	FV62ANNN	FV62Axxx	RV62ANNN	RV62A6xx	FRM_STAFFELBAS_xxx
PNAT		SAPLV61N	LV61NNNN	LV61Nxxx	RV61NNNN	RV61N6xx	FRM_RECHENR_xxx
PRUN		SAPMV13A	FV13ANNN	FV13Axxx	RV13ANNN	RV13Z6xx	FRM_RUNDUNG_xxx
PSTK	VKON	SAPLV61A	FV65ANNN	FV65Axxx	RV65ANNN	RV65A5xx	FRM_GRUPPENKEY_xxx
REAK	VZ	S3VBAKWR	SDVBAKSP	AKSAPxxx	SDVBAKUS	AKUSR6xx	SD_VBAK_xxx
REKA	VZ	SDVBKAWR	SDVBKASP	KASAPxxx	SDVBKAUS	KAUSR6xx	SD_VBKA_xxx
RELK	VZ	S3LIKPWR	SDLIKPSP	DKSAPxxx	SDLIKPUS	DKUSR6xx	SD_LIKP_xxx
RERK	VZ	S3VBRKWR	SDVBRKSP	RKSAPxxx	SDVBRKUS	RKUSR6xx	SD_VBRK_xxx
RISK	VKM	SAPLVKMP	LRISKNNN	LRISKxxx	RRISKNNN	RRISK6xx	ABSICHERUNG_PRUEFEN_xxx
TBED	VT	SAPLV45T	LV45TNNN	LV45Txxx	RV45TNNN	RV45T6xx	BEDINGUNG_PRUEFEN_xxx
TDAT	VT	SAPLV45T	LV45TENN	LV45TExxx	RV45TENN	RV45TE5xx	DATEN_KOPIEREN_xxx
TNAM	VT	SAPLV70T	LV70TNNN	LV70Txxx	RV70TNNN	RV70T6xx	TEXTNAME_xxx
TRAU	VTR	SAPFV56C	FV56CNNN	FV56Cxxx	RV56CNNN	RV56C6xx	DATEN_KOPIEREN_xxx
TXNM	VT	SAPLV45T	FV46TNNN	FV46Txxx	RV46TNNN	RV46T6xx	TEXTNAME_COPY_xxx
VCAU	VFC	SAPLV21F	DVCAUNNN	DVCAUxxx	RVCAUNNN	RVCAU6xx	REQUIREMENT_CHECK_xxx
VFCL	VFSC_1	SAPLV57A	LV57ANNN	FV57Axxx	RV57ANNN	RV57A6xx	SD_VFCL_xxx
VKMP	VKM	SAPLVKMP	LVKMPNNN	LVKMP	RVKMPNNN	RVKMP6xx	BEDINGUNG_PRUEFEN_xxx
VSEL	VV	SAPLV51H	FV51ANNN	FV51Axxx	RV51ANNN	RV51A6xx	DATEN_KOPIEREN_xxx

Routine Activation

Understanding what actually happens when VOFM routines are activated or deactivated is useful knowledge, especially when problems occur and debugging is required.

In the following screen shots, routine 900 will be deactivated to demonstrate the process.

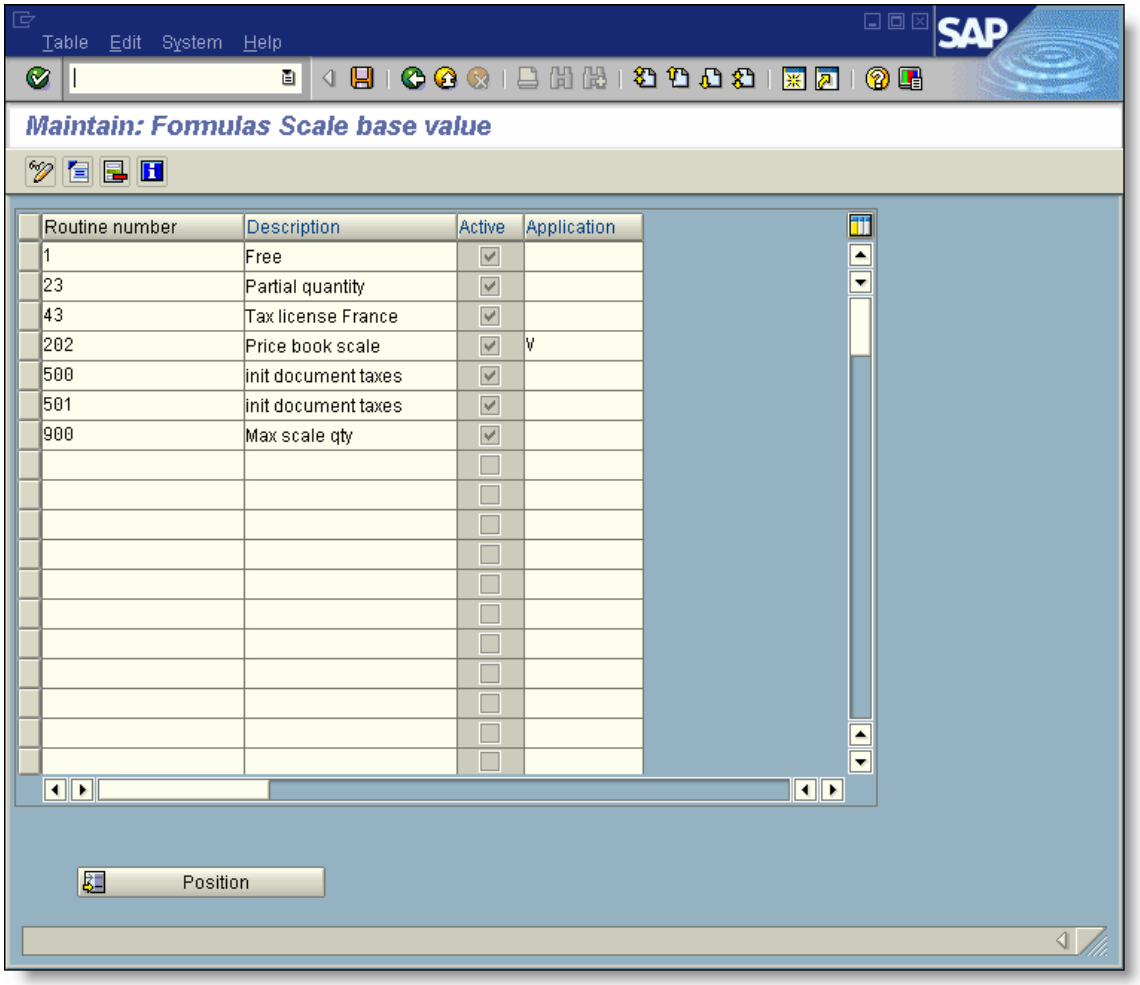
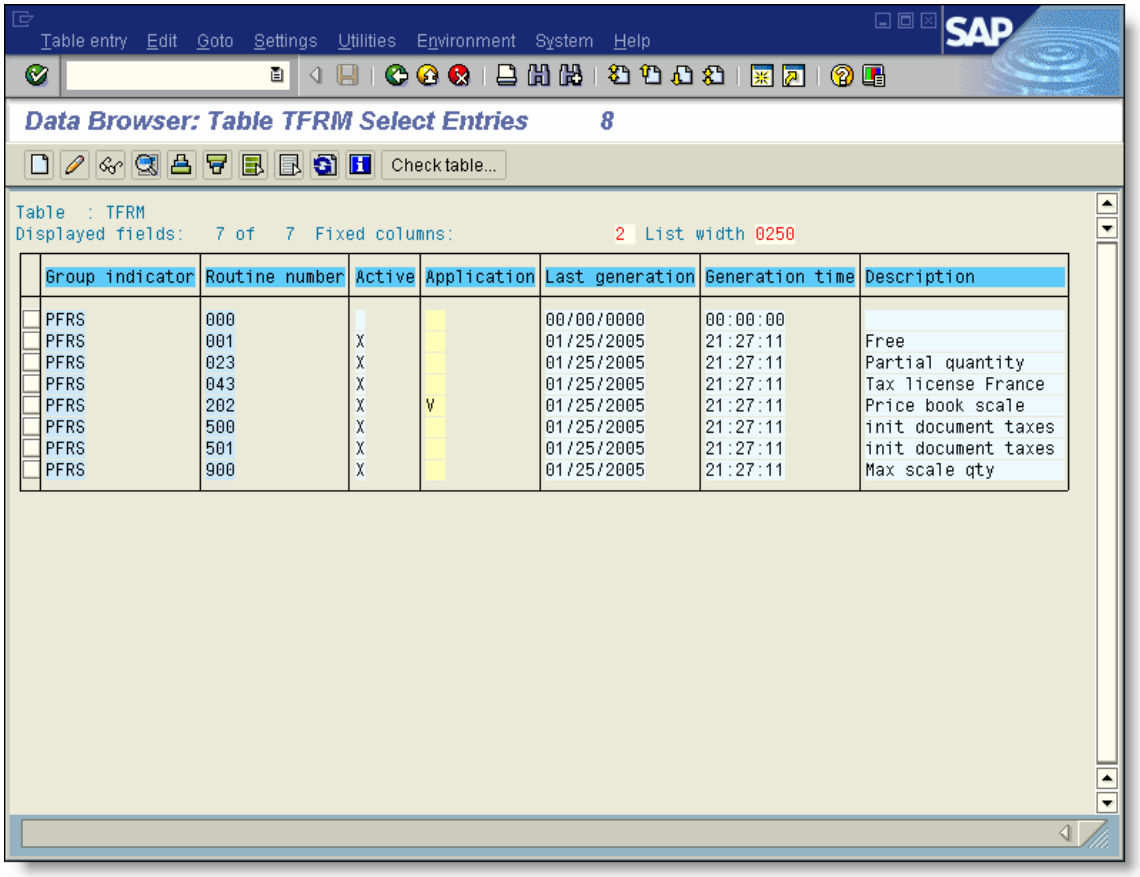


Table TFRM

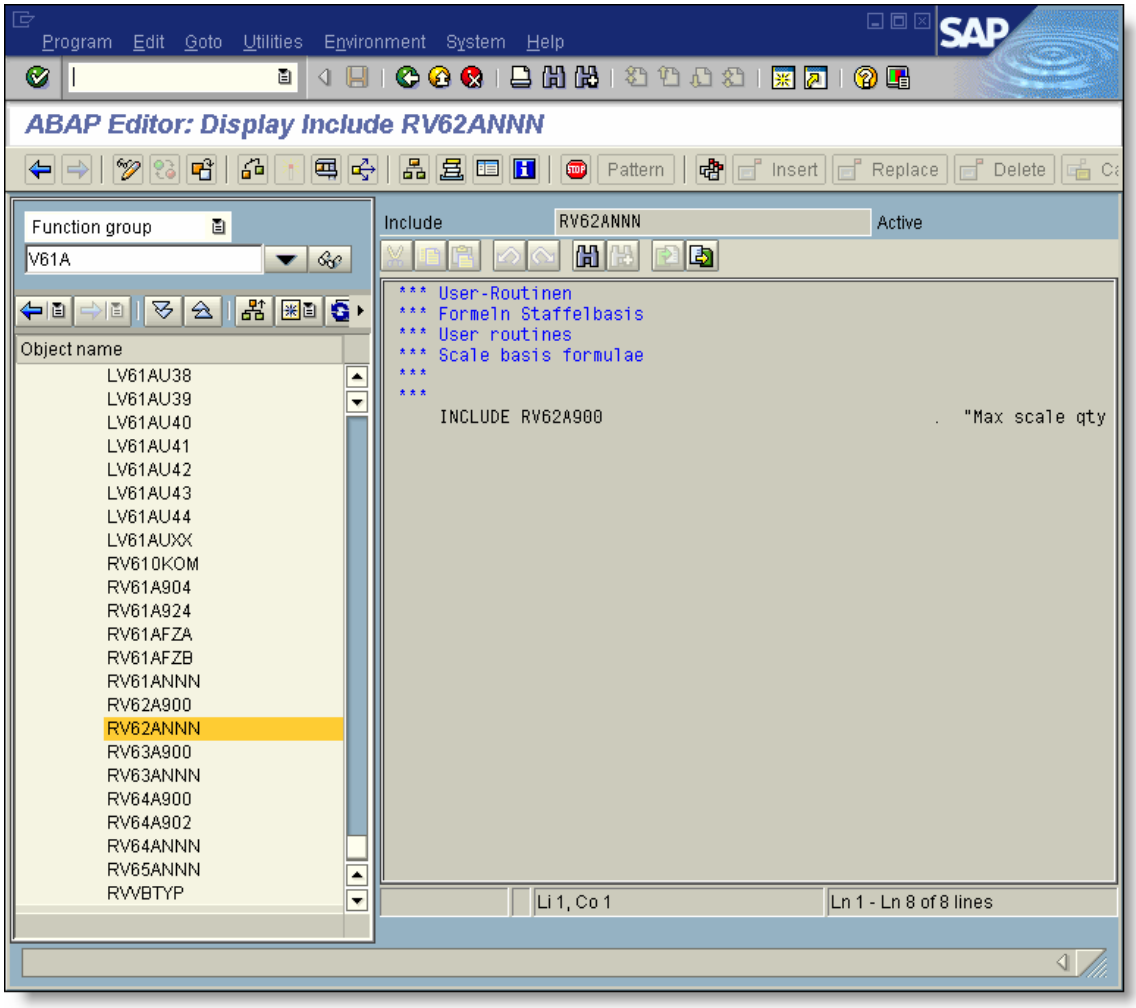
Table TFRM contains one record for each VOFM routine defined in the system. The following list displays all routines in group PFRS (Pricing Scale Base Formulas). For each record, the activation status and date/time stamp of activation is recorded. As denoted in this table, routine 900 is active.



Group indicator	Routine number	Active	Application	Last generation	Generation time	Description
PFRS	000			00/00/0000	00:00:00	
PFRS	001	X		01/25/2005	21:27:11	Free
PFRS	023	X		01/25/2005	21:27:11	Partial quantity
PFRS	043	X		01/25/2005	21:27:11	Tax license France
PFRS	202	X	V	01/25/2005	21:27:11	Price book scale
PFRS	500	X		01/25/2005	21:27:11	init document taxes
PFRS	501	X		01/25/2005	21:27:11	init document taxes
PFRS	900	X		01/25/2005	21:27:11	Max scale qty

Parent Include of the Active Routine

The group indicator for our sample routine is PFRS. Using the tables in the previous sections (or by looking at the program), we determine that the routine exists in function group V61A, the routine is contained in include RV62A900 and the parent include is RV62ANNN.



Deactivating the Routine

The routine is deactivated using the VOFM transaction.

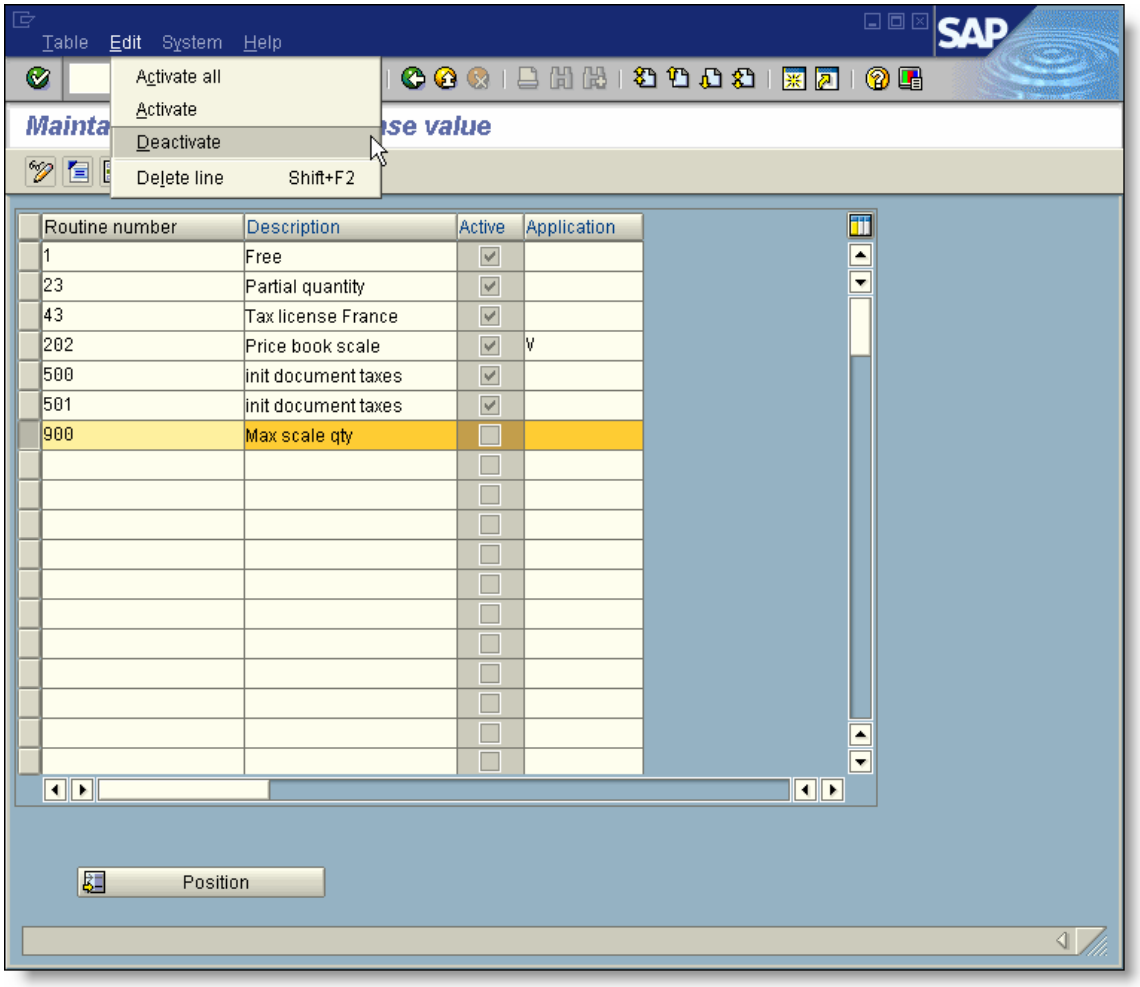
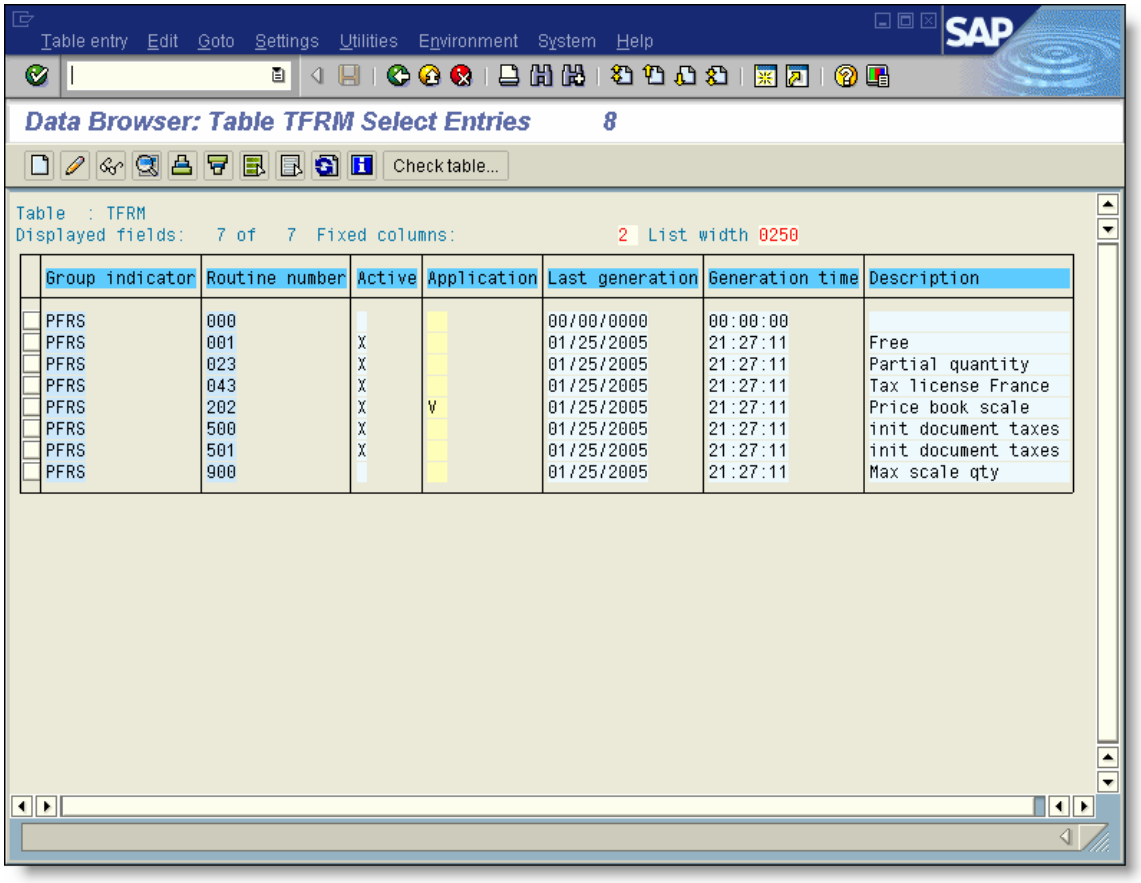


Table TFRM Updated

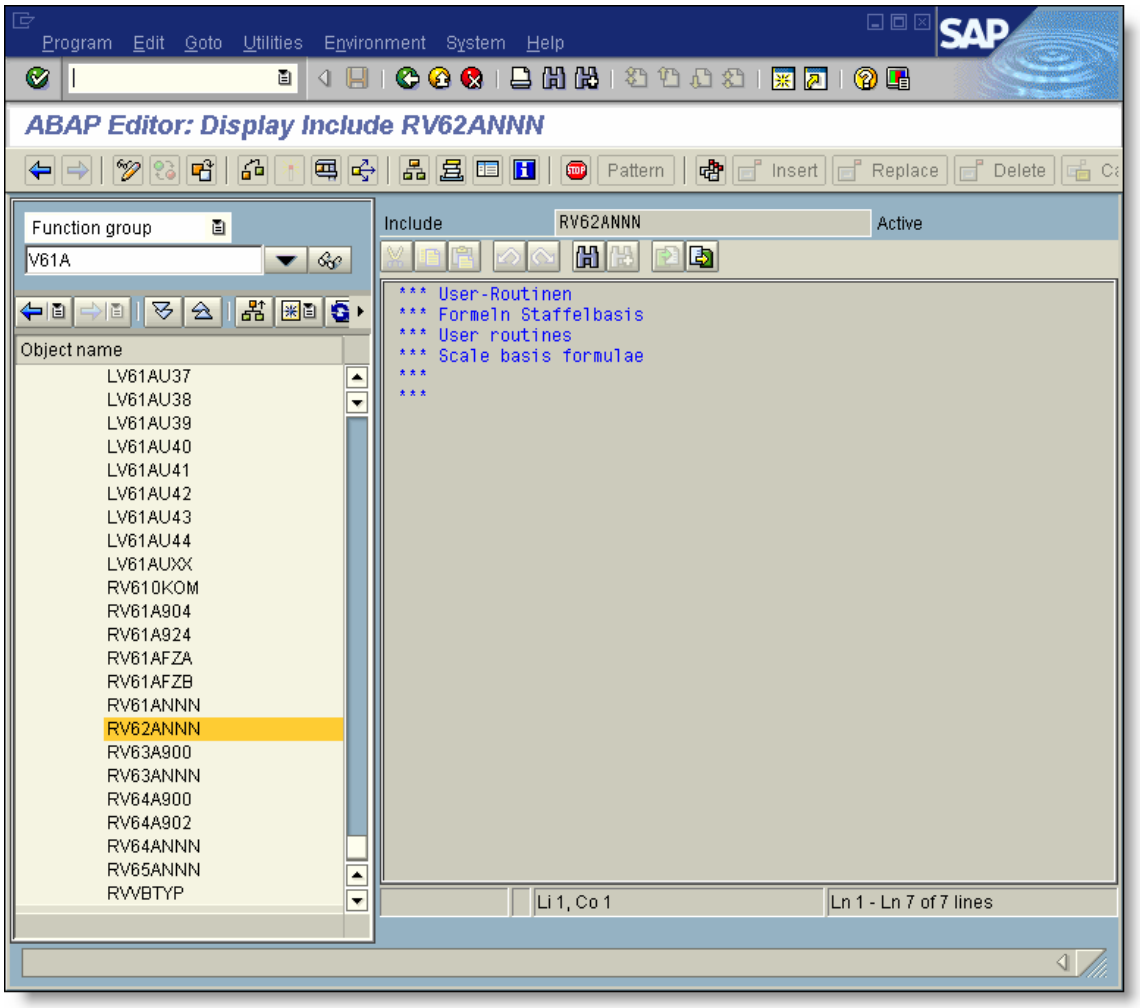
After deactivation, the record in table TFRM is updated.



Group indicator	Routine number	Active	Application	Last generation	Generation time	Description
PFRS	000			00/00/0000	00:00:00	
PFRS	001	X		01/25/2005	21:27:11	Free
PFRS	023	X		01/25/2005	21:27:11	Partial quantity
PFRS	043	X		01/25/2005	21:27:11	Tax license France
PFRS	202	X	V	01/25/2005	21:27:11	Price book scale
PFRS	500	X		01/25/2005	21:27:11	init document taxes
PFRS	501	X		01/25/2005	21:27:11	init document taxes
PFRS	900			01/25/2005	21:27:11	Max scale qty

Parent Include of the inactive Routine

After deactivation, the parent include no longer contains the routine include. The parent include is never deleted.

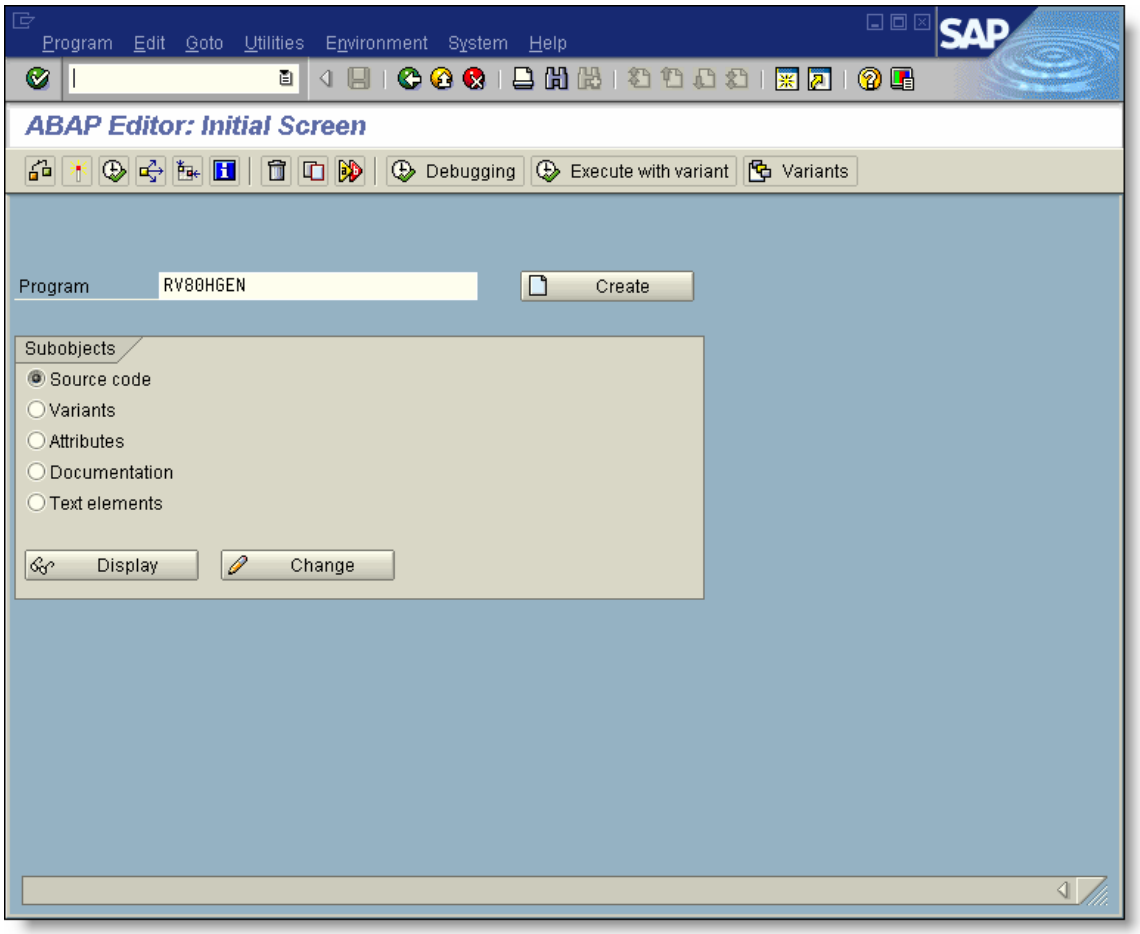


Utility Programs

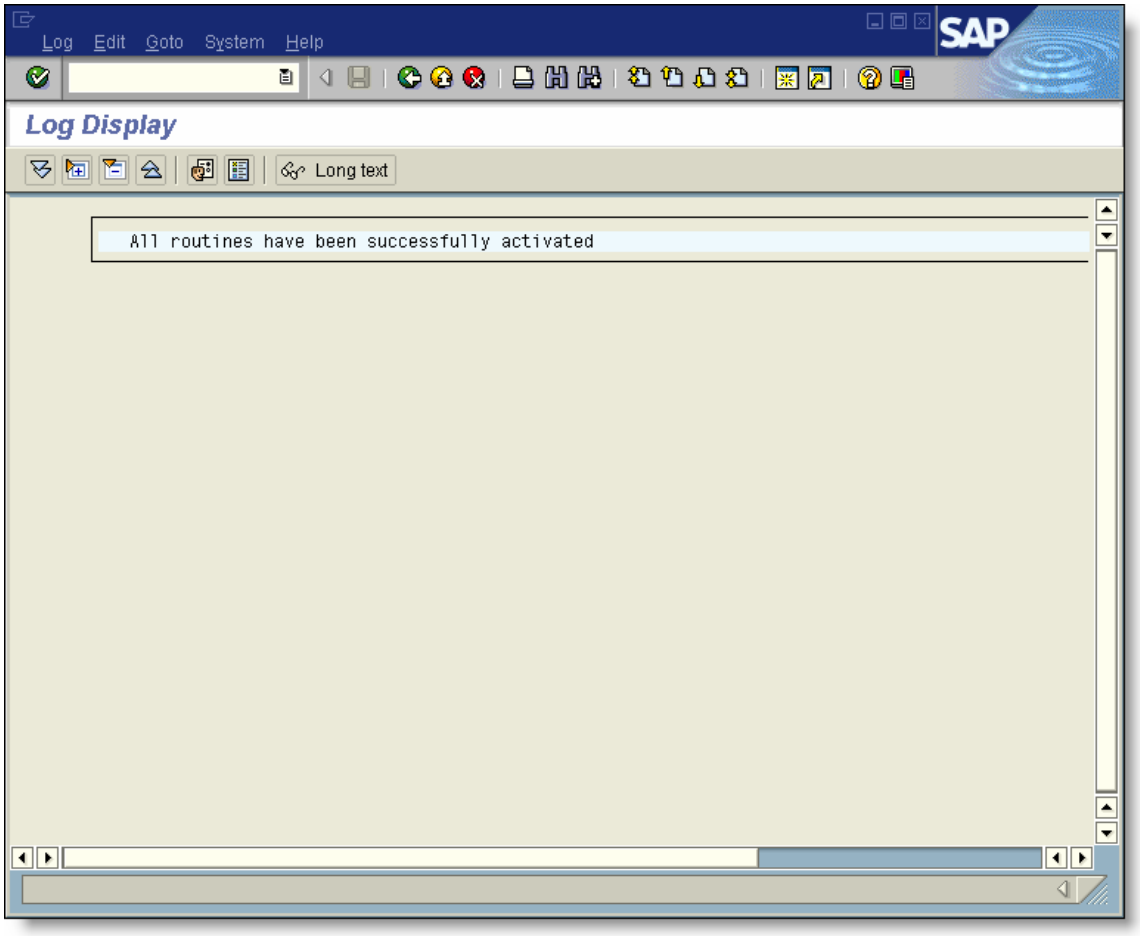
Transport issues with VOFM routines are so common, that SAP has provided utility programs to aid in the repair process.

Program RV80HGEN

Program RV80HGEN can be used to re-generate all VOFM routines. Whenever VOFM routines are transported, this program should be executed in all target systems. It is possible to include an automatic execution of the program in the transport request by applying OSS note 598475.



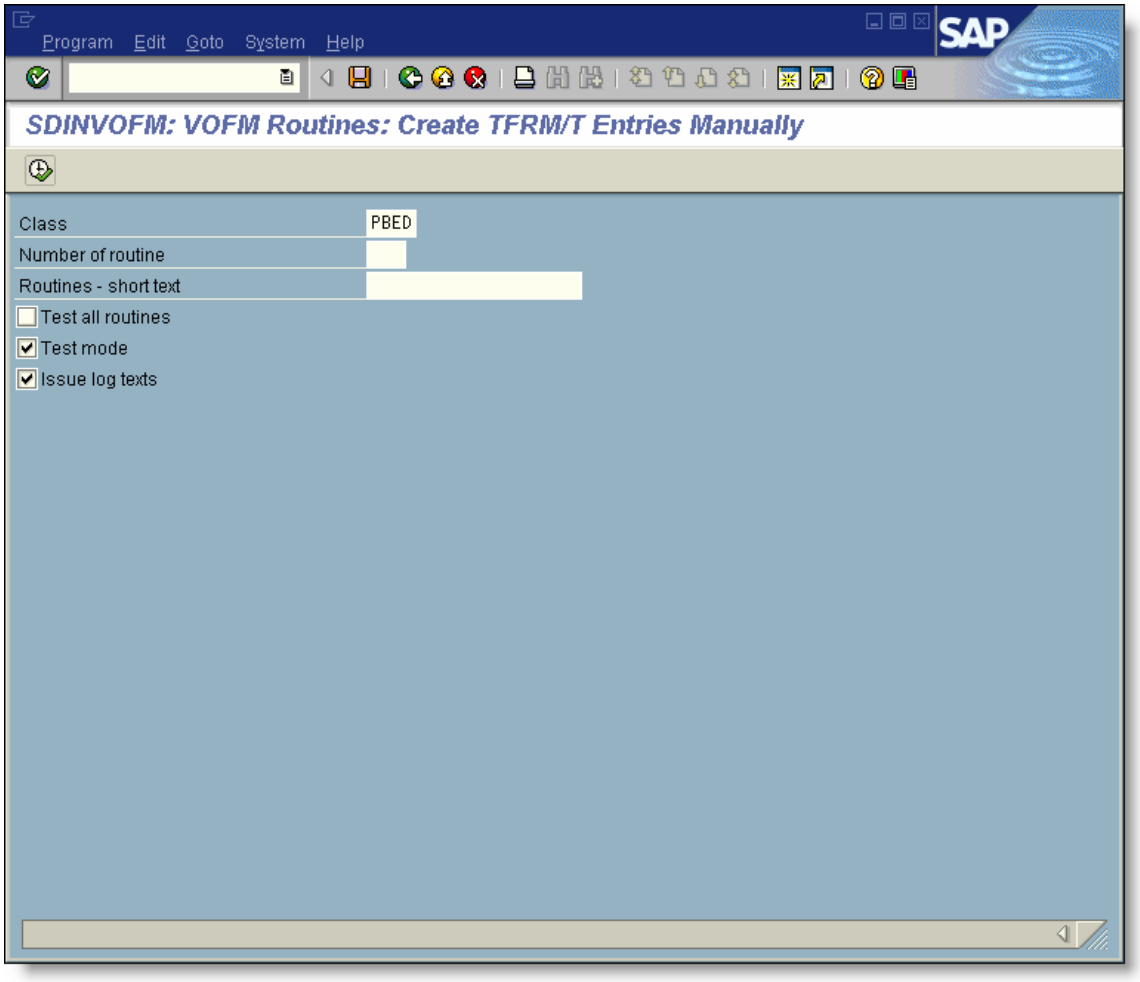
The output report for RV80HGEN denotes that all routines were activated.



Program SDINVOFM

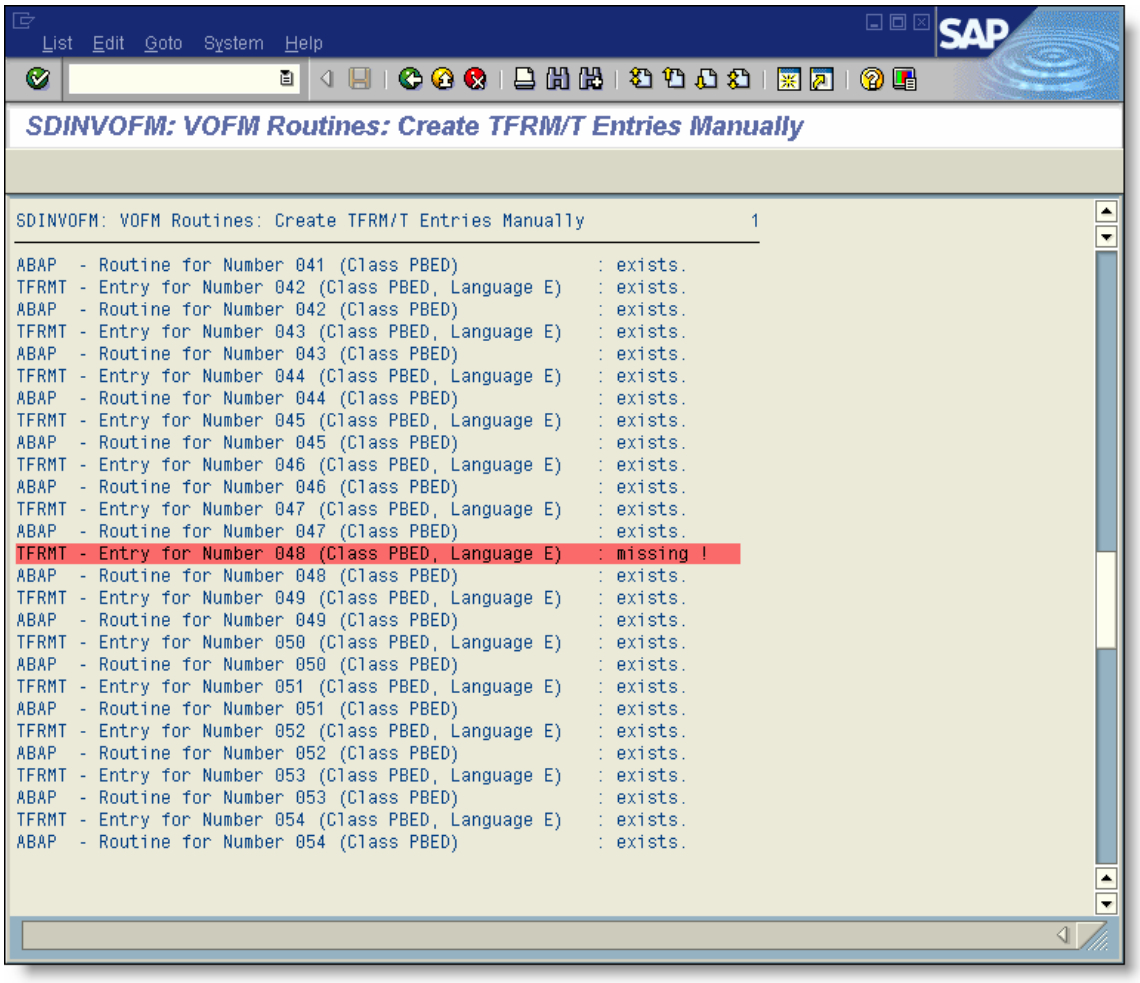
Program SDINVOFM can be used to identify and repair missing table entries in the VOFM control tables TFRM and TFRMT. This program is usually executed after RV80HGEN.

In the following example, the program will analyze the routines in the PBED group (Pricing Requirement Routines).



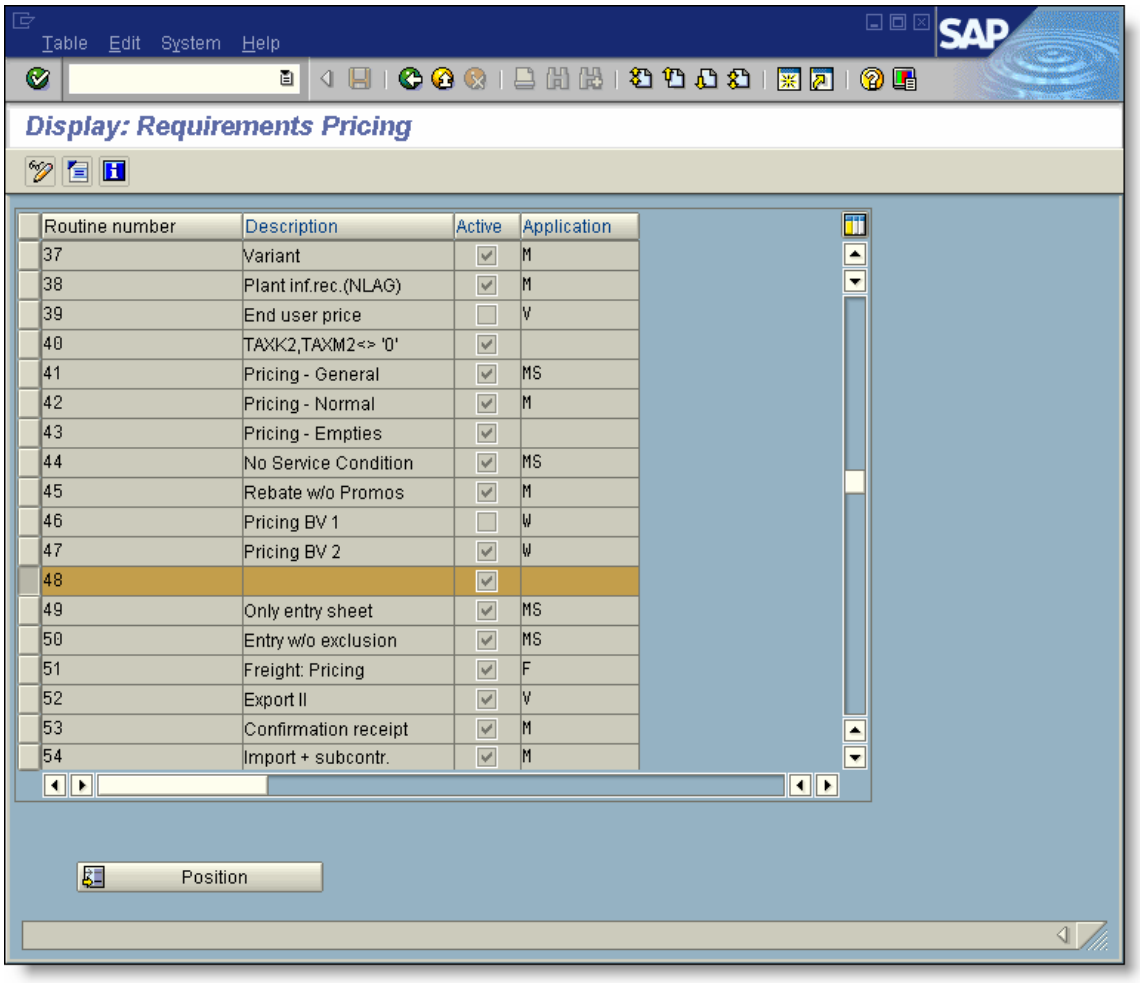
Inside the VOFM Program

The output of the analysis is a report that denotes the status of all items analyzed. In this example, the description (Table TFRMT) is missing for pricing requirement routine 048.



Inside the VOFM Program

Looking at pricing requirement routine 048 via the VOFM transaction, the description is indeed missing and would be corrected here.

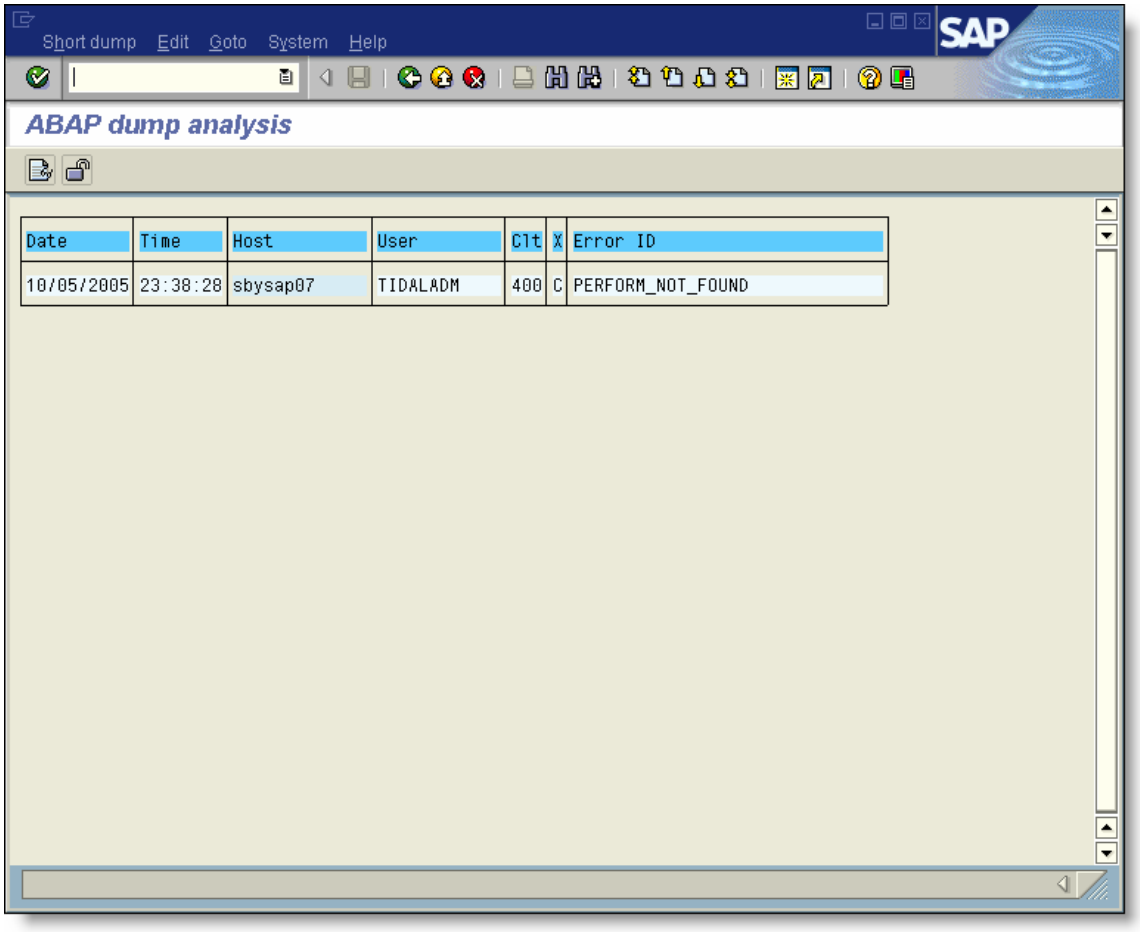


Transport Problems

Problems frequently occur when VOFM routines are transported to test and production instances. It is not uncommon to experience short dumps in these target systems. These problems are not the result of faulty programming or configuration. There has always been a transport issue for VOFM routines that occurs on occasion. The following sections illustrate the type of problems that can occur when VOFM routines do not transport correctly and the procedure to correct them.

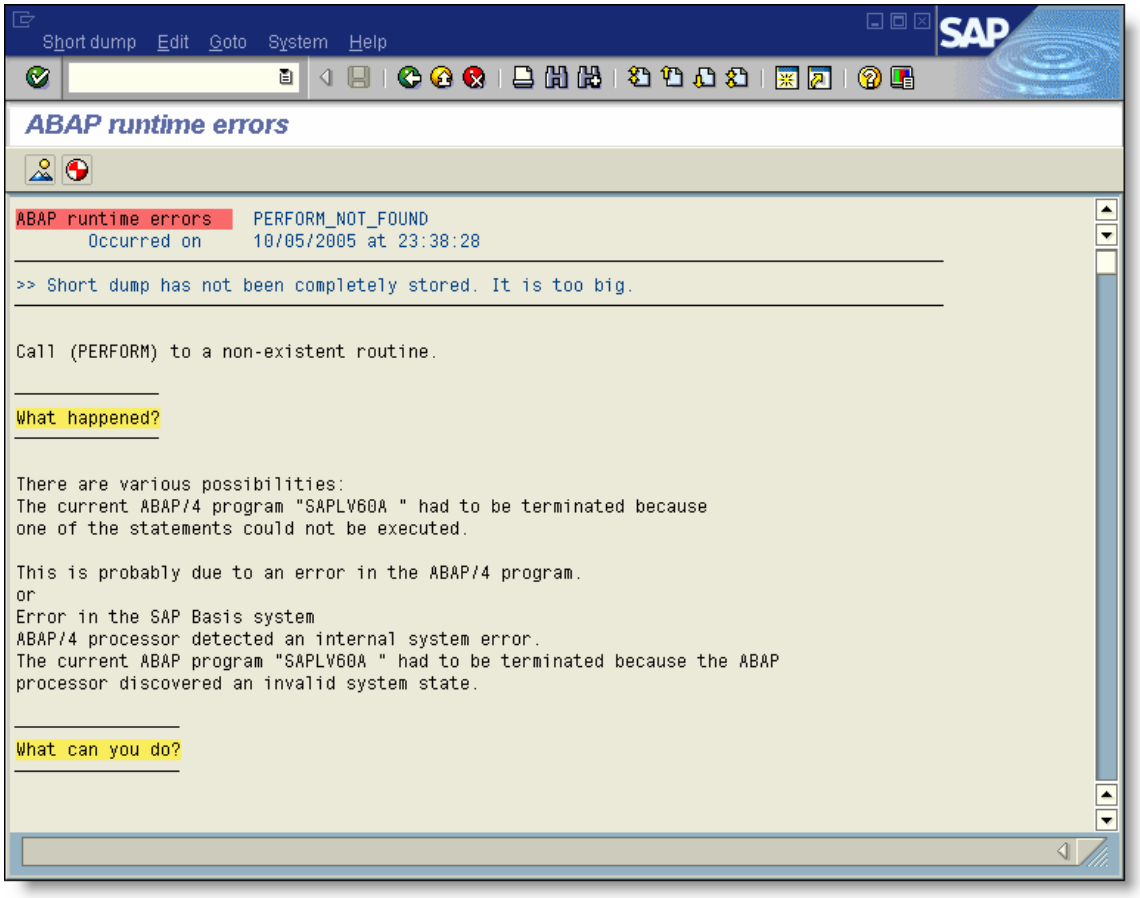
PERFORM_NOT_FOUND Short Dump

In this example, a custom billing data transfer routine was created. It was correctly assigned in copy control and worked flawlessly in the development system in which it was created. Once transported, the billing due list short dumped in the target system because it could not find the new routine.



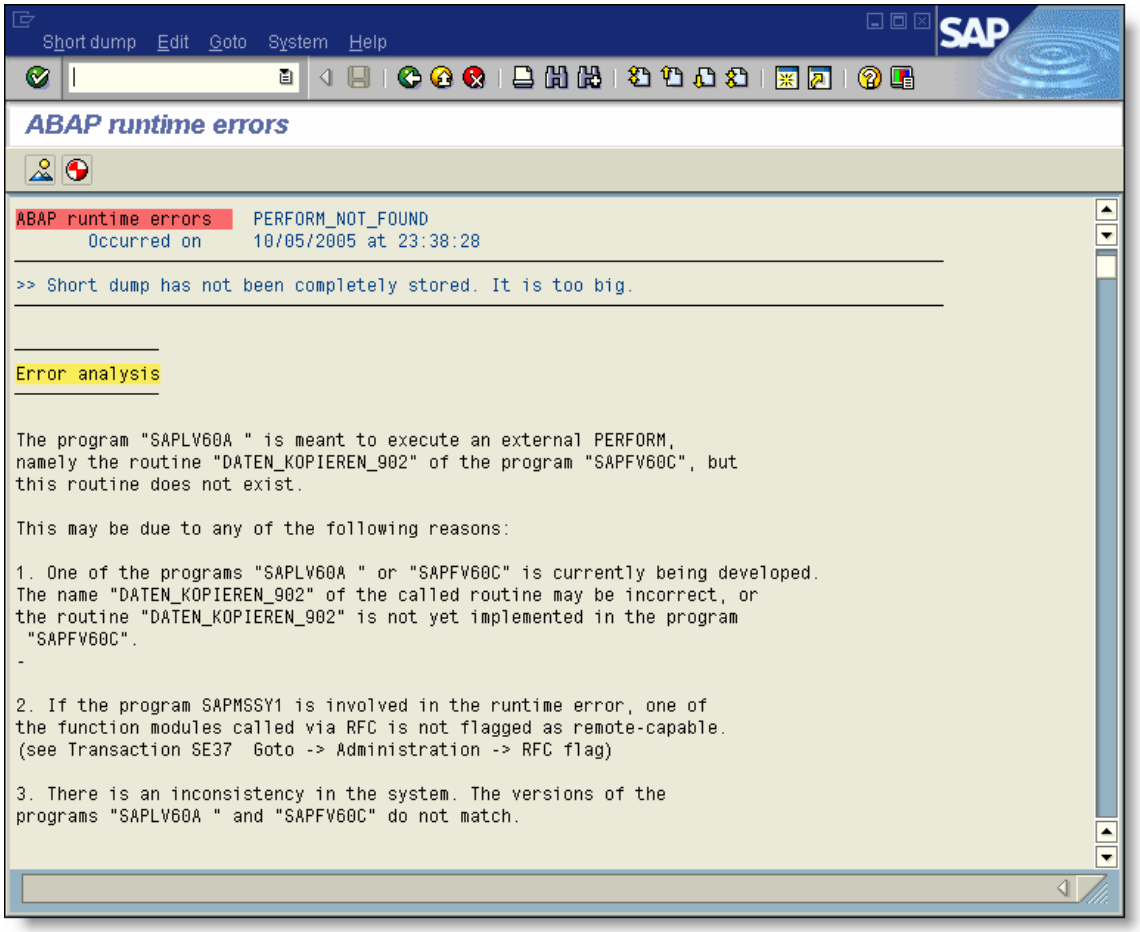
Inside the VOFM Program

Looking at the details of the short dump, it can be determined which routine caused the problem. In this section, it is clear that the billing due list had the problem (program SAPLV60A).



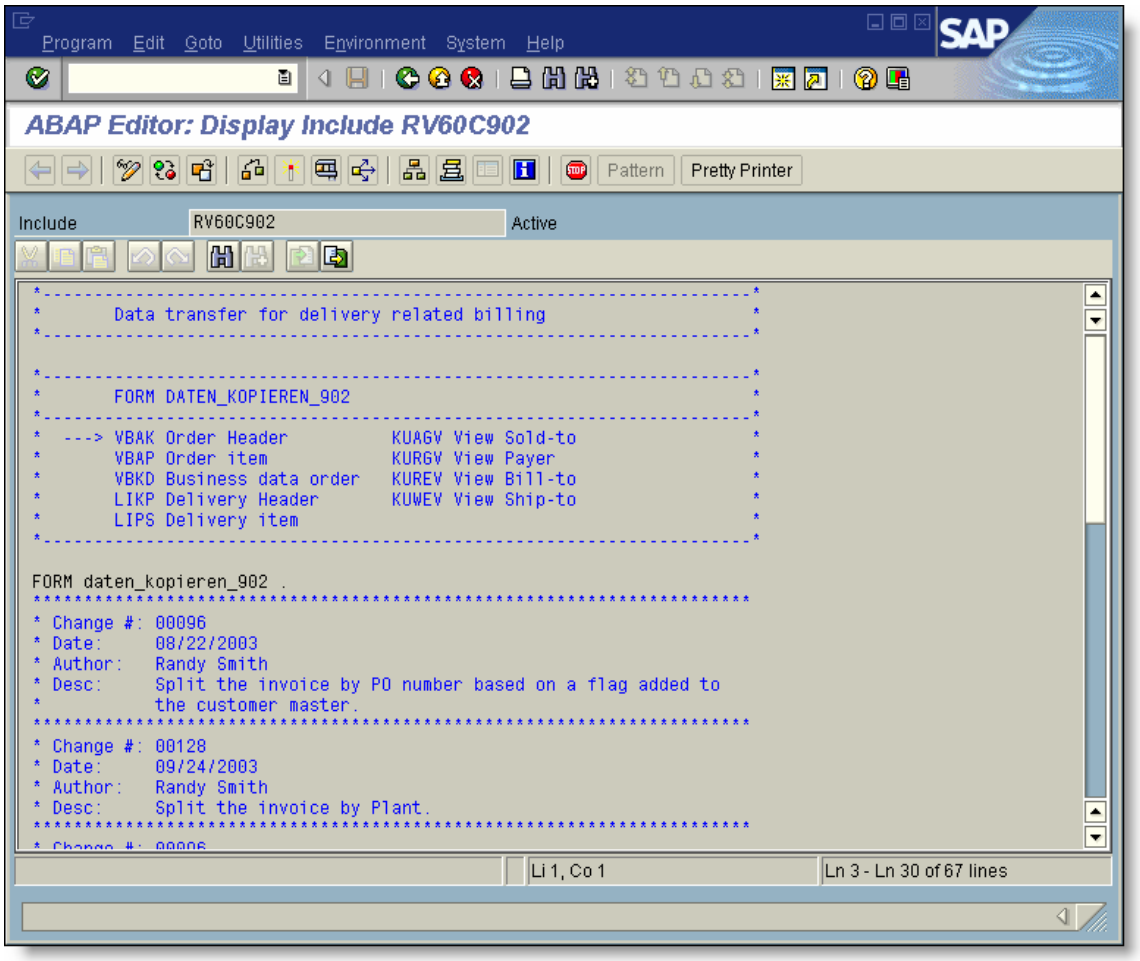
Inside the VOFM Program

Paging down further in the dump, we see that routine DATEN_KOPIEREN_902 in program SAPFV60C was called but not found. This routine is called because of the copy control configuration.



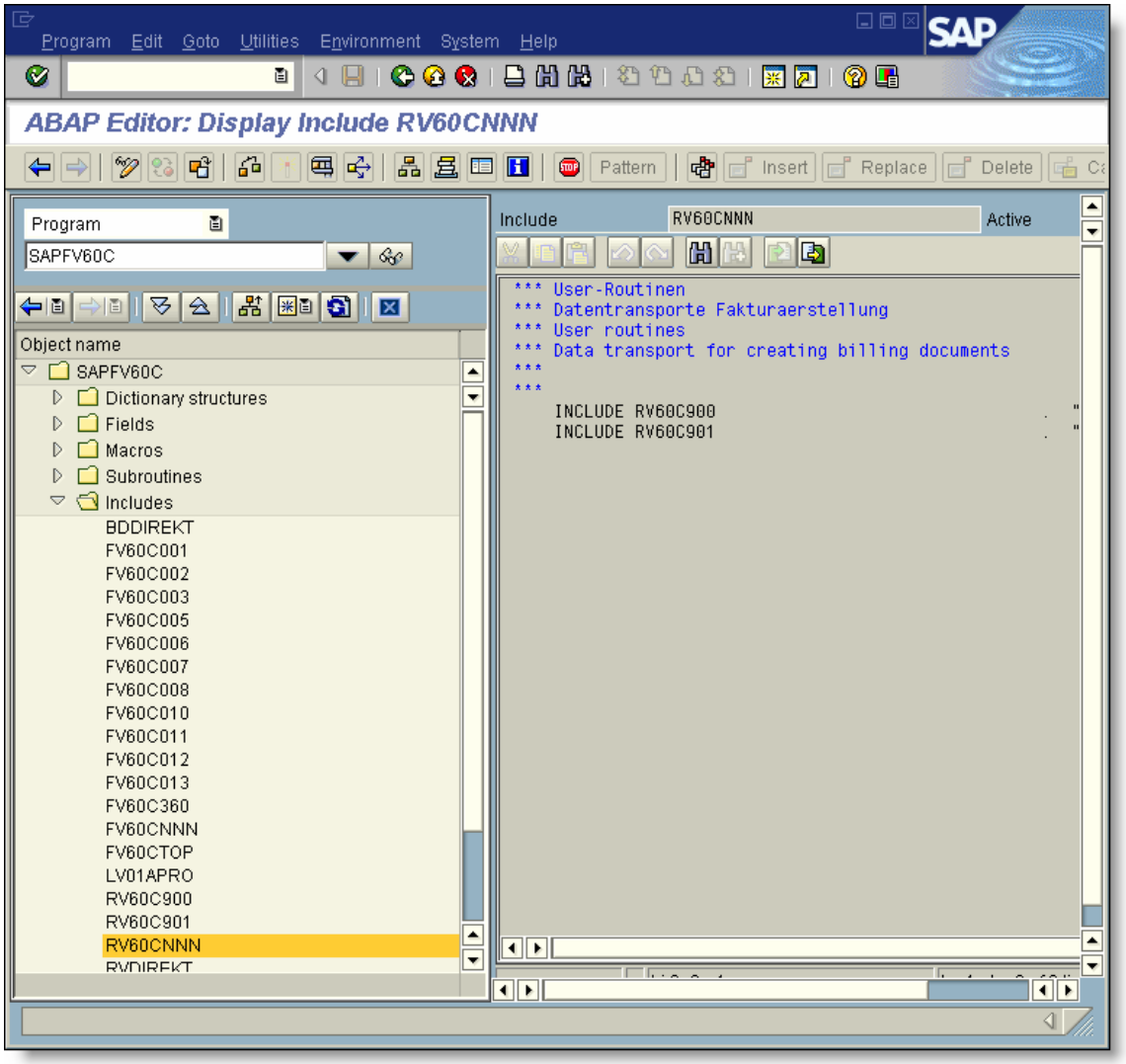
Inside the VOFM Program

Using the Program Editor (SE38) to display the ABAP include that contains the routine, we can see that it does exist.



Inside the VOFM Program

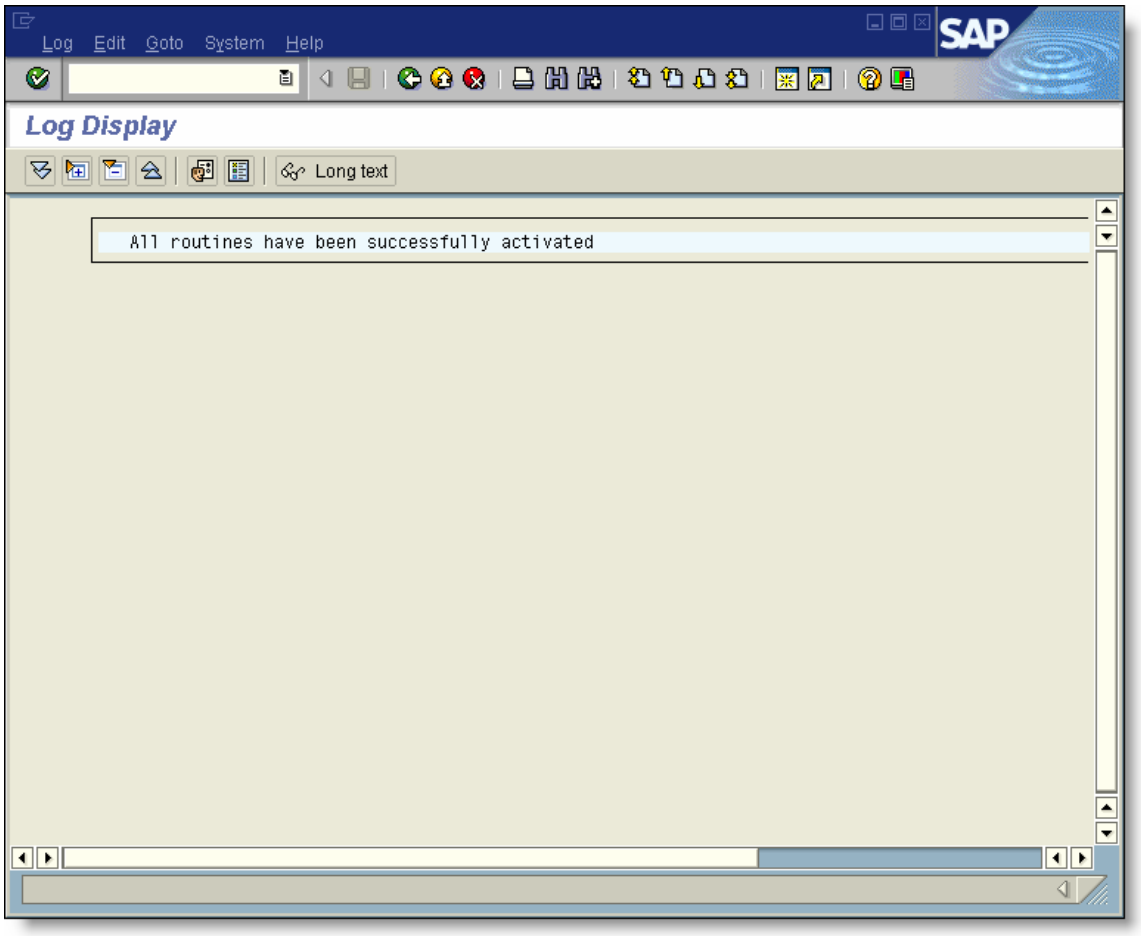
Using the Object Navigator (SE80) to view the entire program for billing data transfer routines, we see that the routine is missing. The include RV60C902 should appear within the RV60CNNN source. In the development system where the routine was created, we would see it here.



Executing Program RV80HGEN

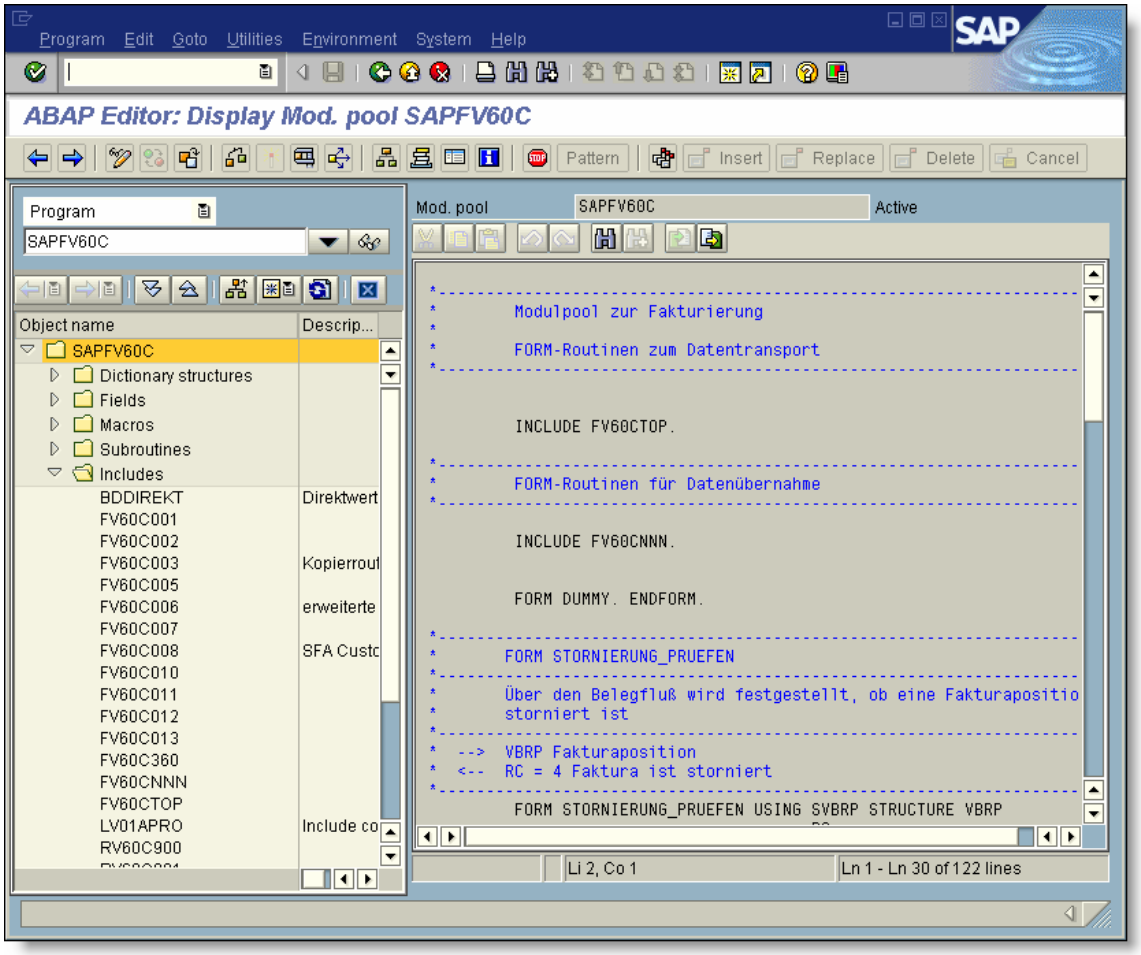
The first step for correcting this type of short dump is to execute the RV80HGEN utility program to re-generate all VOFM routines. Unfortunately, running this program does not always solve the problem. In this example, executing RV80HGEN did not work.

The output report for RV80HGEN denotes that all routines were activated, but the parent include was not repaired.



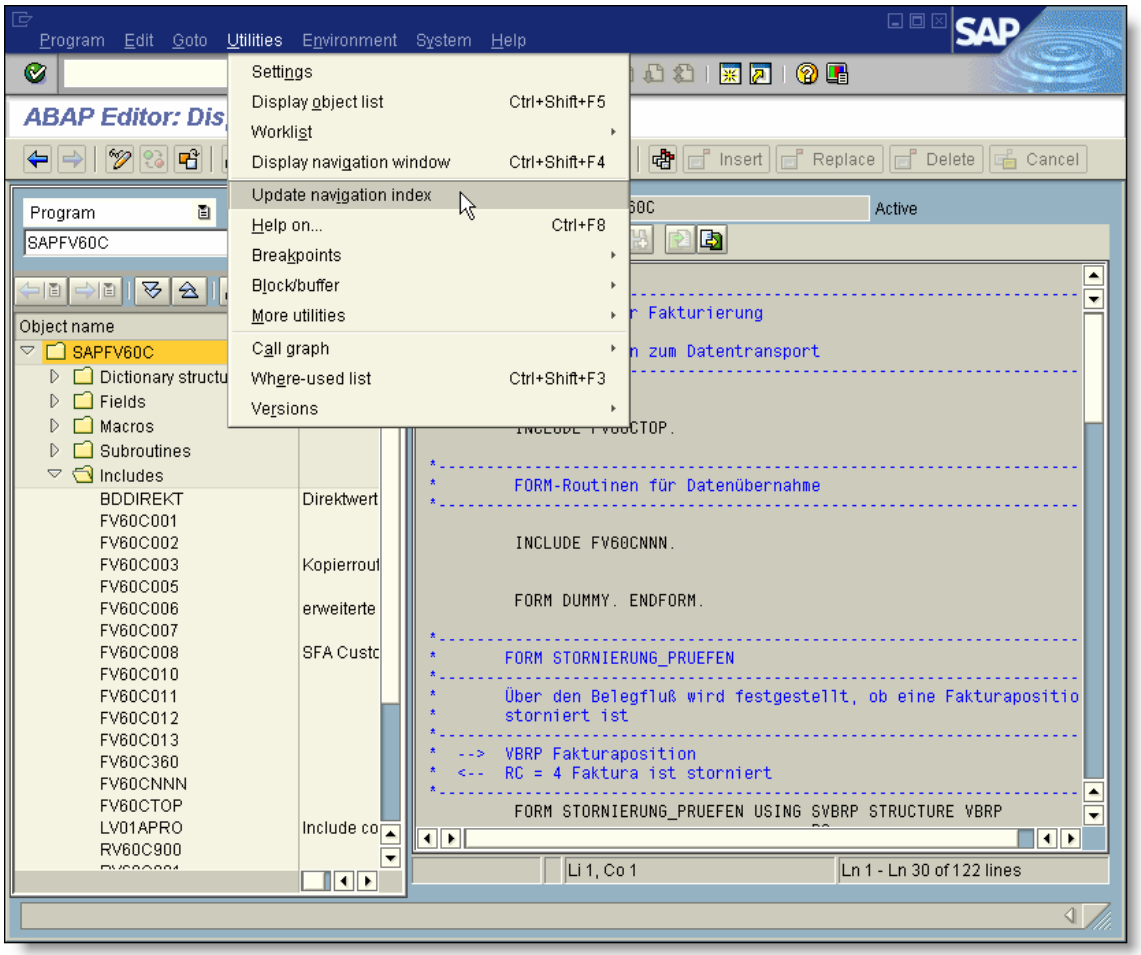
Inside the VOFM Program

Since RV80HGEN did not solve the problem, the next step is to manually regenerate the program index of the program that contains the VOFM routine. For billing data transfer routines, the program is SAPFV60C.



Inside the VOFM Program

From the program editor, navigate using the menu to *Utilities>Update navigation index*. Executing this function will solve the problem.



Inside the VOFM Program

After regenerating the index, RV60C902 include now appears in the parent include and the short dumps will cease.