

## SE16XXL – New Features

November 2019 New version **3.4A** which includes the following features:

- New operation “Perform Group Comparison” [More ...](#)
- New pseudo table \$TABCOUNT [More ...](#)
- New pseudo table \$JOBLOG [More ...](#)
- Improvements of frontend file processing [More ...](#)
- Set/Get parameters for script variants [More ...](#)
- Improved access to the SE16XXL online help [More ...](#)
- Some new features for administrators [More ...](#)
- Additional improvements [More ...](#)

April 2019 New version **3.4** which includes the following features:

- Perform a script step by step [More ...](#)
- Global scripts with multiple roles (perform/maintain) [More ...](#)
- Option “Select with Order by Primary Key” [More ...](#)
- Additional improvements [More ...](#)

November 2018 New version **3.3E** which includes the following features:

- Programs adjusted for latest version of the S/4 system [More ...](#)
- ATNAM as sel. criterion for \$CLASSIF & \$CONFIG [More ...](#)

August 2018 New version **3.3D** which includes the following features:

- Upload of frontend files at any point in a session [More ...](#)
- UPLOAD operation can be edited with the script editor [More ...](#)
- New pseudo table \$CLASSIF [More ...](#)
- New pseudo table \$CONFIG [More ...](#)
- New pseudo table \$E071 [More ...](#)

- New program for scheduling scripts in background [More ...](#)
- Optional use of database join when running a script [More ...](#)
- ABAP statement FIND allowed in a formula [More ...](#)
- Detailed entry display – Prev. entry / Next entry [More ...](#)
- Detailed entry comparison – keys also compared [More ...](#)
- New user settings available [More ...](#)
- Tool for downloading and uploading ALV layouts [More ...](#)
- ALV layout optionally assigned to a script variant [More ...](#)
- Improvement of the operation "Add text columns" [More ...](#)
- NUMC fields as sort criteria for subtotals [More ...](#)
- Other minor improvements [More ...](#)

December 2017 New version **3.3C** which includes the following features:

- CDS views with parameters supported [More ...](#)
- Very long field names (up to 30 characters) supported [More ...](#)
- Multilevel views (views based on views) supported [More ...](#)
- Two new pseudo tables (\$JEST and \$JESTC) available [More ...](#)
- Special program for scheduling a script in background [More ...](#)
- New option "one for all" available for result subscribers [More ...](#)






[More historical entries ...](#)

## New Operation “Perform Group Comparison”

Up to now it was possible by means of a **full-join** to find out which rows were missing on either side of the join. It was, however, not possible to determine which of the rows existing on both sides were “**equal**” or “**different**”. The two attributes have been enclosed in quotes because the concept of equality usually refers to a subset of the fields involved – it depends on what the user is looking for.

In order to close this gap the new operation “**Perform group comparison**” has been implemented. It is only available for a list resulting from one or more joins, i.e. a **join result list**. It allows the user to compare, for each row of the join list, **two groups** of aliases with each other, for example alias **A** with alias **B**, if the list contains only A and B, or **A + B** with **C + D**, if these aliases are present in the list. The user may thereby choose which **pairs of fields** of the two groups are to be compared with each other. The fields to be compared need not necessarily have the same characteristics, they just have to be compatible.

After the operation has been carried out each row of the result list exhibits as first column to the left a special V-field, called **comparison flag**, which shows the result. The comparison flag may assume one of the following values (or, optionally, one of the following icons):

Value	Icon	Description
==		both groups are equal, i.e. all field pairs are equal
<>		the two groups are different, i.e. at least one field pair differs
<.		the right group is initial
.>		the left group is initial
..		both groups are initial

Another important feature of this operation is that it can be “**rerun**” with the same comparison flag, but with different parameters. It is thus possible, by trial and error, to restrict the comparison to the really relevant field pairs, eliminating successively the ones that interfere with the desired result.

For more information please refer to [Group Comparison](#).

[Back to the top](#)

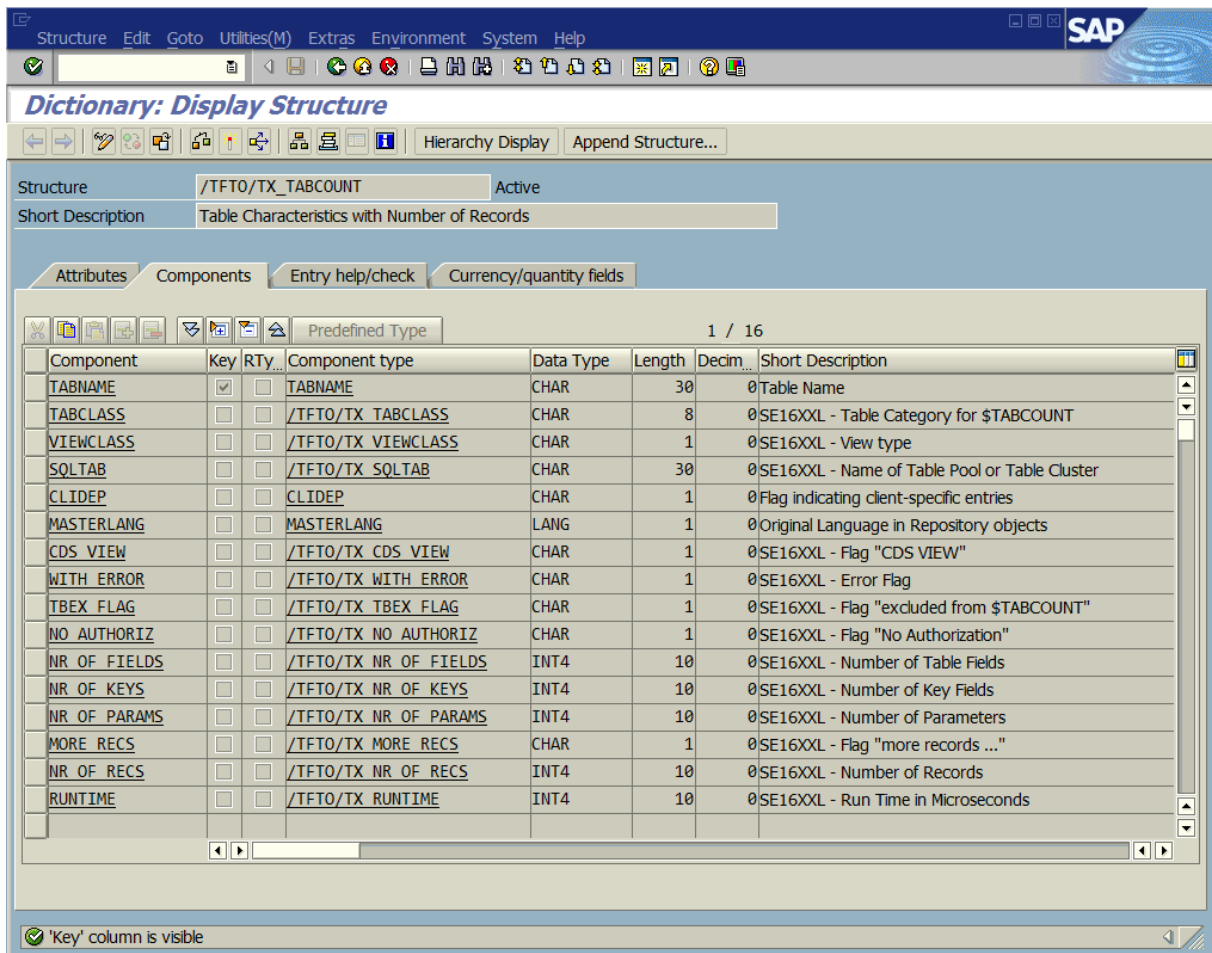
## New pseudo table \$TABCOUNT

This new pseudo table returns for each selected database table (or view) the **number of rows** (up to a specified upper limit) contained, together with other interesting information, like number of keys and number of fields. In this way it is possible for the user to find out which database tables are empty and which contain data.

The following table categories are relevant for this pseudo table:

Table Category	View Type	Description
TRANSP		Transparent database table
POOL		Logical pooled table
CLUSTER		Logical cluster table
TPOOL		Physical pooled table
TCLUSTER		Physical cluster table
VIEW	P	Projection view
VIEW	D	Database view

The structure of the result list is as follows:



Component	Key	RTy	Component type	Data Type	Length	Decim...	Short Description
TABNAME	<input checked="" type="checkbox"/>	<input type="checkbox"/>	TABNAME	CHAR	30		Table Name
TABCLASS	<input type="checkbox"/>	<input type="checkbox"/>	/TFTO/TX TABCLASS	CHAR	8		SE16XXL - Table Category for \$TABCOUNT
VIEWCLASS	<input type="checkbox"/>	<input type="checkbox"/>	/TFTO/TX VIEWCLASS	CHAR	1		SE16XXL - View type
SQLTAB	<input type="checkbox"/>	<input type="checkbox"/>	/TFTO/TX SQLTAB	CHAR	30		SE16XXL - Name of Table Pool or Table Cluster
CLIDEP	<input type="checkbox"/>	<input type="checkbox"/>	CLIDEP	CHAR	1		Flag indicating client-specific entries
MASTERLANG	<input type="checkbox"/>	<input type="checkbox"/>	MASTERLANG	LANG	1		Original Language in Repository objects
CDS_VIEW	<input type="checkbox"/>	<input type="checkbox"/>	/TFTO/TX_CDS_VIEW	CHAR	1		SE16XXL - Flag "CDS VIEW"
WITH_ERROR	<input type="checkbox"/>	<input type="checkbox"/>	/TFTO/TX WITH_ERROR	CHAR	1		SE16XXL - Error Flag
TBEX_FLAG	<input type="checkbox"/>	<input type="checkbox"/>	/TFTO/TX_TBEX_FLAG	CHAR	1		SE16XXL - Flag "excluded from \$TABCOUNT"
NO_AUTHORIZ	<input type="checkbox"/>	<input type="checkbox"/>	/TFTO/TX_NO_AUTHORIZ	CHAR	1		SE16XXL - Flag "No Authorization"
NR_OF_FIELDS	<input type="checkbox"/>	<input type="checkbox"/>	/TFTO/TX_NR_OF_FIELDS	INT4	10		SE16XXL - Number of Table Fields
NR_OF_KEYS	<input type="checkbox"/>	<input type="checkbox"/>	/TFTO/TX_NR_OF_KEYS	INT4	10		SE16XXL - Number of Key Fields
NR_OF_PARAMS	<input type="checkbox"/>	<input type="checkbox"/>	/TFTO/TX_NR_OF_PARAMS	INT4	10		SE16XXL - Number of Parameters
MORE_RECS	<input type="checkbox"/>	<input type="checkbox"/>	/TFTO/TX_MORE_RECS	CHAR	1		SE16XXL - Flag "more records ..."
NR_OF_RECS	<input type="checkbox"/>	<input type="checkbox"/>	/TFTO/TX_NR_OF_RECS	INT4	10		SE16XXL - Number of Records
RUNTIME	<input type="checkbox"/>	<input type="checkbox"/>	/TFTO/TX_RUNTIME	INT4	10		SE16XXL - Run Time in Microseconds

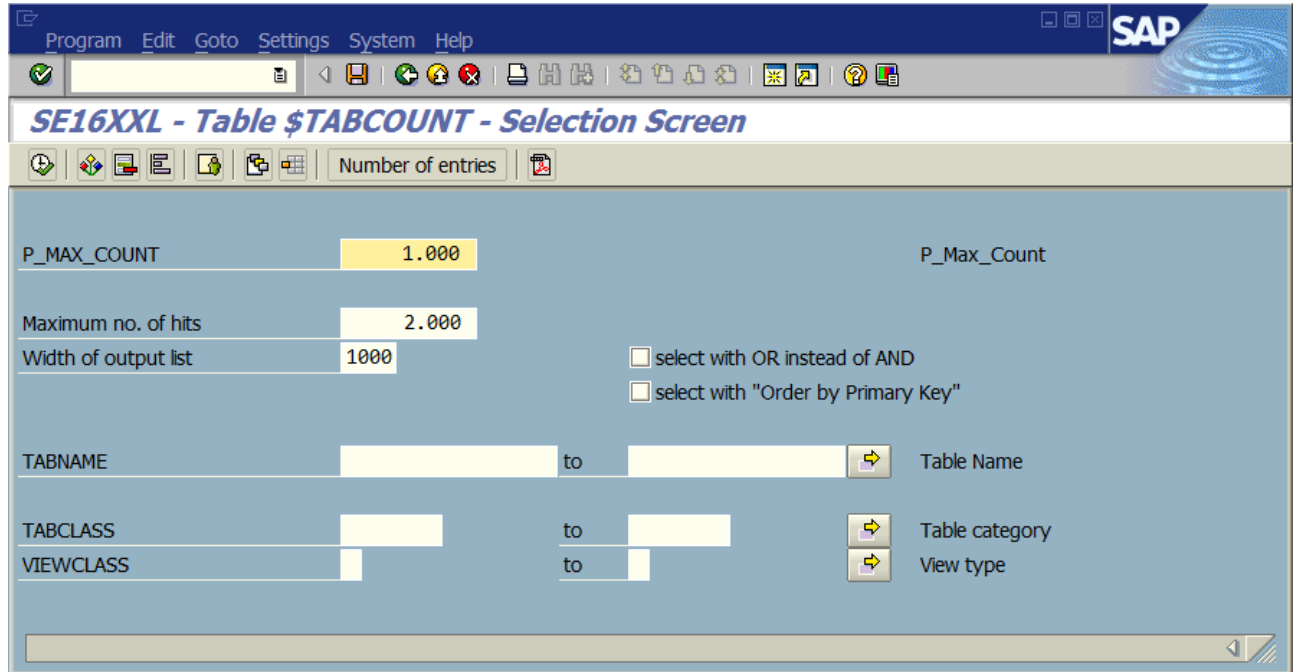
## Description of the fields of \$TABCOUNT

Field	Description
TABNAME	Name of table or view
TABCLASS	Table category
VIEWCLASS	View type
SQLTAB	Name of the physical pool or cluster (only for POOL / CLUSTER tables)
CLIDEP	'X' → the table / view is client dependent
MASTERLANG	Original language of table / view
CDS_VIEW	'X' → this is a CDS view
WITH_ERROR	'X' → something is wrong with the DDIC definition
TBEX_FLAG	'X' → the table / view is excluded from \$TABCOUNT due to extremely long run time when counting the records. In the SE16XXL Settings there is a special dialog to define which tables / views are to be excluded from \$TABCOUNT,
NO_AUTHORIZ	'X' → the user has no access permission for the table / view
NR_OF_FIELDS	Number of fields of the table / view
NR_OF_KEYS	Number of key fields of the table / view
NR_OF_PARAMS	Number of parameters of the table / view
MORE_RECS	'X' → the table / view contains more than P_MAX_COUNT records
NR_OF_RECS	Number of records of the table / view up to a maximum of P_MAX_COUNT
RUNTIME	Runtime (in microseconds) elapsed during the processing

**NOTE:** The **NUMBER\_OF\_RECS** is set to zero if one of the following flags has been set: **WITH\_ERROR**, **TBEX\_FLAG** or **NO\_AUTHORIZ**.

## Selection Screen of Pseudo Table \$TABCOUNT

The selection screen for \$TABCOUNT (with all possible criteria) is as follows:



The parameter **P\_MAX\_COUNT** is the upper limit for counting the records and is set by default to **1000**, which is also the value used implicitly when the parameter is set to zero by the user. An **upper limit** for counting is absolutely necessary, since tables exist in the database which contain millions of records, and counting them all would necessitate a disproportionate amount of time.

A typical result list could be as follows:

SE16XXL - Table \$TABCOUNT - 20 entries selected

Table \$TABCOUNT - Table Characteristics with Number of Records

TABNAME	TABCLASS	VIEWCLASS	SQLTAB	CLIDEP	MASTERLANG	CDS_VIEW	WITH_ERROR	TBEX_FLAG	NO_AUTHORIZ	NR_OF_FIELDS	NR_OF_KEYS	NR_OF_PARAMS	MORE_RECS	NR_OF_RECS	RUNTIME
<input type="checkbox"/> MAKZ	TRANSP			X						11	8	0		85	6.341
<input type="checkbox"/> MALG	TRANSP			X	D					18	5	0		100	4.058
<input type="checkbox"/> MAHT	TRANSP			X	D					7	6	0		553	5.034
<input type="checkbox"/> MAPE	TRANSP			X	D					13	4	0		65	4.157
<input type="checkbox"/> MAPEWG	TRANSP			X	D					14	5	0		0	3.286
<input type="checkbox"/> MAPL	TRANSP			X						25	8	0	X	1.000	4.671
<input type="checkbox"/> MAPOV	VIEW	D		X						89	5	0	X	1.000	18.625
<input type="checkbox"/> MAPR	TRANSP			X						8	3	0	X	1.000	5.194
<input type="checkbox"/> MARA	TRANSP			X	D					208	2	0	X	1.000	10.480
<input type="checkbox"/> MARA1	VIEW	D		X						8	2	0	X	1.000	4.045
<input type="checkbox"/> MARAP	VIEW	D		X						5	2	0	X	1.000	4.405
<input type="checkbox"/> MARAV	VIEW	D		X						134	3	0	X	1.000	8.168
<input type="checkbox"/> MARA_MATNR	VIEW	P		X						2	2	0	X	1.000	3.855
<input type="checkbox"/> MCLIL	POOL		ATAB	X						4	3	0	X	1.000	2.859
<input type="checkbox"/> MCLIM	POOL		ATAB	X						4	3	0		116	3.260
<input type="checkbox"/> MHND	CLUSTER		RFMHN	X						64	15	0		284	10.723
<input type="checkbox"/> MMIM_PRED	CLUSTER		IMPREDOC	X						9	4	0		205	4.612
<input type="checkbox"/> MWCURM	POOL		ATAB	X	D					1	1	0		0	4.142
<input type="checkbox"/> M_MTVMA	POOL		M_MTVM	X						9	9	0		0	3.840
<input type="checkbox"/> M_MTVMB	POOL		M_MTVM	X						9	9	0		0	2.746

[Back to the top](#)

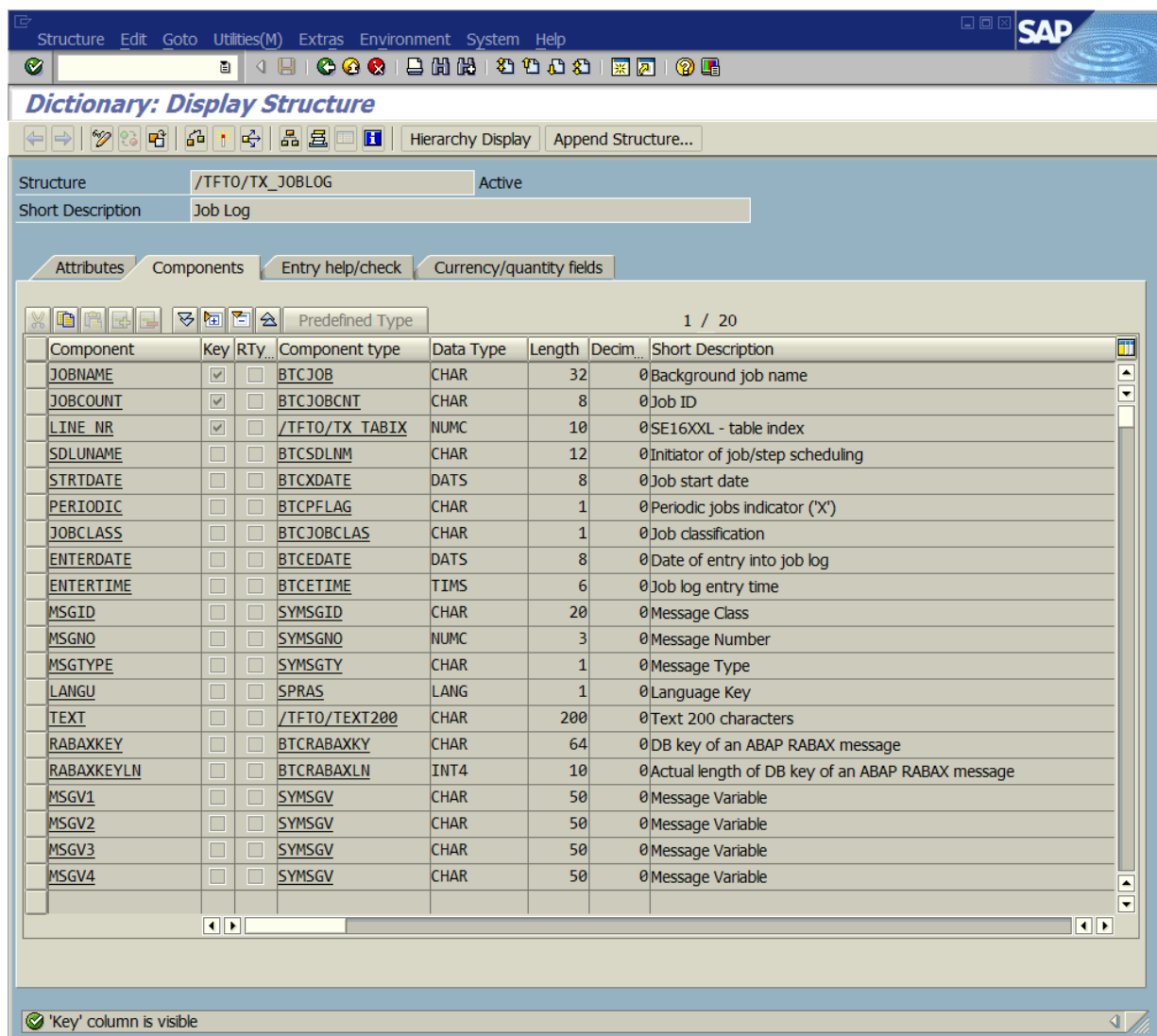
## New pseudo table \$JOBLOG

By means of standard transaction SM37 it is possible to display the job log of individual background jobs, one by one. If just a couple of jobs are to be inspected, this approach is fine. However, if the job logs of a large number of jobs are to be examined, viewing them one by one can be extremely time consuming and tedious.

For this reason the new pseudo table \$JOBLOG has been implemented. It allows the user to read the job logs of many background jobs at one go, making it thus possible to examine all these logs without having to go back and forth as in SM37.

The implementation of the pseudo table is based on table **TBTCO** and function module **'BP\_JOBLOG\_READ'**.

The structure of the result list is as follows:

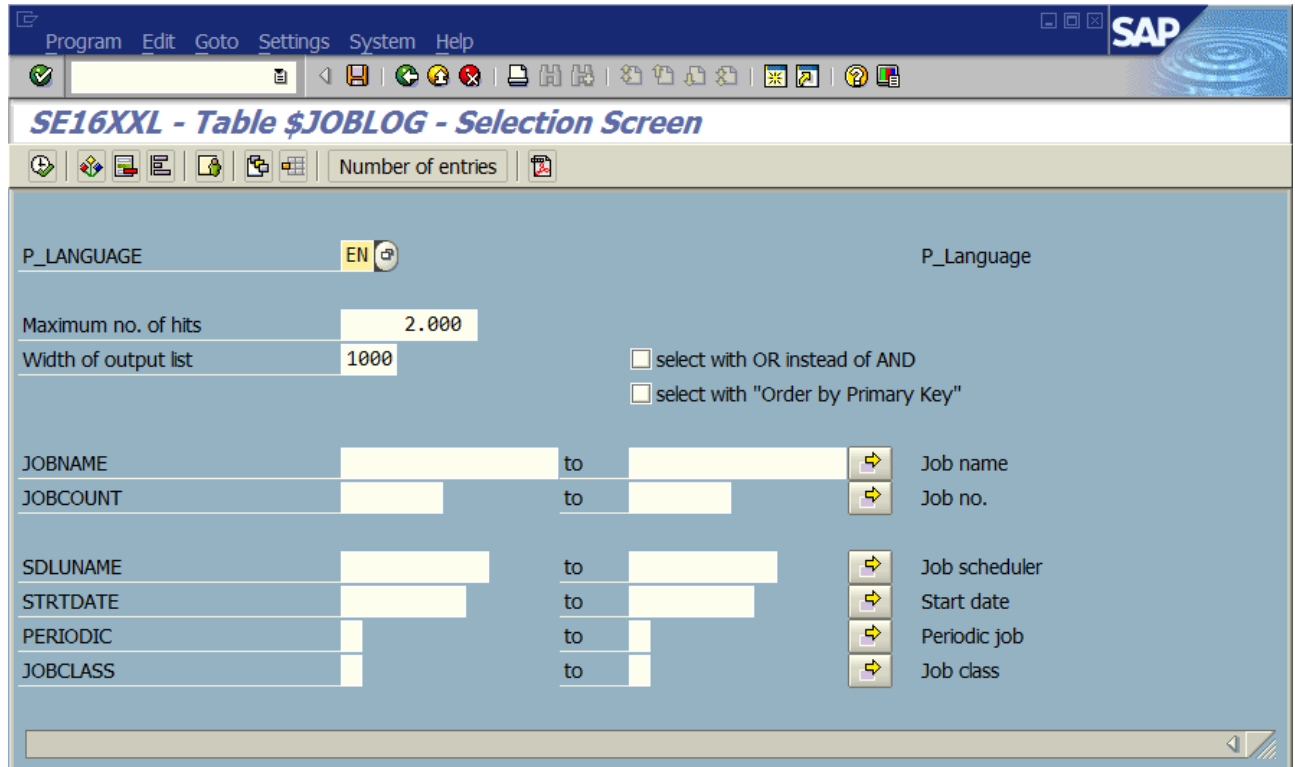


Component	Key	RTy	Component type	Data Type	Length	Decim	Short Description
JOBNAME	<input checked="" type="checkbox"/>	<input type="checkbox"/>	BTCJOB	CHAR	32	0	Background job name
JOBCOUNT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	BTCJOBcnt	CHAR	8	0	Job ID
LINE_Nr	<input checked="" type="checkbox"/>	<input type="checkbox"/>	/TFTO/TX_TABIX	NUMC	10	0	SE16XXL - table index
SDLUNAME	<input type="checkbox"/>	<input type="checkbox"/>	BTCSDLNM	CHAR	12	0	Initiator of job/step scheduling
STRDATE	<input type="checkbox"/>	<input type="checkbox"/>	BTCXDATE	DATS	8	0	Job start date
PERIODIC	<input type="checkbox"/>	<input type="checkbox"/>	BTCPFLAG	CHAR	1	0	Periodic jobs indicator ('X')
JOBCLASS	<input type="checkbox"/>	<input type="checkbox"/>	BTCJOBCLAS	CHAR	1	0	Job classification
ENTERDATE	<input type="checkbox"/>	<input type="checkbox"/>	BTCEDATE	DATS	8	0	Date of entry into job log
ENTERTIME	<input type="checkbox"/>	<input type="checkbox"/>	BTCETIME	TIMS	6	0	Job log entry time
MSGID	<input type="checkbox"/>	<input type="checkbox"/>	SYMSGID	CHAR	20	0	Message Class
MSGNO	<input type="checkbox"/>	<input type="checkbox"/>	SYMSGNO	NUMC	3	0	Message Number
MSGTYPE	<input type="checkbox"/>	<input type="checkbox"/>	SYMSGTY	CHAR	1	0	Message Type
LANGU	<input type="checkbox"/>	<input type="checkbox"/>	SPRAS	LANG	1	0	Language Key
TEXT	<input type="checkbox"/>	<input type="checkbox"/>	/TFTO/TEXT200	CHAR	200	0	Text 200 characters
RABAXKEY	<input type="checkbox"/>	<input type="checkbox"/>	BTCRABAXKY	CHAR	64	0	DB key of an ABAP RABAX message
RABAXKEYLN	<input type="checkbox"/>	<input type="checkbox"/>	BTCRABAXLN	INT4	10	0	Actual length of DB key of an ABAP RABAX message
MSGV1	<input type="checkbox"/>	<input type="checkbox"/>	SYMSGV	CHAR	50	0	Message Variable
MSGV2	<input type="checkbox"/>	<input type="checkbox"/>	SYMSGV	CHAR	50	0	Message Variable
MSGV3	<input type="checkbox"/>	<input type="checkbox"/>	SYMSGV	CHAR	50	0	Message Variable
MSGV4	<input type="checkbox"/>	<input type="checkbox"/>	SYMSGV	CHAR	50	0	Message Variable



## Selection Screen of Pseudo Table \$JOBLOG

The selection screen of \$JOBLOG (with all possible criteria) is as follows:



SE16XXL - Table \$JOBLOG - Selection Screen

Number of entries

P\_LANGUAGE EN P\_Language

Maximum no. of hits 2.000

Width of output list 1000

select with OR instead of AND

select with "Order by Primary Key"

JOBNAME to Job name

JOBCOUNT to Job no.

SDLUNAME to Job scheduler

STRTDATE to Start date

PERIODIC to Periodic job

JOBCCLASS to Job class

In order not to overburden the structure of the pseudo table, only the **major criteria** for background jobs have been made available on the selection screen. If other criteria are deemed necessary, it is advisable to first select table **TBTCO** and then join the result with \$JOBLOG.

The parameter **P\_LANGUAGE** may be used to obtain the job log in a language different from the logon language. It is set by default to the **logon language**.

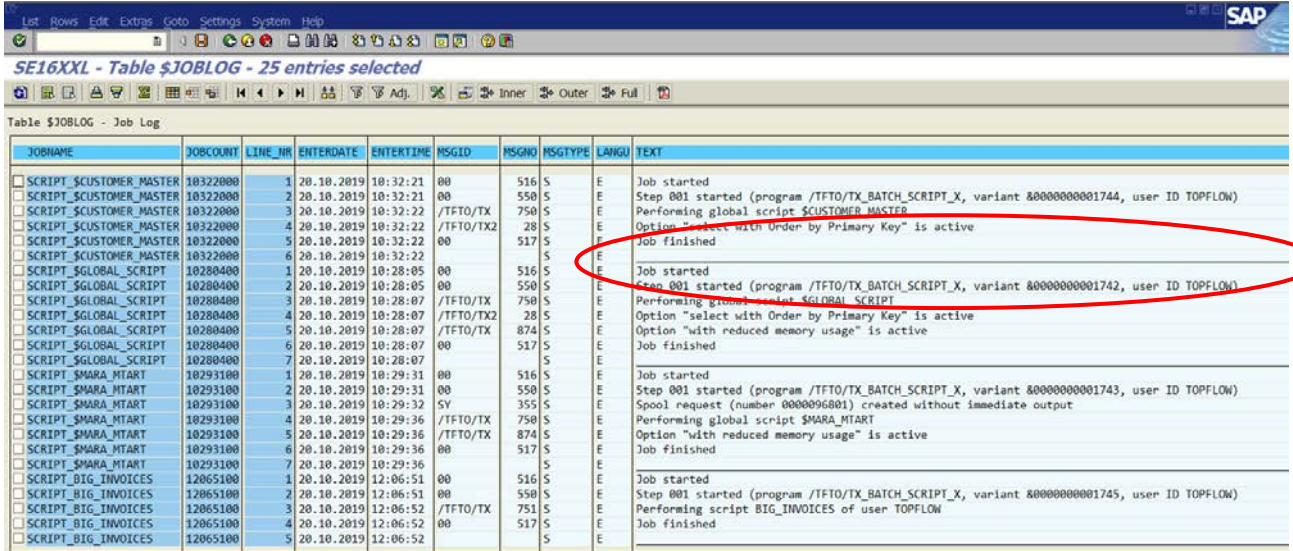
If some of the messages contained in the job logs are not available (table T100) in the requested language, the program uses first the logon language, then English and finally German to find a suitable message text. The result list displays in field LANGU the language that was actually taken.

In the following page we will show an example of the result list of \$JOBLOG.



## Example of Result List with \$JOBLOG

A typical result list with \$JOBLOG could be as follows:



JOBNAME	JOBCOUNT	LINE_Nr	ENTERDATE	ENRTIME	MSGID	MSGNO	MSGTYPE	LANGU	TEXT
SCRIPT_ \$CUSTOMER_MASTER	10322000	1	20.10.2019	10:32:21	00	516	S	E	Job started
SCRIPT_ \$CUSTOMER_MASTER	10322000	2	20.10.2019	10:32:21	00	550	S	E	Step 001 started (program /TFTO/TX_BATCH_SCRIPT_X, variant &0000000001744, user ID TOPFLOW)
SCRIPT_ \$CUSTOMER_MASTER	10322000	3	20.10.2019	10:32:22	/TFTO/TX	750	S	E	Performing global script \$CUSTOMER_MASTER
SCRIPT_ \$CUSTOMER_MASTER	10322000	4	20.10.2019	10:32:22	/TFTO/TX2	28	S	E	Option "select with Order by Primary Key" is active
SCRIPT_ \$CUSTOMER_MASTER	10322000	5	20.10.2019	10:32:22	00	517	S	E	Job finished
SCRIPT_ \$GLOBAL_SCRIPT	10280400	1	20.10.2019	10:28:05	00	516	S	E	Job started
SCRIPT_ \$GLOBAL_SCRIPT	10280400	2	20.10.2019	10:28:05	00	550	S	E	Step 001 started (program /TFTO/TX_BATCH_SCRIPT_X, variant &0000000001742, user ID TOPFLOW)
SCRIPT_ \$GLOBAL_SCRIPT	10280400	3	20.10.2019	10:28:07	/TFTO/TX	750	S	E	Performing global script \$GLOBAL_SCRIPT
SCRIPT_ \$GLOBAL_SCRIPT	10280400	4	20.10.2019	10:28:07	/TFTO/TX2	28	S	E	Option "select with Order by Primary Key" is active
SCRIPT_ \$GLOBAL_SCRIPT	10280400	5	20.10.2019	10:28:07	/TFTO/TX	874	S	E	Option "with reduced memory usage" is active
SCRIPT_ \$GLOBAL_SCRIPT	10280400	6	20.10.2019	10:28:07	00	517	S	E	Job finished
SCRIPT_ \$GLOBAL_SCRIPT	10280400	7	20.10.2019	10:28:07	00	517	S	E	Job finished
SCRIPT_ \$MARA_MTART	10293100	1	20.10.2019	10:29:31	00	516	S	E	Job started
SCRIPT_ \$MARA_MTART	10293100	2	20.10.2019	10:29:31	00	550	S	E	Step 001 started (program /TFTO/TX_BATCH_SCRIPT_X, variant &0000000001743, user ID TOPFLOW)
SCRIPT_ \$MARA_MTART	10293100	3	20.10.2019	10:29:32	5V	355	S	E	Spool request (number 0000096801) created without immediate output
SCRIPT_ \$MARA_MTART	10293100	4	20.10.2019	10:29:36	/TFTO/TX	750	S	E	Performing global script \$MARA_MTART
SCRIPT_ \$MARA_MTART	10293100	5	20.10.2019	10:29:36	/TFTO/TX	874	S	E	Option "with reduced memory usage" is active
SCRIPT_ \$MARA_MTART	10293100	6	20.10.2019	10:29:36	00	517	S	E	Job finished
SCRIPT_ \$MARA_MTART	10293100	7	20.10.2019	10:29:36	00	517	S	E	Job finished
SCRIPT_ \$BIG_INVOICES	12065100	1	20.10.2019	12:06:51	00	516	S	E	Job started
SCRIPT_ \$BIG_INVOICES	12065100	2	20.10.2019	12:06:51	00	550	S	E	Step 001 started (program /TFTO/TX_BATCH_SCRIPT_X, variant &0000000001745, user ID TOPFLOW)
SCRIPT_ \$BIG_INVOICES	12065100	3	20.10.2019	12:06:52	/TFTO/TX	751	S	E	Performing script \$BIG_INVOICES of user TOPFLOW
SCRIPT_ \$BIG_INVOICES	12065100	4	20.10.2019	12:06:52	00	517	S	E	Job finished
SCRIPT_ \$BIG_INVOICES	12065100	5	20.10.2019	12:06:52	00	517	S	E	Job finished

Notice that at the end of the job log of an individual job a **separation line** has been added by the program to improve readability. The **LINE\_Nr** is set by the program to assure a correct sorting of the log lines.

If the same job logs are now selected for P\_LANGUAGE = 'FR' (French), the result list would be as follows (detail):

ME	MSGID	MSGNO	MSGTYPE	LANGU	TEXT
1	00	516	S	F	Job lancé
1	00	550	S	F	Etape 001 lancée (programme /TFTO/TX_BATCH_SCRIPT_X, variante &0000000001744, utilisateur TOPFLOW)
2	/TFTO/TX	750	S	E	Performing global script \$CUSTOMER_MASTER
2	/TFTO/TX2	28	S	E	Option "select with Order by Primary Key" is active
2	00	517	S	F	Job terminé
2			S	F	
5	00	516	S	F	Job lancé
5	00	550	S	F	Etape 001 lancée (programme /TFTO/TX_BATCH_SCRIPT_X, variante &0000000001742, utilisateur TOPFLOW)
7	/TFTO/TX	750	S	E	Performing global script \$GLOBAL_SCRIPT

As may be seen, some of the messages carry LANGU = 'E' instead of 'F' because there is no T100 entry for these messages in French.

[Back to the top](#)

## Improvements of Frontend File Processing

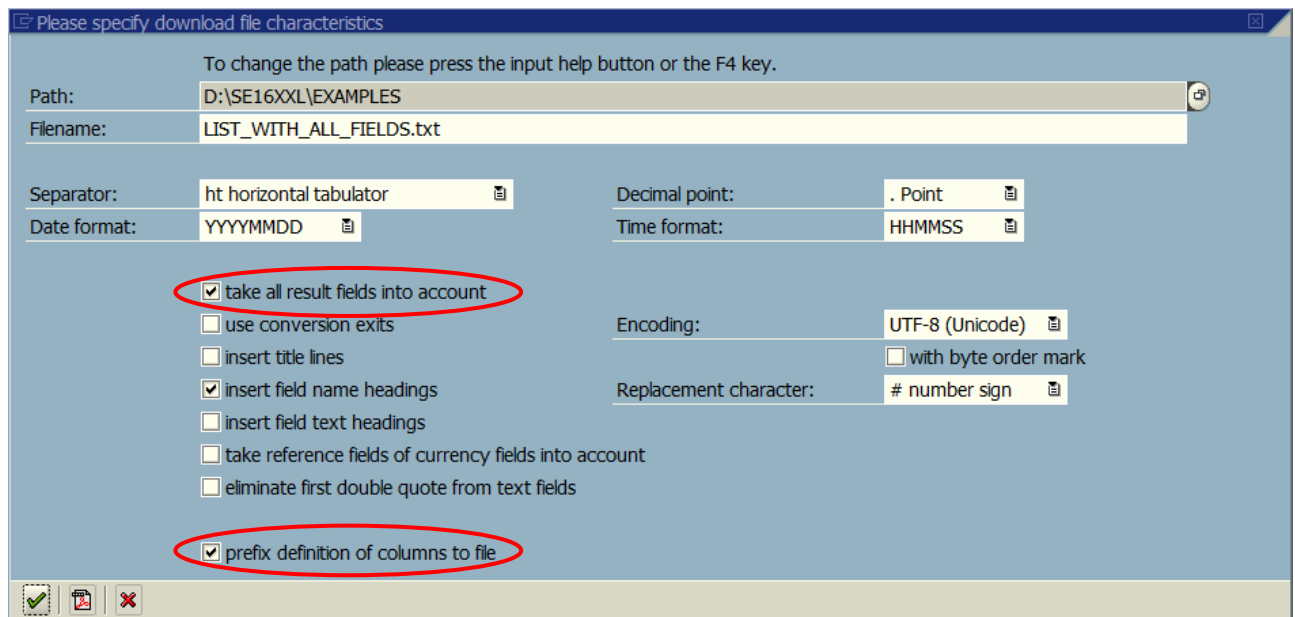
### Download to Frontend File taking all Fields into Account

Up to now the menu function

*List → Download to frontend file → as text with separator characters*

took only the chosen columns of the result list into account. The maximum number of columns for the result list is **256** (in case of ALV list even less – i.e. **90**). This is sufficient in most cases. There are, however, database tables, like for example LIPS, that have more than 256 columns. And if the list results from a series of joins, the number of columns can be even larger, ranging in the thousands.

All this did not pose a problem up to now, with the advent of the “**Perform group comparison**” operation, however, the situation has changed. If a comparison is to be carried out between the result of a script and a frontend file representing the same result at an earlier date, it is **imperative** that the frontend file contains all columns which are also present internally in the result. For this reason the download function has been enhanced accordingly:

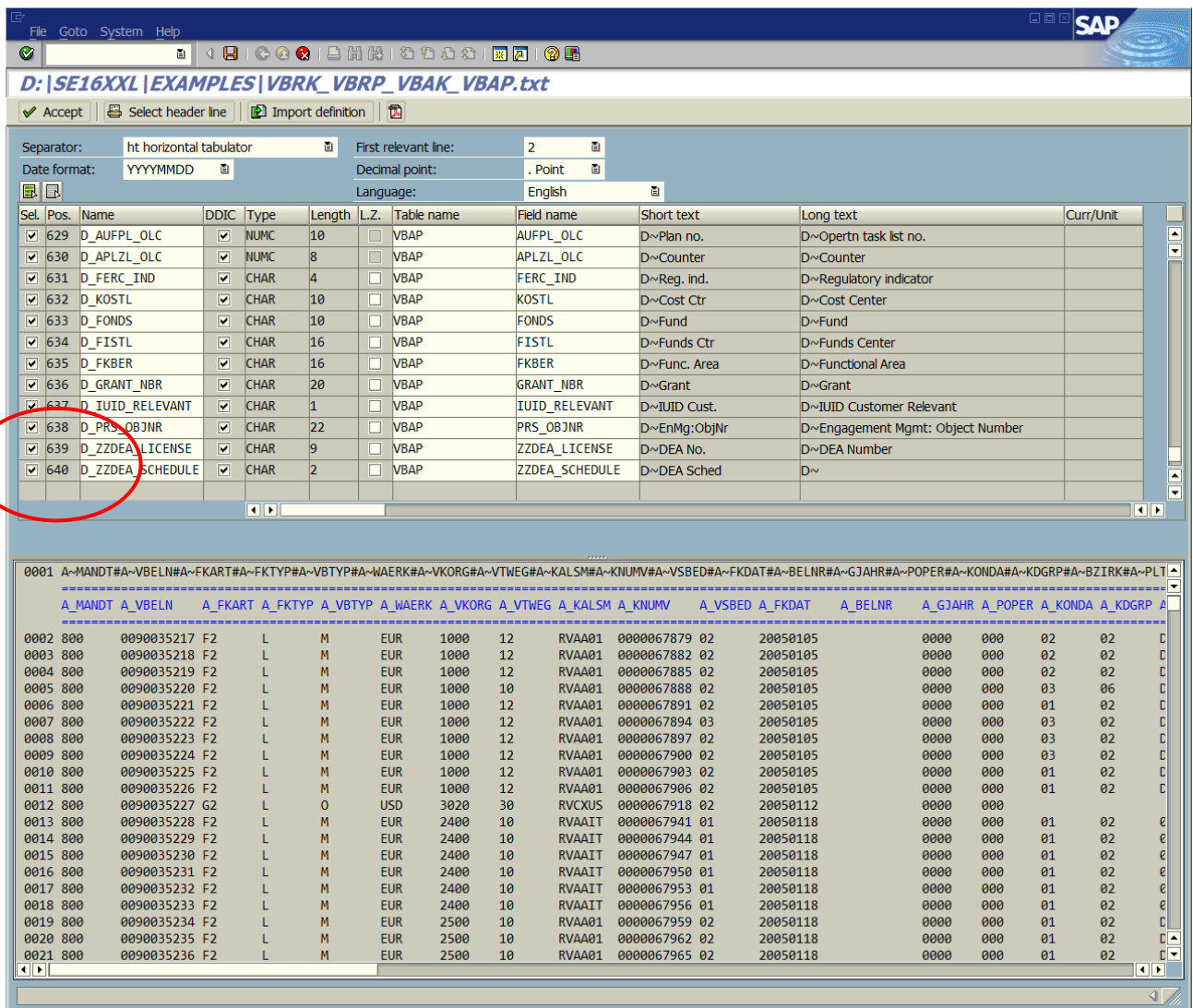


The new option “**take all result fields into account**” is exactly what is needed for performing a group comparison at a later time. The option “**prefix definition of columns to file**” should also be activated for this purpose.

## Upload Start File / Frontend File with up to 4000 Columns

Up to now a start file / frontend file could only be uploaded in SE16XXL if the number of columns did not exceed **256**. This number has now been increased to **4000**, making it thus possible to carry out group comparisons with wide database tables and big join lists.

As an example let us upload a frontend file created by downloading a join list of **VBRK + VBRP + VBAK + VBAP**:



Separator: ht horizontal tabulator    First relevant line: 2

Date format: YYYYMMDD    Decimal point: . Point

Language: English

Sel.	Pos.	Name	DDIC	Type	Length	L.Z.	Table name	Field name	Short text	Long text	Curr/Unit
<input checked="" type="checkbox"/>	629	D_AUFPL_OLC	<input checked="" type="checkbox"/>	NUMC	10	<input type="checkbox"/>	VBAP	AUFPL_OLC	D~Plan no.	D~Opertrn task list no.	
<input checked="" type="checkbox"/>	630	D_APLZL_OLC	<input checked="" type="checkbox"/>	NUMC	8	<input type="checkbox"/>	VBAP	APLZL_OLC	D~Counter	D~Counter	
<input checked="" type="checkbox"/>	631	D_FERC_IND	<input checked="" type="checkbox"/>	CHAR	4	<input type="checkbox"/>	VBAP	FERC_IND	D~Reg. ind.	D~Regulatory Indicator	
<input checked="" type="checkbox"/>	632	D_KOSTL	<input checked="" type="checkbox"/>	CHAR	10	<input type="checkbox"/>	VBAP	KOSTL	D~Cost Ctr	D~Cost Center	
<input checked="" type="checkbox"/>	633	D_FONDS	<input checked="" type="checkbox"/>	CHAR	10	<input type="checkbox"/>	VBAP	FONDS	D~Fund	D~Fund	
<input checked="" type="checkbox"/>	634	D_FISTL	<input checked="" type="checkbox"/>	CHAR	16	<input type="checkbox"/>	VBAP	FISTL	D~Funds Ctr	D~Funds Center	
<input checked="" type="checkbox"/>	635	D_FKBER	<input checked="" type="checkbox"/>	CHAR	16	<input type="checkbox"/>	VBAP	FKBER	D~Func. Area	D~Functional Area	
<input checked="" type="checkbox"/>	636	D_GRANT_NBR	<input checked="" type="checkbox"/>	CHAR	20	<input type="checkbox"/>	VBAP	GRANT_NBR	D~Grant	D~Grant	
<input checked="" type="checkbox"/>	637	D_IUID_RELEVANT	<input checked="" type="checkbox"/>	CHAR	1	<input type="checkbox"/>	VBAP	IUID_RELEVANT	D~IUID Cust.	D~IUID Customer Relevant	
<input checked="" type="checkbox"/>	638	D_PRS_OBJNR	<input checked="" type="checkbox"/>	CHAR	22	<input type="checkbox"/>	VBAP	PRS_OBJNR	D~EnMg:ObjNr	D~Engagement Mgmt: Object Number	
<input checked="" type="checkbox"/>	639	D_ZZDEA_LICENSE	<input checked="" type="checkbox"/>	CHAR	9	<input type="checkbox"/>	VBAP	ZZDEA_LICENSE	D~DEA No.	D~DEA Number	
<input checked="" type="checkbox"/>	640	D_ZZDEA_SCHEDULE	<input checked="" type="checkbox"/>	CHAR	2	<input type="checkbox"/>	VBAP	ZZDEA_SCHEDULE	D~DEA Sched	D~	

0001 A-MANDT#A~VBELN#A~FKART#A~FKTYP#A~VBTP#A~WAERK#A~VKORG#A~VTWEG#A~KALSM#A~KNUMV#A~VSBED#A~FKDAT#A~BELNR#A~GJAHR#A~POPER#A~KONDA#A~KDGRP#A~BZIRK#A~PLT

A_MANDT	A_VBELN	A_FKART	A_FKTP	A_VBTP	A_WAERK	A_VKORG	A_VTWEG	A_KALSM	A_KNUMV	A_VSBED	A_FKDAT	A_BELNR	A_GJAHR	A_POPER	A_KONDA	A_KDGRP	A_BZIRK	A_PLT
0002	800	0090035217	F2	L	M	EUR	1000	12	RVAA01	0000067879	02	20050105	0000	000	02	02		
0003	800	0090035218	F2	L	M	EUR	1000	12	RVAA01	0000067882	02	20050105	0000	000	02	02		
0004	800	0090035219	F2	L	M	EUR	1000	12	RVAA01	0000067885	02	20050105	0000	000	02	02		
0005	800	0090035220	F2	L	M	EUR	1000	10	RVAA01	0000067888	02	20050105	0000	000	03	06		
0006	800	0090035221	F2	L	M	EUR	1000	12	RVAA01	0000067891	02	20050105	0000	000	01	02		
0007	800	0090035222	F2	L	M	EUR	1000	12	RVAA01	0000067894	03	20050105	0000	000	03	02		
0008	800	0090035223	F2	L	M	EUR	1000	12	RVAA01	0000067897	02	20050105	0000	000	03	02		
0009	800	0090035224	F2	L	M	EUR	1000	12	RVAA01	0000067900	02	20050105	0000	000	03	02		
0010	800	0090035225	F2	L	M	EUR	1000	12	RVAA01	0000067903	02	20050105	0000	000	01	02		
0011	800	0090035226	F2	L	M	EUR	1000	12	RVAA01	0000067906	02	20050105	0000	000	01	02		
0012	800	0090035227	G2	L	O	USD	3020	30	RVCXUS	0000067918	02	20050112	0000	000				
0013	800	0090035228	F2	L	M	EUR	2400	10	RVAAIT	0000067941	01	20050118	0000	000	01	02		
0014	800	0090035229	F2	L	M	EUR	2400	10	RVAAIT	0000067944	01	20050118	0000	000	01	02		
0015	800	0090035230	F2	L	M	EUR	2400	10	RVAAIT	0000067947	01	20050118	0000	000	01	02		
0016	800	0090035231	F2	L	M	EUR	2400	10	RVAAIT	0000067950	01	20050118	0000	000	01	02		
0017	800	0090035232	F2	L	M	EUR	2400	10	RVAAIT	0000067953	01	20050118	0000	000	01	02		
0018	800	0090035233	F2	L	M	EUR	2400	10	RVAAIT	0000067956	01	20050118	0000	000	01	02		
0019	800	0090035234	F2	L	M	EUR	2500	10	RVAA01	0000067959	02	20050118	0000	000	01	02		
0020	800	0090035235	F2	L	M	EUR	2500	10	RVAA01	0000067962	02	20050118	0000	000	01	02		
0021	800	0090035236	F2	L	M	EUR	2500	10	RVAA01	0000067965	02	20050118	0000	000	01	02		

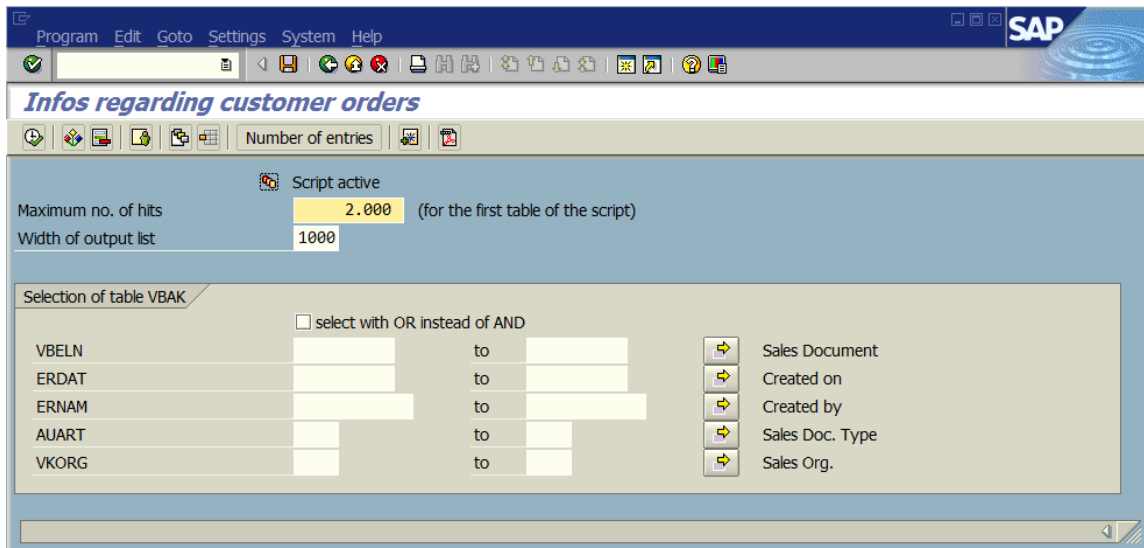
As may be seen, the file has **640** columns.


[Back to the top](#)

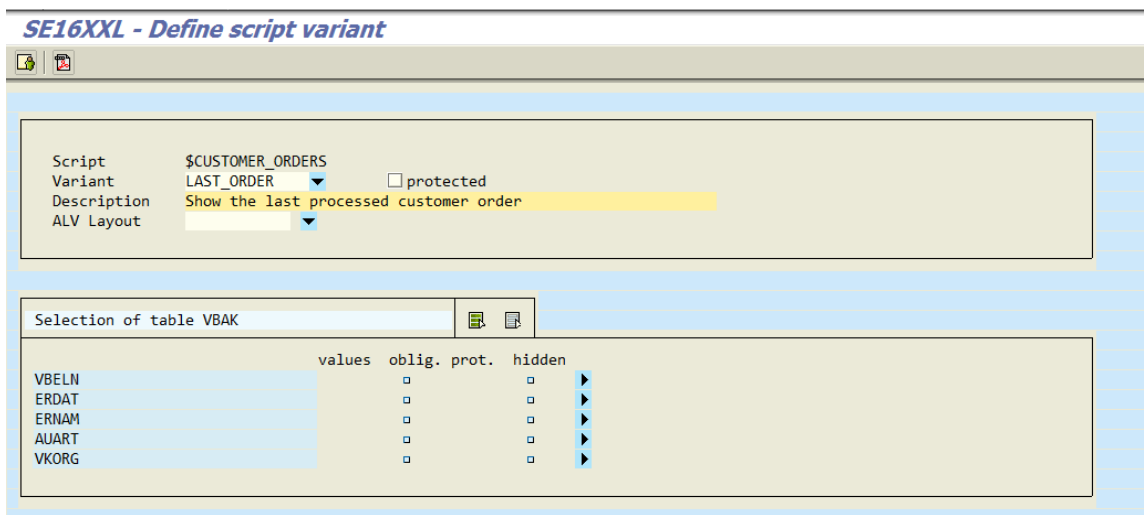
## Set/Get parameters for script variants


Up to now it was possible, when creating a script variant, to assign special **variables** to individual select options or parameters. In order to add still more flexibility to script variants a new kind of variable is now available, i.e. it is now possible to assign a **Set/Get parameter** to a given select option or parameter. An example will show how this works.

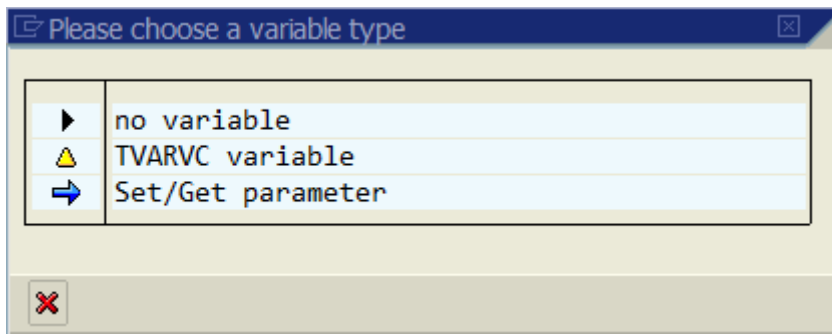
Suppose we have a script which selects information regarding customer orders. We would like to create a variant that **by default** always proposes the **order number** (VBAK-VBELN) of the last customer order processed by the user using a standard transaction like **VA01**, **VA02** or **VA03**. These transactions make use for this purpose of the Set/Get parameter “AUN”. So we start by calling the script. The special selection screen shows up:



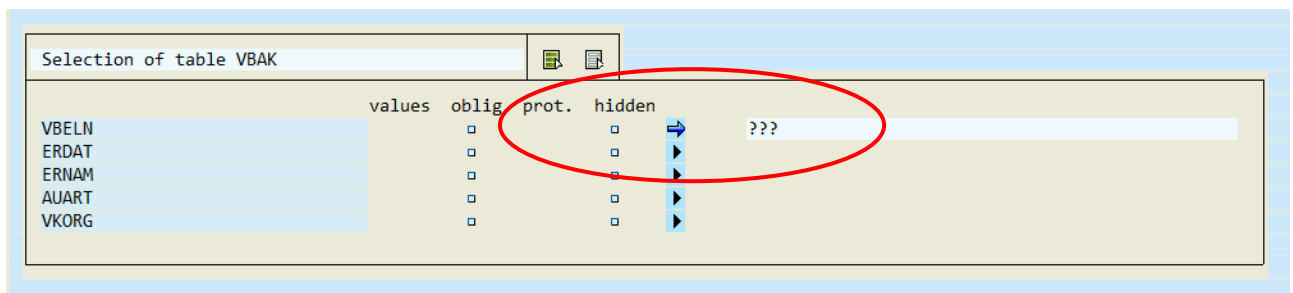
Now we press the  button on the system function bar to create a script variant:



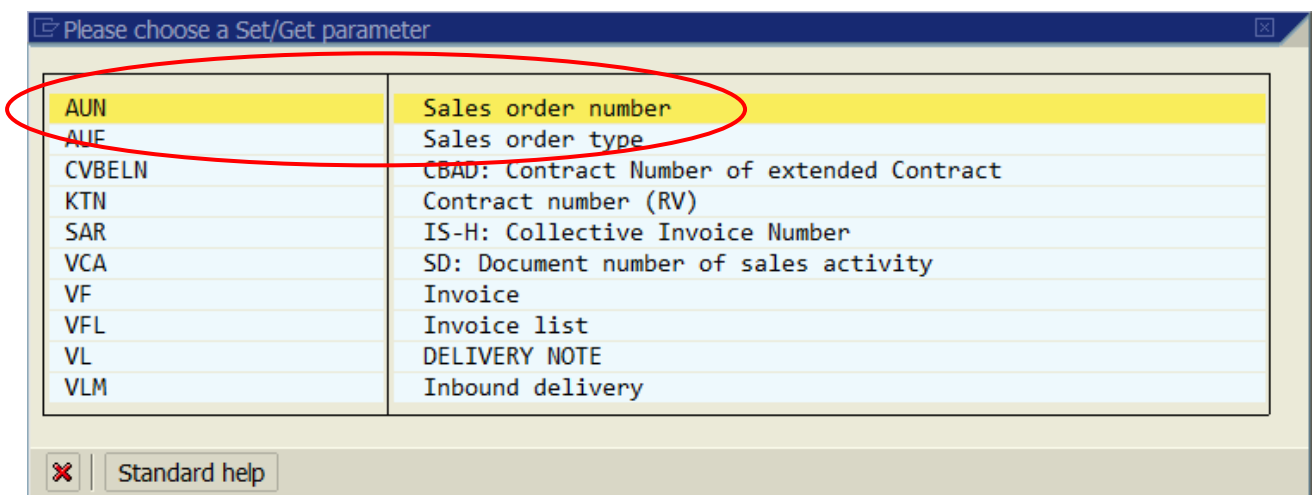
In order to assign the Set/Get parameter to VBELN we click on the  icon to the right to obtain a list of possible choices:

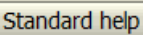


The last type () is the one we are looking for. The variant definition changes accordingly:

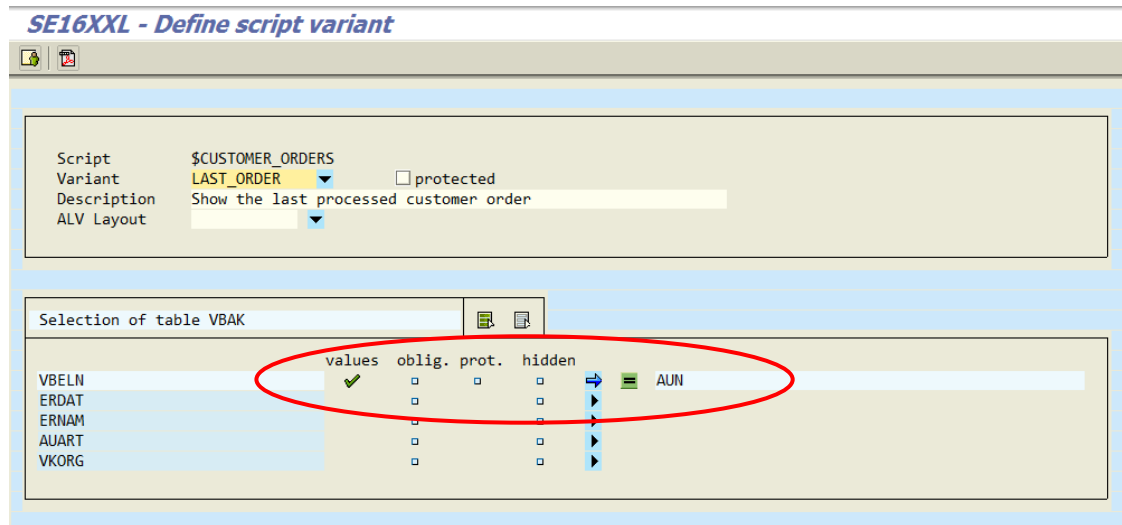



Clicking on  we obtain a list of proposals:



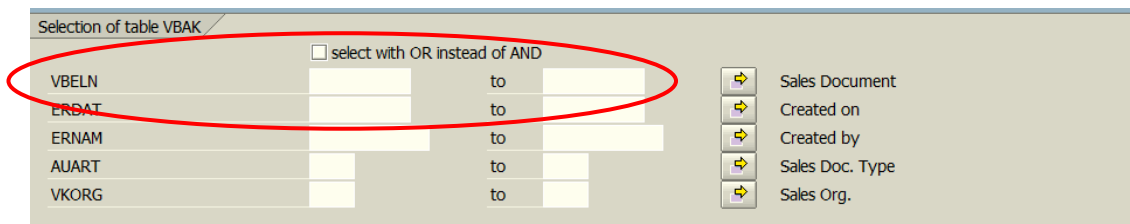
Our Set/Get parameter of choice is the one on top. The others have been found by analyzing the data elements associated with the domain of VBAK-VBELN. If none of the proposed Set/Get parameters seems relevant, it is possible to make use of the standard help function by pressing .

After choosing **AUN** from the list of proposals the variant definition changes as follows:

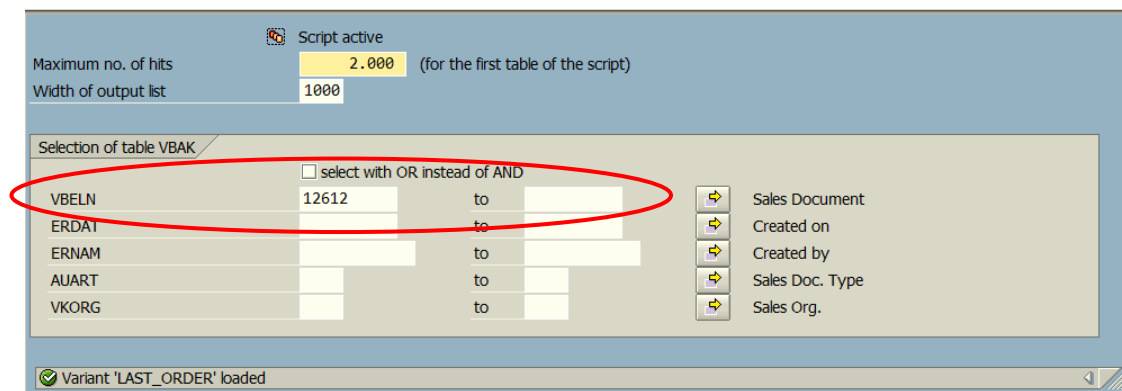


The variant is now complete and we can save it by pressing the  button.

Since no customer order was processed in the current session, the special selection screen still shows an empty VBELN select option:



In order to test if the variant really works, we call transaction **VA03** for customer order number **12612**. Then we perform again our script thereby loading the variant:





As expected the order number now shows up – it works!

[Back to the top](#)




## Improved Access to the SE16XXL Online Help

Up to now if **no path** was assigned for the SE16XXL **online help**, pressing on the  button would cause the program to issue the following message:

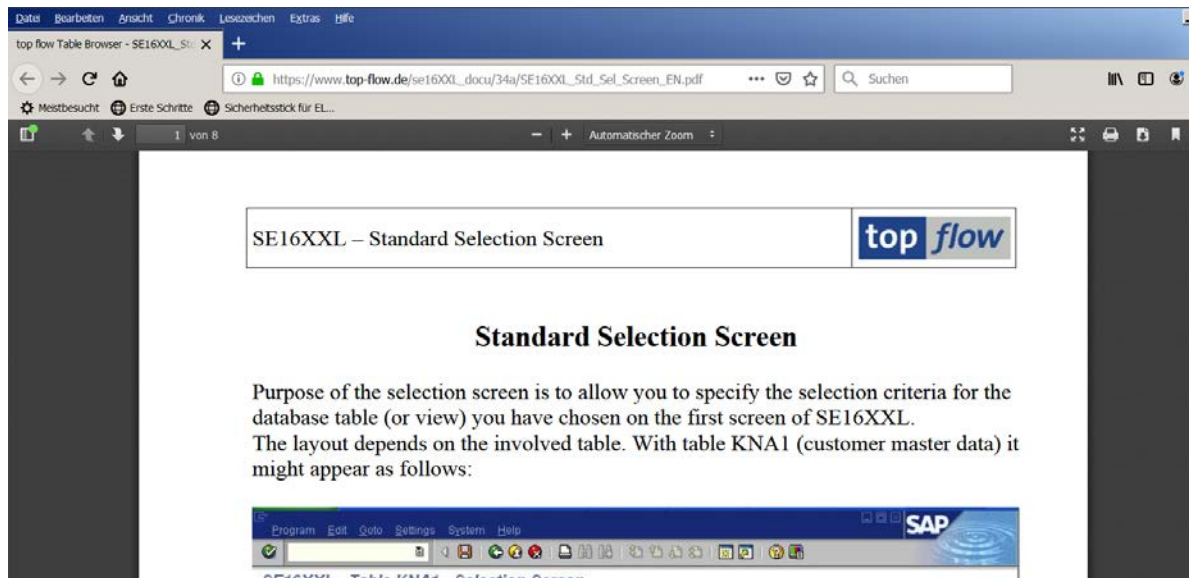
 No path has been defined for the online help files !

If a faulty path was assigned, the message would be similar to the following:

 The help file 'SE16XXL\_Std\_Sel\_Screen\_EN.pdf' is not in directory 'D:\SE16XXL\EXAMPLES' !

Since it turns out that many users have **no access** to the online help because the associated documentation files have either **not been installed at all**, or in a directory which is **only available for some** but not all users, it has become necessary to implement an alternative logic which should always work.

When the desired online help file is not found, for whatever reason, the program now looks for it on the **top-flow home page** on the internet:



**NOTE:** This mechanism does not work, if the defined online help path points to an internet address (**http:// ... or https:// ...**) because in this case the program **cannot determine** if the desired file has been **found or not**. So if the online help still does not work, please assign a **local path** by calling on the initial screen of SE16XXL the menu function **Settings** → **Path for documentation**.

[Back to the top](#)



## Some new features for administrators

The following topics are only relevant for administrators. They are cited here only to inform the normal SE16XXL user of the possible implications at his/her level.

### Access Permissions for Pseudo Tables

Up to now pseudo tables like **\$CLASSIF** or **\$JOBLOG** could not be specified directly when defining the table access permissions in the SE16XXL Settings. Instead the permission logic rested upon the database tables on which these pseudo tables are based. The problem with this logic is that the base table of a pseudo table is not readily visible. In order to increase the transparency of the permission logic, it is now possible to specify pseudo tables directly, but only by name, not explicitly at field level. This improvement has been especially necessary with the introduction of pseudo table **\$JOBLOG**, the access to which possibly needs to be restricted.

### Definition of Tables/Views to be excluded from \$TABCOUNT

As already mentioned regarding pseudo table **\$TABCOUNT**, there exist in the system database tables and views which cannot be easily counted, because any attempt to count the records, even with a very low upper limit, leads to **absolutely impractical run times**. Since this abnormal behavior cannot be detected a priori by taking a look at the characteristics of the tables / views, it has become necessary to implement a **table of exclusions**, which is delivered with only a couple of entries, but can be extended by the administrator as similar cases come to be known.

For normal users this means that, should a particular database table or view exhibit a similar behavior, i.e. an abnormally long run time when counting the records with **\$TABCOUNT**, the administrator should be contacted for a possible exclusion of the involved table / view.

[Back to the top](#)

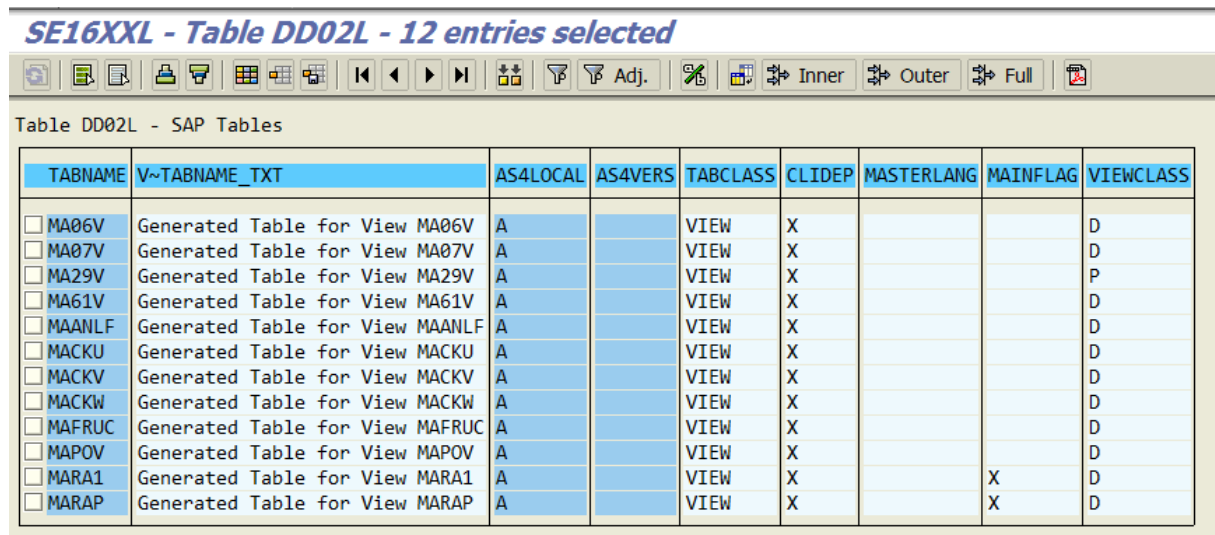
## Additional Improvements

### Improved Text Determination for TABNAME

The menu function *Extras* → *Add text columns* returned up to now, in case of a field based on data element **TABNAME** (for example DD02L-TABNAME etc.) only short texts derived from table **DD02T**, which is the “inofficial” text table of DD02L. The problem thereby is, that views do indeed have an entry in DD02T, but this entry does not correspond to the short text displayed in transaction **SE11** (Data Dictionary). The reason is that the “real” short text of a **view** is to be found in table **DD25T**. This shortcoming has now been eliminated. To make an example, we will show the same views with the “old” short texts and the new ones.

The old texts would be as follows:

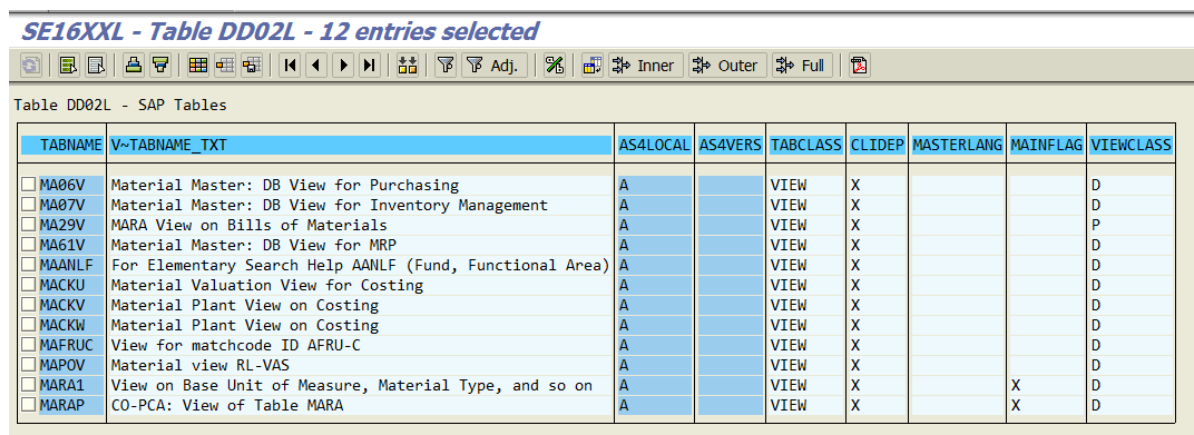
*SE16XXL - Table DD02L - 12 entries selected*



TABNAME	V~TABNAME_TXT	AS4LOCAL	AS4VERS	TABCLASS	CLIDEP	MASTERLANG	MAINFLAG	VIEWCLASS
<input type="checkbox"/> MA06V	Generated Table for View MA06V	A		VIEW	X			D
<input type="checkbox"/> MA07V	Generated Table for View MA07V	A		VIEW	X			D
<input type="checkbox"/> MA29V	Generated Table for View MA29V	A		VIEW	X			P
<input type="checkbox"/> MA61V	Generated Table for View MA61V	A		VIEW	X			D
<input type="checkbox"/> MAANLF	Generated Table for View MAANLF	A		VIEW	X			D
<input type="checkbox"/> MACKU	Generated Table for View MACKU	A		VIEW	X			D
<input type="checkbox"/> MACKV	Generated Table for View MACKV	A		VIEW	X			D
<input type="checkbox"/> MACKW	Generated Table for View MACKW	A		VIEW	X			D
<input type="checkbox"/> MAFRUC	Generated Table for View MAFRUC	A		VIEW	X			D
<input type="checkbox"/> MAPOV	Generated Table for View MAPOV	A		VIEW	X			D
<input type="checkbox"/> MARA1	Generated Table for View MARA1	A		VIEW	X		X	D
<input type="checkbox"/> MARAP	Generated Table for View MARAP	A		VIEW	X		X	D

The new texts are much better:

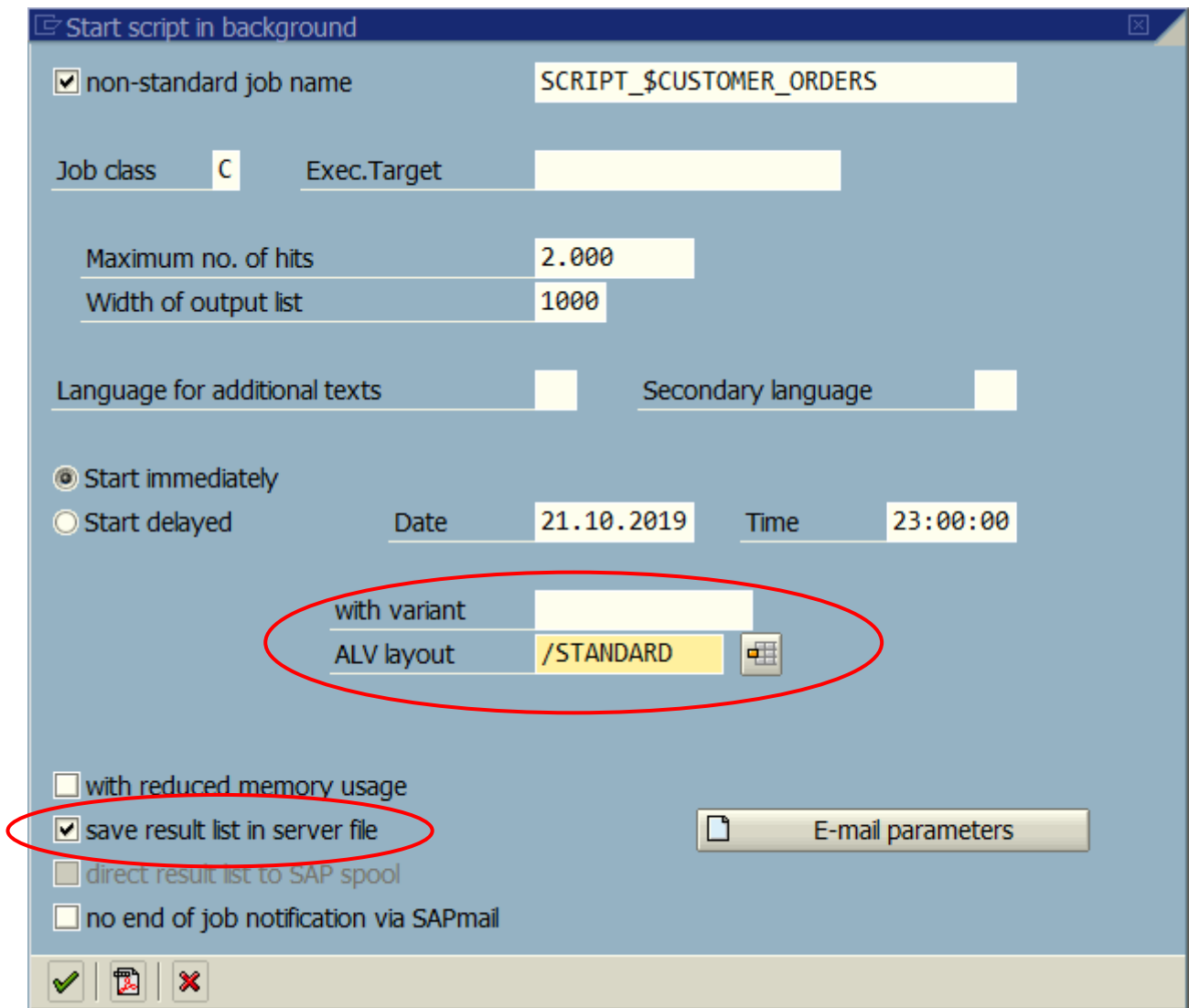
*SE16XXL - Table DD02L - 12 entries selected*



TABNAME	V~TABNAME_TXT	AS4LOCAL	AS4VERS	TABCLASS	CLIDEP	MASTERLANG	MAINFLAG	VIEWCLASS
<input type="checkbox"/> MA06V	Material Master: DB View for Purchasing	A		VIEW	X			D
<input type="checkbox"/> MA07V	Material Master: DB View for Inventory Management	A		VIEW	X			D
<input type="checkbox"/> MA29V	MARA View on Bills of Materials	A		VIEW	X			P
<input type="checkbox"/> MA61V	Material Master: DB View for MRP	A		VIEW	X			D
<input type="checkbox"/> MAANLF	For Elementary Search Help AANLF (Fund, Functional Area)	A		VIEW	X			D
<input type="checkbox"/> MACKU	Material Valuation View for Costing	A		VIEW	X			D
<input type="checkbox"/> MACKV	Material Plant View on Costing	A		VIEW	X			D
<input type="checkbox"/> MACKW	Material Plant View on Costing	A		VIEW	X			D
<input type="checkbox"/> MAFRUC	View for matchcode ID AFRU-C	A		VIEW	X			D
<input type="checkbox"/> MAPOV	Material view RL-VAS	A		VIEW	X			D
<input type="checkbox"/> MARA1	View on Base Unit of Measure, Material Type, and so on	A		VIEW	X		X	D
<input type="checkbox"/> MARAP	CO-PCA: View of Table MARA	A		VIEW	X		X	D

## ALV Layout for Background Scripts with Server File Output

Up to now, when scheduling a script in background, an **ALV layout** could only be specified for an output directed to SAP spool, but not when creating a server file. This shortcoming has now been eliminated. The dialog window has been enhanced accordingly:



Start script in background

non-standard job name SCRIPT\_\$CUSTOMER\_ORDERS

Job class C Exec.Target

Maximum no. of hits 2.000

Width of output list 1000

Language for additional texts Secondary language

Start immediately

Start delayed Date 21.10.2019 Time 23:00:00

with variant

ALV layout /STANDARD

with reduced memory usage

save result list in server file

direct result list to SAP spool

no end of job notification via SAPmail

E-mail parameters

**NOTE:** Only the column layout and the sort criteria specified by the ALV layout are taken into account when creating the server file. Other features of the ALV layout, such as sums etc., are ignored.

[Back to the top](#)

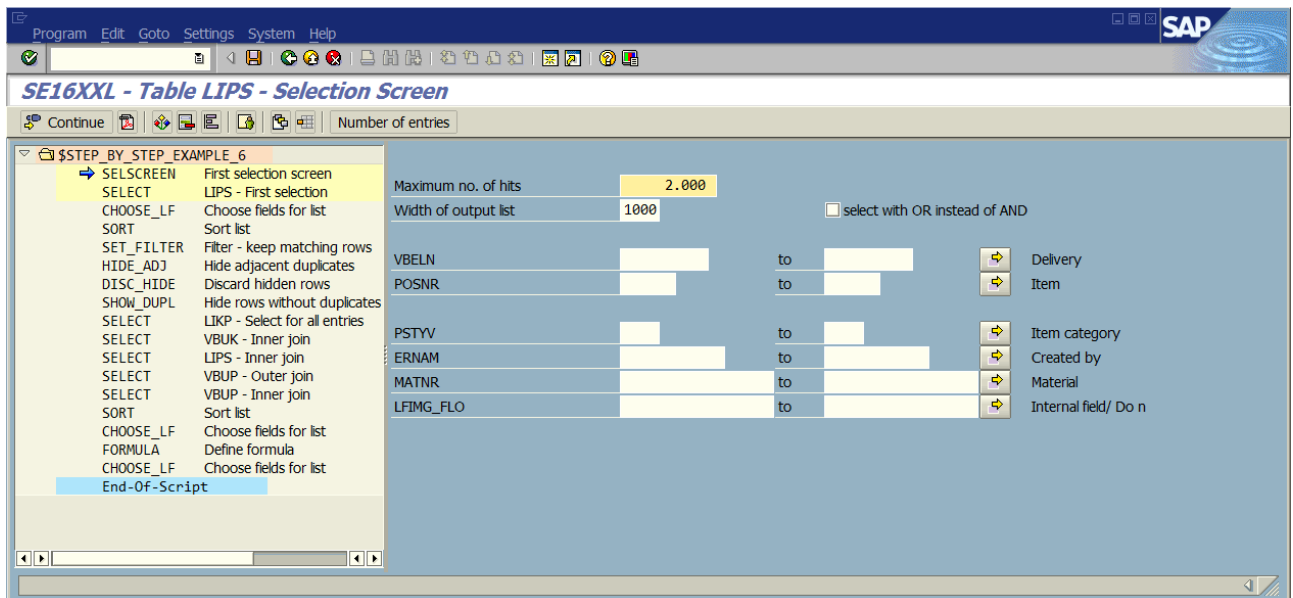
## Perform a Script Step by Step

It is now possible to perform a script step by step. This functionality is helpful to find out why a given script does not behave as expected. In addition it may be used to better understand the internal logic of a complicated script. All this is made possible because the various operations of the script are visibly carried out one by one, showing at every step the intermediate results just as in a normal dialog session. The operations specified by the script may be modified to a certain extent, and additional operations can also be performed. Another useful feature is the possibility to leave the current level and go back to the previous one, i.e. to the last SELECT or UPLOAD operation, in order to perform the last operations once more, possibly with different parameters. Breakpoints are also available to be able to “run” through a series of operations in one big step.

The functionality is available in the Script Catalog by means of the menu function

***Script → Perform script step by step***

An example of a script in step by step mode would be as follows:



For more detailed informations please refer to [Performing a Script Step by Step](#).

[Back to the top](#)

## Global Scripts with multiple Roles

Up to now a global script could have at most one “perform” role (used to check the authorization to perform the script) and one “maintain” role (used to check the authorization to change or delete the script). This applies to “foreign” global scripts, i.e. scripts not created by the current user. This limitation has now been lifted, i.e. starting with SE16XXL version 3.4 it is possible to assign as many roles of both kinds as necessary to each global script.

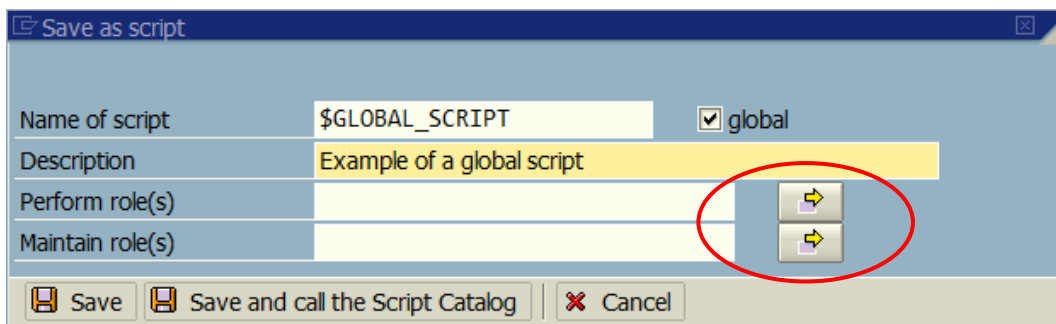
There are two types of roles, single and composite. Until now both types could be assigned to a script, but the dependent roles of composite roles were not taken into account in the authorization logic. The check logic has now been enhanced so as to consider the following four situations:


- 1) A single role is assigned to the script and the user has this role.
- 2) A composite role is assigned to the script and the user has this composite role.
- 3) A single role is assigned to the script and the user has a composite role which contains the single role of the script.
- 4) A composite role is assigned to the script and the user has a single role which is contained in the composite role of the script.

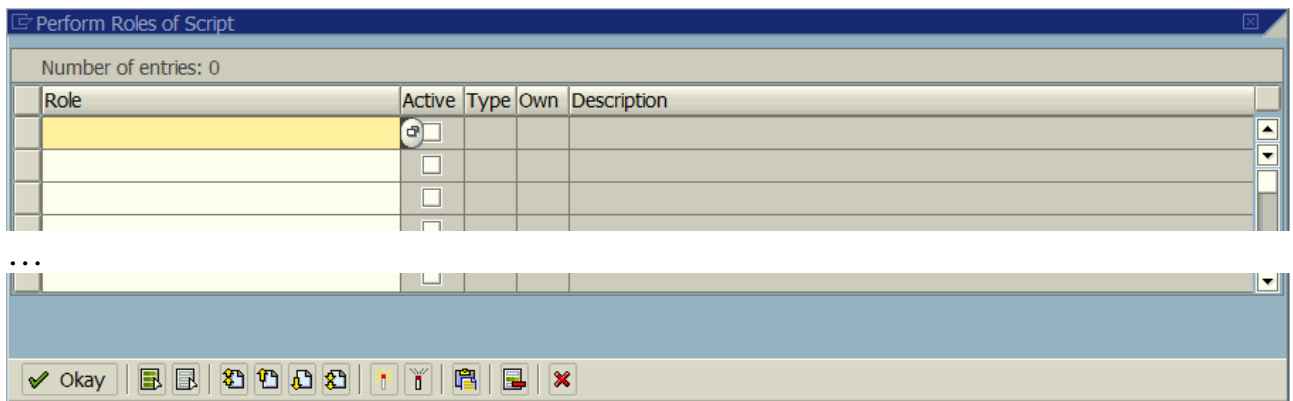
The first two situations were supported in previous versions. Henceforth the last two situations will also be supported.

To be able to perform a given foreign global script, it is sufficient for the user to have **at least one** “perform” role that satisfies one of the four situations listed above. This also applies to changing or deleting the script (“maintain” role).

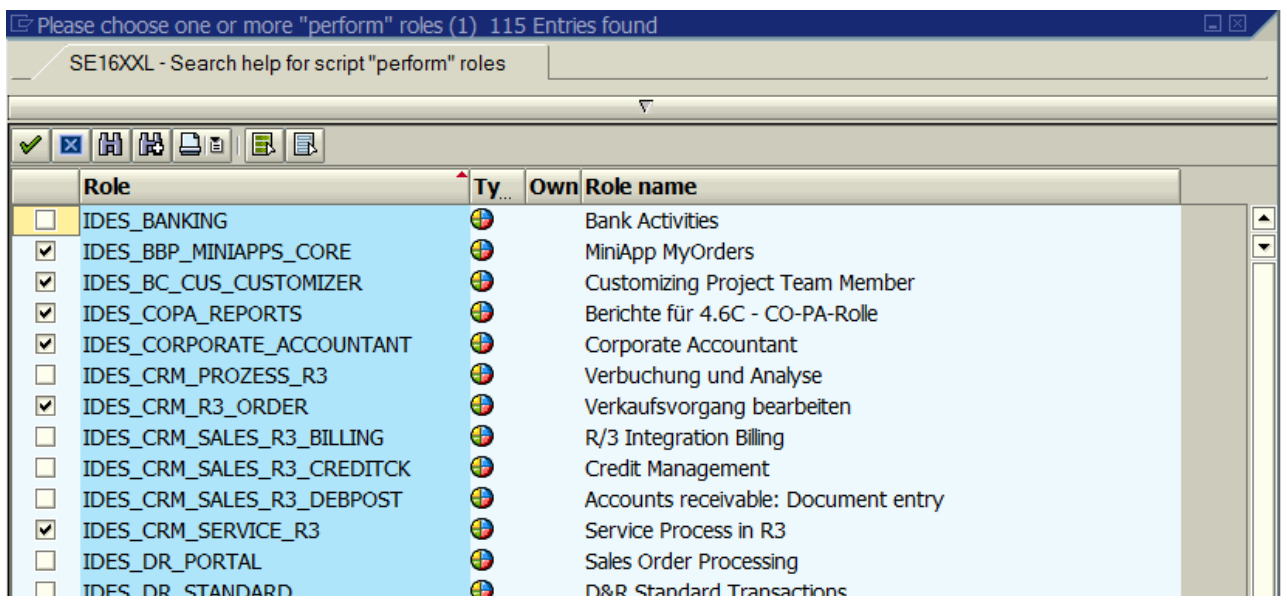
We will now briefly go through the various changes made necessary by the new functionality. Let us start with the dialog screen for saving a script:



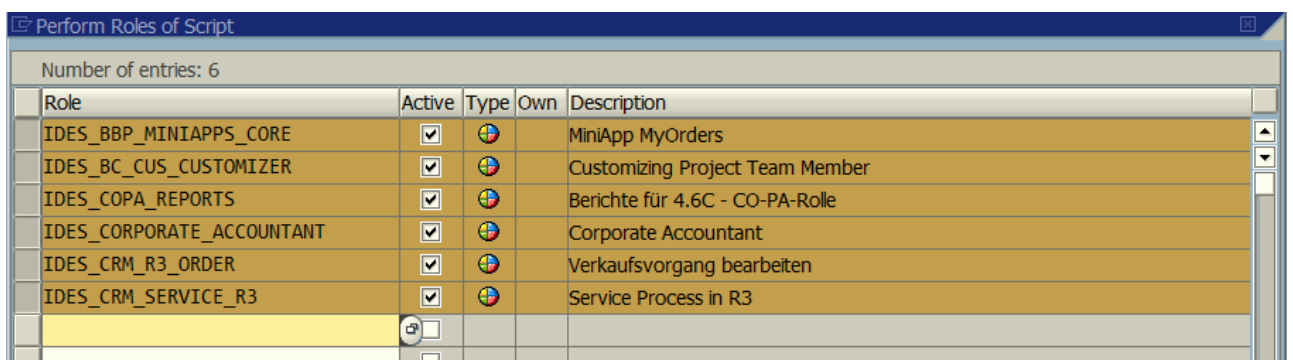
By means of the “more” () buttons additional roles may be specified. In the following example more “perform” roles are being added:



By means of the F4 help it is possible to select more than one role in one pass:

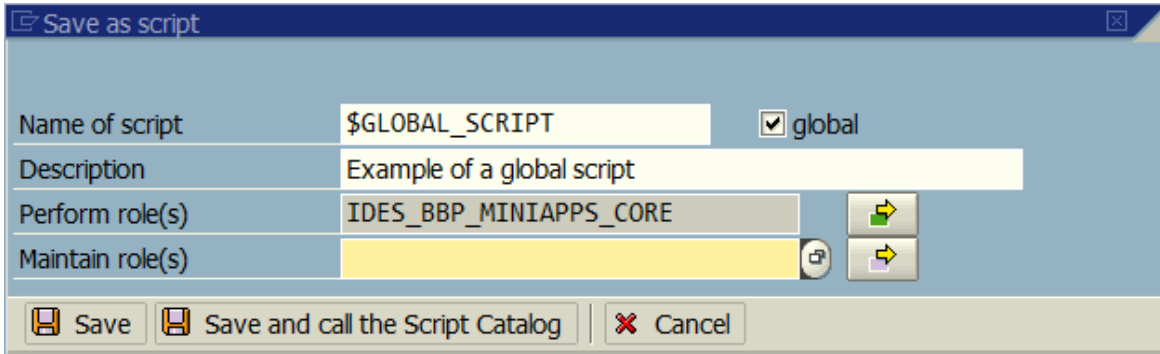


The result would be:

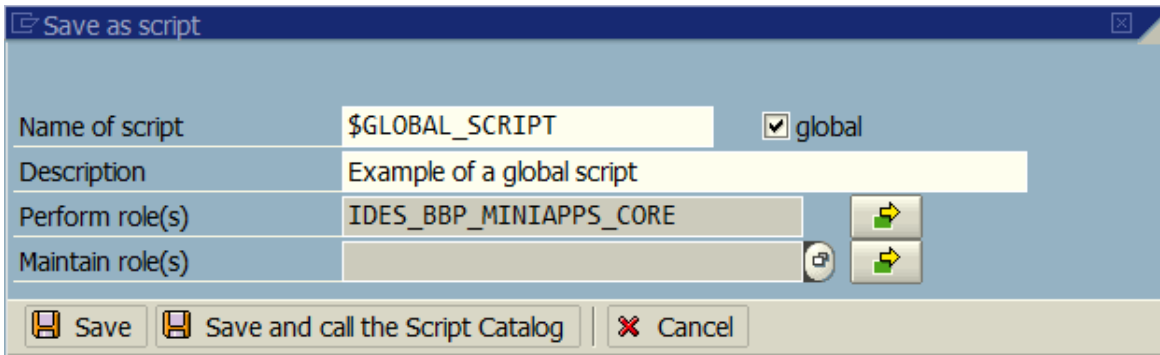


Notice the “Active” flag: only roles marked “active” are taken into account during authorization processing. The flag may be used to eliminate the roles logically, with the possibility of reactivating them at a later time.

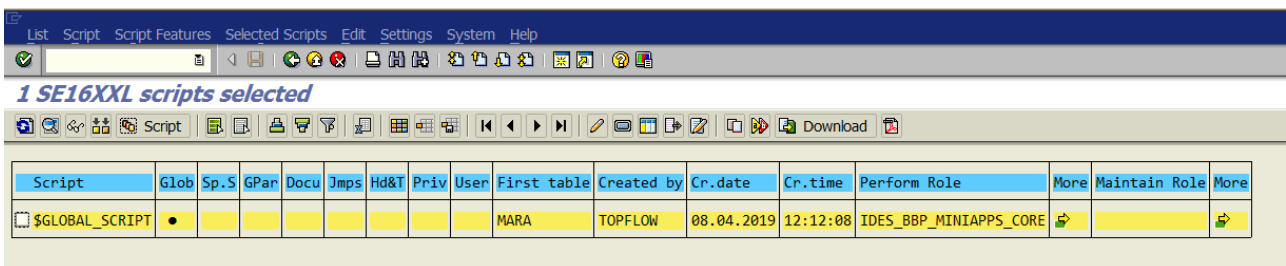
The result on the main saving dialog screen would be as follows:



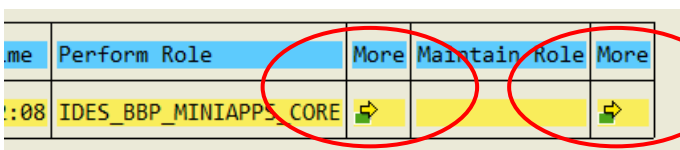
The principle field for the perform roles shows the **first active role** – it is protected, since more than one role has been assigned. If all assigned roles are marked “inactive”, the corresponding field is empty and protected:



The Script Catalog has also been enhanced accordingly:



Notice the new columns for “more” roles:



A double click on the “more” (👉) icon shows the assigned roles in detail.

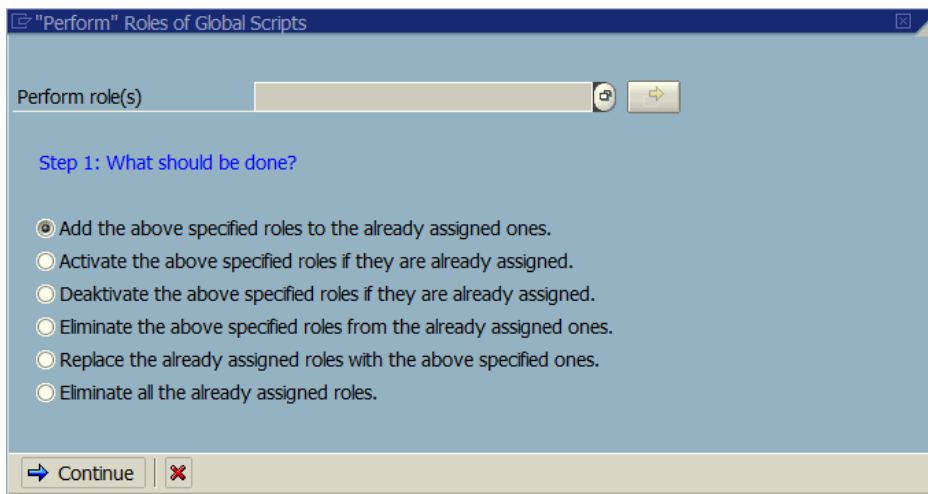


## Changing the roles of several global scripts

The corresponding functions of the Script Catalog have been enhanced so as to take the possibility of multiple roles into account. The two functions

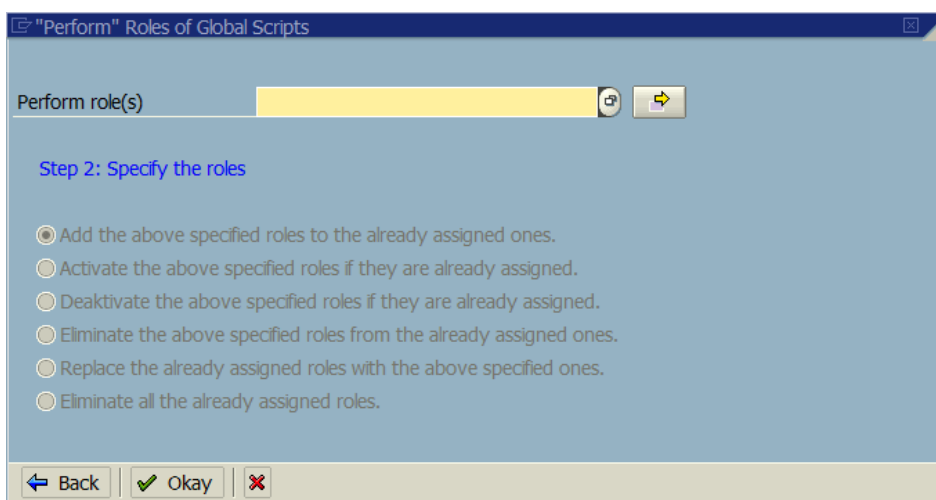
*Selected Scripts* → *Script roles* → *Change perform roles* and  
*Selected Scripts* → *Script roles* → *Change maintain roles*

now offer various alternatives:



First of all the desired activity must be chosen. Make use of the **F1-help** in order to obtain more information about each alternative.

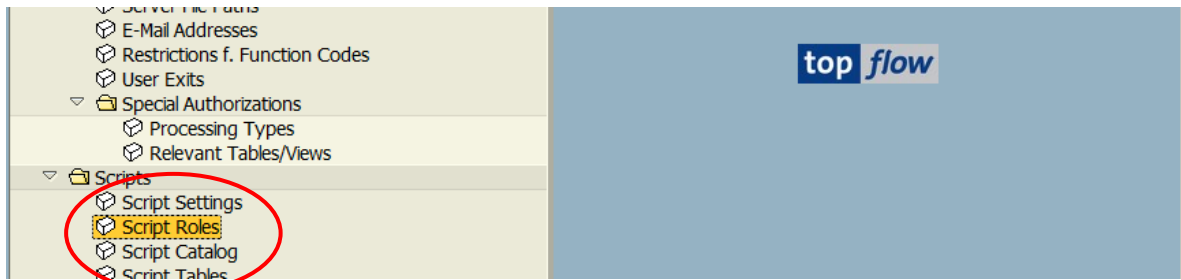
After pressing  the appropriate roles may be specified:



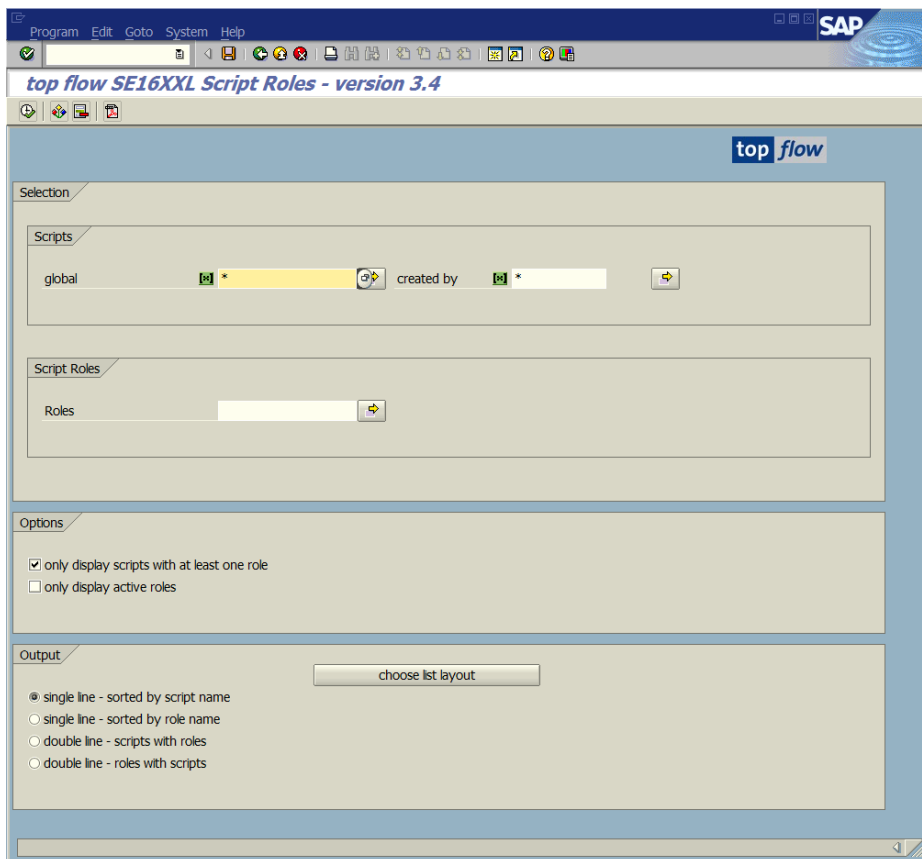
This two step approach has been established since the meaning of the specified roles depends on the selected activity.

## List of Script Roles

Due to the fact that multiple roles may now be assigned to global scripts, the Script Catalog has become inadequate to provide a complete picture of the roles assigned to scripts. For this reason a new utility program has been implemented. It can be called from the SE16XXL Settings:



The selection screen of this tool looks as follows:



For more information please refer to [List of Script Roles](#).

[Back to the top](#)

## Option “Select with Order by Primary Key”

When data are read from the database, the selected rows are not returned in any specific order unless the ORDER BY option is specified in the SELECT statement. Without the ORDER BY option there is no guarantee regarding the sort order of the resulting rows, even if some databases provide this feature implicitly.

The standard form of the above-mentioned option is “ORDER BY PRIMARY KEY”. A SELECT statement with this option will return the rows sorted according to the key fields of the database table.

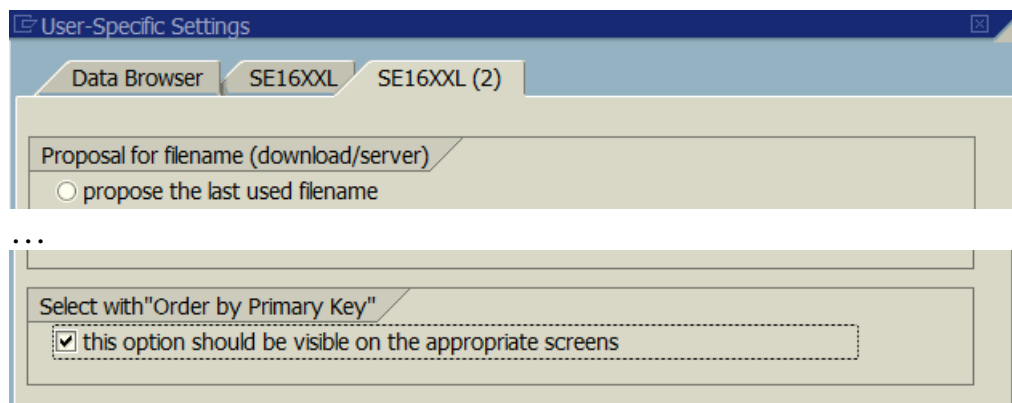
If all the rows that satisfy the selection criteria are selected, there will be no difference in the result except for the sort order (with or without the option).

However, this is not true if fewer rows are selected. An example will illustrate this behavior. Let us suppose that there are 1000 rows in the database that satisfy the selection criteria. If we select only 200 rows without the ORDER BY PRIMARY KEY option, the database will return 200 of the 1000 rows, but not necessarily the first 200. And several SELECTs at different times may return other subsets of 200 rows.

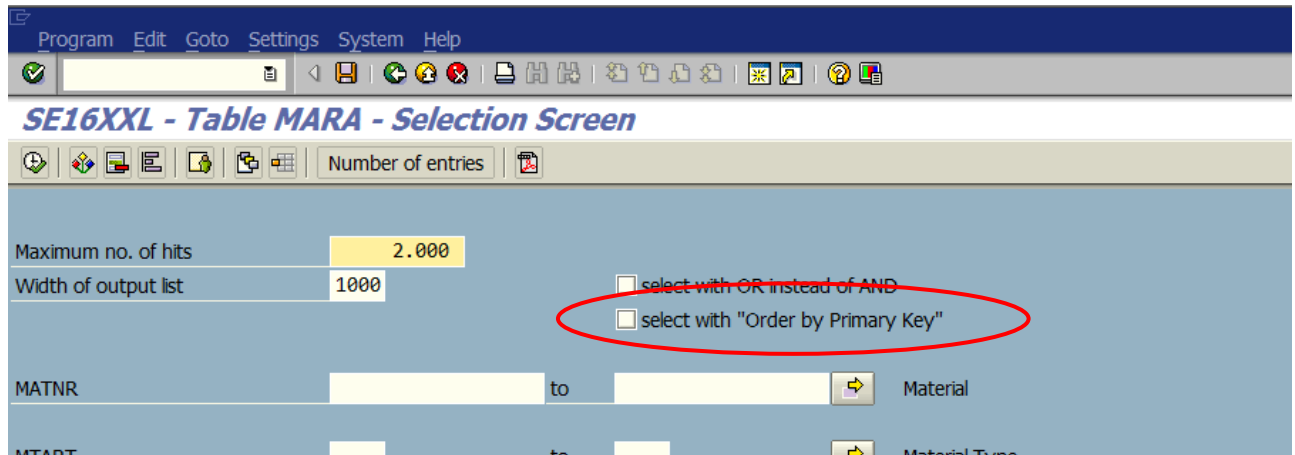
Before the introduction of the SAP HANA<sup>®</sup>, database the recommendation for ABAP<sup>®</sup> programmers was to avoid the ORDER BY option since it puts additional stress on the database server. Instead, it was recommended to use the ABAP<sup>®</sup> SORT statement which runs on the application server.

For SAP HANA<sup>®</sup> the recommendation is exactly the opposite: the ORDER BY PRIMARY KEY option should always be used.

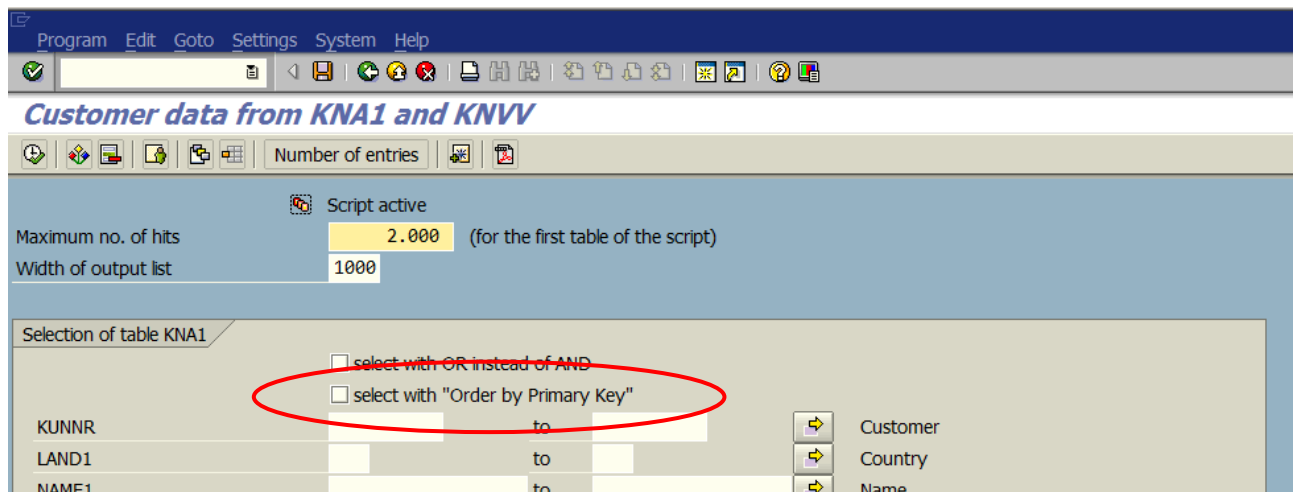
In order to comply with the SAP HANA<sup>®</sup> recommendation, the option “select with Order by Primary Key” has been made available starting with version 3.4 of SE16XXL. The option is normally invisible, since it is only significant in very special cases. It may, however, be made visible by choosing the appropriate setting in the User-Specific Settings:



On the normal selection screen the option shows up as follows:



On a special script selection screen:



The option “select with Order by Primary Key” is only available for the first selection, since only in this case it is possible to specify the maximum number of hits. All subsequent joins (and “select for all entries”) depend on the current result list and therefore select all related data, without any limit. This is also true for scripts – only the first SELECT operation of the script is subject to this option.

For views the ORDER BY option must refer to specific fields – SE16XXL internally uses the key fields of the view explicitly (the first 10 at most).

**NOTE:** The option is preset to ‘X’ for SAP HANA® and to SPACE for all other databases.

[Back to the top](#)

## Additional Improvements

### New File Separator “Vertical Bar Aligned”

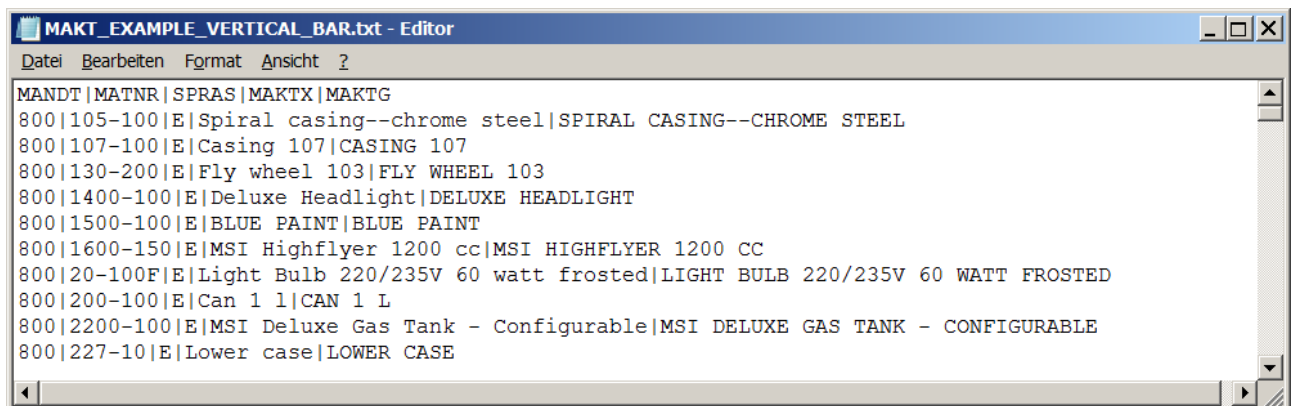
In the SE16XXL result list it is possible to either create a **server file** by means of the following menu function:

*List → Save to server file → As text with separator characters*

or to download the list to a **frontend file** as follows:

*List → Download to frontend file → As text with separator characters.*

Until now, one of the available separators was “| **vertical bar**”. The resulting file would look appropriate viewed on a spreadsheet, but would make a rather jumbled appearance in a text editor:

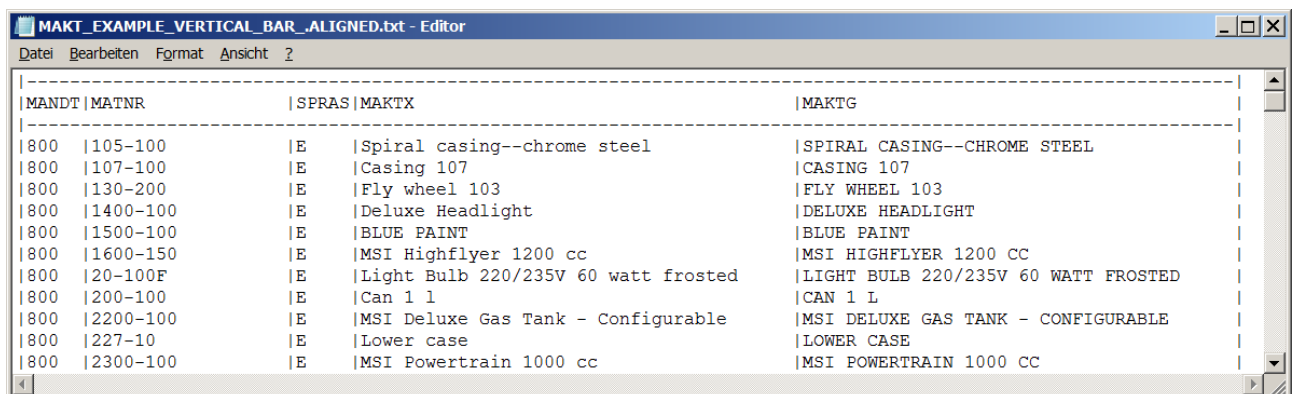


```

MAKT_EXAMPLE_VERTICAL_BAR.txt - Editor
Datei Bearbeiten Format Ansicht ?
MANDT|MATNR|SPRAS|MAKTX|MAKTG
800|105-100|E|Spiral casing--chrome steel|SPIRAL CASING--CHROME STEEL
800|107-100|E|Casing 107|CASING 107
800|130-200|E|Fly wheel 103|FLY WHEEL 103
800|1400-100|E|Deluxe Headlight|DELUXE HEADLIGHT
800|1500-100|E|BLUE PAINT|BLUE PAINT
800|1600-150|E|MSI Highflyer 1200 cc|MSI HIGHFLYER 1200 CC
800|20-100F|E|Light Bulb 220/235V 60 watt frosted|LIGHT BULB 220/235V 60 WATT FROSTED
800|200-100|E|Can 1 l|CAN 1 L
800|2200-100|E|MSI Deluxe Gas Tank - Configurable|MSI DELUXE GAS TANK - CONFIGURABLE
800|227-10|E|Lower case|LOWER CASE

```

With the newly available separator “| **vertical bar aligned**” the file would present itself as follows in a text editor:



```

MAKT_EXAMPLE_VERTICAL_BAR_ALIGNED.txt - Editor
Datei Bearbeiten Format Ansicht ?
-----|-----|-----|-----|-----|
|MANDT|MATNR| |SPRAS|MAKTX| |MAKTG| | |
|---|---|---|---|---|---|---|---|---|
|800| |105-100| |E| |Spiral casing--chrome steel| |SPIRAL CASING--CHROME STEEL|
|800| |107-100| |E| |Casing 107| |CASING 107|
|800| |130-200| |E| |Fly wheel 103| |FLY WHEEL 103|
|800| |1400-100| |E| |Deluxe Headlight| |DELUXE HEADLIGHT|
|800| |1500-100| |E| |BLUE PAINT| |BLUE PAINT|
|800| |1600-150| |E| |MSI Highflyer 1200 cc| |MSI HIGHFLYER 1200 CC|
|800| |20-100F| |E| |Light Bulb 220/235V 60 watt frosted| |LIGHT BULB 220/235V 60 WATT FROSTED|
|800| |200-100| |E| |Can 1 l| |CAN 1 L|
|800| |2200-100| |E| |MSI Deluxe Gas Tank - Configurable| |MSI DELUXE GAS TANK - CONFIGURABLE|
|800| |227-10| |E| |Lower case| |LOWER CASE|
|800| |2300-100| |E| |MSI Powertrain 1000 cc| |MSI POWERTRAIN 1000 CC|

```

## Script Catalog – New Functions for Script Variants

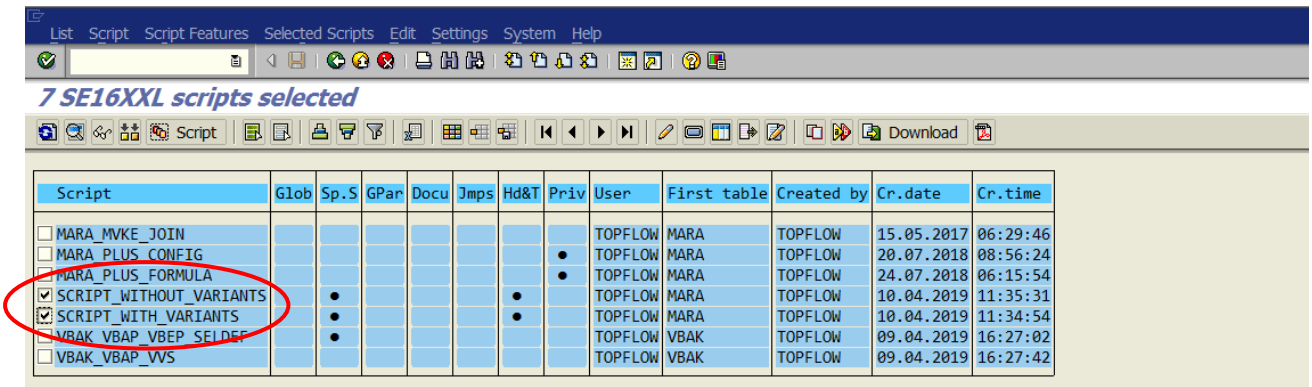
When a script with a special selection screen is copied, the associated script variants may also be copied to the target script. Similarly, when scripts are downloaded, it is possible to store the associated script variants in the same frontend file, from which they can be imported or copied at a later time.

Until now, however, if a script was copied without taking care of the variants, there was no way of copying them at a later time. Since special selection screens tend to be quite complicated, this missing functionality may become a serious problem, especially if numerous script variants have to be recreated manually.

With SE16XXL version 3.4 this gap has been closed. The Script Catalog now offers the following menu function:

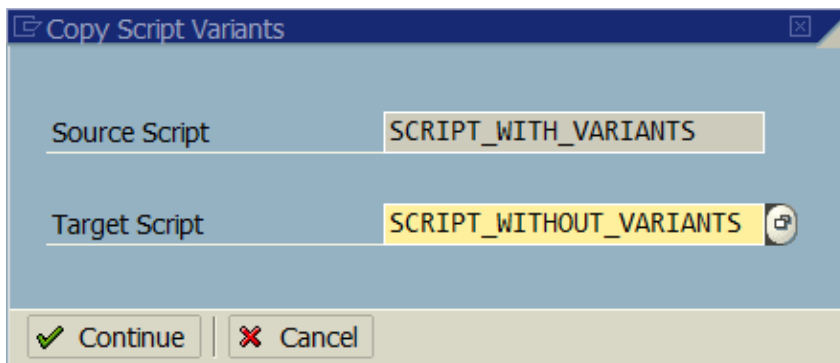
**Script Features** → **Copy selected script variants**

The user may choose between selecting **only** the source script or **both** the source **and** the target script:



Script	Glob	Sp.S	GPar	Docu	Jmps	Hd&T	Priv	User	First table	Created by	Cr.date	Cr.time
<input type="checkbox"/> MARA_MVKE_JOIN								TOPFLOW	MARA	TOPFLOW	15.05.2017	06:29:46
<input type="checkbox"/> MARA_PLUS_CONFIG							•	TOPFLOW	MARA	TOPFLOW	20.07.2018	08:56:24
<input type="checkbox"/> MARA_PLUS_FORMULA							•	TOPFLOW	MARA	TOPFLOW	24.07.2018	06:15:54
<input checked="" type="checkbox"/> SCRIPT_WITHOUT_VARIANTS		•				•		TOPFLOW	MARA	TOPFLOW	10.04.2019	11:35:31
<input checked="" type="checkbox"/> SCRIPT_WITH_VARIANTS		•				•		TOPFLOW	MARA	TOPFLOW	10.04.2019	11:34:54
<input type="checkbox"/> VBAK_VBAP_VBEP_SELDEF		•						TOPFLOW	VBAK	TOPFLOW	09.04.2019	16:27:02
<input type="checkbox"/> VBAK_VBAP_VVS								TOPFLOW	VBAK	TOPFLOW	09.04.2019	16:27:42

When the above mentioned function is called, the following dialog window appears:



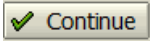
Copy Script Variants

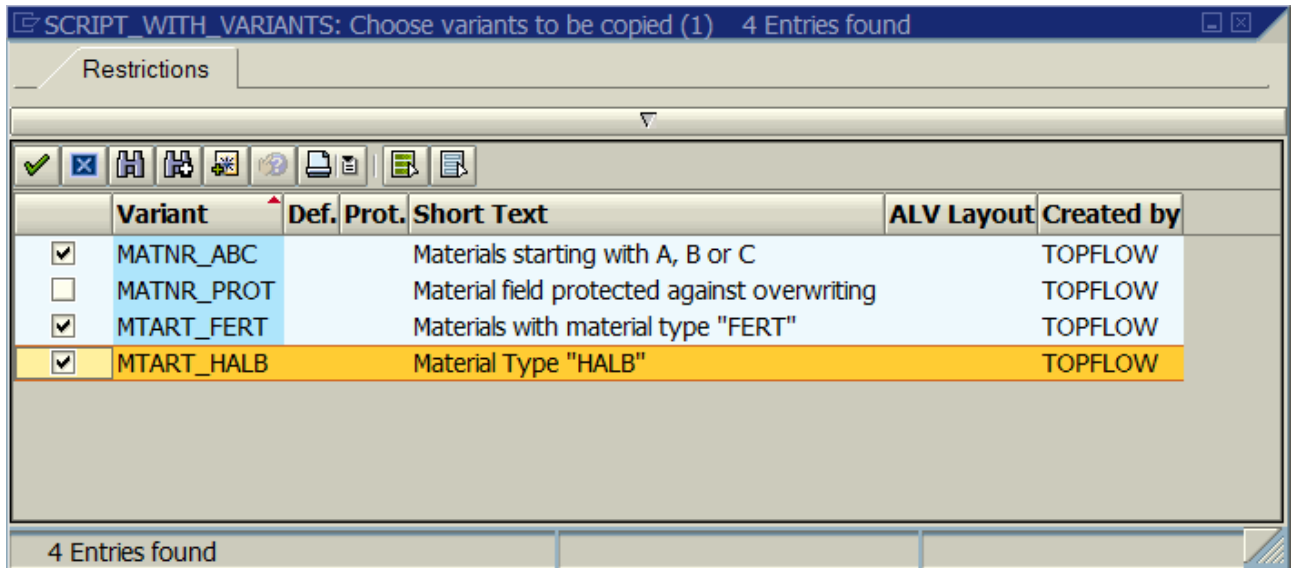
Source Script: SCRIPT\_WITH\_VARIANTS

Target Script: SCRIPT\_WITHOUT\_VARIANTS

Continue Cancel

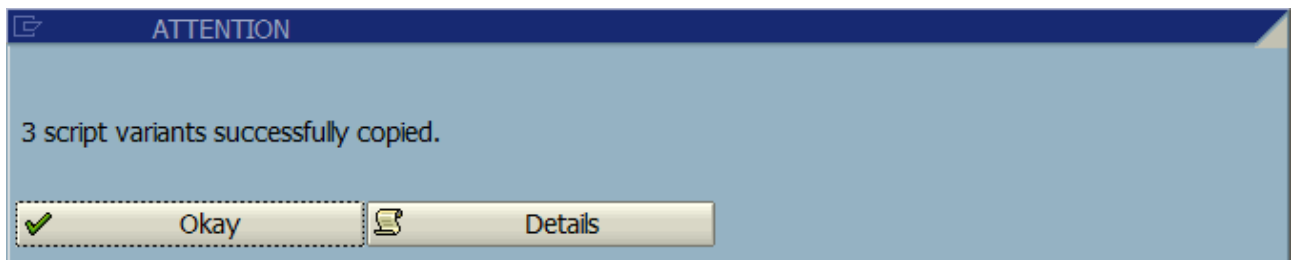
Notice that the program has automatically taken the script with variants as source.

After pressing  a list of the variants of the source script shows up. We select some of them to be copied:

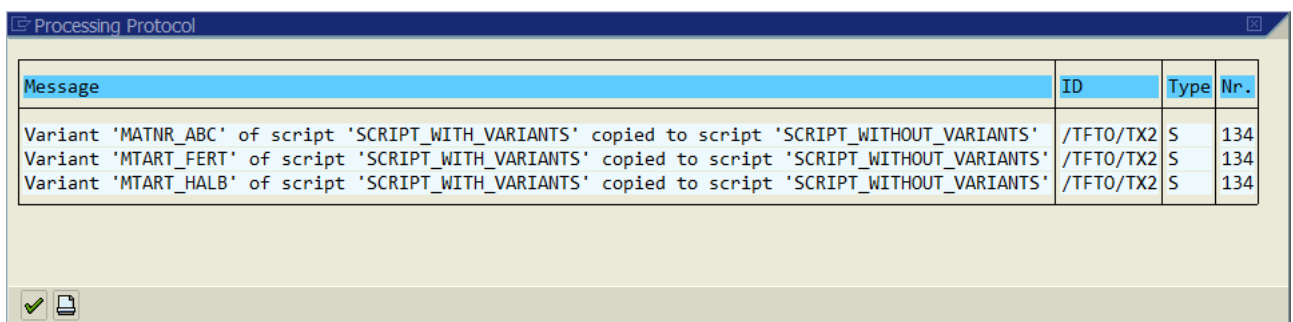


	Variant	Def.	Prot.	Short Text	ALV Layout	Created by
<input checked="" type="checkbox"/>	MATNR_ABC			Materials starting with A, B or C		TOPFLOW
<input type="checkbox"/>	MATNR_PROT			Material field protected against overwriting		TOPFLOW
<input checked="" type="checkbox"/>	MTART_FERT			Materials with material type "FERT"		TOPFLOW
<input checked="" type="checkbox"/>	MTART_HALB			Material Type "HALB"		TOPFLOW

The following message announces the end of the operation:



If  is pressed, a detailed account of the processing appears:

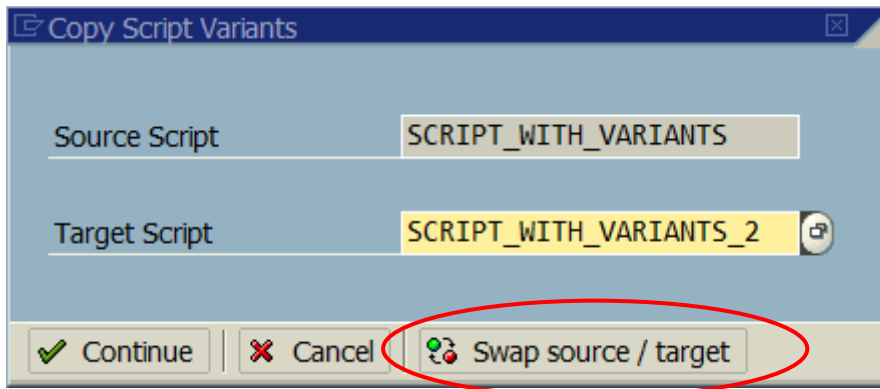


Message	ID	Type	Nr.
Variant 'MATNR_ABC' of script 'SCRIPT_WITH_VARIANTS' copied to script 'SCRIPT_WITHOUT_VARIANTS'	/TFT0/TX2	S	134
Variant 'MTART_FERT' of script 'SCRIPT_WITH_VARIANTS' copied to script 'SCRIPT_WITHOUT_VARIANTS'	/TFT0/TX2	S	134
Variant 'MTART_HALB' of script 'SCRIPT_WITH_VARIANTS' copied to script 'SCRIPT_WITHOUT_VARIANTS'	/TFT0/TX2	S	134

**NOTE:** It must be observed, that the program **does not check** whether the variants to be copied are compatible with the logic of the target script. This is the responsibility of the user. Incompatible variants will be rejected as “obsolete” during the loading process.



In case both involved scripts already have some variants, the dialog window offers the possibility to swap source and target:



### Function for Deleting Script Variants

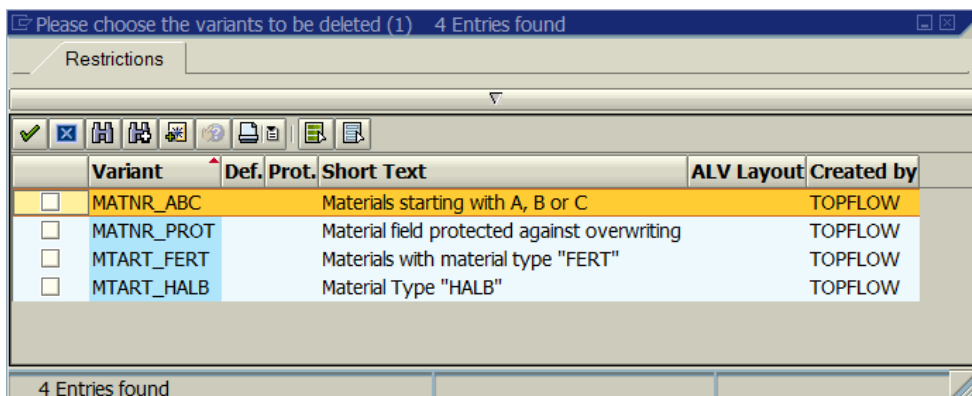
In addition to the function described above, there is a second one for deleting selected variants:

*Script Features* → *Delete selected script variants*

This second function is not strictly needed, since there is already a general function to this effect on the main screen of SE16XXL (*Goto* → *Script variants* ...).

Nonetheless, it is quite convenient to have such a function in the Script Catalog as well.

After calling this function for a given script the list of associated variants shows up:

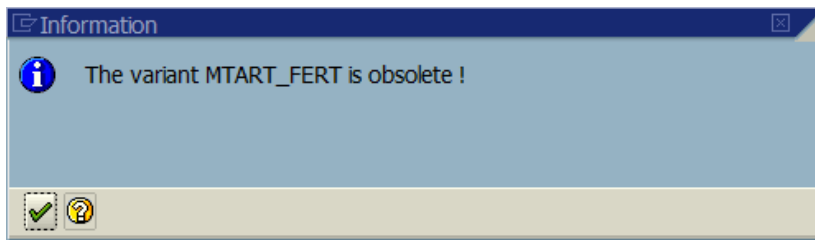


	Variant	Def.	Prot.	Short Text	ALV Layout	Created by
<input type="checkbox"/>	MATNR_ABC			Materials starting with A, B or C		TOPFLOW
<input type="checkbox"/>	MATNR_PROT			Material field protected against overwriting		TOPFLOW
<input type="checkbox"/>	MTART_FERT			Materials with material type "FERT"		TOPFLOW
<input type="checkbox"/>	MTART_HALB			Material Type "HALB"		TOPFLOW

Before carrying out the deletion the program asks the user to confirm the operation.

## Improved Compatibility Check of Script Variants

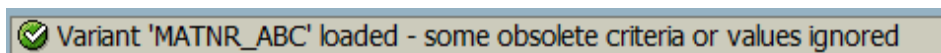
When a script variant is created, it is provided, among other things, with a minimal copy of the operations log of the associated script together with a minimal copy of the selection screen definition. At load time this information is used to check whether the variant is still compatible with the script to be performed. If the data do not match, the program considers the variant to be obsolete and refuses to load it. The following message is issued:



Two new features of SE16XXL version 3.4 have made an improvement of the check logic advisable:

- 1) The possibility of copying variants from one script to another;
- 2) The possibility of performing a script step by step.

In both cases situations may arise, where a script variant only approximately matches the operational structure of the associated script. If the check logic were left unchanged, a lot of variants would end up as “obsolete”, thereby frustrating the efforts of the users. For this reason a higher level of tolerance has been implemented into the check logic. Variants, which in older versions would have been considered “obsolete” are now accepted, possibly with a hint in regard to obsolete criteria and/or values:



**NOTE:** It is still possible for a given script variant to be considered obsolete. This situation, however, should occur less frequently.

[Back to the top](#)

## Programs adjusted for latest version of the S/4 system

Up to now on SAP HANA® based systems there were no cluster or pool tables in the database, only transparent ones. However, the associated Data Dictionary tables were still present, albeit empty. This situation has changed in the latest SAP HANA® based S/4 system: the DDIC tables used for describing clusters and pools (DD06L, DD06T and DD16S) have been eliminated altogether. As a consequence, some of the programs and function groups of the SE16XXL package produced syntax errors when compiled.

One of the guiding principles of the SE16XXL development has always been to have only one source code which runs on all current versions of the SAP system. Therefore, the coding of all programs and function groups involved has been adapted to cope with the new situation, i.e. that on some systems the above mentioned DDIC tables may not be available.

The following coding excerpt gives an idea of how this has been accomplished:

```

CLEAR LL_RC.
TRY.
  CREATE DATA LD_REC_REF TYPE (C_TAB_DD06T).
  CATCH CX_ROOT.                                "#EC CATCH_ALL
    LL_RC = 4.
  ENDTRY.
CHECK LL_RC = 0.

ASSIGN LD_REC_REF->* TO <REC>.
CHECK SY-SUBRC = 0.

CLEAR LL_RC.
TRY.
  CREATE DATA LD_TAB_REF LIKE STANDARD TABLE OF <REC>.
  CATCH CX_ROOT.                                "#EC CATCH_ALL
    LL_RC = 4.
  ENDTRY.
CHECK LL_RC = 0.

ASSIGN LD_TAB_REF->* TO <T_RECS>.
CHECK SY-SUBRC = 0.

CLEAR <T_RECS>[].
CLEAR LL_RC.
TRY.
  SELECT * FROM (C_TAB_DD06T) INTO TABLE <T_RECS>
          WHERE SQLTAB      = LL_SQLTAB
          AND DDLANGUAGE    = PP_LANGUAGE
          AND AS4LOCAL      = 'A'.

  CATCH CX_ROOT.                                "#EC CATCH_ALL
    LL_RC = 4.
  ENDTRY.
CHECK LL_RC = 0.

```

[Back to the top](#)

## ATNAM as selection criterion for \$CLASSIF & \$CONFIG

The field **ATNAM** (Characteristic Name) is now available as selection criterion for the two pseudo tables **\$CLASSIF** and **\$CONFIG**. This feature has been made available in order to offer more flexibility during selection.

Up to now the selection criteria of the two pseudo tables were taken exclusively from the underlying base tables (**KSSK** or **INOB** for **\$CLASSIF**, **IBIN** for **\$CONFIG**). The field **ATNAM** does not belong to these base tables. Its values become known only after the final result has been produced by calling the corresponding function modules and/or methods. This means that no increase in performance is to be expected from restricting the values of **ATNAM** during selection. For experienced users however, the selection process should be easier to handle.

### Selection Criteria for \$CLASSIF

*SE16XXL - Table \$CLASSIF - Selection Screen*

Number of entries

P_CLASSTYPE			P_ClassType
P_OBJECTTABLE			P_ObjectTable
P_LANGUAGE	EN		P_Language
P_KEYDATE	31.10.2018		P_KeyDate
Maximum no. of hits	2.000		
Width of output list	1000	<input type="checkbox"/> select with OR instead of AND	
OBJEK		to	Object
CLINT		to	Int class no.
CLASS		to	Class
ATNAM		to	Characteristic

### Selection Criteria for \$CONFIG

*SE16XXL - Table \$CONFIG - Selection Screen*

Number of entries

P_LANGUAGE	EN		P_Language
Maximum no. of hits	2.000		
Width of output list	1000	<input type="checkbox"/> select with OR instead of AND	
CUOBJ		to	Int. object no.
ATNAM		to	Characteristic

[Back to the top](#)

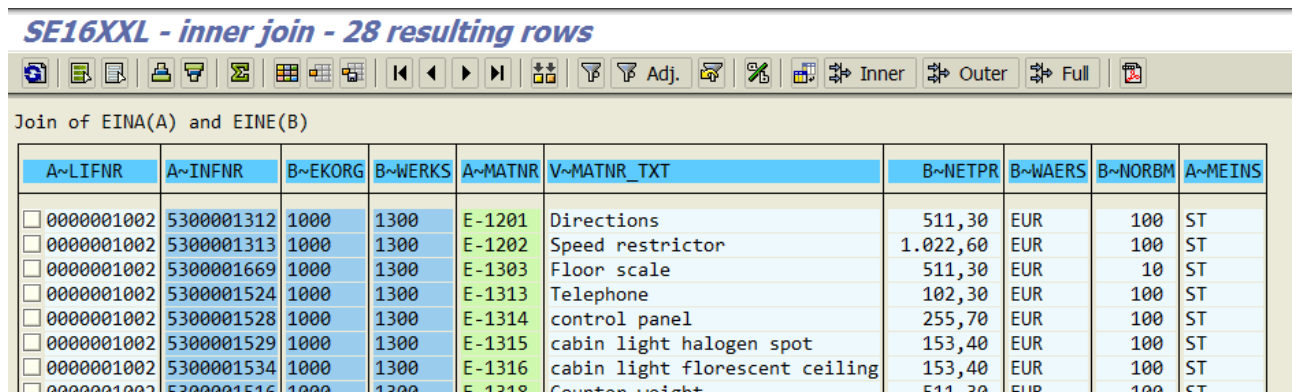
## Upload of Frontend Files at any point

Up to now a frontend file could only be uploaded at the beginning of an SE16XXL session (start file). This restriction has been lifted. It is now possible to upload a frontend file at **any** point in a session, i.e. just like performing a SELECT operation. The operations that are available for database tables (select for all entries, inner join, outer join and full join) can now also be performed with a frontend file.

In order to show how this functionality has been integrated, a simple example will be discussed.

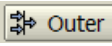
Suppose we have a list of the info records for a particular vendor and plant:

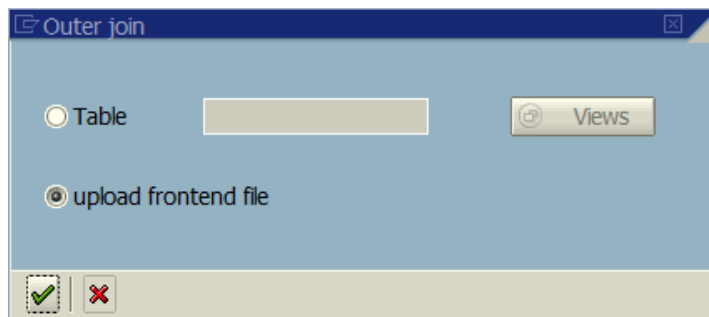
**SE16XXL - inner join - 28 resulting rows**



A~LIFNR	A~INFNR	B~EKORG	B~WERKS	A~MATNR	V~MATNR_TXT	B~NETPR	B~WAERS	B~NORBM	A~MEINS
0000001002	5300001312	1000	1300	E-1201	Directions	511,30	EUR	100	ST
0000001002	5300001313	1000	1300	E-1202	Speed restrictor	1.022,60	EUR	100	ST
0000001002	5300001669	1000	1300	E-1303	Floor scale	511,30	EUR	10	ST
0000001002	5300001524	1000	1300	E-1313	Telephone	102,30	EUR	100	ST
0000001002	5300001528	1000	1300	E-1314	control panel	255,70	EUR	100	ST
0000001002	5300001529	1000	1300	E-1315	cabin light halogen spot	153,40	EUR	100	ST
0000001002	5300001534	1000	1300	E-1316	cabin light florescent ceiling	153,40	EUR	100	ST
0000001002	5300001516	1000	1300	E-1318	Counter weight	511,30	EUR	100	ST

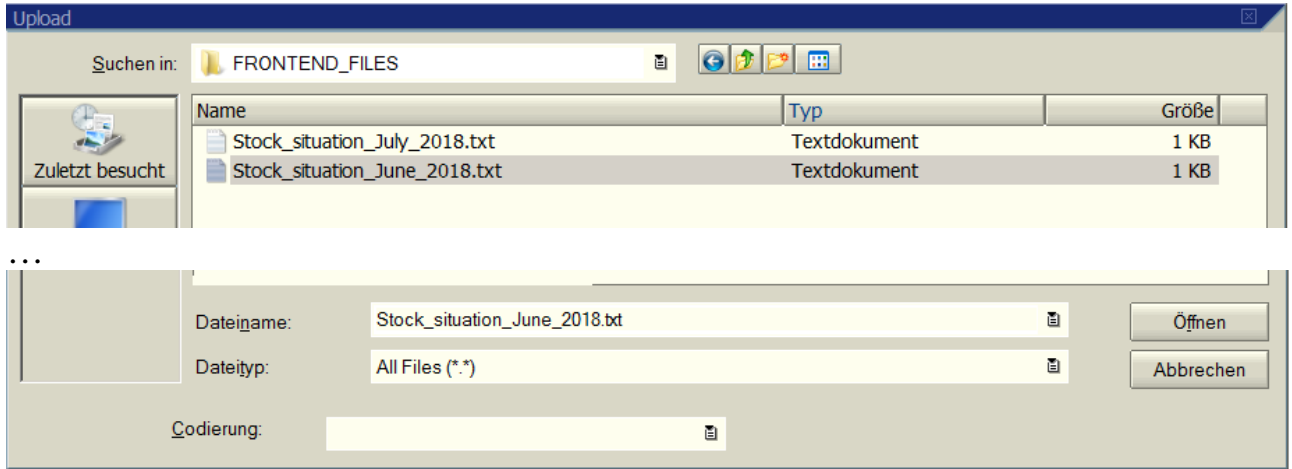
From the vendor we receive each month a file containing the storage situation of the relevant materials. We would now like to add the information of the past two months to our list.

We mark the A~MATNR column and start with the previous month's file. We make use of an outer join ( Outer):

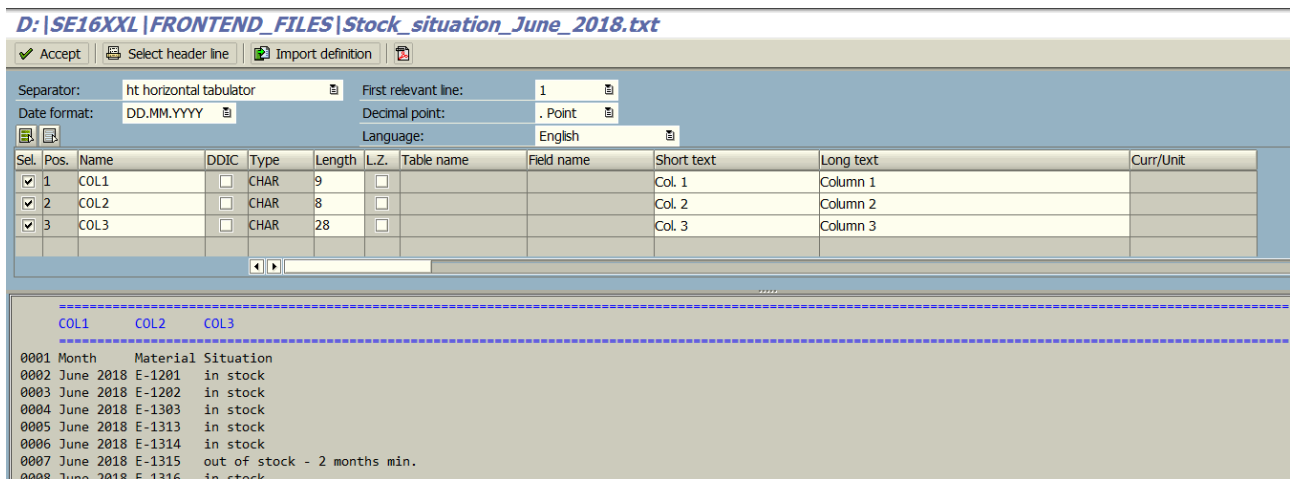


But instead of specifying a database table, we check the "*upload frontend file*" option.

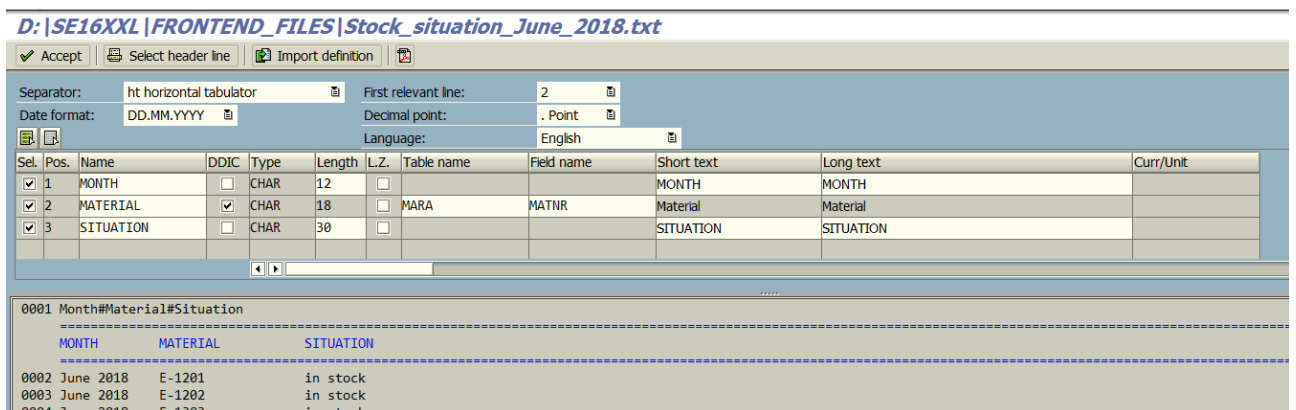
The program responds by showing the file selector:



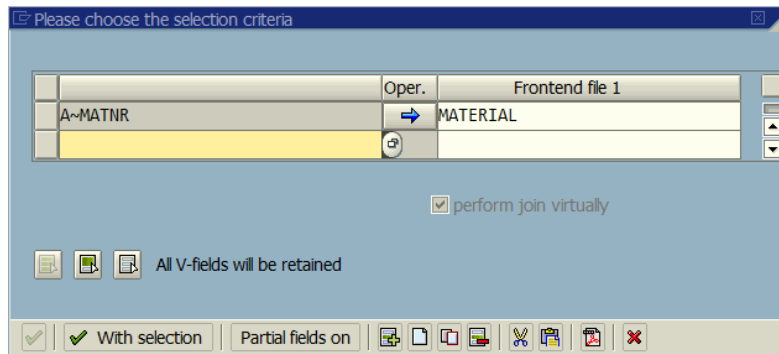
After the frontend file has been chosen the definition screen (which we already know from start files) shows up:



We make some necessary adjustments to the definition:

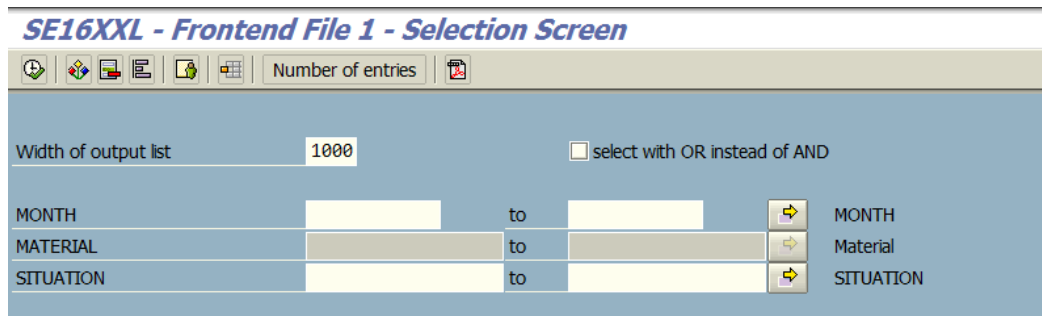


When the column definition appears to be satisfactory we press to proceed. Now the dialog window for specifying the join criteria pops up:



Notice that the option "perform join virtually" is active. This means that the frontend file will be completely uploaded and the join carried out in memory.

At this point the selection screen for the frontend file is displayed:



We acknowledge without entering any restrictions by pressing the execute () button. Finally our join result shows up:

**SE16XXL - outer join - 28 resulting rows**

Join of EINA(A) EINE(B) and Frontend file 1(C)

A~LIFNR	A~INFNR	B~EKORG	B~WERKS	A~MATNR	V~MATNR_TXT	B~NETPR	B~WAERS	B~NORBM	A~MEINS	C~MONTH	C~MATERIAL	C~SITUATION	
<input type="checkbox"/>	0000001002	5300001312	1000	1300	E-1201	Directions	511,30	EUR	100	ST	June 2018	E-1201	in stock
<input type="checkbox"/>	0000001002	5300001313	1000	1300	E-1202	Speed restrictor	1.022,60	EUR	100	ST	June 2018	E-1202	in stock
<input type="checkbox"/>	0000001002	5300001669	1000	1300	E-1303	Floor scale	511,30	EUR	10	ST	June 2018	E-1303	in stock
<input type="checkbox"/>	0000001002	5300001524	1000	1300	E-1313	Telephone	102,30	EUR	100	ST	June 2018	E-1313	in stock
<input type="checkbox"/>	0000001002	5300001528	1000	1300	E-1314	control panel	255,70	EUR	100	ST	June 2018	E-1314	in stock
<input type="checkbox"/>	0000001002	5300001529	1000	1300	E-1315	cabin light halogen spot	153,40	EUR	100	ST	June 2018	E-1315	out of stock - 2 months min.
<input type="checkbox"/>	0000001002	5300001534	1000	1300	E-1316	cabin light florescent ceiling	153,40	EUR	100	ST	June 2018	E-1316	in stock
<input type="checkbox"/>	0000001002	5300001516	1000	1300	E-1318	Counter weight	511,30	EUR	100	ST	June 2018	E-1318	in stock
<input type="checkbox"/>	0000001002	5300001708	1000	1300	E-1501	Engine unit 5 KW 380V AC	5.112,80	EUR	20	ST	June 2018	E-1501	in stock
<input type="checkbox"/>	0000001002	5300001709	1000	1300	E-1502	Engine unit 10 KW 380V AC	6.135,36	EUR	20	ST	June 2018	E-1502	in stock
<input type="checkbox"/>	0000001002	5300001710	1000	1300	E-1503	Engine unit 20 KW 380V AC	7.157,92	EUR	20	ST	June 2018	E-1503	in stock
<input type="checkbox"/>	0000001002	5300001515	1000	1300	E-1504	motor 5 KW 380V AC 3 phases	511,30	EUR	100	ST	June 2018	E-1504	in stock
<input type="checkbox"/>	0000001002	5300001514	1000	1300	E-1505	Motor 5 KW 460V AC 3 phases	511,30	EUR	100	ST	June 2018	E-1505	in stock
<input type="checkbox"/>	0000001002	5300001513	1000	1300	E-1506	Motor 10 KW 380V AC 3 phases	766,90	EUR	100	ST	June 2018	E-1506	in stock
<input type="checkbox"/>	0000001002	5300001512	1000	1300	E-1507	Motor 10 KW 460V AC 3 phases	766,90	EUR	100	ST	June 2018	E-1507	in stock
<input type="checkbox"/>	0000001002	5300001511	1000	1300	E-1508	Motor 20 KW 380V AC 3 phases	1.278,20	EUR	100	ST	June 2018	E-1508	in stock
<input type="checkbox"/>	0000001002	5300001510	1000	1300	E-1509	Motor 20 KW 460V AC 3 phases	1.278,20	EUR	100	ST	June 2018	E-1509	in stock
<input type="checkbox"/>	0000001002	5300001637	1000	1300	E-1510	Motor 50 KW 380V AC 3 phases	1.789,48	EUR	1	ST	June 2018	E-1510	no longer available
<input type="checkbox"/>	0000001002	5300001656	1000	1300	E-1511	Motor 50 KW 460V AC 3 phases	1.789,48	EUR	1	ST	June 2018	E-1511	only a few pieces left
<input type="checkbox"/>	0000001002	5300001509	1000	1300	E-1512	Gearbox 1	766,90	EUR	100	ST	June 2018	E-1512	in stock
<input type="checkbox"/>	0000001002	5300001508	1000	1300	E-1513	Gearbox 2	1.278,20	EUR	100	ST	June 2018	E-1513	out of stock - 3 weeks
<input type="checkbox"/>	0000001002	5300001507	1000	1300	E-1514	Gearbox 3	1.789,50	EUR	100	ST	June 2018	E-1514	in stock

As may be seen, the fields with alias **C** belong to the uploaded frontend file.



Now we repeat the same procedure with the second frontend file containing the data for the current month. The final result list is as follows:

**SE16XXL - outer join - 28 resulting rows**

Join of EINA(A) EINE(B) Frontend file 1(C) and Frontend file 2(D)

A-LIFNR	B-EKORG	B-WERKS	A-MATNR	V-MATNR_TXT	B-NETPR	B-WAERS	C-MONTH	C-MATERIAL	C-SITUATION	D-MONTH	D-MATERIAL	D-SITUATION
0000001002	1000	1300	E-1201	Directions	511,30	EUR	June 2018	E-1201	in stock	July 2018	E-1201	in stock
0000001002	1000	1300	E-1202	Speed restrictor	1.022,60	EUR	June 2018	E-1202	in stock	July 2018	E-1202	in stock
0000001002	1000	1300	E-1303	Floor scale	511,30	EUR	June 2018	E-1303	in stock	July 2018	E-1303	out of stock - 3 to 6 weeks
0000001002	1000	1300	E-1313	Telephone	102,30	EUR	June 2018	E-1313	in stock	July 2018	E-1313	in stock
0000001002	1000	1300	E-1314	control panel	255,70	EUR	June 2018	E-1314	in stock	July 2018	E-1314	in stock
0000001002	1000	1300	E-1315	cabin light halogen spot	153,40	EUR	June 2018	E-1315	out of stock - 2 months min.	July 2018	E-1315	out of stock - 4 weeks
0000001002	1000	1300	E-1316	cabin light florescent ceiling	153,40	EUR	June 2018	E-1316	in stock	July 2018	E-1316	in stock
0000001002	1000	1300	E-1318	Counter weight	511,30	EUR	June 2018	E-1318	in stock	July 2018	E-1318	in stock
0000001002	1000	1300	E-1501	Engine unit 5 KW 380V AC	5.112,80	EUR	June 2018	E-1501	in stock	July 2018	E-1501	in stock
0000001002	1000	1300	E-1502	Engine unit 10 KW 380V AC	6.135,36	EUR	June 2018	E-1502	in stock	July 2018	E-1502	in stock
0000001002	1000	1300	E-1503	Engine unit 20 KW 380V AC	7.157,92	EUR	June 2018	E-1503	in stock	July 2018	E-1503	no longer available
0000001002	1000	1300	E-1504	motor 5 KW 380V AC 3 phases	511,30	EUR	June 2018	E-1504	in stock	July 2018	E-1504	in stock
0000001002	1000	1300	E-1505	Motor 5 KW 460V AC 3 phases	511,30	EUR	June 2018	E-1505	in stock	July 2018	E-1505	in stock
0000001002	1000	1300	E-1506	Motor 10 KM 380V AC 3 phases	766,90	EUR	June 2018	E-1506	in stock	July 2018	E-1506	in stock
0000001002	1000	1300	E-1507	Motor 10 KM 460V AC 3 phases	766,90	EUR	June 2018	E-1507	in stock	July 2018	E-1507	in stock
0000001002	1000	1300	E-1508	Motor 20 KM 380V AC 3 phases	1.278,20	EUR	June 2018	E-1508	in stock	July 2018	E-1508	in stock
0000001002	1000	1300	E-1509	Motor 20 KM 460V AC 3 phases	1.278,20	EUR	June 2018	E-1509	in stock	July 2018	E-1509	in stock
0000001002	1000	1300	E-1510	Motor 50 KM 380V AC 3 phases	1.789,48	EUR	June 2018	E-1510	no longer available	July 2018	E-1510	no longer available
0000001002	1000	1300	E-1511	Motor 50 KM 460V AC 3 phases	1.789,48	EUR	June 2018	E-1511	only a few pieces left	July 2018	E-1511	no longer available
0000001002	1000	1300	E-1512	Gearbox 1	766,90	EUR	June 2018	E-1512	in stock	July 2018	E-1512	in stock
0000001002	1000	1300	E-1513	Gearbox 2	1.278,20	EUR	June 2018	E-1513	out of stock - 3 weeks	July 2018	E-1513	in stock
0000001002	1000	1300	E-1514	Gearbox 3	1.789,50	EUR	June 2018	E-1514	in stock	July 2018	E-1514	in stock
0000001002	1000	1300	E-1515	engine unit frame	255,70	EUR	June 2018	E-1515	in stock	July 2018	E-1515	in stock
0000001002	1000	1300	E-1516	Assembly parts	255,70	EUR	June 2018	E-1516	only a few left	July 2018	E-1516	out of stock - 6 weeks
0000001002	1000	1300	E-1517	control unit / 220V 50Hz	766,90	EUR	June 2018	E-1517	no longer available	July 2018	E-1517	no longer available
0000001002	1000	1300	E-1518	Control unit / 110V 50Hz	766,90	EUR	June 2018	E-1518	in stock	July 2018	E-1518	in stock
0000001002	1000	1300	E-1520	Engine unit 10 KW 460V AC	7.157,92	EUR	June 2018	E-1520	in stock	July 2018	E-1520	in stock
0000001002	1000	1300	E-1521	Engine unit 20 KW 460V AC	12.270,72	EUR	June 2018	E-1521	no longer available	July 2018	E-1521	no longer available

(Some columns have been hidden in order to better show the file contents).

If we take a look at the operations log (*Goto* → *Show operations log*) we will see the two UPLOAD operations:

**SE16XXL - operations log**

List of the operations log:

Nr. 16	SELESCREEN	Selection screen	SEL_MODE=0 - TABNAME=\$FEFILE1 - SEL_WITH_OR= - ALIAS=A	Join criteria: A-MATNR 000000 000000 C → MATERIAL C	Select fields: MONTH MATERIAL SITUATION
Nr. 17	UPLOAD	Frontend file - Outer join	SEL_MODE=0 - TABNAME=\$FEFILE1 - SEL_WITH_OR= - ALIAS=C	Join criteria: A-MATNR 000000 000000 C → MATERIAL C	List fields: A-LIFNR A-INFNR B-EKORG B-WERKS A-MATNR V-MATNR_TXT B-NETPR B-WAERS B-NORBM A-MEINS C-MONTH C-MATERIAL C-SITUATION Involved tables: (A)EINA (B)EINE (C)\$FEFILE1
Nr. 18	SELESCREEN	Selection screen	SEL_MODE=0 - TABNAME=\$FEFILE2 - SEL_WITH_OR= - ALIAS=A	Join criteria: A-MATNR 000000 000000 C → MATERIAL C	Select fields: MONTH MATERIAL SITUATION
Nr. 19	UPLOAD	Frontend file - Outer join	SEL_MODE=0 - TABNAME=\$FEFILE2 - SEL_WITH_OR= - ALIAS=B	Join criteria: A-MATNR 000000 000000 C → MATERIAL C	List fields: A-LIFNR A-INFNR B-EKORG B-WERKS A-MATNR V-MATNR_TXT B-NETPR B-WAERS B-NORBM A-MEINS C-MONTH C-MATERIAL C-SITUATION D-MONTH D-MATERIAL D-SITUATION Involved tables: (A)EINA (B)EINE (C)\$FEFILE1 (D)\$FEFILE2

Notice that the frontend files are called \$FEFILE1 and \$FEFILE2.

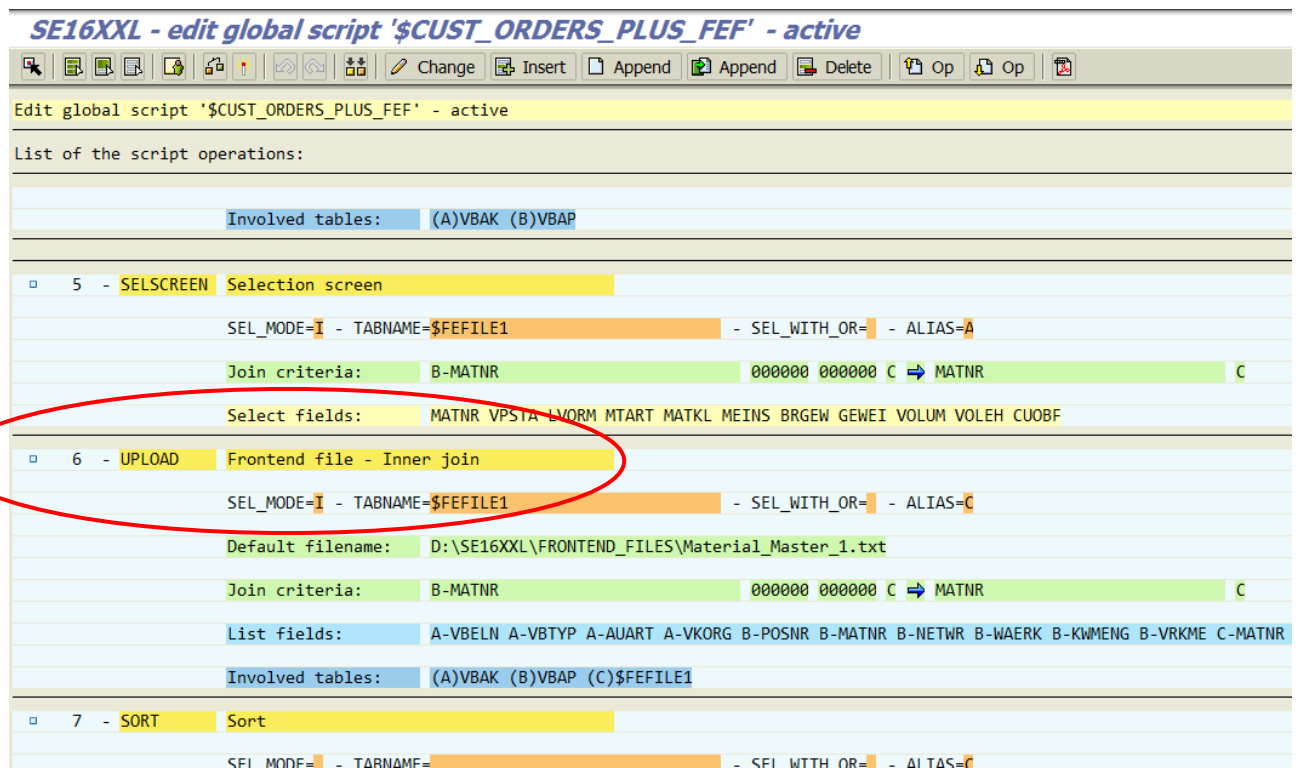
[Back to the top](#)

## UPLOAD Operation can be edited with the Script Editor

Up to now an UPLOAD operation could not be edited. This restriction has now been lifted. This means that UPLOAD operations at the beginning of a script (with pseudo table \$STARTFILE) as well as the ones placed somewhere else in a script (with pseudo table \$FEFILEn) may all be edited.

An example will illustrate the new functionality.

We edit a script called \$CUST\_ORDERS\_PLUS\_FEF which contains an inner join with a frontend file:



**SE16XXL - edit global script '\$CUST\_ORDERS\_PLUS\_FEF' - active**

Edit global script '\$CUST\_ORDERS\_PLUS\_FEF' - active

List of the script operations:

Involved tables: (A)VBAK (B)VBAP

5 - SELSCREEN Selection screen

SEL\_MODE=I - TABNAME=\$FEFILE1 - SEL\_WITH\_OR= - ALIAS=A

Join criteria: B-MATNR 000000 000000 C ⇒ MATNR C

Select fields: MATNR VPSTA LVORM MTART MATKL MEINS BRGEW GEWEI VOLUM VOLEH CUOBF

6 - **UPLOAD Frontend file - Inner join**

SEL\_MODE=I - TABNAME=\$FEFILE1 - SEL\_WITH\_OR= - ALIAS=C

Default filename: D:\SE16XXL\FRONTEND\_FILES\Material\_Master\_1.txt

Join criteria: B-MATNR 000000 000000 C ⇒ MATNR C

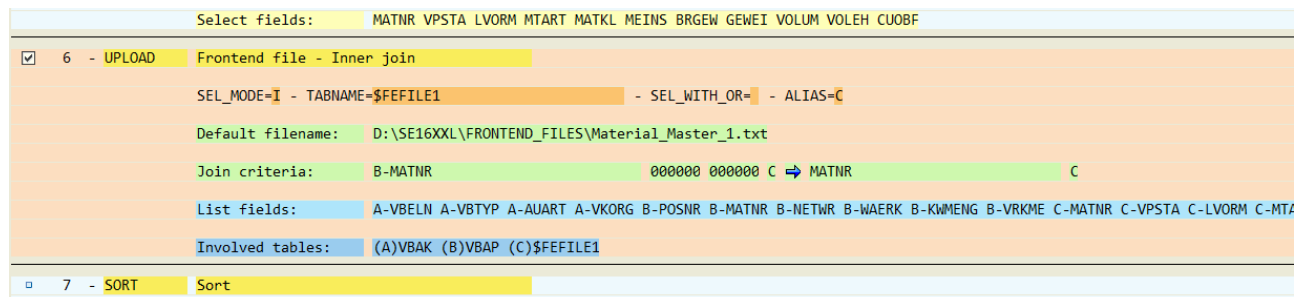
List fields: A-VBELN A-VBTYP A-AUART A-VKORG B-POSNR B-MATNR B-NETWR B-WAERK B-KWMENG B-VRKME C-MATNR

Involved tables: (A)VBAK (B)VBAP (C)\$FEFILE1

7 - SORT Sort

SFI MODF= - TARNAMF= - SFI WITH OR= - ΔI TAS=C

We mark the UPLOAD operation:



Select fields: MATNR VPSTA LVORM MTART MATKL MEINS BRGEW GEWEI VOLUM VOLEH CUOBF

6 - **UPLOAD Frontend file - Inner join**

SEL\_MODE=I - TABNAME=\$FEFILE1 - SEL\_WITH\_OR= - ALIAS=C


Default filename: D:\SE16XXL\FRONTEND\_FILES\Material\_Master\_1.txt

Join criteria: B-MATNR 000000 000000 C ⇒ MATNR C

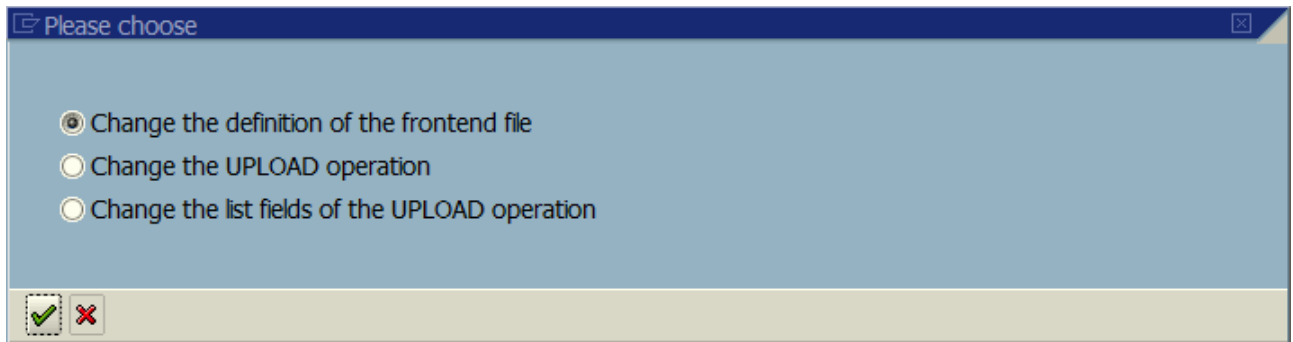
List fields: A-VBELN A-VBTYP A-AUART A-VKORG B-POSNR B-MATNR B-NETWR B-WAERK B-KWMENG B-VRKME C-MATNR C-VPSTA C-LVORM C-MTART

Involved tables: (A)VBAK (B)VBAP (C)\$FEFILE1

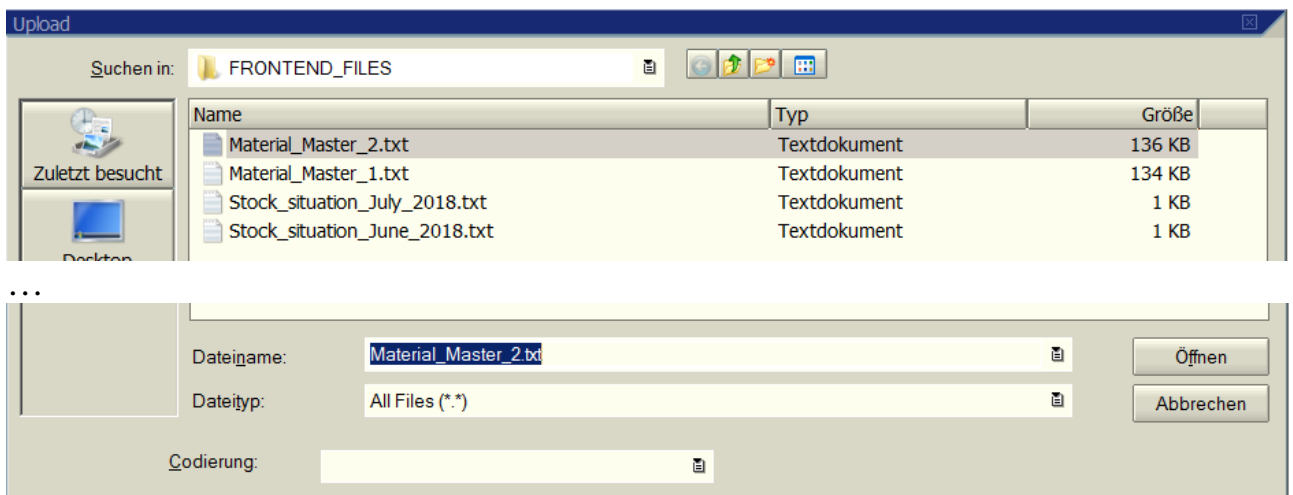
7 - SORT Sort

And then make use of the  Change function on the toolbar.

The following dialog window shows up:



We acknowledge the choice and proceed with . The file selector pops up:



Instead of selecting the original file, we choose another one, which contains more information. The program reacts by displaying the definition of the file columns. Since the file was created with meta data ("**prefix definition of columns to file**") the columns are already assigned to a DDIC table field:

*D: |SE16XXL |FRONTEND\_FILES |Material\_Master\_2.txt*

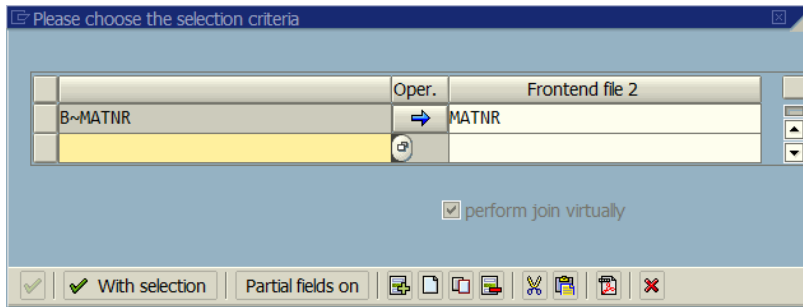
Accept  
  Select header line  
  Import definition

Separator: ht horizontal tabulator  
 First relevant line: 2  
 Date format: YYYYMMDD  
 Decimal point: . Point  
 Language: English

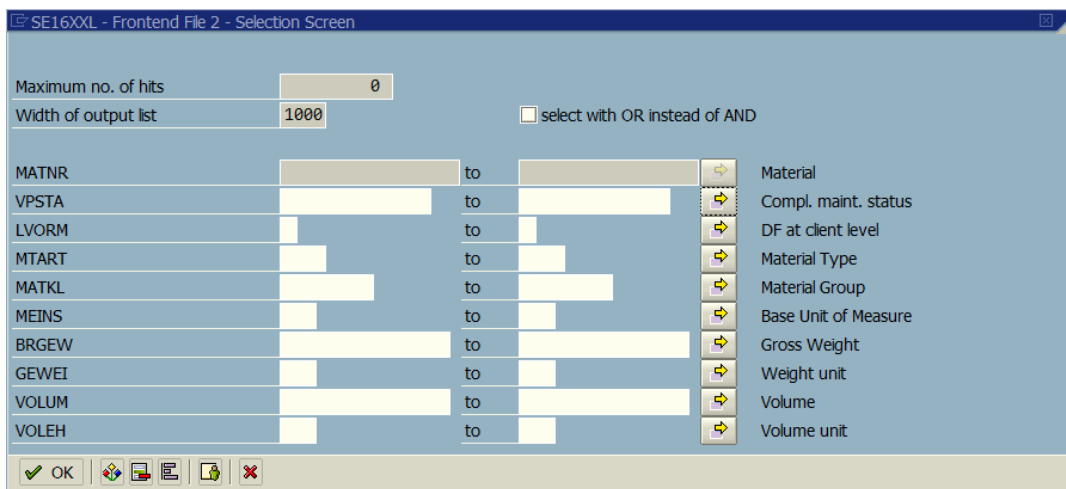
Sel.	Pos.	Name	DDIC	Type	Length	L.Z.	Table name	Field name	Short text	Long text	Curr/Unit
<input checked="" type="checkbox"/>	1	MATNR	<input checked="" type="checkbox"/>	CHAR	18	<input type="checkbox"/>	MARA	MATNR	Material	Material	
<input checked="" type="checkbox"/>	2	ERSDA	<input checked="" type="checkbox"/>	DATS	8	<input type="checkbox"/>	MARA	ERSDA	Created	Created On	
<input checked="" type="checkbox"/>	3	VPSTA	<input checked="" type="checkbox"/>	CHAR	15	<input type="checkbox"/>	MARA	VPSTA	Maint.stat	Compl. maint. status	
<input checked="" type="checkbox"/>	4	LVORM	<input checked="" type="checkbox"/>	CHAR	1	<input type="checkbox"/>	MARA	LVORM	Client lvl	DF at client level	
<input checked="" type="checkbox"/>	5	MTART	<input checked="" type="checkbox"/>	CHAR	4	<input type="checkbox"/>	MARA	MTART	Matl Type	Material Type	
<input checked="" type="checkbox"/>	6	MBRSH	<input checked="" type="checkbox"/>	CHAR	1	<input type="checkbox"/>	MARA	MBRSH	Industry	Industry sector	
<input checked="" type="checkbox"/>	7	MATVL	<input checked="" type="checkbox"/>	CHAR	9	<input type="checkbox"/>	MARA	MATVL	Matl Group	Material Group	

The new file contains more columns than the original one. It would be possible to deactivate or to rename some of them. But in this example we accept the new structure as it is.

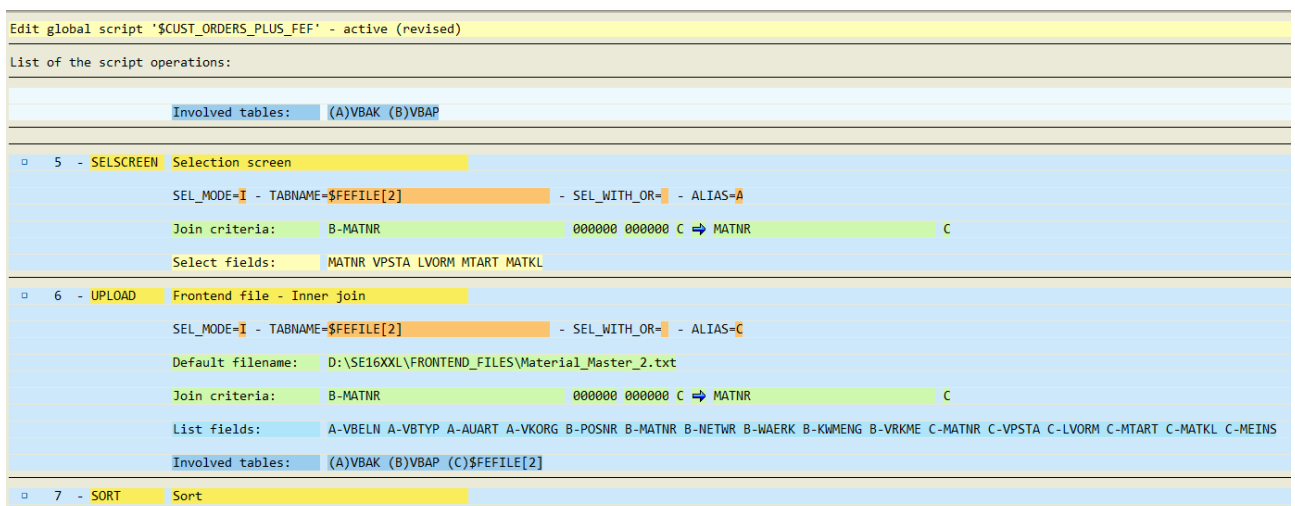
Now the dialog window with the join criteria shows up:




We leave them unchanged and proceed to the selection screen with :




We accept the selection screen as it is and come finally to the conclusion of the change operation:



Notice that the name of the table has changed from **\$FEFILE1** to **\$FEFILE[2]**. It will return again to **\$FEFILE1** after the script has been activated.

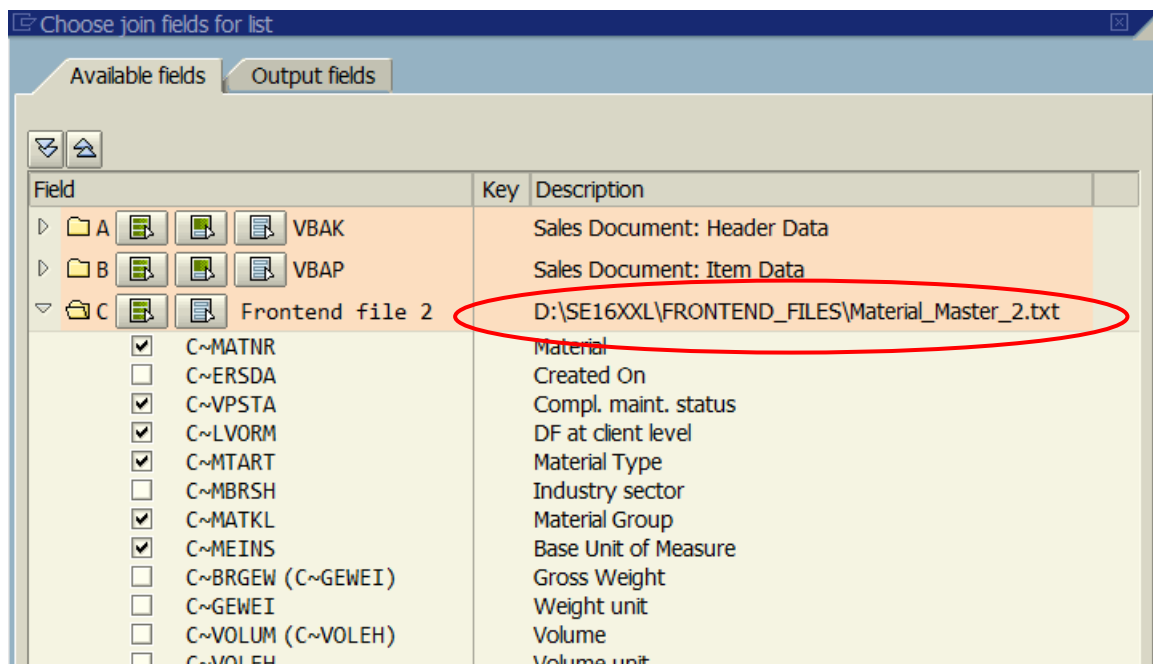
If we now check the consistency of the operations with , the editor will respond with:

 The script operations are consistent

This is due to the fact that the new structure of the frontend file is compatible with the old one, only new columns having been added. If, however, the new structure is not compatible with the old one, or some of the original columns are deactivated or renamed, then the check would most probably detect some inconsistency.

If the file structure is completely changed, all succeeding operations must probably be **adjusted manually**. In such cases it may be more convenient to recreate the script from scratch.

Once the UPLOAD operation has been changed, the new structure of the frontend file is available. Just to give an idea we insert a new CHOOSE\_LF operation directly after the UPLOAD:



Notice that the new file name is displayed.

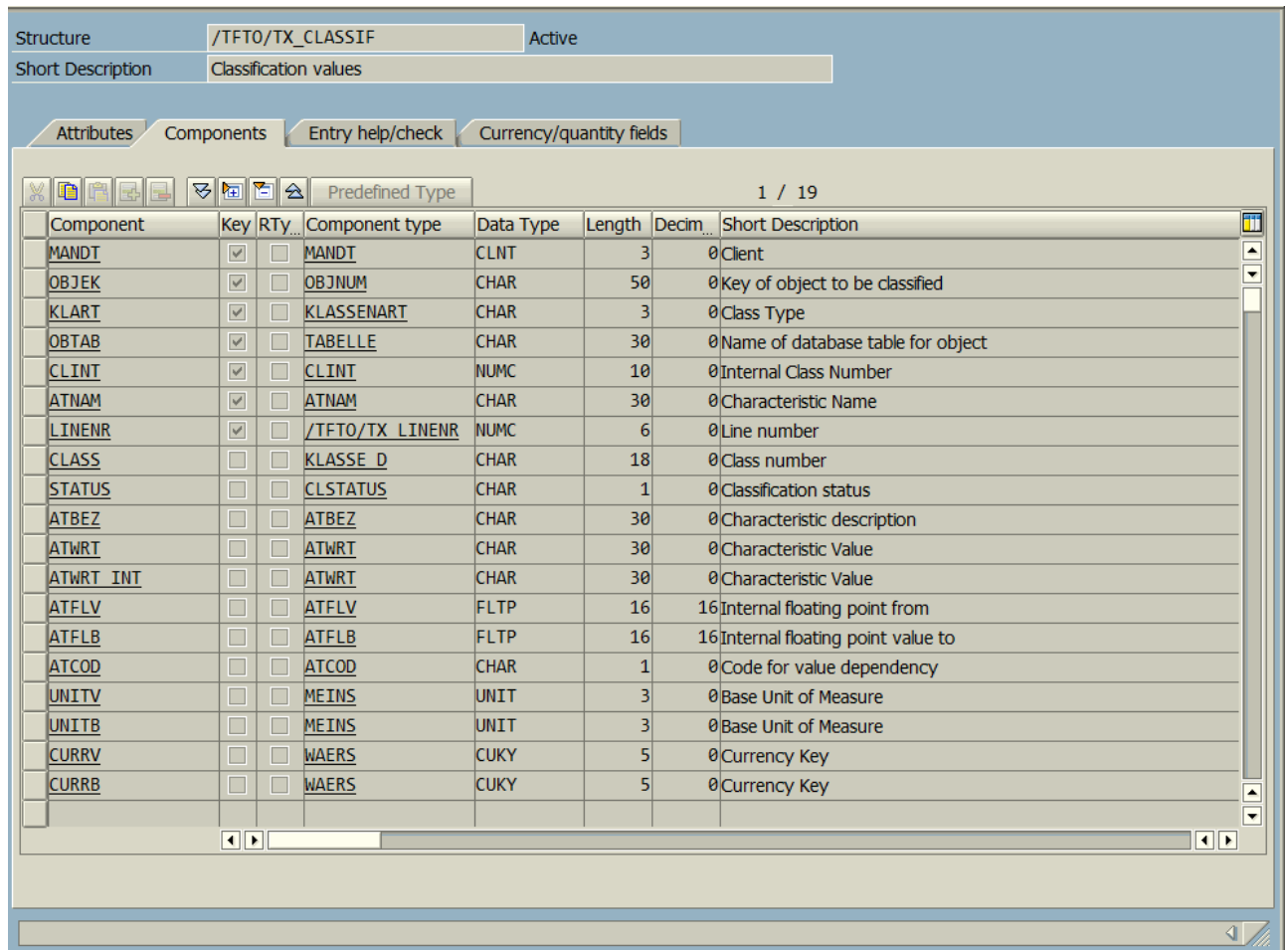
[Back to the top](#)

## New Pseudo Table \$CLASSIF

The SAP classification system is quite complicated. It is based on several database tables (KLAH, SWOR, CABN, CAWN, KSSK, INOB etc.) which interact in a non straightforward way. In order to offer the classification data in a readable form, a new pseudo table (\$CLASSIF) has been implemented. The program starts by selecting internally either table **KSSK**, if the class type is associated with a single object table, or table **INOB**, if more than one object table is involved. Then the following function modules are called:

**BAPI\_OBJCL\_GETCLASSES**  
**BAPI\_OBJCL\_GETDETAIL**

The structure of the result list is as follows:



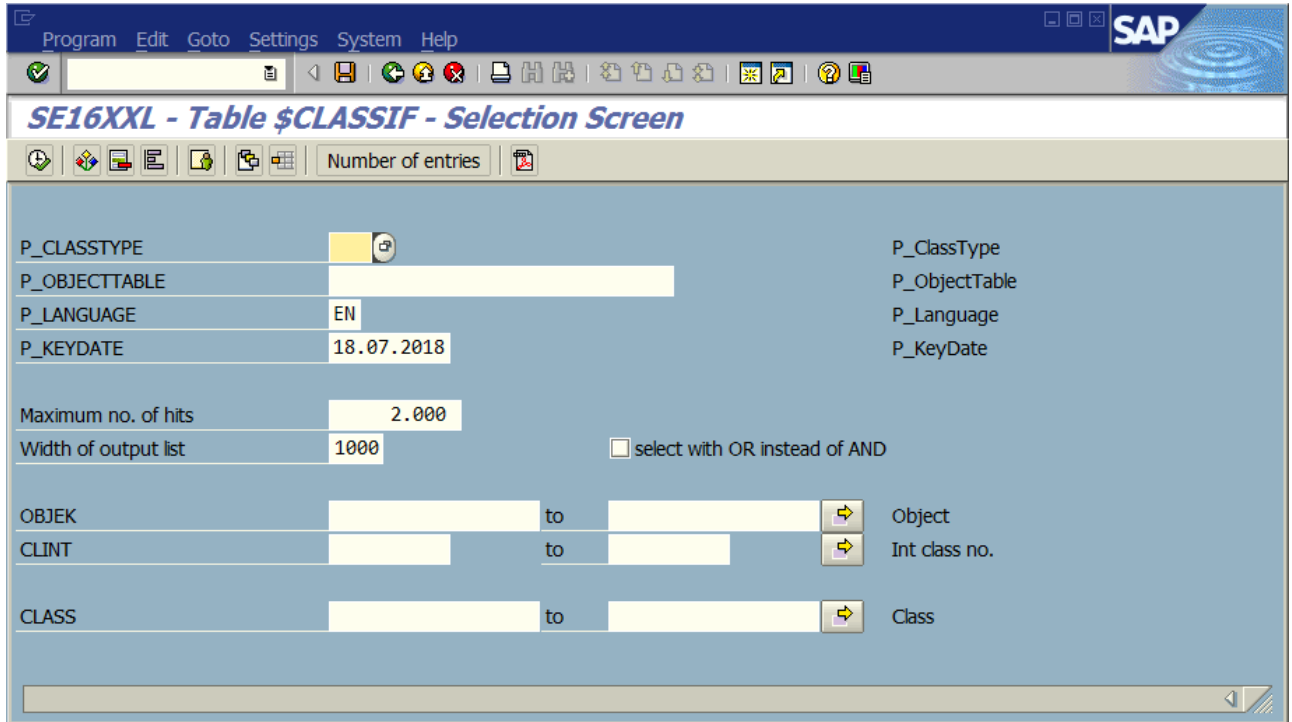
Component	Key	RTY	Component type	Data Type	Length	Decim	Short Description
MANDT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	MANDT	CLNT	3	0	Client
OBJEK	<input checked="" type="checkbox"/>	<input type="checkbox"/>	OBJNUM	CHAR	50	0	Key of object to be classified
KLART	<input checked="" type="checkbox"/>	<input type="checkbox"/>	KLASSENART	CHAR	3	0	Class Type
OBTAB	<input checked="" type="checkbox"/>	<input type="checkbox"/>	TABELLE	CHAR	30	0	Name of database table for object
CLINT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	CLINT	NUMC	10	0	Internal Class Number
ATNAM	<input checked="" type="checkbox"/>	<input type="checkbox"/>	ATNAM	CHAR	30	0	Characteristic Name
LINENR	<input checked="" type="checkbox"/>	<input type="checkbox"/>	/TFTO/TX LINENR	NUMC	6	0	Line number
CLASS	<input type="checkbox"/>	<input type="checkbox"/>	KLASSE D	CHAR	18	0	Class number
STATUS	<input type="checkbox"/>	<input type="checkbox"/>	CLSTATUS	CHAR	1	0	Classification status
ATBEZ	<input type="checkbox"/>	<input type="checkbox"/>	ATBEZ	CHAR	30	0	Characteristic description
ATWRT	<input type="checkbox"/>	<input type="checkbox"/>	ATWRT	CHAR	30	0	Characteristic Value
ATWRT INT	<input type="checkbox"/>	<input type="checkbox"/>	ATWRT	CHAR	30	0	Characteristic Value
ATFLV	<input type="checkbox"/>	<input type="checkbox"/>	ATFLV	FLTP	16	16	Internal floating point from
ATFLB	<input type="checkbox"/>	<input type="checkbox"/>	ATFLB	FLTP	16	16	Internal floating point value to
ATCOD	<input type="checkbox"/>	<input type="checkbox"/>	ATCOD	CHAR	1	0	Code for value dependency
UNITV	<input type="checkbox"/>	<input type="checkbox"/>	MEINS	UNIT	3	0	Base Unit of Measure
UNITB	<input type="checkbox"/>	<input type="checkbox"/>	MEINS	UNIT	3	0	Base Unit of Measure
CURRV	<input type="checkbox"/>	<input type="checkbox"/>	WAERS	CUKY	5	0	Currency Key
CURRB	<input type="checkbox"/>	<input type="checkbox"/>	WAERS	CUKY	5	0	Currency Key

Each row of the result list contains a class plus a characteristic with value.

The description of the class may be added by calling the function "Add text columns" for field **CLINT**.

## Selection Screen of \$CLASSIF

The selection screen is as follows (with all possible criteria):



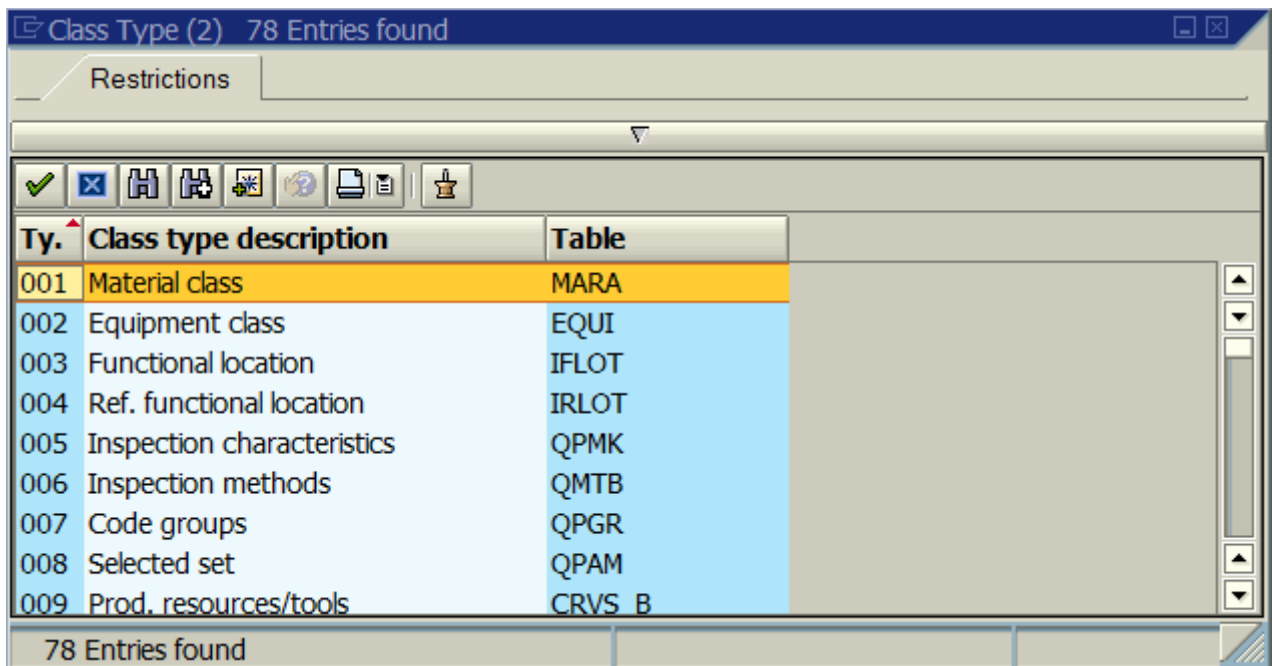
SE16XXL - Table \$CLASSIF - Selection Screen

Number of entries

P_CLASSTYPE		P_ClassType		
P_OBJECTTABLE		P_ObjectTable		
P_LANGUAGE	EN	P_Language		
P_KEYDATE	18.07.2018	P_KeyDate		
Maximum no. of hits	2.000			
Width of output list	1000	<input type="checkbox"/> select with OR instead of AND		
OBJEK		to		Object
CLINT		to		Int class no.
CLASS		to		Class

The parameters (in imitation of a CDS view) must be specified (at least the first two).

Please make use of the F4 help to choose the P\_CLASSTYPE value:



Class Type (2) 78 Entries found

Restrictions

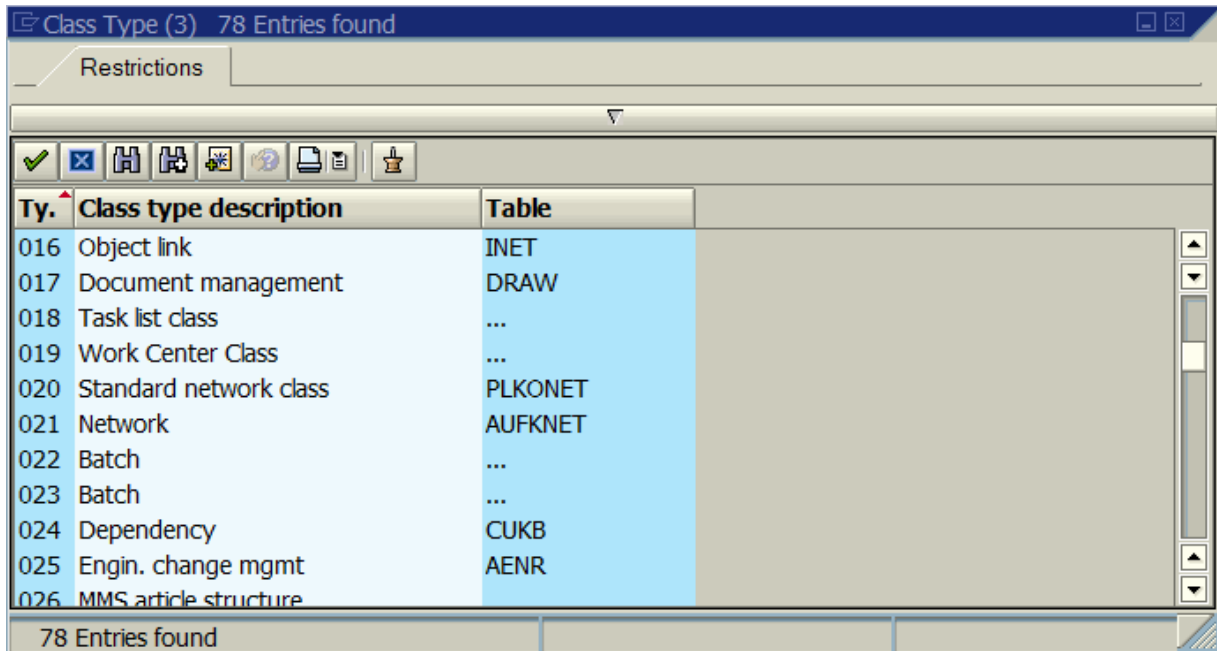
Ty.	Class type description	Table
001	Material class	MARA
002	Equipment class	EQUI
003	Functional location	IFLOT
004	Ref. functional location	IRLOT
005	Inspection characteristics	QPMK
006	Inspection methods	QMTB
007	Code groups	QPGR
008	Selected set	QPAM
009	Prod. resources/tools	CRVS B

78 Entries found



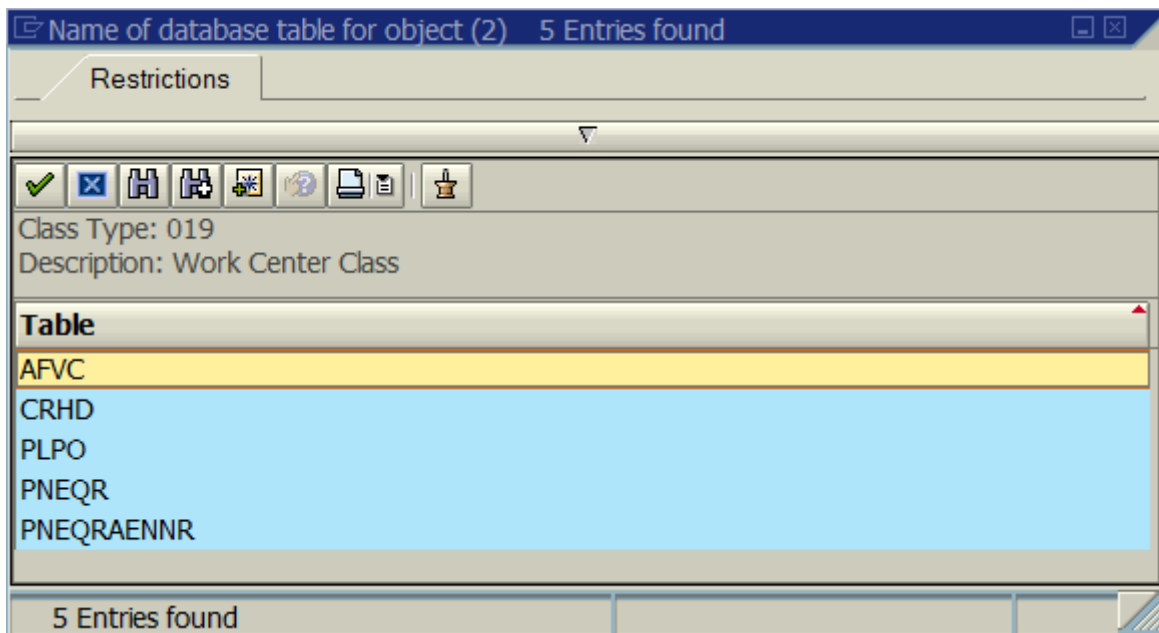
If the class type is associated with a single object table, P\_OBJECTTABLE will be filled in automatically.

Some class types are associated with **more than one** object table (...):



Ty.	Class type description	Table
016	Object link	INET
017	Document management	DRAW
018	Task list class	...
019	Work Center Class	...
020	Standard network class	PLKONET
021	Network	AUFKNET
022	Batch	...
023	Batch	...
024	Dependency	CUKB
025	Engin. change mgmt	AENR
026	MMS article structure	

If for example you choose **class type 019** P\_OBJECTTABLE will be filled with "..."  
and you must make use of the F4 help to select the desired object table:



Class Type: 019  
Description: Work Center Class

Table
AFVC
CRHD
PLPO
PNEQR
PNEQRAENR

The last two parameters (P\_LANGUAGE and P\_KEYDATE) can be left empty, in which case the **logon language** and the **current date** are implicitly used.

## Result list of \$CLASSIF

A typical result list could be as follows:

MANDT	OBJEK	KLART	OBTAB	CLINT	ATNAM	LINENR	CLASS	STATUS	ATBEZ	ATWRT	ATWRT_INT	
800	0001MIXER-01	019	CRHD	1210	MIXER_MATERIAL	1	MIXER	1	Material mixer is made out of	Stainless Steel	STSTL	0,00000000
800	0001MIXER-01	019	CRHD	1210	MIXER_SPEED	1	MIXER	1	Speed of Mixer	0,0 - 2.000,0	0,0 - 2.000,0	0,00000000
800	0001OVEN-01	019	CRHD	1211	OVEN_TEMP	1	OVEN	1	Oven temperature	0,00 - 300,00 °C	0,00 - 300,00 °C	0,00000000
800	0099TP99	019	CRHD	261	IDES_003	1	IDES	1	Internal IDES: Course no.	Basic data discrete production	L0205	0,00000000
800	0099TP99	019	CRHD	261	IDES_003	2	IDES	1	Internal IDES: Course no.	Make to order production	L0215	0,00000000
800	0099TP99	019	CRHD	261	IDES_003	3	IDES	1	Internal IDES: Course no.	Capacity planning/scheduling	L0230	0,00000000
800	10001110	019	CRHD	267	CH1000	1	CL0001	1	Work procedure	Milling	01	0,00000000
800	10001110	019	CRHD	267	CH1001	1	CL0001	1	Type of processing	by hand	01	0,00000000
800	10001110	019	CRHD	267	CH1002	1	CL0001	1	Range drilling/turning			0,00000000
800	10001110	019	CRHD	267	CH1003	1	CL0001	1	Coordinates X, presses			0,00000000
800	10001110	019	CRHD	267	CH1004	1	CL0001	1	Coordinates Y, presses			0,00000000
800	10001111	019	CRHD	261	IDES_003	1	IDES	1	Internal IDES: Course no.	Make to order production	L0215	0,00000000
800	10001111	019	CRHD	267	CH1000	1	CL0001	1	Work procedure	Rotate	03	0,00000000
800	10001111	019	CRHD	267	CH1001	1	CL0001	1	Type of processing	numeric (NC)	02	0,00000000
800	10001111	019	CRHD	267	CH1002	1	CL0001	1	Range drilling/turning			0,00000000
800	10001111	019	CRHD	267	CH1003	1	CL0001	1	Coordinates X, presses			0,00000000

## Special Join Criteria for \$CLASSIF

In order to be able to perform a join with the same entries using a different language, all fields of \$CLASSIF can be used as join criteria:

\$CLASSIF	Oper.	\$CLASSIF
OBJEK	→	OBJEK
KLART	→	KLART
OBTAB	→	OBTAB
CLINT	→	CLINT
ATNAM	→	ATNAM
LINENR	→	LINENR

perform join virtually

With selection Partial fields on

**NOTE:** This functionality is available for all pseudo tables.

[Back to the top](#)

## New Pseudo Table \$CONFIG

Configurations are a very complicated part of SAP. The data involved in a configuration are distributed among several database tables. In addition a lot of customizing and special processing is involved. Up to now it was very difficult in SE16XXL to select and display in a reasonable way the data associated with a configuration.

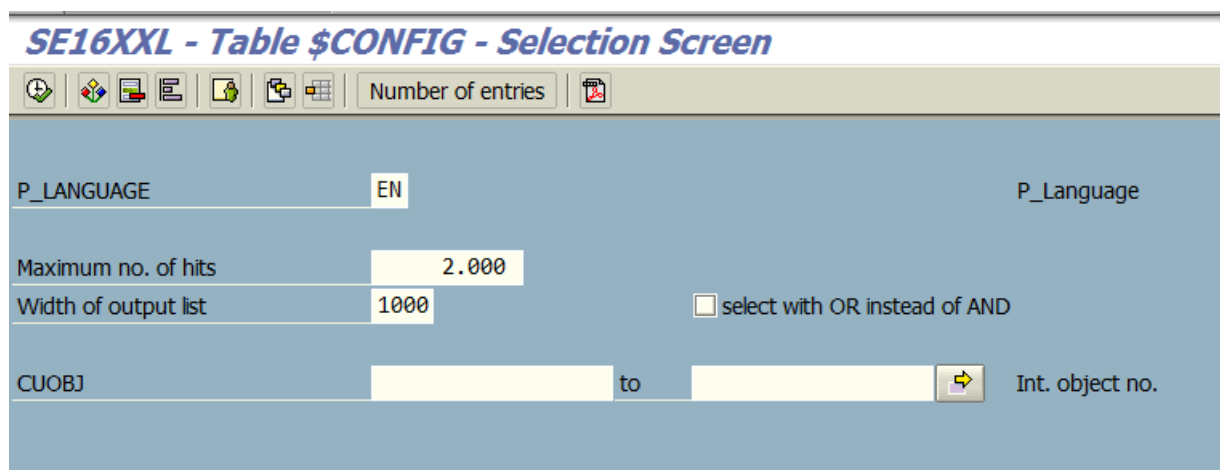
For this reason a new pseudo table **\$CONFIG** has been implemented. This feature makes use internally of several SAP methods and functions. Among others:

Method **CL\_CBASE=>GET\_CURRENT\_CBASE\_BY\_INSTANCE**  
 Function module **VC\_I\_GET\_CONFIGURATION\_IBASE**

Since some of these SAP functions contain a lot of MESSAGE X statements, which could possibly precipitate the whole session into an abnormal end, the main function is called via RFC with destination "NONE".

### Selection Screen of \$CONFIG

The selection screen of this pseudo table is quite unspectacular:



The screenshot shows the selection screen for the pseudo table \$CONFIG. The title is "SE16XXL - Table \$CONFIG - Selection Screen". The screen contains several input fields and a checkbox:

- P\_LANGUAGE**: EN
- Maximum no. of hits**: 2.000
- Width of output list**: 1000
- select with OR instead of AND
- CUOBJ**: [ ] to [ ] Int. object no.

**P\_LANGUAGE** is a parameter (in imitation of CDS views).

**CUOBJ** is the internal identifier of a configuration.

**NOTE:** It does not make much sense to start a selection with \$CONFIG, because the main object of the configuration, for example a material, would not show up in the results. A better idea is to start with a main table, for example MARA, and then join the configuration data to the master records.

## Structure of \$CONFIG

The structure of the entries of this pseudo table is as follows:

**Dictionary: Display Structure**

Structure: /TFTO/TX\_CONFIG Active  
 Short Description: Configuration values

Attributes Components Entry help/check Currency/quantity fields

Predefined Type 1 / 9

Component	Key	RTY...	Component type	Data Type	Length	Decim...	Short Description
MANDT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	MANDT	CLNT	3	0	Client
CUOBJ	<input checked="" type="checkbox"/>	<input type="checkbox"/>	CUOBJ	NUMC	18	0	Configuration (internal object number)
ATINN	<input checked="" type="checkbox"/>	<input type="checkbox"/>	ATINN	NUMC	10	0	Internal characteristic
LINENR	<input checked="" type="checkbox"/>	<input type="checkbox"/>	/TFTO/TX LINENR	NUMC	6	0	Line number
ATNAM	<input type="checkbox"/>	<input type="checkbox"/>	ATNAM	CHAR	30	0	Characteristic Name
ATBEZ	<input type="checkbox"/>	<input type="checkbox"/>	ATBEZ	CHAR	30	0	Characteristic description
ATWRT	<input type="checkbox"/>	<input type="checkbox"/>	ATWRT	CHAR	30	0	Characteristic Value
ATWTB	<input type="checkbox"/>	<input type="checkbox"/>	ATWTB	CHAR	30	0	Characteristic value description
ATAUT	<input type="checkbox"/>	<input type="checkbox"/>	ATAUT	CHAR	1	0	Classification: Author

## Result List with \$CONFIG

A typical result list could have the following appearance (MARA plus \$CONFIG):

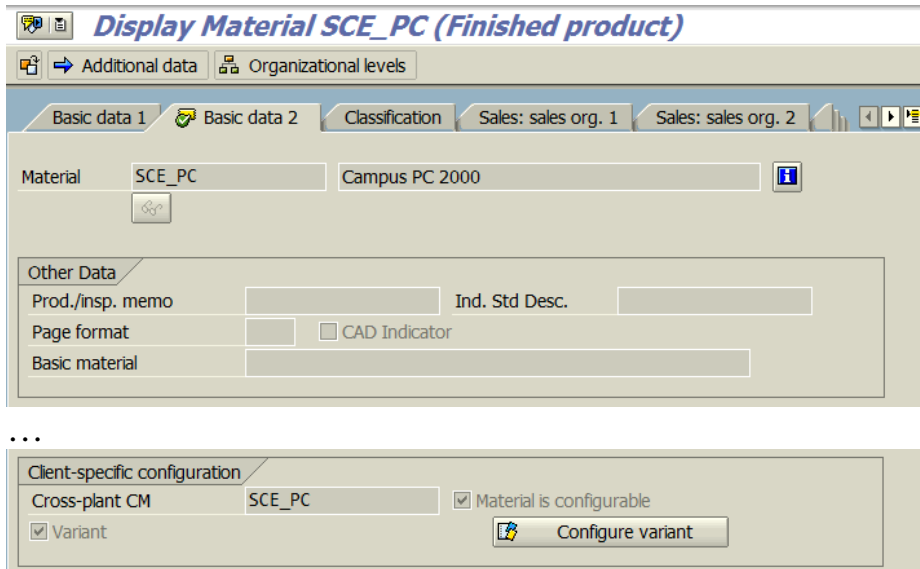
**SE16XXL - inner join - 104 resulting rows**

Join of MARA(A) and \$CONFIG(B)

A~MATNR	A~MTART	A~MEINS	A~CUOBF	B~ATINN	B~LINENR	B~ATNAM	B~ATBEZ	B~ATWRT	B~ATWTB	B~ATAUT
<input type="checkbox"/> SCE_PC	FERT	ST	3209	0000001950	1	SCE_PC_PROD	Processor	SCE_400MH_W0	400 MHz Pentium II w/MS Work	8
<input type="checkbox"/> SCE_PC	FERT	ST	3209	0000001951	1	SCE_MEM	Memory	SCE_64MB	64 MB SDRAM	8
<input type="checkbox"/> SCE_PC	FERT	ST	3209	0000001952	1	SCE_HD	Hard Disk	SCE_HD_064	6.4 GB Ultra DMA Hard Drive	8
<input type="checkbox"/> SCE_PC	FERT	ST	3209	0000001953	1	SCE_CD_OPT	CD Options	SCE_CD_32	32X Speed CD ROM Drive	8
<input type="checkbox"/> SCE_PC	FERT	ST	3209	0000001954	1	SCE_RSTOR_0	Removable Storage	SCE_ZIP_100_FDD	100 MB Zip Drive & 1.44 FDD	8
<input type="checkbox"/> SCE_PC	FERT	ST	3209	0000001955	1	SCE_CD_WRITE4	4X HP CDWriter 811 Rewritable	YES	4x HP+ 8110 Rewritable Drive	8
<input type="checkbox"/> SCE_PC	FERT	ST	3209	0000001956	1	SCE_GRAPH_16	16 MB Dia Viper V550 Nvidi RI	YES	V550 Nvidia RIVA TNT w/TV-out	8
<input type="checkbox"/> SCE_PC	FERT	ST	3209	0000001957	1	SCE_MODEM_56	56 kbps K56Flex V.90 Data/Fax	YES	Data/Fax Modem	8
<input type="checkbox"/> SCE_PC	FERT	ST	3209	0000001958	1	SCE_ETHERNETC	10/100Base-T Ethernet	YES	3Com(r) Ethernet Network Card	8
<input type="checkbox"/> SCE_PC	FERT	ST	3209	0000001959	1	SCE_KEYB	Keyboard	SCE_KEYB_SCR_M	One-Touch KB & Scrolling Mouse	8
<input type="checkbox"/> SCE_PC	FERT	ST	3209	0000001960	1	SCE_SOUND	Sound card	SCE_SOUND_BL	Creative Sound Blaster AudioPC	8
<input type="checkbox"/> SCE_PC	FERT	ST	3209	0000001962	1	SCE_SP	Speaker	SCE_SP_STEREO	Polk Audio Stereo Speakers	8
<input type="checkbox"/> SCE_PC	FERT	ST	3209	0000001963	1	SCE_MON	Monitor	SCE_MON15	15 Inch 550 Monitor	8
<input type="checkbox"/> SCE_PC	FERT	ST	3209	0000001964	1	SCE_SOFTWARE	Software	SCE_MSWORK	MS Works	8
<input type="checkbox"/> SCE_PC	FERT	ST	3209	0000001965	1	SCE_DIGCAM_PORT	Digital Camera Port	YES	Digital Port	8
<input type="checkbox"/> SCE_PC	FERT	ST	3209	0000001966	1	SCE_WARRANTY	Warranty for PC	INCLUDED	INCLUDED	8
<input type="checkbox"/> T-PPE-MV1	FERT	ST	9638	0000003651	1	T_PPE_COL	Colors for iPPE Car	R	red	
<input type="checkbox"/> T-PPE-MV1	FERT	ST	9638	0000003652	1	T_PPE_CHA	Chassis Styles	STD	standard	
<input type="checkbox"/> T-PPE-MV1	FERT	ST	9638	0000003654	1	T_PPE_WHE	Wheel Types	STL	Steel	
<input type="checkbox"/> T-PPE-MV1	FERT	ST	9638	0000003656	1	T_PPE_ENG	Engine Types	150	150 HP / 110 kW	
<input type="checkbox"/> T-PPE-MV2	FERT	ST	9640	0000003651	1	T_PPE_COL	Colors for iPPE Car	B	blue	
<input type="checkbox"/> T-PPE-MV2	FERT	ST	9640	0000003652	1	T_PPE_CHA	Chassis Styles	SPC	special	
<input type="checkbox"/> T-PPE-MV2	FERT	ST	9640	0000003654	1	T_PPE_WHE	Wheel Types	ALU	Aluminum	
<input type="checkbox"/> T-PPE-MV2	FERT	ST	9640	0000003656	1	T_PPE_ENG	Engine Types	200	200 HP / 147 kW	

## Transaction View of the Data

Just to give an idea of what has been discussed, we will take a look at the configuration of material **SCE\_PC** as seen using standard transaction **MM03**:



**Display Material SCE\_PC (Finished product)**

Additional data | Organizational levels

Basic data 1 | Basic data 2 | Classification | Sales: sales org. 1 | Sales: sales org. 2

Material: SCE\_PC | Campus PC 2000

Other Data

Prod./insp. memo: | Ind. Std Desc.: |

Page format: |  CAD Indicator

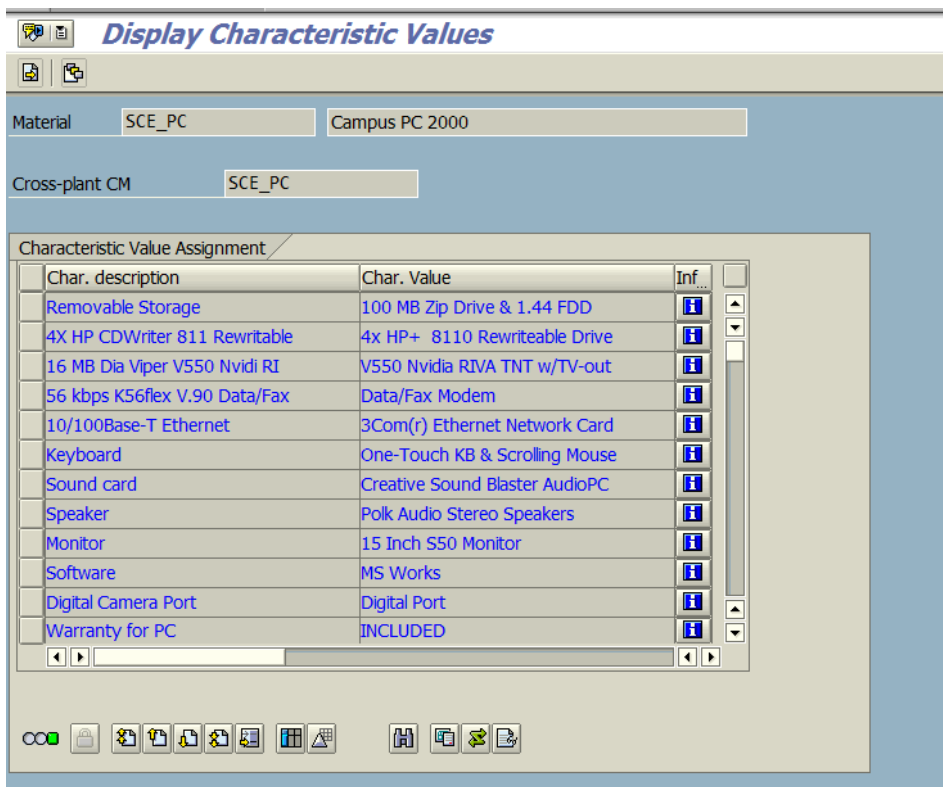
Basic material: |

Client-specific configuration

Cross-plant CM: SCE\_PC  Material is configurable

Variant

When  is pressed the details of the configuration show up:



**Display Characteristic Values**

Material: SCE\_PC | Campus PC 2000

Cross-plant CM: SCE\_PC

Char. description	Char. Value	Inf...
Removable Storage	100 MB Zip Drive & 1.44 FDD	<input type="button" value="i"/>
4X HP CDWriter 811 Rewritable	4x HP+ 8110 Rewriteable Drive	<input type="button" value="i"/>
16 MB Dia Viper V550 Nvidi RI	V550 Nvidia RIVA TNT w/TV-out	<input type="button" value="i"/>
56 kbps K56flex V.90 Data/Fax	Data/Fax Modem	<input type="button" value="i"/>
10/100Base-T Ethernet	3Com(r) Ethernet Network Card	<input type="button" value="i"/>
Keyboard	One-Touch KB & Scrolling Mouse	<input type="button" value="i"/>
Sound card	Creative Sound Blaster AudioPC	<input type="button" value="i"/>
Speaker	Polk Audio Stereo Speakers	<input type="button" value="i"/>
Monitor	15 Inch S50 Monitor	<input type="button" value="i"/>
Software	MS Works	<input type="button" value="i"/>
Digital Camera Port	Digital Port	<input type="button" value="i"/>
Warranty for PC	INCLUDED	<input type="button" value="i"/>

[Back to the top](#)

## New Pseudo Table \$E071

Database table **E071** contains the object entries of a transport request / task. This is essentially what you see when you display the contents of a transport request / task in transaction SE10:

*Display request/task ZE5K901261*

Request/Task: ZE5K901261      Development/Correction

Properties    Objects    Documentation

1 / 16

Short Description	Pro...	Obj	Object Name	Fun...	Lock/Import Status	L...	IMG Activity
GUI Definition	LIMU	CUAD	/TFTO/SAPLTX_DISP		Object Locked		
Documentation	LIMU	DOCU	NA/TFTO/TX2062		Object Locked		
Function Module	LIMU	FUNC	/TFTO/TX_COMPARE_JOIN_RECS_INT		Object Locked		
Function Module	LIMU	FUNC	/TFTO/TX_COMPARE_RECORDS_INT		Object Locked		
Function Module	LIMU	FUNC	/TFTO/TX_DISPLAY_RECORD		Object Locked		
Function Module	LIMU	FUNC	/TFTO/TX_DISPLAY_RECORD_INT		Object Locked		
Function Module	LIMU	FUNC	/TFTO/TX_FORMULA_EDITOR		Object Locked		
Single Message	LIMU	MESS	/TFTO/TX2062		Object Locked		
Report Source Code	LIMU	REPS	/TFTO/LTX_DISPF01		Object Locked		
Report Source Code	LIMU	REPS	/TFTO/LTX_DISPTOP		Object Locked		
Report Source Code	LIMU	REPS	/TFTO/LTX_DISP_JOINF01		Object Locked		
Report Source Code	LIMU	REPS	/TFTO/LTX_FORMULATMA		Object Locked		
Report Source Code	LIMU	REPS	/TFTO/TX_SEL_AND_I_FORMS		Object Locked		
Report Source Code	LIMU	REPS	/TFTO/TX_SEL_AND_I_FORMS2		Object Locked		
Report Source Code	LIMU	REPS	/TFTO/TX_SEL_AND_SHOW_RECORDS		Object Locked		
Table Type	R3TR	TTYP	/TFTO/TX_IT_TABIX		Object Locked		

The same information displayed as SE16XXL result list with table E071:

*SE16XXL - Table E071 - 16 entries selected*

Table E071 - Change & Transport System: Object Entries of Requests/Tasks

TRKORR	AS4POS	PGMID	OBJECT	OBJ_NAME	OBJFUNC	LOCKFLAG	GENNUM	LANG	ACTIVITY
<input type="checkbox"/>	ZE5K901261	1	LIMU	CUAD	/TFTO/SAPLTX_DISP		X		
<input type="checkbox"/>	ZE5K901261	2	LIMU	DOCU	NA/TFTO/TX2062		X		
<input type="checkbox"/>	ZE5K901261	3	LIMU	FUNC	/TFTO/TX_COMPARE_JOIN_RECS_INT		X		
<input type="checkbox"/>	ZE5K901261	4	LIMU	FUNC	/TFTO/TX_COMPARE_RECORDS_INT		X		
<input type="checkbox"/>	ZE5K901261	5	LIMU	FUNC	/TFTO/TX_DISPLAY_RECORD		X		
<input type="checkbox"/>	ZE5K901261	6	LIMU	FUNC	/TFTO/TX_DISPLAY_RECORD_INT		X		
<input type="checkbox"/>	ZE5K901261	7	LIMU	FUNC	/TFTO/TX_FORMULA_EDITOR		X		
<input type="checkbox"/>	ZE5K901261	8	LIMU	MESS	/TFTO/TX2062		X		
<input type="checkbox"/>	ZE5K901261	9	LIMU	REPS	/TFTO/LTX_DISPF01		X		
<input type="checkbox"/>	ZE5K901261	10	LIMU	REPS	/TFTO/LTX_DISPTOP		X		
<input type="checkbox"/>	ZE5K901261	11	LIMU	REPS	/TFTO/LTX_DISP_JOINF01		X		
<input type="checkbox"/>	ZE5K901261	12	LIMU	REPS	/TFTO/LTX_FORMULATMA		X		
<input type="checkbox"/>	ZE5K901261	13	LIMU	REPS	/TFTO/TX_SEL_AND_I_FORMS		X		
<input type="checkbox"/>	ZE5K901261	14	LIMU	REPS	/TFTO/TX_SEL_AND_I_FORMS2		X		
<input type="checkbox"/>	ZE5K901261	15	LIMU	REPS	/TFTO/TX_SEL_AND_SHOW_RECORDS		X		
<input type="checkbox"/>	ZE5K901261	16	R3TR	TTYP	/TFTO/TX_IT_TABIX		X		



Each transport object is characterized by three values: **PGMID**, **OBJECT** and **OBJ\_NAME**.

PGMID has two main values:

**R3TR** denotes a complete repository object  
**LIMU** denotes a subobject of a repository object

An R3TR object may have several LIMU subobjects. If a transport request contains a main object and some of its subobjects, the latter will disappear if the function "*Request/Task → Object List → Aggregate*" is performed in SE10.

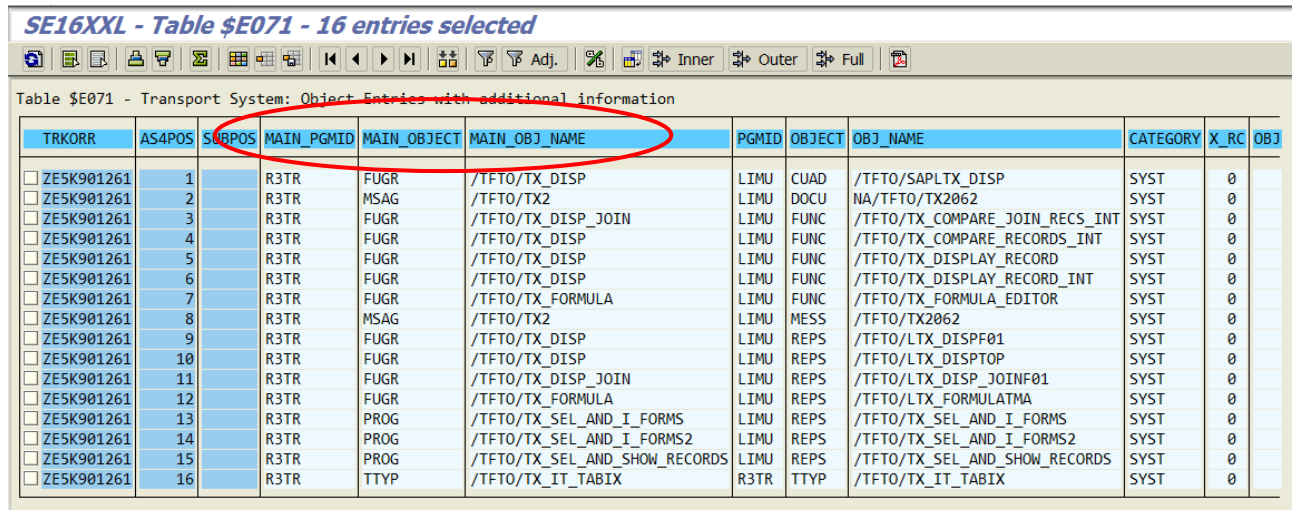
The implicit assignment of a subobject to its main object is not always evident. The subobject **LIMU REPS** has the same **OBJ\_NAME** as its object **R3TR PROG**. But **LIMU FUNC** (function module) has no visible connection to its main object **R3TR FUGR** (function group). The function group has to be looked for in transaction SE37 under "Attributes". Other (sub)objects present similar difficulties.

In order to bridge this gap a new pseudo table **SE071** has been implemented. Its structure contains all the fields of E071 plus the three fields of the main object:

**MAIN\_PGMID**  
**MAIN\_OBJECT**  
**MAIN\_OBJ\_NAME**

The above contents of transport ZE5K901261 displayed with **SE071** are as follows:

*SE16XXL - Table SE071 - 16 entries selected*



TRKORR	AS4POS	SUBPOS	MAIN_PGMID	MAIN_OBJECT	MAIN_OBJ_NAME	PGMID	OBJECT	OBJ_NAME	CATEGORY	X_RC	OBJ
<input type="checkbox"/>	ZE5K901261	1	R3TR	FUGR	/TFTO/TX_DISP	LIMU	CUAD	/TFTO/SAPLTX_DISP	SYST	0	
<input type="checkbox"/>	ZE5K901261	2	R3TR	MSAG	/TFTO/TX2	LIMU	DOCU	NA/TFTO/TX2062	SYST	0	
<input type="checkbox"/>	ZE5K901261	3	R3TR	FUGR	/TFTO/TX_DISP_JOIN	LIMU	FUNC	/TFTO/TX_COMPARE_JOIN_RECS_INT	SYST	0	
<input type="checkbox"/>	ZE5K901261	4	R3TR	FUGR	/TFTO/TX_DISP	LIMU	FUNC	/TFTO/TX_COMPARE_RECORDS_INT	SYST	0	
<input type="checkbox"/>	ZE5K901261	5	R3TR	FUGR	/TFTO/TX_DISP	LIMU	FUNC	/TFTO/TX_DISPLAY_RECORD	SYST	0	
<input type="checkbox"/>	ZE5K901261	6	R3TR	FUGR	/TFTO/TX_DISP	LIMU	FUNC	/TFTO/TX_DISPLAY_RECORD_INT	SYST	0	
<input type="checkbox"/>	ZE5K901261	7	R3TR	FUGR	/TFTO/TX_FORMULA	LIMU	FUNC	/TFTO/TX_FORMULA_EDITOR	SYST	0	
<input type="checkbox"/>	ZE5K901261	8	R3TR	MSAG	/TFTO/TX2	LIMU	MESS	/TFTO/TX2062	SYST	0	
<input type="checkbox"/>	ZE5K901261	9	R3TR	FUGR	/TFTO/TX_DISP	LIMU	REPS	/TFTO/LTX_DISPF01	SYST	0	
<input type="checkbox"/>	ZE5K901261	10	R3TR	FUGR	/TFTO/TX_DISP	LIMU	REPS	/TFTO/LTX_DISPTOP	SYST	0	
<input type="checkbox"/>	ZE5K901261	11	R3TR	FUGR	/TFTO/TX_DISP_JOIN	LIMU	REPS	/TFTO/LTX_DISP_JOINF01	SYST	0	
<input type="checkbox"/>	ZE5K901261	12	R3TR	FUGR	/TFTO/TX_FORMULA	LIMU	REPS	/TFTO/LTX_FORMULATMA	SYST	0	
<input type="checkbox"/>	ZE5K901261	13	R3TR	PROG	/TFTO/TX_SEL_AND_I_FORMS	LIMU	REPS	/TFTO/TX_SEL_AND_I_FORMS	SYST	0	
<input type="checkbox"/>	ZE5K901261	14	R3TR	PROG	/TFTO/TX_SEL_AND_I_FORMS2	LIMU	REPS	/TFTO/TX_SEL_AND_I_FORMS2	SYST	0	
<input type="checkbox"/>	ZE5K901261	15	R3TR	PROG	/TFTO/TX_SEL_AND_SHOW_RECORDS	LIMU	REPS	/TFTO/TX_SEL_AND_SHOW_RECORDS	SYST	0	
<input type="checkbox"/>	ZE5K901261	16	R3TR	TTYP	/TFTO/TX_IT_TABIX	R3TR	TTYP	/TFTO/TX_IT_TABIX	SYST	0	

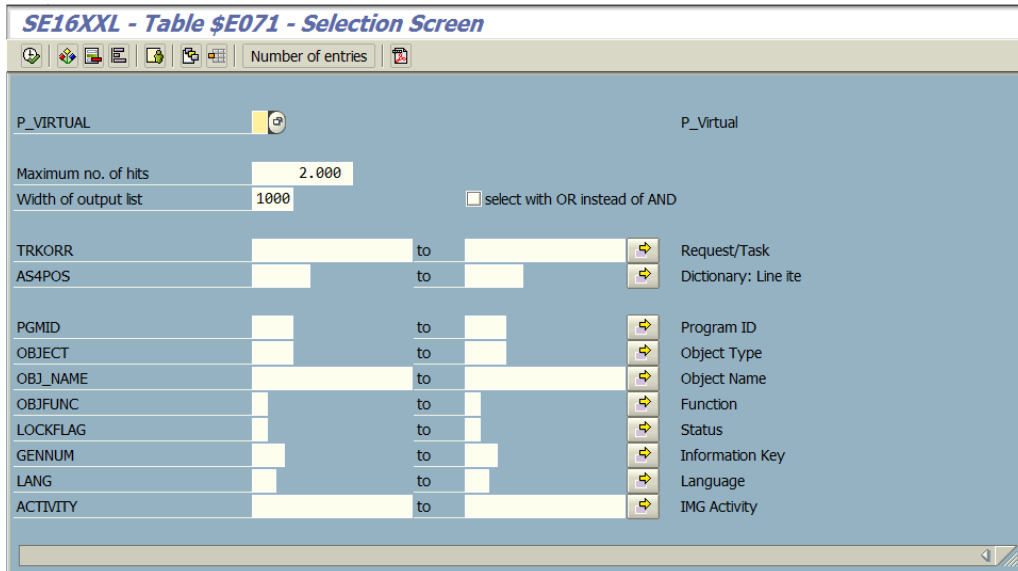
The connection between a LIMU FUNC and its R3TR FUGR is now immediately visible. It is fetched internally by means of function module '**TR\_CHECK\_TYPE**'.

This function also returns the **CATEGORY** ("SYST" or "CUST"). Possible problems are returned in field **X\_RC**.



## Selection Screen of \$E071

The selection screen is as follows (with all possible criteria):



The parameter P\_VIRTUAL (in imitation of a CDS view) has a special purpose.

Suppose that transport ZE5K901314 contains R3TR FUGR /TFTO/TX\_DISP. If both ZE5K901261 and ZE5K901314 are selected together, and P\_VIRTUAL is set to 'X', SE16XXL will create "virtual" LIMU entries to match the real ones. The corresponding entries (after sorting and filtering) would be as follows (the virtual entries have a **SUBPOS** value):

SE16XXL - Table \$E071 - 13 entries selected

TRKORR	AS4POS	SUBPOS	MAIN_PGMID	MAIN_OBJECT	MAIN_OBJ_NAME	PGMID	OBJECT	OBJ_NAME	CATEGORY	X_RC	OBJFUNC	LOCKFLAG	GENNUM	LANG	ACTIVITY
<input type="checkbox"/> ZE5K901261	1		R3TR	FUGR	/TFTO/TX_DISP	LIMU	CUAD	/TFTO/SAPLTX_DISP	SYST	0		X			
<input type="checkbox"/> ZE5K901314	79	1	R3TR	FUGR	/TFTO/TX_DISP	LIMU	CUAD	/TFTO/SAPLTX_DISP	SYST	0		X			
<input type="checkbox"/> ZE5K901261	4		R3TR	FUGR	/TFTO/TX_DISP	LIMU	FUNC	/TFTO/TX_COMPARE_RECORDS_INT	SYST	0		X			
<input type="checkbox"/> ZE5K901314	79	2	R3TR	FUGR	/TFTO/TX_DISP	LIMU	FUNC	/TFTO/TX_COMPARE_RECORDS_INT	SYST	0		X			
<input type="checkbox"/> ZE5K901261	5		R3TR	FUGR	/TFTO/TX_DISP	LIMU	FUNC	/TFTO/TX_DISPLAY_RECORD	SYST	0		X			
<input type="checkbox"/> ZE5K901314	79	3	R3TR	FUGR	/TFTO/TX_DISP	LIMU	FUNC	/TFTO/TX_DISPLAY_RECORD	SYST	0		X			
<input type="checkbox"/> ZE5K901261	6		R3TR	FUGR	/TFTO/TX_DISP	LIMU	FUNC	/TFTO/TX_DISPLAY_RECORD_INT	SYST	0		X			
<input type="checkbox"/> ZE5K901314	79	4	R3TR	FUGR	/TFTO/TX_DISP	LIMU	FUNC	/TFTO/TX_DISPLAY_RECORD_INT	SYST	0		X			
<input type="checkbox"/> ZE5K901261	9		R3TR	FUGR	/TFTO/TX_DISP	LIMU	REPS	/TFTO/LTX_DISPF01	SYST	0		X			
<input type="checkbox"/> ZE5K901314	79	5	R3TR	FUGR	/TFTO/TX_DISP	LIMU	REPS	/TFTO/LTX_DISPF01	SYST	0		X			
<input type="checkbox"/> ZE5K901261	10		R3TR	FUGR	/TFTO/TX_DISP	LIMU	REPS	/TFTO/LTX_DISPTOP	SYST	0		X			
<input type="checkbox"/> ZE5K901314	79	6	R3TR	FUGR	/TFTO/TX_DISP	LIMU	REPS	/TFTO/LTX_DISPTOP	SYST	0		X			
<input type="checkbox"/> ZE5K901314	79		R3TR	FUGR	/TFTO/TX_DISP	R3TR	FUGR	/TFTO/TX_DISP	SYST	0		3			

This makes it easier to compare the two transports.

[Back to the top](#)

## New Program /TFTO/TX\_SCHED\_SCRIPT

In a previous version of SE16XXL program /TFTO/TX\_SCHED\_SCRIPT\_SF has been introduced which makes it possible to schedule a script in background by means of a scheduling tool (possibly outside of SAP). This program works fine, but is restricted to the case of producing a server file as script result.

In order to lift this restriction a **new** program (/TFTO/TX\_SCHED\_SCRIPT) has been implemented, which **supports all kinds** of background runs, i.e. the result can be a server file, or be written to SAP spool, or can be saved to be analyzed at a later time using SE16XXL. The previous program may still be used, but its functionality is completely covered by the new one.

The mode of operation of the new program is the same as for the old one. Each time the program runs, it starts immediately the script in background. It is up to the user to define when the (scheduling) program should run.

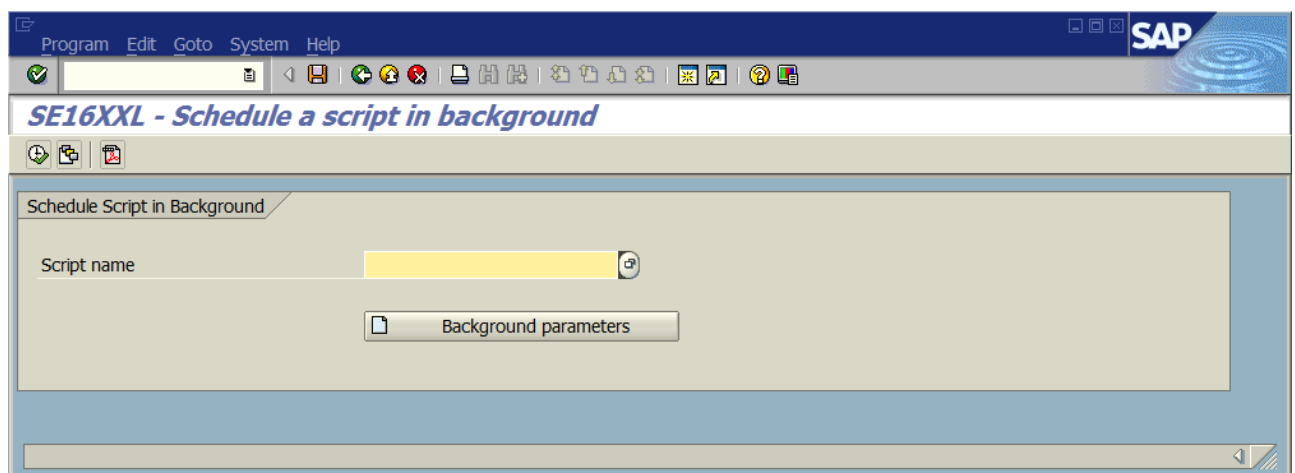
There are still some restrictions. Only the following options are available:

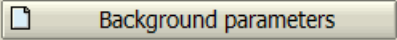
- with referenced script
- immediate start

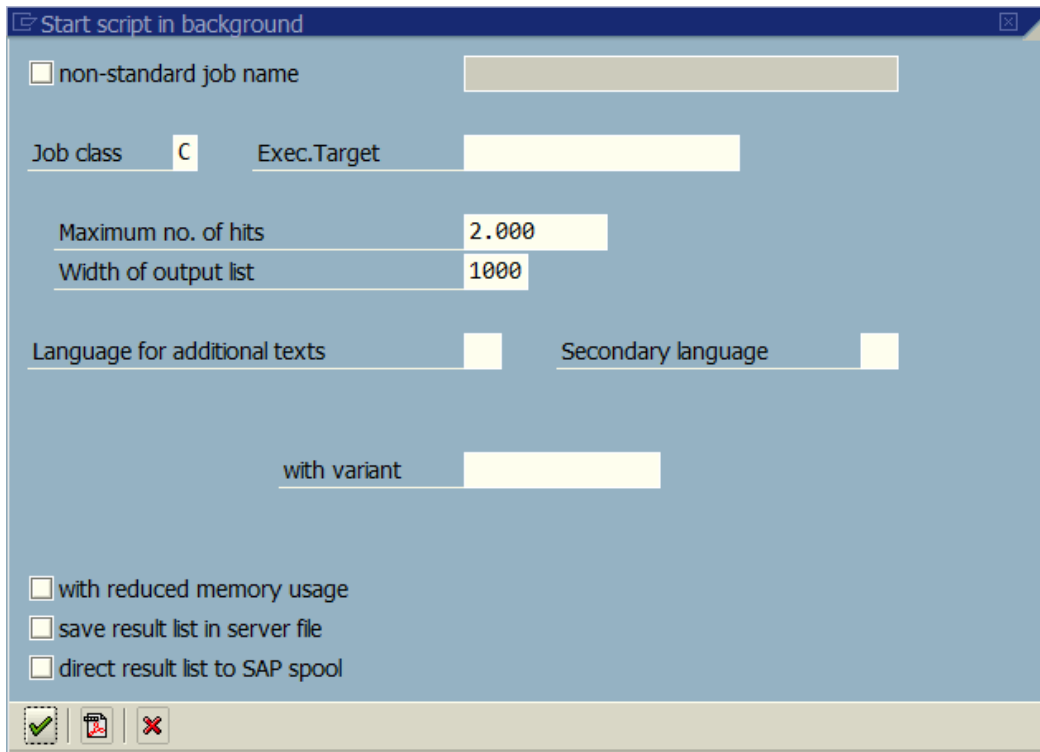
The above listed options correspond to scheduling a script in background from the first screen of SE16XXL with immediate start.

## Selection Screen of /TFTO/TX\_SCHED\_SCRIPT


The selection screen, which is needed to create a variant for running the program in the background, is (in contrast to the previous program) extremely simple:

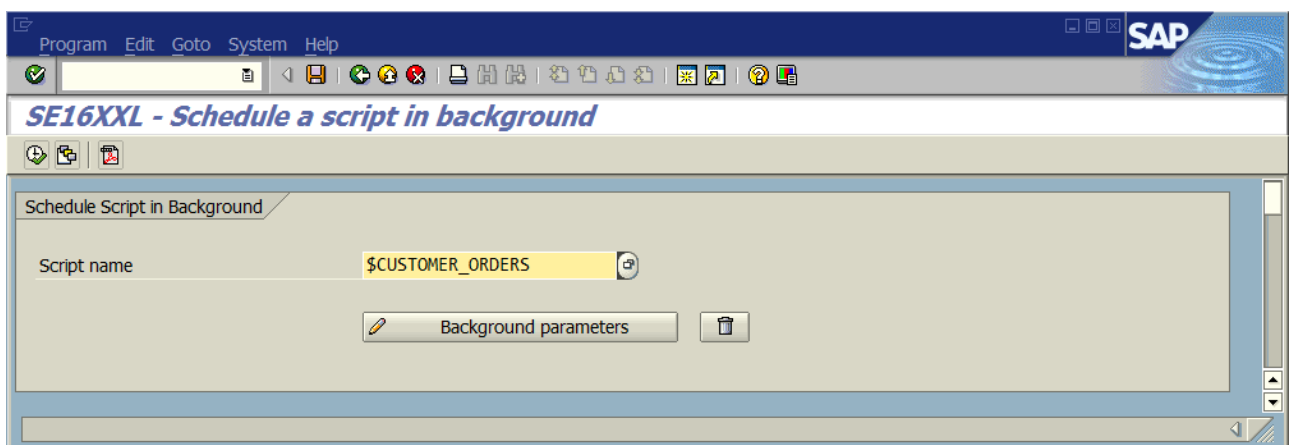


After typing in the name of the script press the  button to specify the required parameters. The usual popup for background scheduling appears;



Notice that no parameters are available for defining the start. This is because the start parameters and options, like **start date**, **start time**, **scheduling period** etc. are associated with the outer scheduling program and not with the script.

When all parameters have been specified, the selection screen shows up again to allow you to save these parameters in a variant (by pressing ):



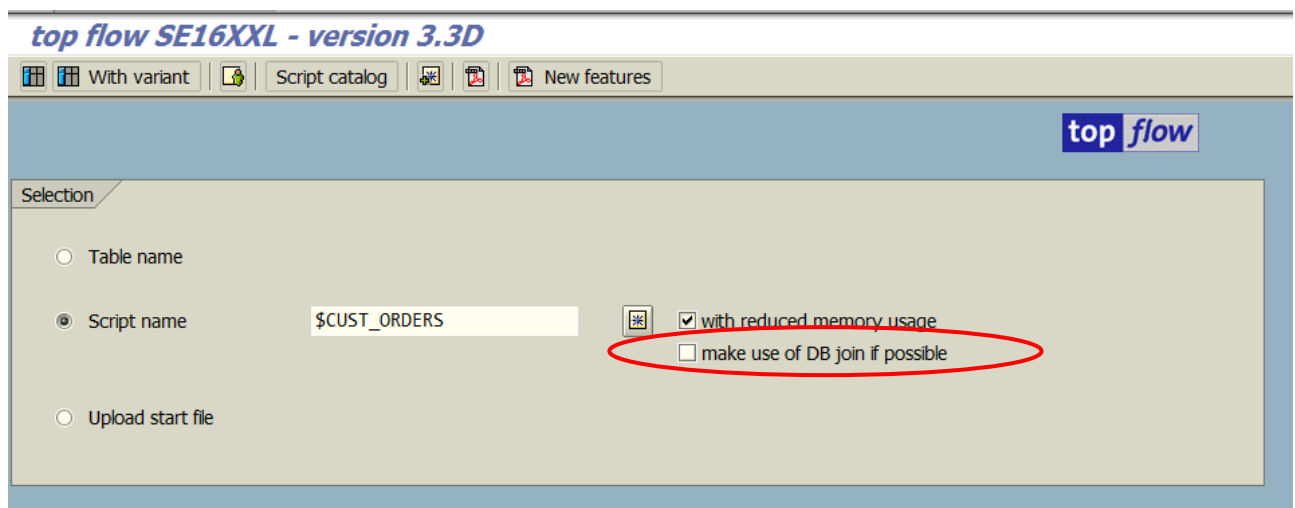
[Back to the top](#)

## Optional Use of Database Join when running a Script

Up to now all SELECT operations of a script were carried out internally one by one, i.e. each SQL selection was performed separately. This is normally a good strategy. It makes it possible to inspect the intermediate results, and is easy to understand. There are, however, situations which result in bad response times, since large amounts of intermediate data have to be transferred from the database server to the application server. In these cases it would be desirable to take advantage of the **inner join capabilities** of the database.

A **new option** is now available for this purpose.

When a script is to be run "*with reduced memory usage*" the new option shows up:



Its wording contains "*if possible*" because the logic is subject to the following restrictions:

- Only the SELECT operations at the beginning of the script are considered;
- Only transparent tables are eligible;
- Only a series of inner joins after the first SELECT operation is eligible;
- No other operations (except "Choose List Fields" and "Sort") may separate the various SELECT operations in the series of inner joins.

If the first n SELECT operations of the script fulfill the above listed restrictions and the option "*make use of DB join if possible*" has been activated, then all these SELECTs are carried out with **one** big SQL operation containing "inner join" clauses.

An appropriate example will illustrate what has been discussed.

The global script \$CUST\_ORDERS starts with the selection of table VBAK followed by four inner joins (VBUK, VBAP, VBUP, VBEP):



**SE16XXL - display global script '\$CUST\_ORDERS'**

Global script \$CUST\_ORDERS - Customer Orders with Details

This script contains ALV layout information

List of the script operations:

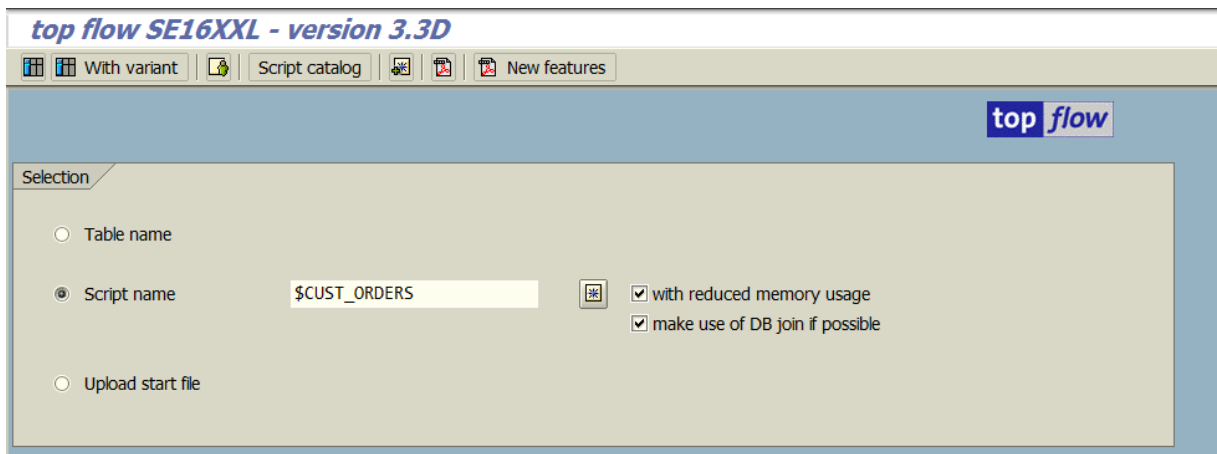
Nr. 1 **SESCREEN** First selection screen  
 SEL\_MODE=N - TABNAME-VBAK - SEL\_WITH\_OR= - ALIAS=A  
 Select fields: VBELN ERDAT ERZET ERNAM ANGOT BNDOT AUMAT VBTYP TRVOG AUART AUGRU GMLDT SUBMI LIFS K FAKSK NETHR WAERK VKORG VTWEG SPART VKGRP VKBUR GSBER GSKST QUEBG

Nr. 2 **SELECT** First selection  
 SEL\_MODE=N - TABNAME-VBAK - SEL\_WITH\_OR= - ALIAS=A  
 List fields: MANDT VBELN ERDAT ERZET VBTYP AUART WAERK VKORG VTWEG SPART KVGR1 KVGR2 KVGR3 KVGR4 KVGR5  
 Involved tables: VBAK

Nr. 3 **SELECT** Inner join  
 SEL\_MODE=I - TABNAME-VBUK - SEL\_WITH\_OR= - ALIAS=B  
 Join criteria: VBELN 000000 000000 C => VBELN C  
 List fields: A-MANDT A-VBELN A-ERDAT A-ERZET A-VBTYP A-AUART A-WAERK A-VKORG A-VTNEG A-SPART A-KVGR1 A-KVGR2 A-KVGR3 A-KVGR4 A-KVGR5 B-MANDT B-VBELN B-RFSTK B-RFGS  
 Involved tables: (A)VBAK (B)VBUK

Nr. 4 **SELECT** Inner join  
 SEL\_MODE=I - TABNAME-VBAP - SEL\_WITH\_OR= - ALIAS=C  
 Join criteria: A-VBELN 000000 000000 C => VBELN C

We run this script with the above described option:



**top flow SE16XXL - version 3.3D**

With variant | Script catalog | New features

**top flow**

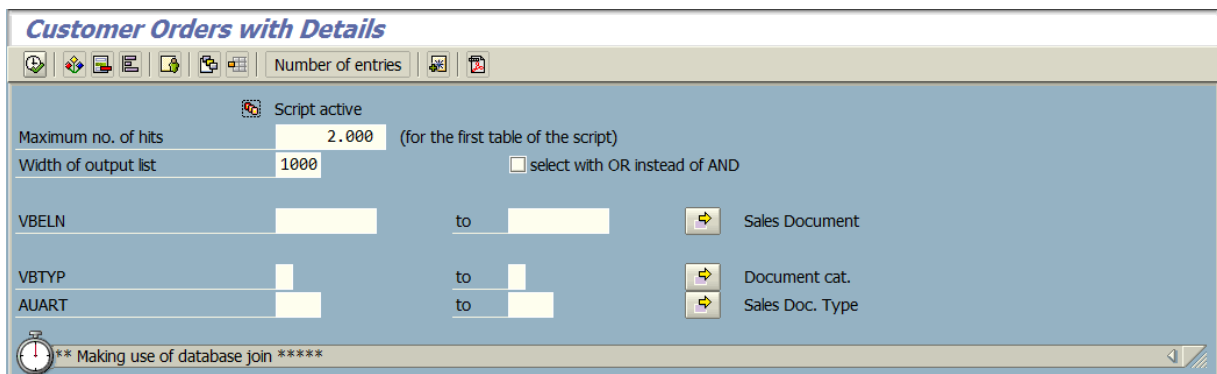
Selection

Table name

Script name    with reduced memory usage  
 make use of DB join if possible

Upload start file

On the ensuing selection screen we press  (execute):



**Customer Orders with Details**

Number of entries

Script active

Maximum no. of hits  (for the first table of the script)

Width of output list   select with OR instead of AND

VBELN  to   Sales Document

VBTYP  to   Document cat.

AUART  to   Sales Doc. Type

**\*\* Making use of database join \*\*\*\*\***

Notice the "**\*\*\*\*\* Making use of database join \*\*\*\*\***" at the bottom of the screen.

The dynamic coding contains in this case the following SQL statement:

```
SELECT
  A~MANDT A~VBELN A~VBTYP A~AUART A~VKORG A~VTWEG A~SPART
  A~VKGRP
  B~MANDT B~VBELN B~WBSTK B~ABSTK B~GBSTK
  C~MANDT C~VBELN C~POSNR C~MATNR C~PSTYV C~ABGRU
  D~MANDT D~VBELN D~POSNR D~FKSTA D~GBSTA
  E~MANDT E~VBELN E~POSNR E~ETENR E~ETTYP E~LFREL E~WMENG
  E~BMENG E~VRKME
  INTO TABLE PT_DBJOIN
  FROM ( ( ( ( VBAK AS A
              INNER JOIN VBUK AS B
                ON B~VBELN =
                  A~VBELN )
          INNER JOIN VBAP AS C
            ON C~VBELN =
              A~VBELN )
      INNER JOIN VBUP AS D
        ON D~VBELN =
          C~VBELN
        AND D~POSNR =
          C~POSNR )
    INNER JOIN VBEP AS E
      ON E~VBELN =
        C~VBELN
      AND E~POSNR =
        C~POSNR )
  UP TO PP_MAX_ROWS ROWS
  .
```

#### NOTE:

Under normal circumstances the response times with or without this option are almost identical. This is in my opinion due to the enormous computing power of the current servers.

If, however, a very large amount of data has to be selected in order to produce a very small result, then the DB join option may result in a significant improvement of the response time.

The best thing to do is to try out both situations, with and without the new option.

[Back to the top](#)

## ABAP Statement FIND allowed in a Formula

Up to now the ABAP statement FIND was not allowed in a SE16XXL formula. This restriction has now been lifted. It is thus possible to make use of **regular expressions** in a formula.

This new feature will be illustrated with an example.

Suppose you wish to find out if some of the messages of a message class begin with a lower case letter. This can be easily achieved with a little regular expression.

For our example we select the messages of class '13':

**SE16XXL - Table T100 - 185 entries selected**

Table T100 - Messages

SPRSL	ARBGB	MSGNR	TEXT
<input type="checkbox"/> E	13	001	Error when creating a change document
<input type="checkbox"/> E	13	002	Rate changed with document number \$ \$
<input type="checkbox"/> E	13	003	DYNPRO is not required because of selection, please check (?N)
<input type="checkbox"/> E	13	004	DYNPRO is in the loop (?N)
<input type="checkbox"/> E	13	005	IMPL./EXPL. The length EATAB differs (?N)
<input type="checkbox"/> E	13	006	Rate deleted with document number \$ \$

Now we define an appropriate formula by pressing  on the toolbar:

**Edit formula - inactive**

```

* Please enter your formula statements:
FF_DEF LOWERCASE_BEGIN TYPE SY-DATAR.
FIND FIRST OCCURRENCE OF REGEX '^[a-z]' IN ROW-TEXT.
IF SY-SUBRC = 0.
  LOWERCASE_BEGIN = 'X'.
ELSE.
  LOWERCASE_BEGIN = SPACE.
ENDIF.

```

Li 11, Co 1      Ln 1 - Ln 11 of 11 lines

Notice the simple regular expression.



After activating the formula we obtain the enhanced result list:

**SE16XXL - Table T100 - 185 entries selected**

Table T100 - Messages

SPRSL	ARGBG	MSGNR	TEXT	X~LOWERCASE_BEGIN	
<input type="checkbox"/>	E	13	001	Error when creating a change document	
<input type="checkbox"/>	E	13	002	Rate changed with document number \$ \$	
<input type="checkbox"/>	E	13	003	DYNPRO is not required because of selection, please check (?N)	
<input type="checkbox"/>	E	13	004	DYNPRO is in the loop (?N)	
<input type="checkbox"/>	E	13	005	IMPL./EXPL. The length EATAB differs (?N)	
<input type="checkbox"/>	E	13	006	Rate deleted with document number \$ \$	
<input type="checkbox"/>	E	13	007	Error when initializing VT block (?N)	
<input type="checkbox"/>	E	13	008	Table error in SAPmmKON/SAPmmSTA (?N)	
<input type="checkbox"/>	E	13	009	changed with document numbers of \$ \$ to \$ \$	X
<input type="checkbox"/>	E	13	010	Rate is not available : \$ \$ \$ (?N)	
<input type="checkbox"/>	E	13	011	Rate is already available : \$ \$ \$ (?N)	

By keeping by means of a filter only the relevant rows we finally obtain:

**SE16XXL - Table T100 - 23 entries selected**

Table T100 - Messages

SPRSL	ARGBG	MSGNR	TEXT	X~LOWERCASE_BEGIN	
<input type="checkbox"/>	E	13	009	changed with document numbers of \$ \$ to \$ \$	X
<input type="checkbox"/>	E	13	014	no changes made (?N)	X
<input type="checkbox"/>	E	13	017	incorrect key when locking (?N)	X
<input type="checkbox"/>	E	13	042	exceeded maximum discount for discount key \$	X
<input type="checkbox"/>	E	13	050	please check price translation (?N)	X
<input type="checkbox"/>	E	13	051	please check planned data (?N)	X
<input type="checkbox"/>	E	13	080	no rate found (?N)	X
<input type="checkbox"/>	E	13	102	please enter discount price unit for discount keys - \$ - (?N)	X
<input type="checkbox"/>	E	13	103	please enter discount translation ratio for discount keys - \$ - (?N)	X
<input type="checkbox"/>	E	13	104	please enter discount quantity unit for discount keys - \$ - (?N)	X
<input type="checkbox"/>	E	13	105	please check discount and net price	X
<input type="checkbox"/>	E	13	107	please check article description	X
<input type="checkbox"/>	E	13	108	please enter art.no., quant./percent rate,min. quant. together (?N)	X
<input type="checkbox"/>	E	13	109	no date 'GUELTIG AB' entered for rebate in kind (?N)	X
<input type="checkbox"/>	E	13	110	only date 'GUELTIG AB' entered , however no future values (?N)	X
<input type="checkbox"/>	E	13	111	with KZ M/% =M must be entered 'per' (?N)	X
<input type="checkbox"/>	E	13	112	for KZ M/% = P may no 'per' be entered (?N)	X
<input type="checkbox"/>	E	13	114	future price unit must be set	X
<input type="checkbox"/>	E	13	115	please check sorting	X
<input type="checkbox"/>	E	13	136	for PREISLISTE/KONDITION \$, is the same period of validity defined (?N)	X
<input type="checkbox"/>	E	13	503	no record found : VB\$ Module : \$ / \$	X
<input type="checkbox"/>	E	13	504	duplicate record : VB\$ Module : \$ / \$	X
<input type="checkbox"/>	E	13	619	during entry of a max. quantity must also a quantity unit be entered	X

These are the messages we were looking for.

[Back to the top](#)

## Detailed Entry Display – Prev. entry / Next entry

A double click on a row of the SE16XXL result list usually leads to the detailed list of the row (except in case a jump has been defined for the column).

In order to view the next row in detail the user had up to now to leave the detailed list and doubleclick on the next row, and so on. Rather nasty, if done repeatedly.

Scrolling along the detailed rows can now be done without leaving the detailed view. Two new functions – **Prev. entry** and **Next entry** – have been implemented.

This new feature will be illustrated by means of an example.

Suppose the result list to contain an inner join of invoice headers and items:

*SE16XXL - inner join - 30 resulting rows*

A~VBELN	A~FKART	A~FKTYP	A~VBTYP	A~VKORG	A~VTWEG	A~NETWR	A~WAERK	B~POSNR	B~FKIMG	B~VRKME	B~NETWR	B~MATNR
0090023610	F2	L	M	3020	30	2.995,00	USD	10	1	ST	902,00	ISA-0021
0090023610	F2	L	M	3020	30	2.995,00	USD	20	1	ST	833,00	ISA-0025
0090023610	F2	L	M	3020	30	2.995,00	USD	30	1	ST	32,00	ISA-1038
0090023610	F2	L	M	3020	30	2.995,00	USD	40	1	ST	103,00	ISA-1018
0090023610	F2	L	M	3020	30	2.995,00	USD	50	4	ST	100,00	ISA-1034
0090023610	F2	L	M	3020	30	2.995,00	USD	60	1	EA	921,00	ISA-0019
0090023610	F2	L	M	3020	30	2.995,00	USD	70	2	ST	104,00	ISA-1029
0090026208	F2	L	M	3020	30	8.415,50	USD	10	3	ST	6.009,90	HT-1010
0090026208	F2	L	M	3020	30	8.415,50	USD	20	2	ST	479,60	HT-1023
0090026208	F2	L	M	3020	30	8.415,50	USD	30	1	ST	1.447,00	HT-1037
0090026208	F2	L	M	3020	30	8.415,50	USD	40	2	ST	357,60	HT-1052
0090026208	F2	L	M	3020	30	8.415,50	USD	50	1	ST	56,10	HT-1104
0090026208	F2	L	M	3020	30	8.415,50	USD	60	1	ST	34,90	HT-1106
0090026208	F2	L	M	3020	30	8.415,50	USD	70	1	ST	30,40	HT-1107
0090026211	F2	L	M	3020	30	10.542,10	USD	10	3	ST	6.909,30	HT-1011

We start by doubleclicking on the first row:

*Join of VBRK(A) and VBRP(B)*

Field	Value	Description
A~MANDT	'800'	Client
A~VBELN	'0090023610'	Billing Document
A~FKART	'F2'	Billing Type
A~FKTYP	'L'	Billing category
A~VBTYP	'M'	SD document category
A~WAERK	'USD'	SD Document Currency
A~VKORG	'3020'	Sales Organization
A~VTWEG	'30'	Distribution Channel
A~KALSM	'RVCXUS'	Sales and Distribution: Pricing Procedure in Pricing
A~KNIMV	'0000042316'	Number of the document condition

Since all headers are alike, we compress the A - VBRK part:

### Join of VBRK(A) and VBRP(B)

Join of VBRK(A) and VBRP(B)		
A - VBRK - TRANSP - Billing Document: Header Data		
B - VBRP - TRANSP - Billing Document: Item Data		
B~MANDT	'800'	Client
B~VBELN	'0090023610'	Billing Document
B~POSNR	'000010'	Billing item
B~UEPOS	'000000'	Higher-level item in bill of material structures
B~FKIMG (B~VRKME)	1,000	Actual Invoiced Quantity
B~VRKME	'ST'	Sales unit
B~UMVKZ	1	Numerator (factor) for conversion of sales quantity into SKU
B~UMVKN	1	Denominator (Divisor) for Conversion of Sales Qty into SKU
B~MEINS	'ST'	Base Unit of Measure
B~SMENG (B~MEINS)	0,000	Scale quantity in base unit of measure
B~FKLMG (B~MEINS)	1,000	Billing quantity in stockkeeping unit
B~LMENG (B~MEINS)	1,000	Required quantity for mat.management in stockkeeping units
B~NTGEW (B~GEWEI)	23,000	Net weight
B~BRGEW (B~GEWEI)	23,000	Gross weight
B~GEWEI	'KG'	Weight Unit

By pressing  we obtain the next item of the invoice:

### Join of VBRK(A) and VBRP(B)

Join of VBRK(A) and VBRP(B)		
A - VBRK - TRANSP - Billing Document: Header Data		
B - VBRP - TRANSP - Billing Document: Item Data		
B~MANDT	'800'	Client
B~VBELN	'0090023610'	Billing Document
B~POSNR	'000020'	Billing item
B~UEPOS	'000000'	Higher-level item in bill of material structures
B~FKIMG (B~VRKME)	1,000	Actual Invoiced Quantity
B~VRKME	'ST'	Sales unit
B~UMVKZ	1	Numerator (factor) for conversion of sales quantity into SKU
B~UMVKN	1	Denominator (Divisor) for Conversion of Sales Qty into SKU
B~MEINS	'ST'	Base Unit of Measure
B~SMENG (B~MEINS)	0,000	Scale quantity in base unit of measure
B~FKLMG (B~MEINS)	1,000	Billing quantity in stockkeeping unit
B~LMENG (B~MEINS)	1,000	Required quantity for mat.management in stockkeeping units
B~NTGEW (B~GEWEI)	34,000	Net weight
B~BRGEW (B~GEWEI)	34,000	Gross weight
B~GEWEI	'KG'	Weight Unit

Notice that the compressed state of A – VBRK has remained unchanged. The current position and compression state of the detailed view are kept unchanged during scrolling.

**NOTE:** if only some rows are selected on the result list, only these rows will take part in the scrolling. If no rows are selected, all rows take part.

[Back to the top](#)

## Detailed Entry Comparison – Keys also compared

Up to now the detailed comparison of rows (🔍) did not compare the values of the key fields, since it was assumed that they always differ. This behavior is reasonable for normal tables with a limited number of key fields. In SAP, however, there are database views whose fields **are all marked** as keyfields. For such views the above restriction leads to no comparison at all – an unsatisfactory situation.

For this reason the above restriction has been lifted.

An example with database view **DD08VV** will illustrate the old and new behavior.

### Old Behavior:

5 entries of view DD08VV - Foreign keys

5 entries of view DD08VV - Foreign keys						
TABNAME	T001Q	T001Q	T001Q	T001Q	T001Q	Table Name
FIELDNAME	BUKRS	BUKRS	MANDT	SUBST	SUBST	Name of the field to be checked
AS4LOCAL	A	A	A	A	A	Activation Status of a Repository Object
AS4VERS	0000	0000	0000	0000	0000	Version of the entry (not used)
PRIMPOS	0001	0002	0001	0001	0002	Position of corresp. foreign key field in primary key
CHECKTABLE	T001	T001	T000	G892	G892	Check table name of the foreign key
FRKART	ID	ID	ID	OBL	OBL	Dependency factor for semantic foreign keys
CLASFIELD						Internal classification of foreign keys: partial, etc.
CLASVALUE						Values for classif. field of CLAS relationship type
CARDLEFT						Cardinality of a relationship
CARD	CN	CN	CN	CN	CN	Cardinality of a relationship
CHECKFLAG						No check in Screen Painter
ARBGB						Application Area
MSGNR						Message number
NOINHERIT						DD: Flag whether inheritance is switched off for the FRK
FORTABLE	T001Q	T001Q	T001Q	T001Q	T001Q	Table of the foreign key field
FORKEY	MANDT	BUKRS	MANDT	MANDT	SUBST	Names of Foreign Key Fields
FORSTRING						Foreign Key Part in Other Table

### New Behavior:

5 entries of view DD08VV - Foreign keys


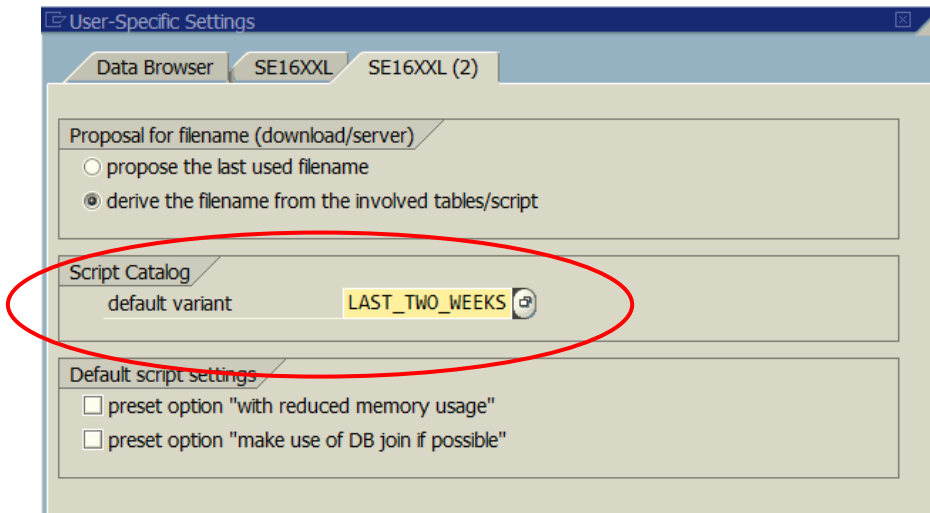
5 entries of view DD08VV - Foreign keys						
TABNAME	T001Q	T001Q	T001Q	T001Q	T001Q	Table Name
FIELDNAME	BUKRS	BUKRS	MANDT	SUBST	SUBST	Name of the field to be checked
AS4LOCAL	A	A	A	A	A	Activation Status of a Repository Object
AS4VERS	0000	0000	0000	0000	0000	Version of the entry (not used)
PRIMPOS	0001	0002	0001	0001	0002	Position of corresp. foreign key field in primary key
CHECKTABLE	T001	T001	T000	G892	G892	Check table name of the foreign key
FRKART	ID	ID	ID	OBL	OBL	Dependency factor for semantic foreign keys
CLASFIELD						Internal classification of foreign keys: partial, etc.
CLASVALUE						Values for classif. field of CLAS relationship type
CARDLEFT						Cardinality of a relationship
CARD	CN	CN	CN	CN	CN	Cardinality of a relationship
CHECKFLAG						No check in Screen Painter
ARBGB						Application Area
MSGNR						Message number
NOINHERIT						DD: Flag whether inheritance is switched off for the FRK
FORTABLE	T001Q	T001Q	T001Q	T001Q	T001Q	Table of the foreign key field
FORKEY	MANDT	BUKRS	MANDT	MANDT	SUBST	Names of Foreign Key Fields
FORSTRING						Foreign Key Part in Other Table

**NOTE:** The comparison of key field values is now **always** carried out, not only for database views but for all kinds of tables as well.

[Back to the top](#)

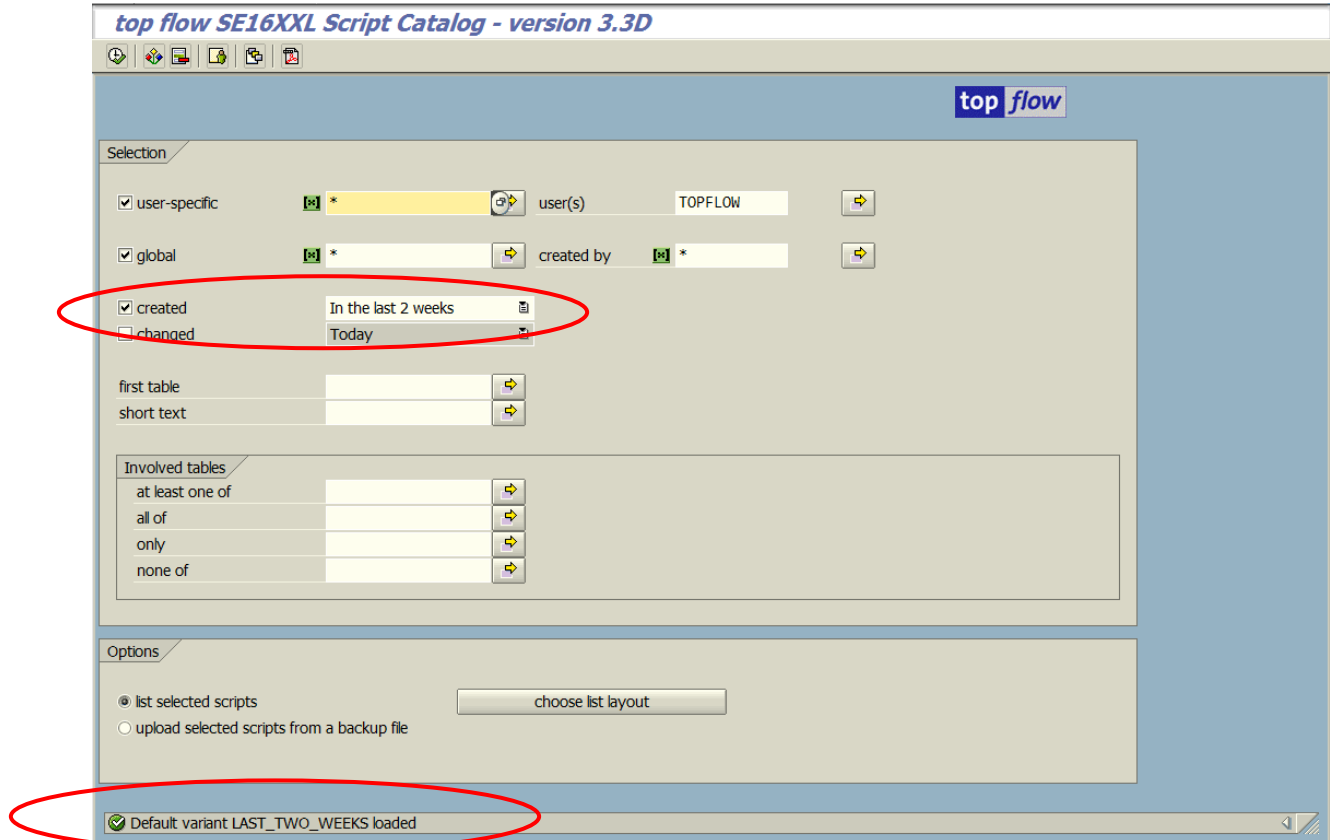
## New User Settings

### Default Variant for the Script Catalog


This new setting is available on the **third tab** of the SE16XXL User Settings (


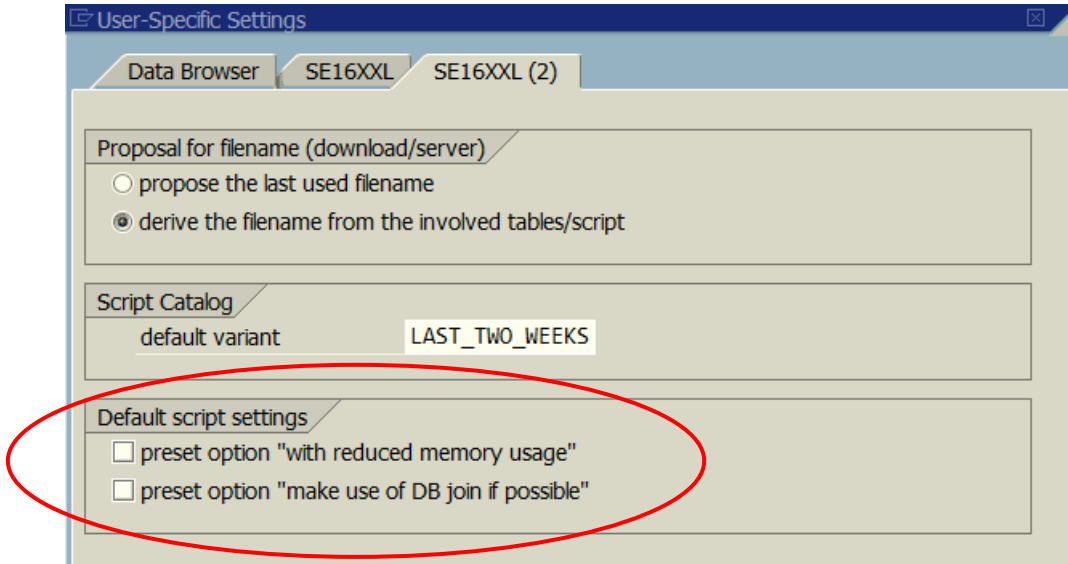
The screenshot shows the 'User-Specific Settings' dialog box with three tabs: 'Data Browser', 'SE16XXL', and 'SE16XXL (2)'. The 'SE16XXL (2)' tab is active. The 'Script Catalog' section is circled in red and contains a 'default variant' dropdown menu set to 'LAST\_TWO\_WEEKS'. Other sections include 'Proposal for filename (download/server)' with radio buttons for 'propose the last used filename' and 'derive the filename from the involved tables/script', and 'Default script settings' with checkboxes for 'preset option "with reduced memory usage"' and 'preset option "make use of DB join if possible"'. The 'LAST\_TWO\_WEEKS' dropdown is also circled in red.

When the Script Catalog is called it loads automatically the default variant:



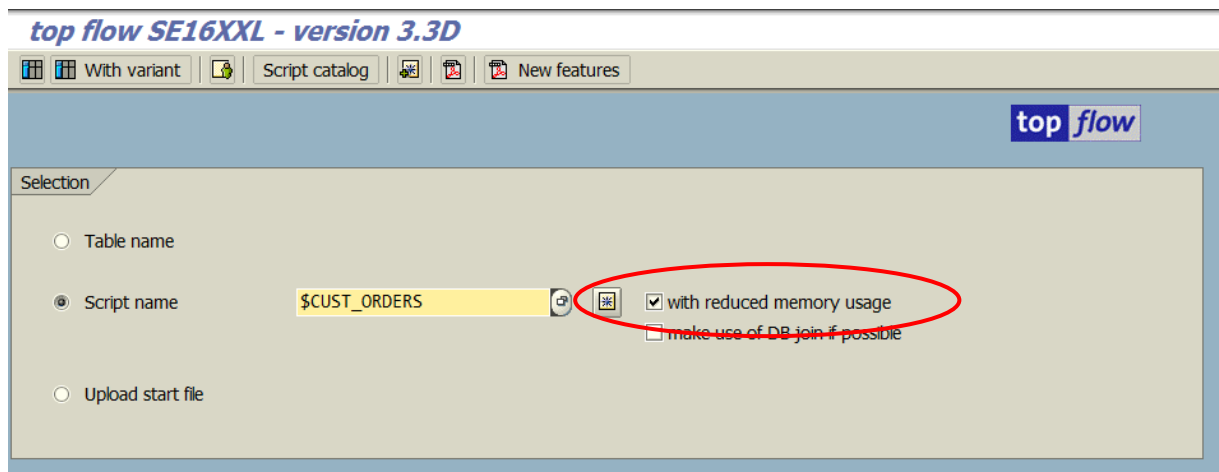
## Default Settings for Performing a Script

These settings are also available on the third tab of the User Settings ():



### Preset option "with reduced memory usage"

If this setting is activated the option "*with reduced memory usage*" on the SE16XXL initial screen will be preset to "**activated**":



In this way the user does not have to activate the option each time. It **does not mean** that all scripts will be performed with reduced memory usage.

The **second** setting in regard to the use of a DB join **works analogously**.

[Back to the top](#)



## Tool for Downloading and Uploading ALV Layouts

SE16XXL result lists are mostly displayed by means of **ALV grid** or **ALV list**. It is thus possible to make use of **ALV layouts**, which give a personal touch to the list.

**ALV** is a **standard** functionality of SAP for displaying lists. It is **not** part of the SE16XXL add-on. Neither are ALV layouts. They can be used to render SE16XXL result lists more attractive, but are **not** part of SE16XXL. They are stored in SAP standard tables, not in the namespace of SE16XXL.

### SE16XXL and ALV Layouts

The basic assumption of ALV layouts is that a particular ABAP report produces one or more ALV lists. For each kind of list ALV layouts can be created and loaded at a later time. Thus an ALV layout is characterized (among other things) by the name of the **report**, the name of the **layout**, and possibly the logon name of the **user**, if the layout is not standard but user-specific. This assumption works well with normal reports, but is **totally inadequate** for SE16XXL. Why? Because in SE16XXL the **same program** produces all kinds of result lists. If the real program name were used for saving the ALV layouts, **all existing** SE16XXL ALV layouts would pop up in the F4 help, even if they would have absolutely nothing to do with the current result list.

Fortunately ALV **does not check** if the specified report actually exists.

SE16XXL groups ALV layouts by **result list structure**. This means that if the result list contains a join of MARA and MVKE, all ALV layouts defined for this combination (MARA + MVKE) are available for loading – and only these. This grouping (or separation) is achieved internally by creating a **pseudo report name** derived from the combination of tables. In our example the derived report name would be **/TFTO/TX~~MARA~MVKE**. The report name can be up to 40 characters long. Join structures may involve up to 20 tables. The logic for creating the pseudo report name fails if the name gets too long. For such situations a special logic has been implemented to create a unique report name.

At this point it must be emphasized that ALV layouts **are not associated** directly to SE16XXL scripts. Rather they are associated to the result list structure of a script. If the structure of the result list of a script is changed, owing for example to the insertion of an additional SELECT operation, the "old" ALV layouts disappear from the F4 help of this script, to be substituted by new ones, if for the new table combination any exist.

After this brief introduction we may proceed to the main topic, the tool for downloading and uploading ALV layouts.

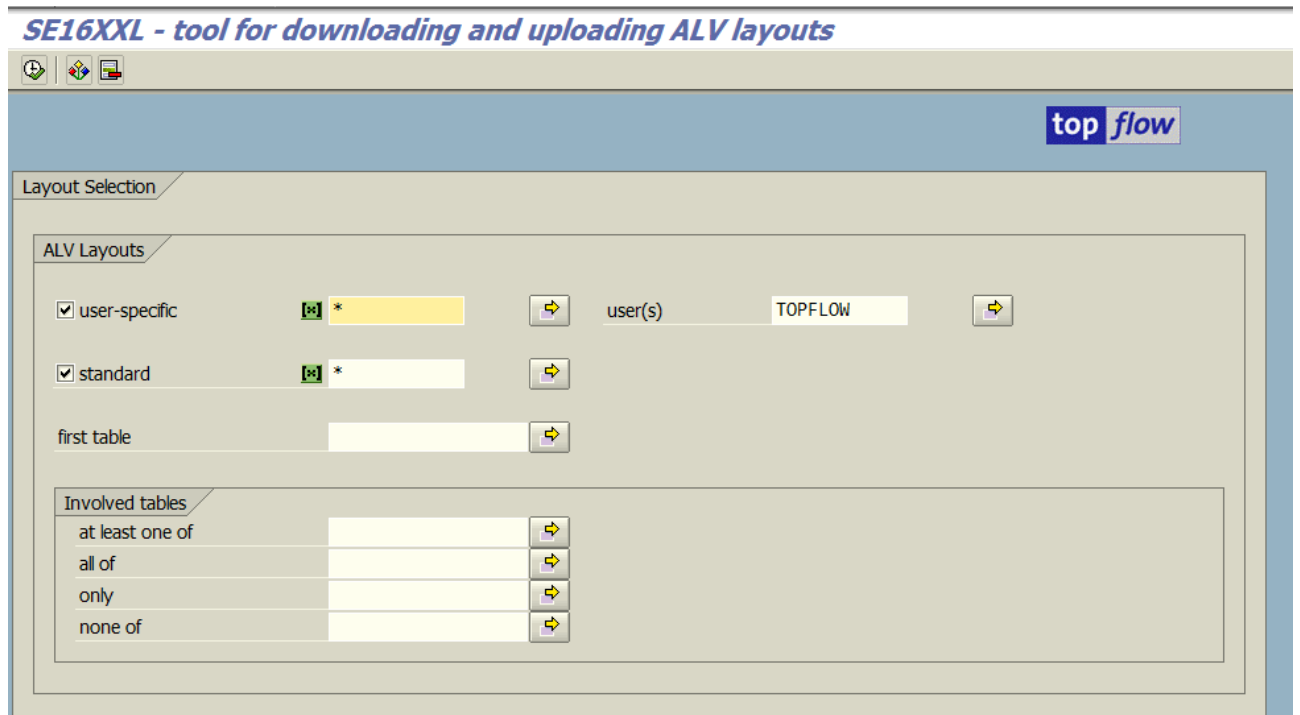


## Description of the Tool

The tool can be reached from the initial screen of SE16XXL by making use of the menu function

*Goto* → *ALV layouts ...*

The main part of the selection screen of the tool is as follows:



**SE16XXL - tool for downloading and uploading ALV layouts**

Layout Selection

ALV Layouts

user-specific  user(s) TOPFLOW

standard

first table

Involved tables

at least one of

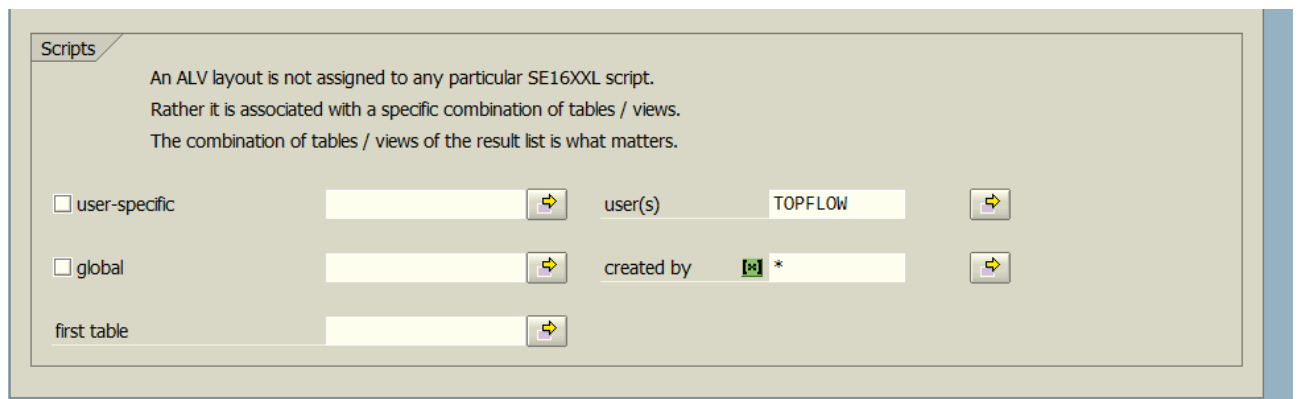
all of

only

none of

This allows to select ALV layouts according to the involved tables.

In some cases it may be desirable to restrict the selection to particular scripts. For this reason a second block of criteria is also available:



Scripts

An ALV layout is not assigned to any particular SE16XXL script.  
Rather it is associated with a specific combination of tables / views.  
The combination of tables / views of the result list is what matters.

user-specific  user(s) TOPFLOW

global  created by

first table

The standard option "download ALV layouts" leads to the following popup window:

Please choose the ALV layouts to be downloaded

No.	User	Layout	Layout description	First Table	2nd Table	3rd Table	4th Table	5th Table	6th Table	7th Table	...	Created on	Creat. time	Creator
1		/ALLE_FELDER		VBAK								17.07.2016	14:36:07	TOPFLOW
2		/KD_GRUPPEN		VBAK								17.07.2016	14:40:26	TOPFLOW
3		/MINIMAL		VBAK								17.07.2016	14:37:24	TOPFLOW
4		/STUEKCLASS		VBAK								17.07.2016	14:39:15	TOPFLOW
5		/ALL_FIELDS	with all fields	VBRK								22.07.2018	09:58:14	TOPFLOW
6		/MINIMAL	Minimal list with very few columns	VBRK								22.07.2018	09:58:14	TOPFLOW
7		/TAX_CLASS	With tax classification fields	VBRK								22.07.2018	09:58:14	TOPFLOW
8		/WITH_INCO	With incoterms	VBRK								22.07.2018	09:58:14	TOPFLOW
9		/KUNAG_SUBT	xx	VBRK	VBRP							22.07.2018	09:58:14	TOPFLOW
10		/KUNAG_SUMS	xx	VBRK	VBRP							22.07.2018	09:58:14	TOPFLOW
11		/KUNAG_XUBT	xx	VBRK	VBRP							22.07.2018	09:58:14	TOPFLOW
12		/KUNAG_XUMX	xxx	VBRK	VBRP							22.07.2018	09:58:14	TOPFLOW
13		/KUNAG_XXXX	xxxx	VBRK	VBRP							22.07.2018	09:58:14	TOPFLOW
14		/TOTALS	xx	VBRK	VBRP							22.07.2018	09:58:14	TOPFLOW
15		/GAMBA	xx	VBRP								06.07.2016	13:32:03	TOPFLOW
16		/MATNR	xx	VBRP								06.07.2016	13:40:12	TOPFLOW
17		/STD	Standard	VBRP								06.07.2016	13:33:02	TOPFLOW
18	TOPFLOW	MEIN_LAYOUT		VBAK								17.07.2016	14:33:58	TOPFLOW
19	TOPFLOW	LONG1	xx	VBAK	VBAP	VBEP	TVKOT	TSPAT	TVTWT	MAKT		22.07.2018	10:00:55	TOPFLOW
20	TOPFLOW	LONG1_ONLY	xx	VBAK	VBAP	VBEP	TVKOT	TSPAT	TVTWT	MAKT		22.07.2018	10:00:55	TOPFLOW
21	TOPFLOW	LONG1_SUBTOT	xx	VBAK	VBAP	VBEP	TVKOT	TSPAT	TVTWT	MAKT		22.07.2018	10:00:55	TOPFLOW
22	TOPFLOW	LONG2	xx	VBAK	VBAP	VBEP	TVKOT	TSPAT	TVTWT	MAKT		22.07.2018	10:00:55	TOPFLOW
23	TOPFLOW	LONG2_ONLY	xx	VBAK	VBAP	VBEP	TVKOT	TSPAT	TVTWT	MAKT	...	22.07.2018	10:00:55	TOPFLOW
24	TOPFLOW	LONG2_SUBTOT	xx	VBAK	VBAP	VBEP	TVKOT	TSPAT	TVTWT	MAKT	...	22.07.2018	10:00:55	TOPFLOW
25	TOPFLOW	ONLY_SUBTOTS	xx	VBAK	VBAP	VBEP	TVKOT	TSPAT	TVTWT	MAKT	...	22.07.2018	10:00:56	TOPFLOW
26	TOPFLOW	KUNAG_SUBTOT	xx	VBRK								22.07.2018	10:00:56	TOPFLOW
27	TOPFLOW	MY_LAYOUT	My personal default layout	VBRK								22.06.2016	10:19:03	TOPFLOW
28	TOPFLOW	ONLY_KUN_SUB	xx	VBRK								22.07.2018	10:00:56	TOPFLOW
29	TOPFLOW	KUNAG_SUBT	xx	VBRK	VBRP							22.07.2018	10:00:56	TOPFLOW
30	TOPFLOW	KUNAG_SUMS	xx	VBRK	VBRP							22.07.2018	10:00:56	TOPFLOW
31	TOPFLOW	MY_STANDARD	xxx	VBRK	VBRP							22.07.2018	10:00:56	TOPFLOW

These ALV layouts to be downloaded can be individually marked.

## Upload

Options

download ALV layouts  
 upload ALV layouts

If the upload option is chosen the program first asks for an appropriate upload file. The contents of this file are then presented in a similar popup window:

Please choose the ALV layouts to be imported

No.	User	Layout	Layout description	First Table	2nd Table	3rd Table	4th Table
1		/ALLE_FELDER		VBAK			
2		/KD_GRUPPEN		VBAK			
3		/MINIMAL		VBAK			
4		/STUEKCLASS		VBAK			
5		/ALL_FIELDS	with all fields	VBRK			
6		/MINIMAL	Minimal list with very few columns	VBRK			
7		/TAX_CLASS	With tax classification fields	VBRK			

If any layouts already exist the tool asks for permission before overwriting them.

[Back to the top](#)



When we now make use of the F4 help for script variants ( ) on the selection screen, our new variant will also show up:

Please choose a variant (1) 3 Entries found

Restrictions

Variant	Def.	Short Text	ALV Layout	Created by
MEINS_M2		Materials with basic unit of measure M2	/SOME_FIELDS	TOPFLOW
MEINS_PIECES		Only unit of measure "pieces"	TOP_LAYOUT	TOPFLOW
STANDARD		Standard Selection	/AEND	TOPFLOW

3 Entries found

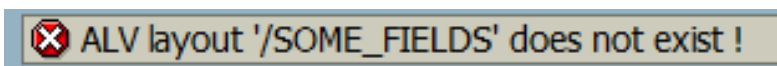
The result list makes use, as expected, of the selected ALV layout:

SE16XXL - Table MARA - 97 entries selected

Table MARA - General Material Data

MATNR	ERSDA	ERNAM	LAEDA	AENAM	VPSTA	PSTAT	MEINS	MTART	MATKL	NORMT	VOLUM	VOLEH	NTGEW	BRGEW	GEWEI
<input type="checkbox"/> 00000000000000821	10.09.2003	MERTZLUFFT			K	K	M2	HAWA	00107				1	1	KG
<input type="checkbox"/> 100-700	08.11.1994	BALLER	17.10.2002	VANDIJK	KCEDPLSQBGXZ	KCEDPLSQBG	M2	ROH	001				0	1	KG
<input type="checkbox"/> 102-700	14.02.2002	KUNITZ	17.10.2002	VANDIJK	KCEDPLSQBGXZ	KCEDPLSQBG	M2	ROH	001				0	1	KG
<input type="checkbox"/> 350-150	10.09.2003	MERTZLUFFT	10.09.2003	MERTZLUFFT	KCVBGX	KCVBG	M2	HAWA	00107				1	1	KG
<input type="checkbox"/> 350-170	10.09.2003	MERTZLUFFT	10.09.2003	MERTZLUFFT	KBCV	KBCV	M2	HAWA	00107				1	1	KG
<input type="checkbox"/> 400-700	24.05.1995	D002772	18.02.1999	FISCHER	KCEDPLQBGXZ	KCEDPLQBG	M2	ROH	001						
<input type="checkbox"/> 401-700	17.06.1996	PIPERMUSIOL	06.03.1997	FUNKE	KDLBGXZ	KDLBG	M2	ROH	001						
<input type="checkbox"/> AI002	16.08.2006	BIRNLEY			KEDLBG	KEDLBG	M2	ROH	001				1	1	KG
<input type="checkbox"/> DB-100	23.12.1994	BALLER	13.09.1996	MACXAS	KEDLSBXZGPQC	KEDLSBGPQC	M2	ROH	010						
<input type="checkbox"/> E-1302	11.05.1995	STEINHORST	25.08.1997	STADEL	KVEDLBGXZC	KVEDLBGC	M2	HALB	013				100	100	KG
<input type="checkbox"/> E-1304	11.05.1995	STEINHORST	25.08.1997	STADEL	KVEDLBGXZC	KVEDLBGC	M2	HALB	013				100	100	KG
<input type="checkbox"/> E-1305	11.05.1995	STEINHORST	22.08.1997	STADEL	KVEDLBGXZC	KVEDLBGC	M2	HALB	013				100	100	KG
<input type="checkbox"/> E-1306	11.05.1995	STEINHORST	22.08.1997	STADEL	KVEDLBGXZC	KVEDLBGC	M2	HALB	013				100	100	KG
<input type="checkbox"/> E-1308	11.05.1995	STEINHORST	22.08.1997	STADEL	KVEDLBGXZ	KVEDLBG	M2	HALB	013				10	10	KG
<input type="checkbox"/> E-1308A	23.05.1995	PROUSE	20.08.1997	STADEL	KVEDQBGXZ	KVEDQBG	M2	HALB	013				10	10	KG
<input type="checkbox"/> E-1309	11.05.1995	STEINHORST	26.08.1997	STADEL	KVEDLBGXZ	KVEDLBG	M2	HALB	013				10	10	KG
<input type="checkbox"/> E-1309A	23.05.1995	PROUSE	26.08.1997	STADEL	KVEDQBGXZ	KVEDQBG	M2	HALB	013				10	10	KG
<input type="checkbox"/> E-1310	11.05.1995	STEINHORST	22.08.1997	STADEL	KVEDLBGXZ	KVEDLBG	M2	HALB	013				10	10	KG
<input type="checkbox"/> E-1310A	23.05.1995	PROUSE	20.08.1997	STADEL	KVEDQBGXZ	KVEDQBG	M2	HALB	013				10	10	KG
<input type="checkbox"/> E-1311	11.05.1995	STEINHORST	22.08.1997	STADEL	KVEDLBGXZC	KVEDLBGC	M2	HALB	013				10	10	KG
<input type="checkbox"/> E-1312	11.05.1995	STEINHORST	22.08.1997	STADEL	KVEDLBGXZ	KVEDLBG	M2	HALB	013				10	10	KG
<input type="checkbox"/> E-1317	11.05.1995	STEINHORST	05.09.1997	STADEL	KVEDLBGXZC	KVEDLBGC	M2	HALB	013				50	50	KG

**NOTE:** It must be kept in mind that if the structure of the result list is changed, for example by inserting a new join into the script operations, the "old" ALV layouts will not be available any more. In this case when we select the script variant the following message will show up:



[Back to the top](#)

## Improvement of the Operation "Add text columns"

In a previous version of SE16XXL the function "*Add text columns*" was introduced, which adds descriptive texts to the result list. So-called **text tables** are the most frequent source of descriptive texts. Each text table is defined in the Data Dictionary and is distinguished by a **language field** among its key fields. But this condition is not sufficient – it must also be declared **explicitly** to be the text table of another database table. For example **TVOKT** is the text table of **TVKO**, the master table of Sales Organizations. In the Data Dictionary (transaction SE11) a function is available to find the text table of a given database table: *Goto → Text Table*.

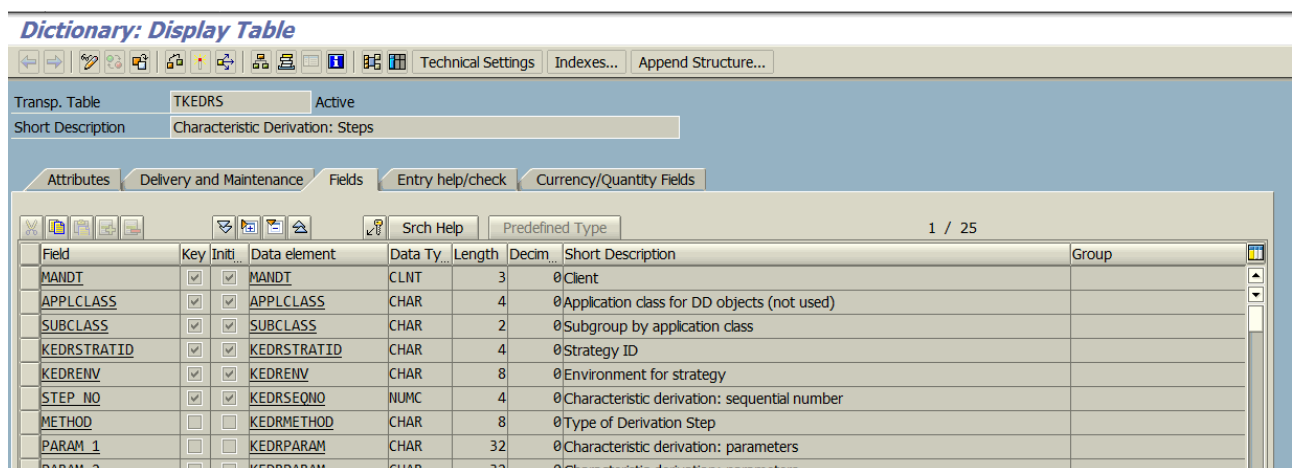
In SAP however, there are database tables which satisfy the criteria of a text table, **without** being declared as such. These text tables have been **ignored** up to now by the function "Add text columns".

In order to close (at least partially) this gap the following logic improvement has been implemented:

If for a given database table there exists another one named **xxxxT** (where **xxxx** is the name of the main table) which has exactly the **same key fields** as the main table **plus a language field**, then this second table is considered to be a text table, even if it is **not explicitly** declared as such.

An appropriate example will illustrate what we are talking about.

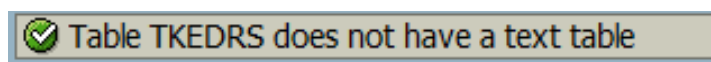
Let us consider database table **TKEDRS** (here shown in transaction SE11):



The screenshot shows the SAP Dictionary: Display Table for table TKEDRS. The table is active and has the short description "Characteristic Derivation: Steps". The Fields tab is selected, showing a list of fields with their properties.

Field	Key	Initi	Data element	Data Ty	Length	Decim	Short Description	Group
MANDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3		Client	
APPLCLASS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	APPLCLASS	CHAR	4		Application class for DD objects (not used)	
SUBCLASS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	SUBCLASS	CHAR	2		Subgroup by application class	
KEDRSTRATID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	KEDRSTRATID	CHAR	4		Strategy ID	
KEDRENV	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	KEDRENV	CHAR	8		Environment for strategy	
STEP_NO	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	KEDRSEQNO	NUMC	4		Characteristic derivation: sequential number	
METHOD	<input type="checkbox"/>	<input type="checkbox"/>	KEDRMETHOD	CHAR	8		Type of Derivation Step	
PARAM_1	<input type="checkbox"/>	<input type="checkbox"/>	KEDRPARAM	CHAR	32		Characteristic derivation: parameters	
PARAM_2	<input type="checkbox"/>	<input type="checkbox"/>	KEDRPARAM	CHAR	32		Characteristic derivation: parameters	

If we call function *Goto → Text Table* we obtain the following message:



There exists, however, the following **non-declared** text table **TKEDRST**:

*Dictionary: Display Table*

Transp. Table: TKEDRST Active  
 Short Description: Characteristic Derivation: Steps, Texts

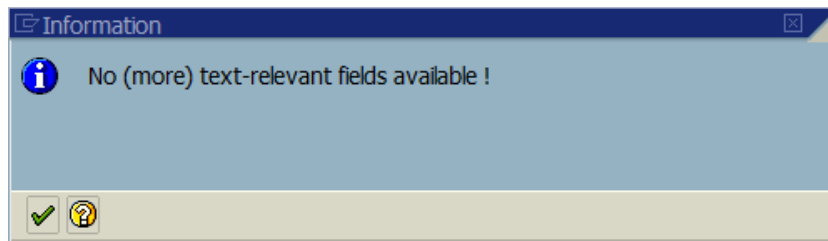
Attributes | Delivery and Maintenance | Fields | Entry help/check | Currency/Quantity Fields

Srch Help | Predefined Type | 1 / 8

Field	Key	Initi.	Data element	Data Ty.	Length	Decim.	Short Description	Group
MANDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3		Client	
APPLCLASS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	APPLCLASS	CHAR	4		Application class for DD objects (not used)	
SUBCLASS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	SUBCLASS	CHAR	2		Subgroup by application class	
KEDRSTRATID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	KEDRSTRATID	CHAR	4		Strategy ID	
KEDRENV	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	KEDRENV	CHAR	8		Environment for strategy	
STEP_NO	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	KEDRSEONO	NUMC	4		Characteristic derivation: sequential number	
LANG	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	DDLANGUAGE	LAWS	1		Language Key	
TEXT	<input type="checkbox"/>	<input type="checkbox"/>	KEDRSTEXT	CHAR	60		Derivation step description	

It exhibits the same key fields as TKEDRS plus the language field **LANG**. Its name is **TKEDRS plus T** at the end. It satisfies thus all our conditions for an unofficial text table.

In a previous SE16XXL version function "**Add text columns**" would have returned the following message for column STEP\_NO:



The same function **now returns** the result list enriched with the descriptive texts:


*SE16XXL - Table TKEDRS - 349 entries selected*

Table TKEDRS - Characteristic Derivation: Steps

MANDT	APPLCLASS	SUBCLASS	KEDRSTRATID	KEDRENV	STEP_NO	V~STEP_NO_TXT	METHOD	PARAM_1	
<input type="checkbox"/>	800	FILA	01	NDUR	NDUR	1	Derivation of Useful Life	DRULE	FILADTF730082
<input type="checkbox"/>	800	FM	01	FMOA	SAP	1	Commitment Item from Account	PERFORM	COMMIT_ITEM_FROM_AC
<input type="checkbox"/>	800	FM	01	FMOA	SAP	2	Cost Element to Commitment Item	DRULE	FMDERIVE001
<input type="checkbox"/>	800	FM	01	FMOA	SAP	3	Funds Center from Commitment Item Master Data	PERFORM	FUND_CENTER_FROM_CC
<input type="checkbox"/>	800	FOTP	01	BASE	FILA	1	One Time Postings Acc.Princ. 01	DRULE	FOTPDTF730088
<input type="checkbox"/>	800	FOTP	01	BASE	FILA	2	One Time Postings Acc.Princ. 60	DRULE	FOTPDTF730089
<input type="checkbox"/>	800	KC	01	BP01	880	1	Revaluation 3	DRULE	K9RID3800022
<input type="checkbox"/>	800	KC	01	BP01	880	2	Derivation Rule: Revaluation No. 1	DRULE	K9RID3800016
<input type="checkbox"/>	800	KC	01	BP02	880	1	Derivation Rule: Distribution No. 1	DRULE	K9RID3800017
<input type="checkbox"/>	800	KC	01	BP03	880	1	Derivation Rule: Forecast No. 1	DRULE	K9RID3800018
<input type="checkbox"/>	800	KE	01	DERI	8500	1	Division from product	TABLE	MARA
<input type="checkbox"/>	800	KE	01	DERI	8500	2	Company code from plant	PERFORM	WERKS—BUKRS
<input type="checkbox"/>	800	KE	01	DERI	8500	3	Company code from sales organization	TABLE	TVKO

[Back to the top](#)

## NUMC Fields as Sort Criteria for Subtotals

Normally columns of type **NUMC** cannot be summed up in ALV (function ) .

A special setting is necessary when calling ALV in order to achieve this.


With SE16XXL version **2.1L** this setting was activated permanently. This means that up to now all fields of type **NUMC** could be summed up in the result list.

It turned out, however, that this setting also has a drawback. If a **NUMC** field is part of the sort criteria, no subtotals can be defined at this level. And since in most cases it does not make sense to sum up **NUMC** values, this is indeed a big drawback.

All this sounds quite theoretical, so a suitable example might help understand the problem.

Suppose we have a result list of table **BKPF** joined with **BSEG**, conveniently sorted:



*SE16XXL - inner join - 4042 resulting rows*

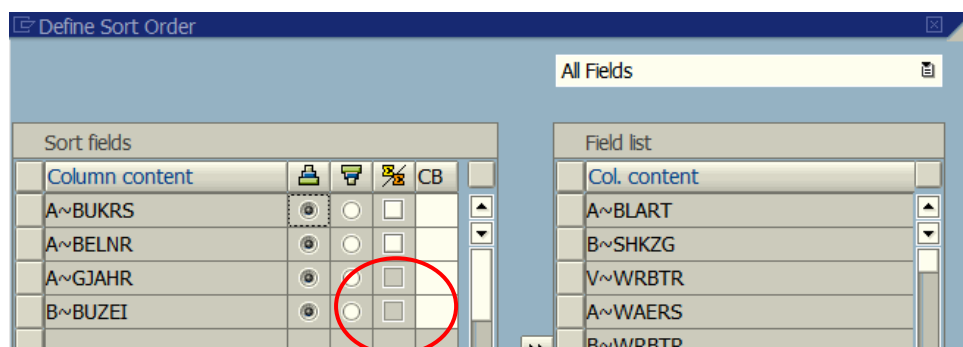



Join of BKPF(A) and BSEG(B)

A~BUKRS	A~BELNR	A~GJAHR	B~BUZEI	A~BLART	B~SHKZG	V~WRBTR	A~WAERS
<input type="checkbox"/> 1000	4900000000	2010	1 WA	S		632,50-	EUR
<input type="checkbox"/> 1000	4900000000	2010	2 WA	H		632,50	EUR
<input type="checkbox"/> 1000	4900000000	2012	1 WA	H		10,50	EUR
<input type="checkbox"/> 1000	4900000000	2012	2 WA	S		10,50-	EUR
<input type="checkbox"/> 1000	4900000000	2012	3 WA	H		10,50	EUR
<input type="checkbox"/> 1000	4900000000	2012	4 WA	S		10,50-	EUR


We would like to sum the column **V~WRBTR** and build subtotals at the level of the header record, i.e. for each combination of **A~BUKRS**, **A~BELNR** and **A~GJAHR**.

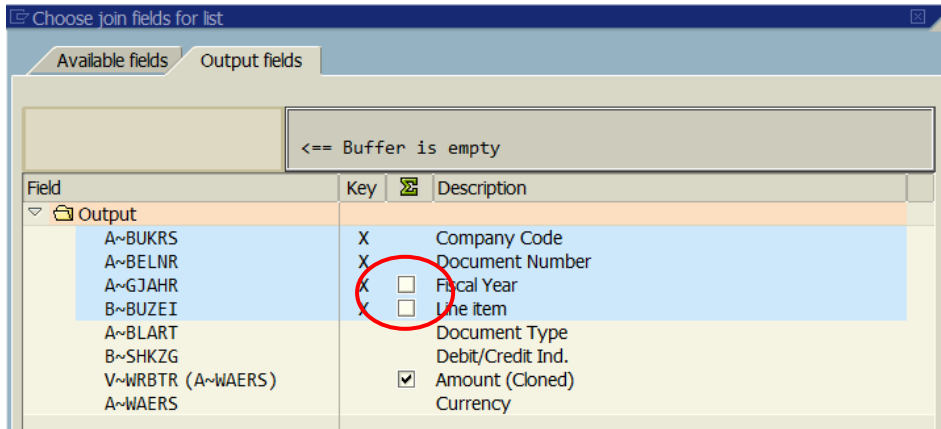
We start by marking column **V~WRBTR** and pressing  to get the total. As soon as the function  shows up on the application toolbar we call it in order to define the subtotals. Up to now the dialog window would have had the following appearance:




The option for subtotal  **could not be chosen** for the last two sort criteria. This was due to the fact that both **GJAHR** and **BUZEI** are of type **NUMC**.

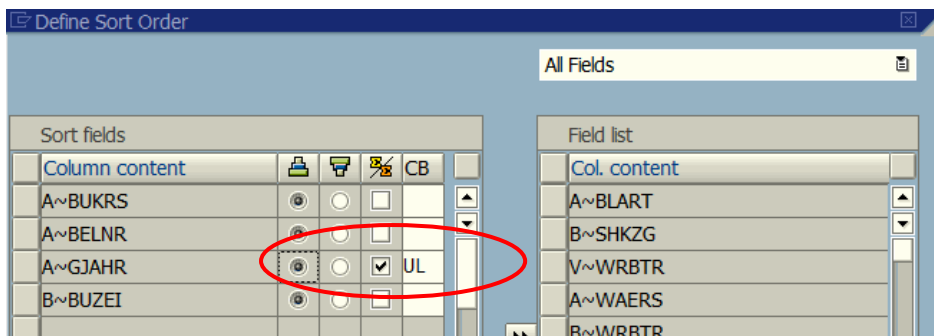


Starting with the current SE16XXL version, the logic of NUMC fields has been revised. From now on NUMC fields are normally non-summable. They may, however, be individually marked as summable. This is done in the unified field selector (function ) , on the "Output fields" tab:



In our example we leave the option unchecked.

When we call the subtotal function () , we observe that the corresponding options are now available:



The final result list is as follows:

**SE16XXL - inner join - 4042 resulting rows**

Join of BKPF(A) and BSEG(B)

A~BUKRS	A~BELNR	A~GJAHR	B~BUZEI	A~BLART	B~SHKZG	V~WRBTR	A~WAERS
<input type="checkbox"/> 1000	4900000002	2015	1	WA	H	3.600,00	EUR
<input type="checkbox"/> 1000	4900000002	2015	2	WA	S	3.600,00-	EUR
<input type="checkbox"/> 1000	4900000002	2015	3	WA	H	3.900,00	EUR
<input type="checkbox"/> 1000	4900000002	2015	4	WA	S	3.900,00-	EUR
* 1000	4900000002	2015				0,00	EUR
<input type="checkbox"/> 1000	4900000003	2012	1	WA	H	810,26	EUR
<input type="checkbox"/> 1000	4900000003	2012	2	WA	S	810,26-	EUR
* 1000	4900000003	2012				0,00	EUR
<input type="checkbox"/> 1000	4900000003	2015	1	WA	H	2.000.000,00	EUR

[Back to the top](#)

## Other Minor Improvements

### Improved Formula Debugging

It has always been possible to **debug** a formula. In the Formula Editor there is a menu function *Formula* → *Debug formula*. When this function is called before activating the formula, a **breakpoint** is set at the statement where the formula is performed – provided that the user has debugging authorization and the system is correspondingly configured.

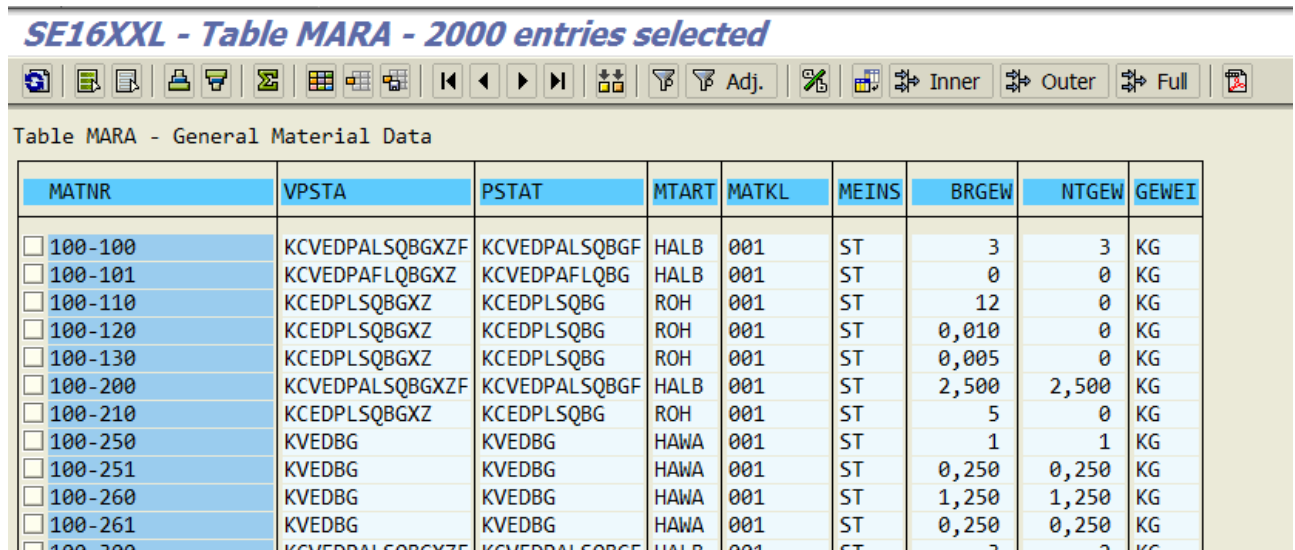
In the generated dynamic coding the formula statements are placed in a **TRY – ENDTRY** block to catch any catchable exceptions. Up to now only **an error flag** was set when an exception was triggered, and all involved formula fields reset to initial. This worked well with small formulas consisting of a few statements.

If, however, formulas are defined that span several pages of coding, that single error flag would not tell much about the statement responsible for the exception. For this reason **two new variables** have been added to the TRY block logic: one contains the **source code line** where the exception took place, the other contains a **message** related to the error situation.


A short example will illustrate what has been discussed.

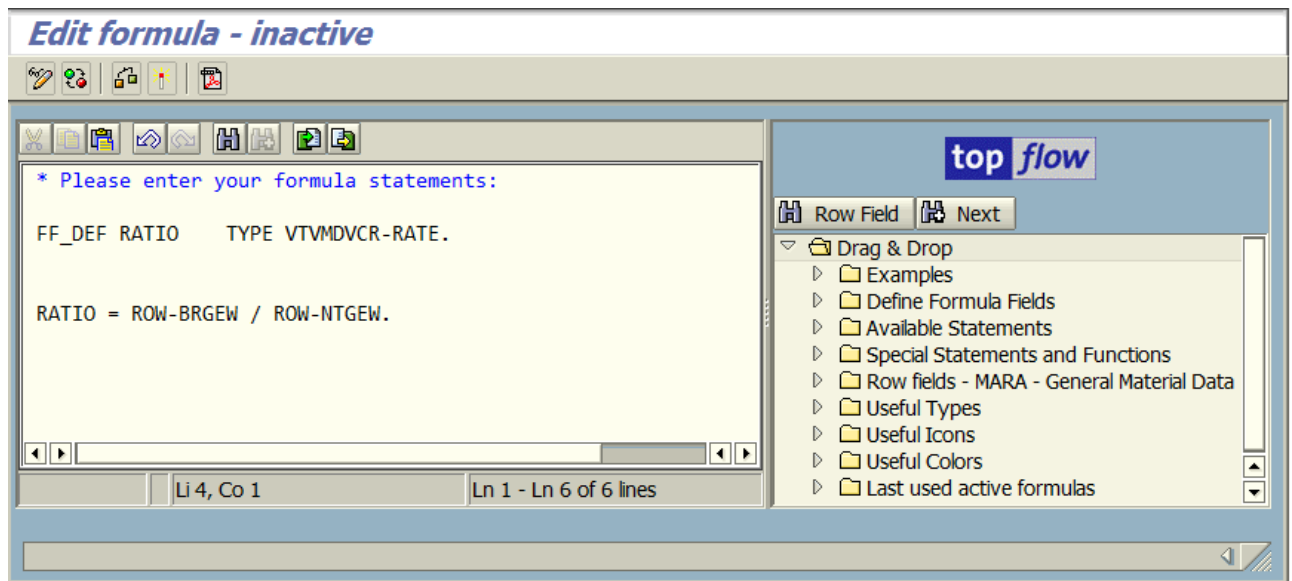
We begin with a result list containing entries of MARA:

*SE16XXL - Table MARA - 2000 entries selected*

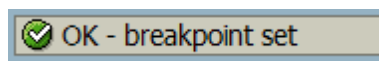


MATNR	VPSTA	PSTAT	MTART	MATKL	MEINS	BRGEW	NTGEW	GEWEI
<input type="checkbox"/> 100-100	KCVEDPALSQBGXZF	KCVEDPALSQBGF	HALB	001	ST	3	3	KG
<input type="checkbox"/> 100-101	KCVEDPAFLQBGXZ	KCVEDPAFLQBG	HALB	001	ST	0	0	KG
<input type="checkbox"/> 100-110	KCEDPLSQBGXZ	KCEDPLSQBG	ROH	001	ST	12	0	KG
<input type="checkbox"/> 100-120	KCEDPLSQBGXZ	KCEDPLSQBG	ROH	001	ST	0,010	0	KG
<input type="checkbox"/> 100-130	KCEDPLSQBGXZ	KCEDPLSQBG	ROH	001	ST	0,005	0	KG
<input type="checkbox"/> 100-200	KCVEDPALSQBGXZF	KCVEDPALSQBGF	HALB	001	ST	2,500	2,500	KG
<input type="checkbox"/> 100-210	KCEDPLSQBGXZ	KCEDPLSQBG	ROH	001	ST	5	0	KG
<input type="checkbox"/> 100-250	KVEDBG	KVEDBG	HAWA	001	ST	1	1	KG
<input type="checkbox"/> 100-251	KVEDBG	KVEDBG	HAWA	001	ST	0,250	0,250	KG
<input type="checkbox"/> 100-260	KVEDBG	KVEDBG	HAWA	001	ST	1,250	1,250	KG
<input type="checkbox"/> 100-261	KVEDBG	KVEDBG	HAWA	001	ST	0,250	0,250	KG
<input type="checkbox"/> 100-300	KCVEDPALSORCYZE	KCVEDPALSORCFE	HALB	001	ST	3	3	KG

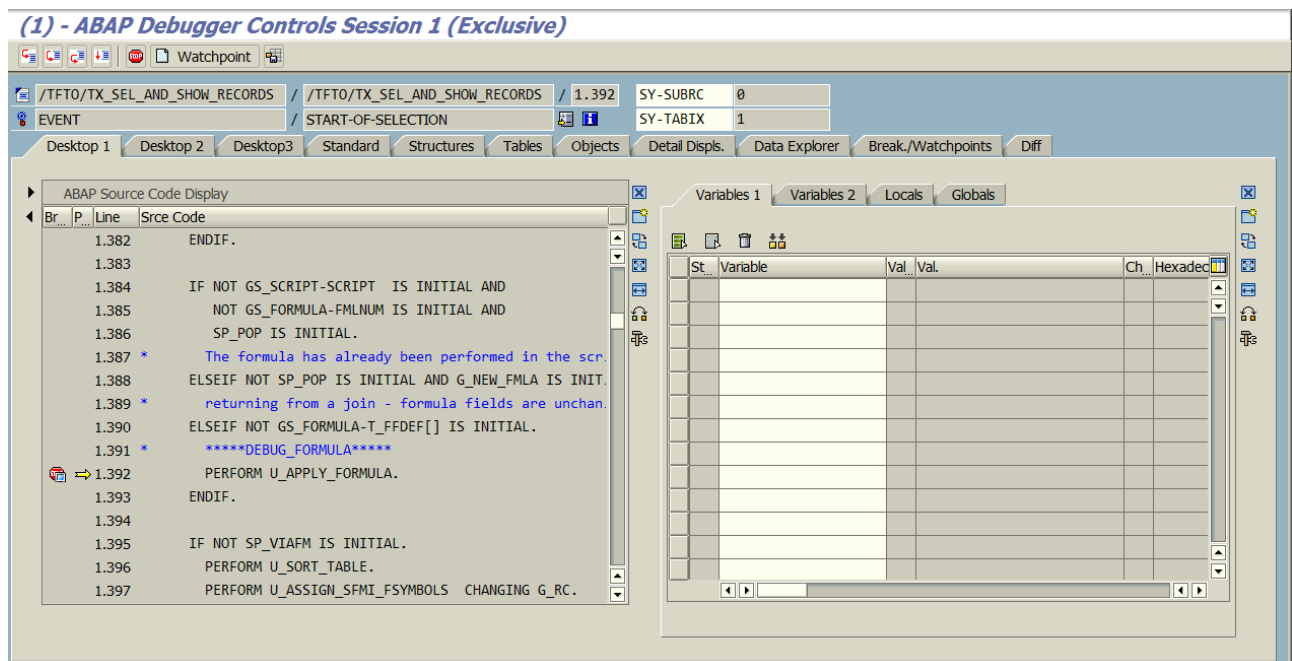
Then we define a formula () to calculate the ratio of Gross Weight to Net Weight. Clearly in the real world the formula would be a lot more complicated.



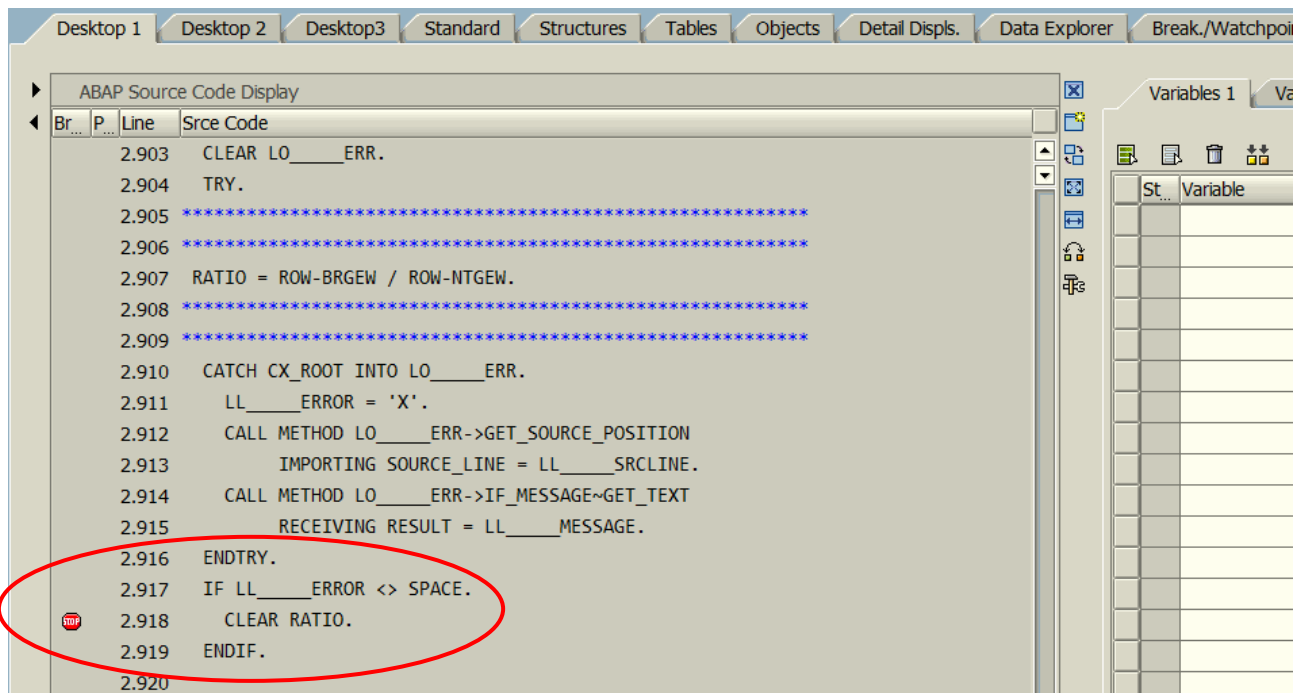
Before activating the formula we call menu function *Formula* → *Debug formula*. The program responds with the following message:



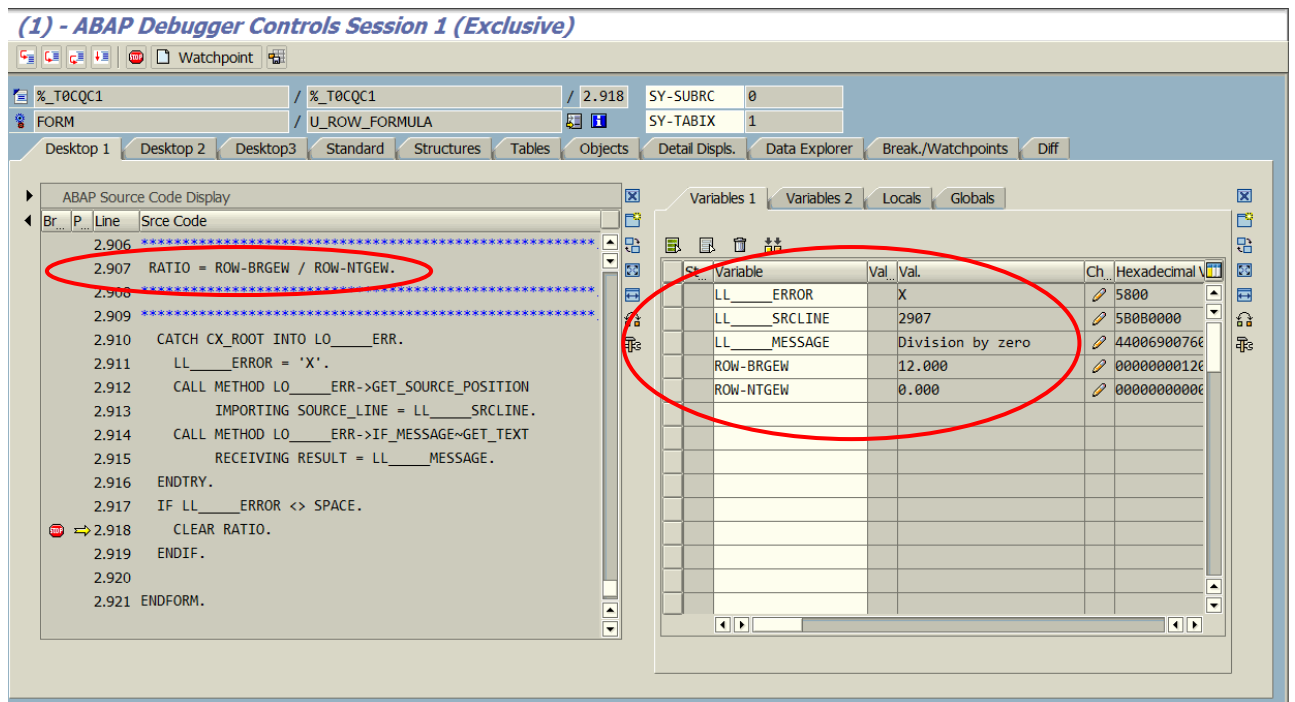
After activating the formula the program switches to debugging mode:



We proceed **step by step** until we come to the formula itself. There we set a breakpoint at the end of the TRY block:




At this point we press F8 (continue) and see what happens:



From the help variables it is easy to see that the exception (Division by zero) took place at line number **2907**. In case of a large formula this would be a great help.

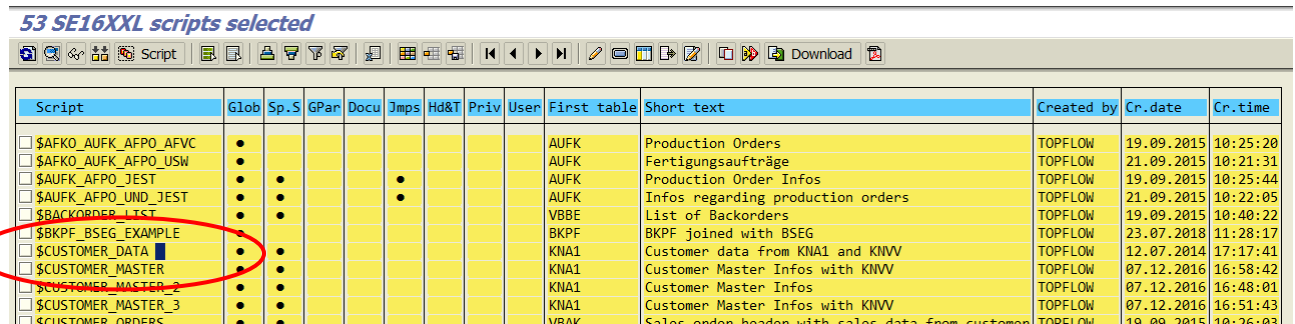
## Simplified Download of a Single Script

Up to now the Download function () of the Script Catalog required that the scripts to be downloaded – and be it only one – be selected (i.e. marked):

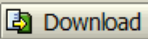
 Please select the scripts to be downloaded !

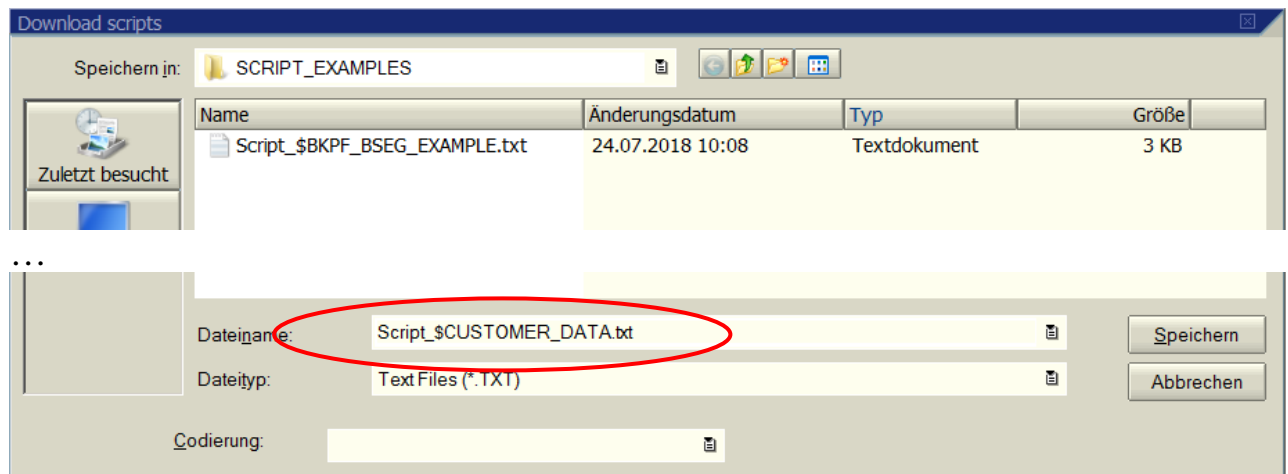
This may seem a trifle if done once, but if you are used to download single scripts very often, it becomes annoying. For this reason it is now possible to download a single script **without having to mark it**. It now suffices to place the **cursor** on it:

53 SE16XXL scripts selected



Script	Glob	Sp.S	GPar	Docu	Jmps	Hd&T	Priv	User	First table	Short text	Created by	Cr.date	Cr.time
<input type="checkbox"/> \$AFKO_AUFG_AFPO_AFVC	•								AUFG	Production Orders	TOPFLOW	19.09.2015	10:25:20
<input type="checkbox"/> \$AFKO_AUFG_AFPO_USW	•								AUFG	Fertigungsaufträge	TOPFLOW	21.09.2015	10:21:31
<input type="checkbox"/> \$AUFG_AFPO_0EST	•				•				AUFG	Production Order Infos	TOPFLOW	19.09.2015	10:25:44
<input type="checkbox"/> \$AUFG_AFPO_UND_0EST	•	•							AUFG	Infos regarding production orders	TOPFLOW	21.09.2015	10:22:05
<input type="checkbox"/> \$BACKORDER_LIST	•	•							VBBE	List of Backorders	TOPFLOW	19.09.2015	10:40:22
<input type="checkbox"/> \$BKPF_BSEG_EXAMPLE	•	•							BKPF	BKPF joined with BSEG	TOPFLOW	23.07.2018	11:28:17
<input type="checkbox"/> \$CUSTOMER_DATA	•	•							KNA1	Customer data from KNA1 and KNWV	TOPFLOW	12.07.2014	17:17:41
<input type="checkbox"/> \$CUSTOMER_MASTER	•	•							KNA1	Customer Master Infos with KNWV	TOPFLOW	07.12.2016	16:58:42
<input type="checkbox"/> \$CUSTOMER_MASTER_2	•	•							KNA1	Customer Master Infos	TOPFLOW	07.12.2016	16:48:01
<input type="checkbox"/> \$CUSTOMER_MASTER_3	•	•							KNA1	Customer Master Infos with KNWV	TOPFLOW	07.12.2016	16:51:43
<input type="checkbox"/> \$CUSTOMER_ORDERS	•	•							VRK1	Sales order header with sales data from customer	TOPFLOW	19.09.2015	10:26:03

After pressing , the file selector shows up. The name of the file is in this case **derived** from the name of the script:



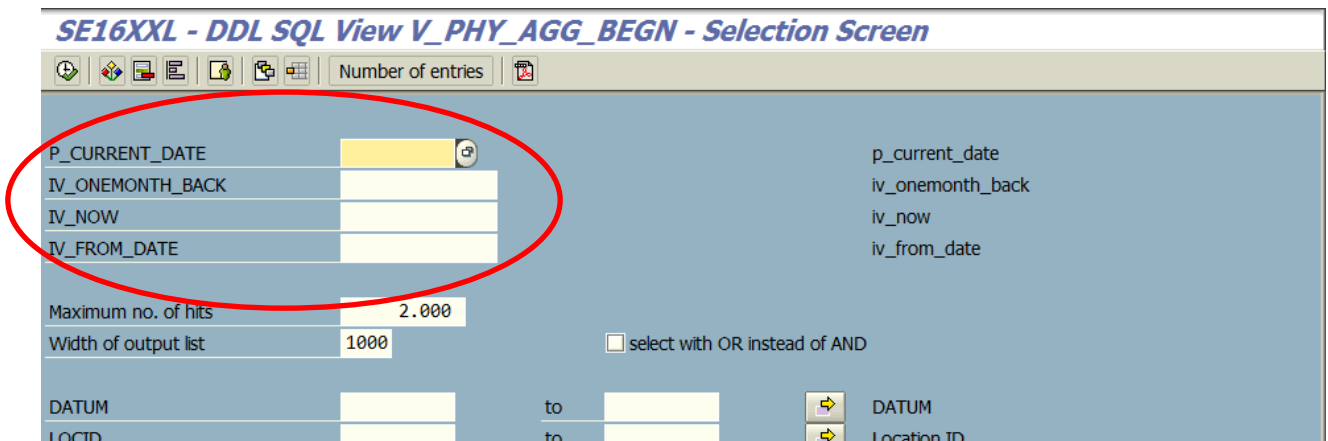
[Back to the top](#)

## CDS Views with Parameters

The so-called **CDS views** – which in the Data Dictionary are displayed as *DDL SQL views* – have been introduced in one of the latest SAP versions. Most of these views could be read by means of SE16XXL like ordinary database views. Only the ones with **parameters** refused to be processed, because this kind of CDS view needs a special SQL syntax.

With version **3.3C** of SE16XXL this gap has been closed. CDS views with up to **30** parameters are now supported. The maximum of 30 should be more than sufficient (most CDS views have **only one or two** parameters).

On the SE16XXL selection screen the parameters are always placed **at the top**:



The screenshot shows the selection screen for the CDS view V\_PHY\_AGG\_BEGN. The parameters are listed at the top, with P\_CURRENT\_DATE, IV\_ONEMONTH\_BACK, IV\_NOW, and IV\_FROM\_DATE highlighted by a red oval. Below the parameters, there are fields for 'Maximum no. of hits' (2.000) and 'Width of output list' (1000). A checkbox labeled 'select with OR instead of AND' is also visible. At the bottom, there are fields for 'DATUM' and 'LOCTID' with 'to' indicators and arrows.

This has been done for clarity – the interpretation of the parameter values **does not depend** on the "select with OR instead of AND" option.

### NOTES:

- 1) There seems to be no standard behavior for parameters. Some are implicitly mandatory, others can be left unspecified. Apparently each CDS view has its own logic. The user will have to try out the best approach.
- 2) If a CDS view has a parameter associated with the **client**, this parameter is kept invisible and is always filled with the value of the current client (**SY-MANDT**).
- 3) All parameters are shown on the standard selection screen (except the client). Only on a special script selection screen can the visible parameters be chosen (at definition time).

Using CDS view **SADL\_V4\_REPO** we will now go through the various situations where selection parameters might show up.

## Standard Selection Screen

**SE16XXL - DDL SQL View SADL\_V4\_REPO - Selection Screen**

Number of entries

P\_LANGUAGE EN

Maximum no. of hits 2.000

Width of output list 1000  select with OR instead of AND

DDLNAME C\* to Name DDL Source

STRUCOBJECT to Object name

VIEWNAME to Object name

## Operations Log (Goto → Show operations log)

**SE16XXL - operations log**

List of the operations log:

Nr. 1 SELSCREEN First selection screen

SEL\_MODE=N - TABNAME=SADL\_V4\_REPO - SEL\_WITH\_OR= - ALIAS=A

Parameter values: P\_LANGUAGE = E

Select fields: DDLNAME STRUCOBJECT VIEWNAME CHANGED\_BY CHANGED\_DATE CHANGED\_TIME DESCRIPTION

Selection criteria: DDLNAME I CP C\*

Nr. 2 SELECT First selection

SEL\_MODE=N - TABNAME=SADL\_V4\_REPO - SEL\_WITH\_OR= - ALIAS=A

List fields: DDLNAME VIEWNAME DESCRIPTION

Parameter values: P\_LANGUAGE = E

Selection criteria: DDLNAME I CP C\*

Involved tables: SADL\_V4\_REPO

## Selection Criteria (List → Display selection criteria)

**Initial Selection Criteria**

Selection criteria

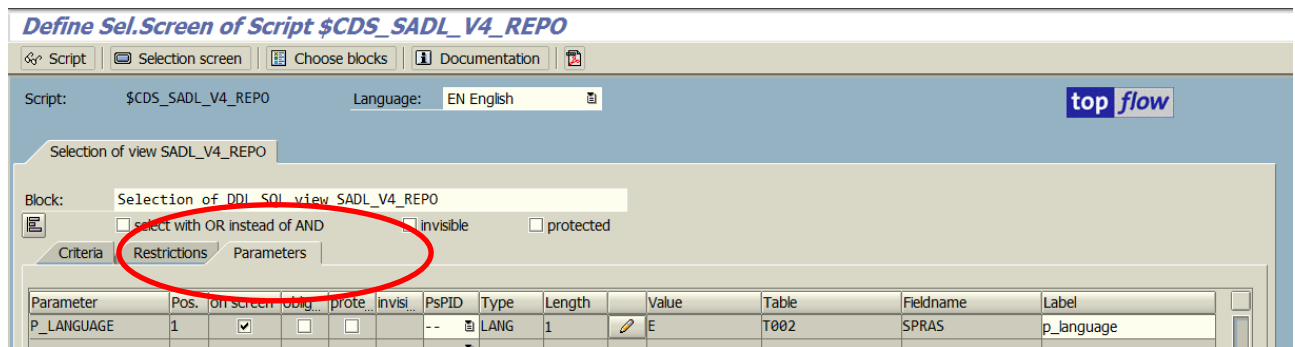
P\_LANGUAGE : E

Maximum no. of hits : 2.000

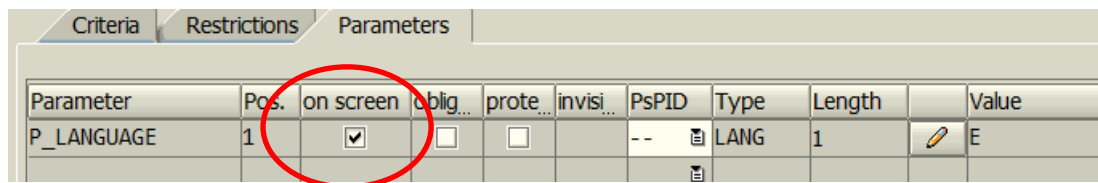
DDLNAME : [C\*]




## Definition of a Script Selection Screen

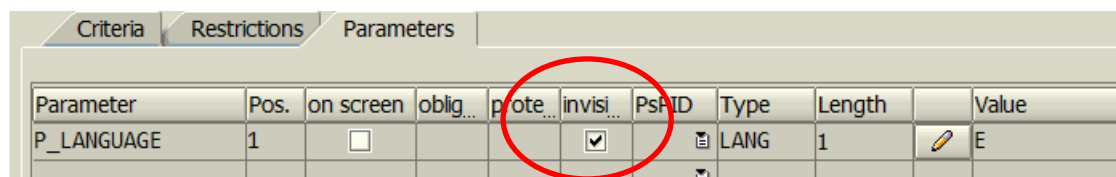



Enlarged the parameter definition is as follows:



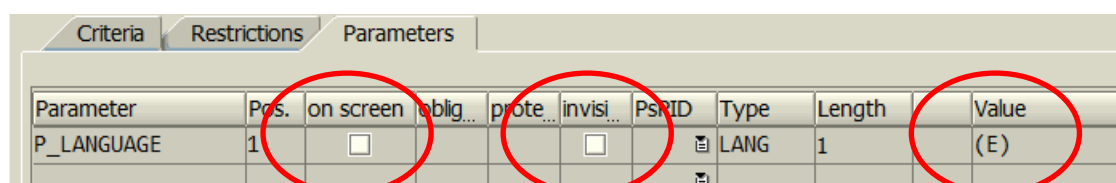
If the parameter is visible on the screen (option "**on screen**"), it can be set to "**obligatory**" or to "**protected**", just like the usual select-options. The value can be set by either pressing the  button or on the simulated selection screen. If any parameters are visible on the selection screen the maximum number of blocks drops from 12 to 8 – this is due to technical constraints.

A parameter can also be set to invisible:



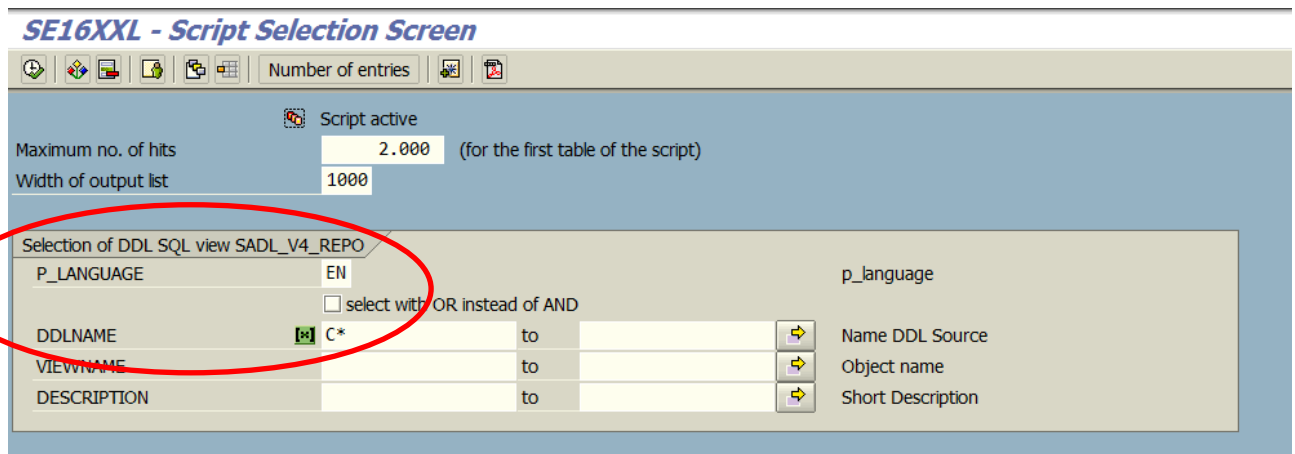
In this case the value can only be set by means of the  button.

If none of the above options is set, the parameter will **not show up** on the selection screen and **no value** will be assigned to the parameter at SELECT time:



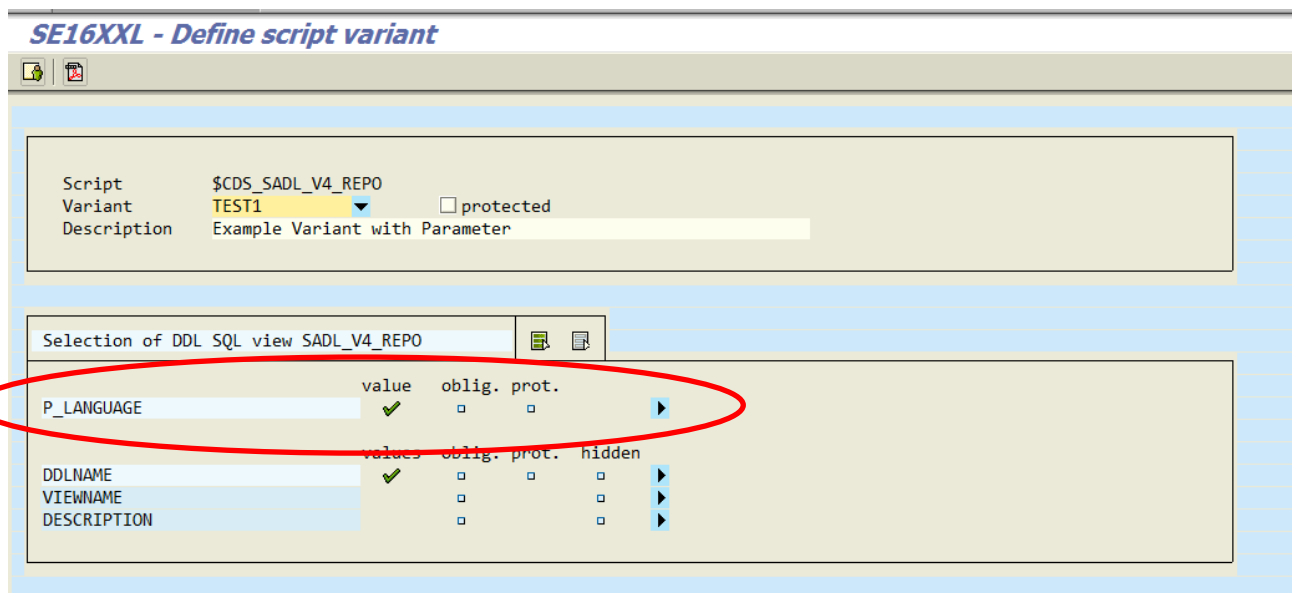
In this case the original value is shown in parentheses to signify that it will **not** be taken into account.

## Script Selection Screen



Notice that the parameter is **at the top** of the selection block and that the option "select with OR instead of AND" is placed beneath it.

## Definition of a Script Variant



As with normal select-options, it is possible to assign a variable to a parameter (▶).

[Back to the top](#)

## Very long Field Names supported

Until recently only DDIC structures could have fields with names up to 30 characters long – for fields of transparent tables and views the maximum allowed length was **16 characters** – in my opinion more than sufficient.

Beginning with one of the latest SAP versions this limit has been lifted – it is now possible to define transparent tables and database views with **field names up to 30 characters long**.

Just to give an idea of what this looks like, a couple of examples will be presented:

- 123456789012345678901234567890 (characters)
- BINDINGPERIODVALIDITYSTARTDATE
- CUSTOMERACCOUNTASSIGNMENTGROUP
- CUSTOMERPURCHASEORDERSUPPLEMNT
- OVERALLPRICINGINCOMPLETIONSTS
- OVERALLSDDOCUMENTREJECTIONSTS

The above are **real** field names, taken from view **ISDSALESDOC**.

It is easy to see from the above examples that, given a maximum length, there will always be a situation where a name has to be shortened (in the last two examples STS stands for STATUS). It is also apparent that to concatenate together several words without a separating character does not necessarily lead to more clarity.

So this is the new situation with transparent tables and database views in SAP.

The reader will ask: **what's the problem?** The data element **FIELDNAME** is already 30 characters long, so the above names, even if very long, should easily fit.

This is true. Unfortunately SE16XXL also deals with joins. In a join list, each table is represented by an alias, and its fields are prefixed with it, as in **A~MATNR**. Internally this is implemented as **A-MATNR**, representing field **MATNR** of substructure **A**. Up to now this concatenation led to names 18 characters long. But if the field itself is 30 characters long, the concatenation **needs 32 characters**.

*"No problem, substitute data element **FIELDNAME** with a 32 character data element and all will be well"* conjectures the interested reader.

**Unfortunately not.** There are several constraints that cannot be overcome. For example in the **ALV field catalog** the field name is only 30 characters long, and it cannot be changed, being an official part of the ALV interface.

## How the problem has been solved

The problem has been solved in the following way:

All field name information stored in the database (scripts, script variants, background results etc.) is kept in its original length, i.e. up to 32 characters long.

When a script (or something else) is loaded, the field names longer than 28 characters are shortened dynamically. The necessary conversion tables are kept in virtual memory. The shortened field names are then used internally where necessary (for example at the ALV interface).

On the outside the user does not notice anything of this complicated game. There is only one instance where the shortened names are visible, i.e. on a **filter popup** window with a join-list. Such a filter window makes use of a standard SAP functionality and the available fields are only 30 characters long. In all other situations the original field names are shown.

## Additional Adjustments

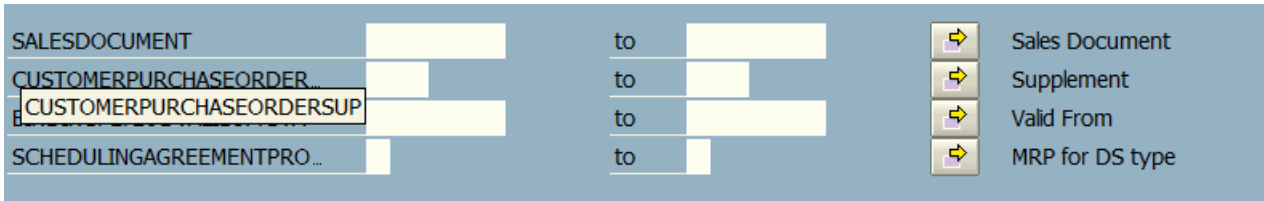
Apart from the internal conversions, several other adjustments have become necessary (or at least advisable) due to the exceedingly long field names, owing to the fact that up to now all dialogs were trimmed for a maximum length of 18 characters. Whenever possible the visible length has been increased, either absolutely or depending on the maximum length of the fields being considered.

In the following pages some examples of these visible adjustments will be given.

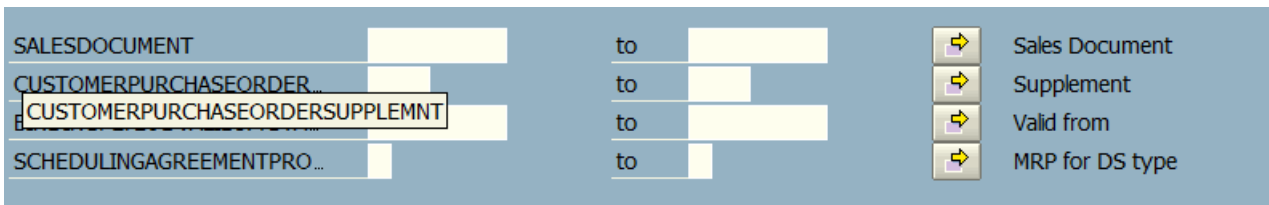
### Standard Selection Screen

Due to technical constraints there is not enough space on the selection screen for displaying the field names in their full length. But hovering with the mouse over a given field will now display the complete name:

Old version (the name is truncated):



New version (the name appears in full):



### Detailed View of a List Entry

Old version:

Entry of view ISDSALESDOC - Sales Document		
MANDT	'800'	Client
SALESDOCUMENT	'0000000012'	Sales Document
SDDOCUMENTCATEGORY	'C'	SD Document Category (Long)
SALESDOCUMENTTYPE	'TA'	Sales Document Type
SALESDOCUMENTPROCESSINGTYPE	' '	SD document indicator
CREATEDBYUSER	'WIESE'	Name of Person who Created the Object
CREATIONDATE	'04.05.2016'	Date on which the record was created
CREATIONTIME	'17:33:22'	Entry time
LASTCHANGEDATE	'00.00.0000'	Date of Last Change
LASTCHANGEDATETIME	'20.160.822.144.938,0240000'	UTC Time Stamp in Long Form (YYYYMMDDhhmmssmmuuun)
SALESORGANIZATION	'0001'	Sales Organization
DISTRIBUTIONCHANNEL	'01'	Distribution Channel
ORGANIZATIONDIVISION	'01'	Division
SALESGROUP	' '	Sales group
SALESOFFICE	' '	Sales office
SOLDTOPARTY	'0000000002'	Sold-to party
CUSTOMERGROUP	' '	Customer Group
ADDITIONALCUSTOMERGROUP1	' '	Customer group 1
ADDITIONALCUSTOMERGROUP2	' '	Customer group 2
ADDITIONALCUSTOMERGROUP3	' '	Customer group 3

New version:

Entry of view ISDSALESDOC - Sales Document

MANDT	'800'	Client
SALESDOCUMENT	'0000000012'	Sales Document
SDDOCUMENTCATEGORY	'C'	SD document category
SALESDOCUMENTTYPE	'TA'	Sales Document Type
CREATEDBYUSER	'WIESE'	Name of Person Who Created the Object
CREATIONDATE	'28.07.2016'	Date on Which Record Was Created
CREATIONTIME	'16:31:09'	Entry time
LASTCHANGEDATE	'00.00.0000'	Date of Last Change
SALESORGANIZATION	'0001'	Sales Organization
DISTRIBUTIONCHANNEL	'01'	Distribution Channel
ORGANIZATIONDIVISION	'01'	Division
SALESGROUP	' '	Sales Group
SALESOFFICE	' '	Sales Office
SOLDTOPARTY	'0000000001'	Sold-To Party
CUSTOMERGROUP	' '	Customer group
ADDITIONALCUSTOMERGROUP1	' '	Customer group 1
ADDITIONALCUSTOMERGROUP2	' '	Customer group 2
ADDITIONALCUSTOMERGROUP3	' '	Customer group 3

### Selection of List Fields

Old version (name column not wide enough):

Field	Key	Description
<input checked="" type="checkbox"/> CUSTOMERREBATEAGREEMEN	X	Agreement
<input checked="" type="checkbox"/> SALESDOCUMENTDATE	X	Document Date
<input checked="" type="checkbox"/> SERVICESRENDEREDDATE	X	Serv. Rendered Date
<input checked="" type="checkbox"/> SDDOCUMENTREASON	X	Order Reason
<input checked="" type="checkbox"/> PURCHASEORDERBYCUSTOMER	X	Customer Reference
<input checked="" type="checkbox"/> SDDOCUMENTCOLLECTIVENUM	X	Collective Number
<input checked="" type="checkbox"/> CUSTOMERPURCHASEORDERT	X	Purchase Order Type
<input checked="" type="checkbox"/> CUSTOMERPURCHASEORDERD	X	Customer Ref. Date
<input checked="" type="checkbox"/> CUSTOMERPURCHASEORDERS	X	Supplement
<input checked="" type="checkbox"/> SALESDISTRICT	X	Sales District


New version:

Field	Key	Description
<input type="checkbox"/> CUSTOMERREBATEAGREEMENT	X	Agreement
<input type="checkbox"/> SALESDOCUMENTDATE	X	Document Date
<input type="checkbox"/> SERVICESRENDEREDDATE	X	Serv. rendered date
<input type="checkbox"/> SDDOCUMENTREASON	X	Order reason
<input type="checkbox"/> PURCHASEORDERBYCUSTOMER	X	PO Number
<input type="checkbox"/> SDDOCUMENTCOLLECTIVENUMBER	X	Collective number
<input type="checkbox"/> CUSTOMERPURCHASEORDERTYPE	X	Purchase order type
<input type="checkbox"/> CUSTOMERPURCHASEORDERDATE	X	Purchase Order Date
<input type="checkbox"/> CUSTOMERPURCHASEORDERSUPPLEMNT	X	Supplement
<input type="checkbox"/> SALESDISTRICT	X	Sales district

## Script Selection Screen Definition

Old version (columns not wide enough):

*Define Sel.Screen of Script CDS\_ISDSALESDOC*

Script: CDS\_ISDSALESDOC Language: EN English 

Selection of table ISDSALESDOC

Block: Selection of table ISDSALESDOC


select with OR instead of AND  invisible  protected

Criteria Restrictions

Criterion	Pos.	Type	Length	Table	Fieldname	Values	Label
SALESDOCUMENT	1	CHAR	10	ISDSALESDOC	SALESDOCUMENT	<input type="checkbox"/>	Sales Document
CUSTOMERPURCHAS...	2	CHAR	4	ISDSALESDOC	CUSTOMERPURCHAS...	<input type="checkbox"/>	Supplement
BINDINGPERIODVA...	3	DATS	8	ISDSALESDOC	BINDINGPERIODVA...	<input type="checkbox"/>	Valid From
BINDINGPERIODVA...	4	DATS	8	ISDSALESDOC	BINDINGPERIODVA...	<input type="checkbox"/>	Valid To
HDRORDERPROBABI...	5	NUMC	3	ISDSALESDOC	HDRORDERPROBABI...	<input type="checkbox"/>	Probability
SCHEDULINGAGREE...	6	CHAR	1	ISDSALESDOC	SCHEDULINGAGREE...	<input type="checkbox"/>	MRP for DS type

New version:

*Define Sel.Screen of Script CDS\_ISDSALESDOC*

Script: CDS\_ISDSALESDOC Language: EN English 

Selection of view ISDSALESDOC

Block: Selection of DDL SQL view ISDSALESDOC

select with OR instead of AND  invisible  protected

Criteria Restrictions

Criterion	Pos.	Type	Length	Table	Fieldname	Values	Label
SALESDOCUMENT	1	CHAR	10	ISDSALESDOC	SALESDOCUMENT	<input type="checkbox"/>	Sales Document
CUSTOMERPURCHASEORDERSUPPLEMNT	2	CHAR	4	ISDSALESDOC	CUSTOMERPURCHASEORDERSUPPLEMNT	<input type="checkbox"/>	Supplement
BINDINGPERIODVALIDITYSTARTDATE	3	DATS	8	ISDSALESDOC	BINDINGPERIODVALIDITYSTARTDATE	<input type="checkbox"/>	Valid from
BINDINGPERIODVALIDITYENDDATE	4	DATS	8	ISDSALESDOC	BINDINGPERIODVALIDITYENDDATE	<input type="checkbox"/>	Valid to
HDRORDERPROBABILITYINPERCENT	5	NUMC	3	ISDSALESDOC	HDRORDERPROBABILITYINPERCENT	<input type="checkbox"/>	Probability
SCHEDULINGAGREEMENTPROFILECODE	6	CHAR	1	ISDSALESDOC	SCHEDULINGAGREEMENTPROFILECODE	<input type="checkbox"/>	MRP for DS type

These examples should suffice to give an idea of the adjustments that have been implemented.

[Back to the top](#)

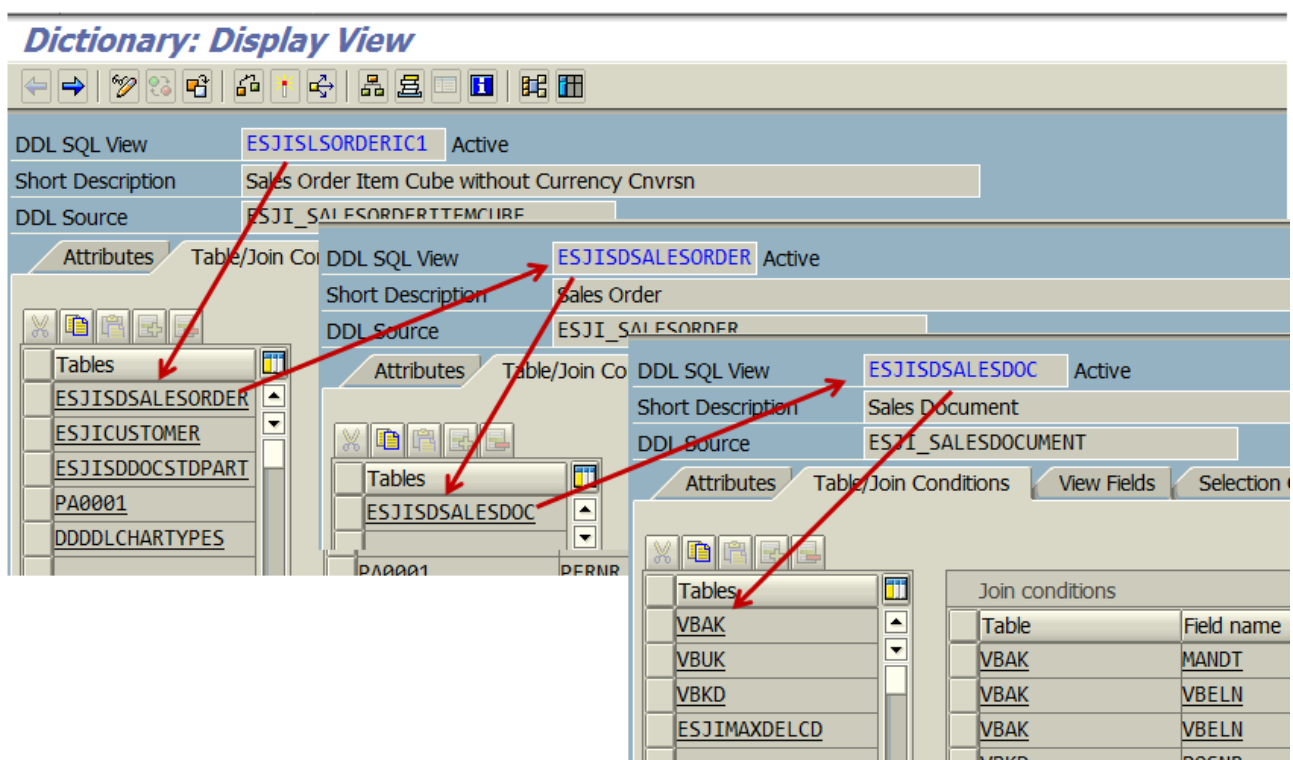


## Multilevel Views (Views based on Views) supported

Until recently a **database view** could only be based on a table of category **transparent, pool or cluster**, but not on another database view.

In the newest SAP versions this restriction **has been lifted**. In these SAP versions views exist, which are **based on other views**, in a recursive fashion. In some cases the real database table is found after **25 levels** of views, or even deeper.

For example:



The screenshot shows the SAP Dictionary 'Display View' interface. It displays a hierarchy of views and tables. Red arrows indicate the following dependencies:

- View **ESJISLSORDERIC1** (Short Description: Sales Order Item Cube without Currency Cnvrns) is based on view **ESJISDSALESORDER**.
- View **ESJISDSALESORDER** (Short Description: Sales Order) is based on view **ESJISDSALESDOC**.
- View **ESJISDSALESDOC** (Short Description: Sales Document) is based on database table **VBAK**.

The 'Join conditions' table for view **ESJISDSALESDOC** is also visible:

Table	Field name
VBAK	MANDT
VBAK	VBELN
VBAK	VBELN
VBKD	POSNR

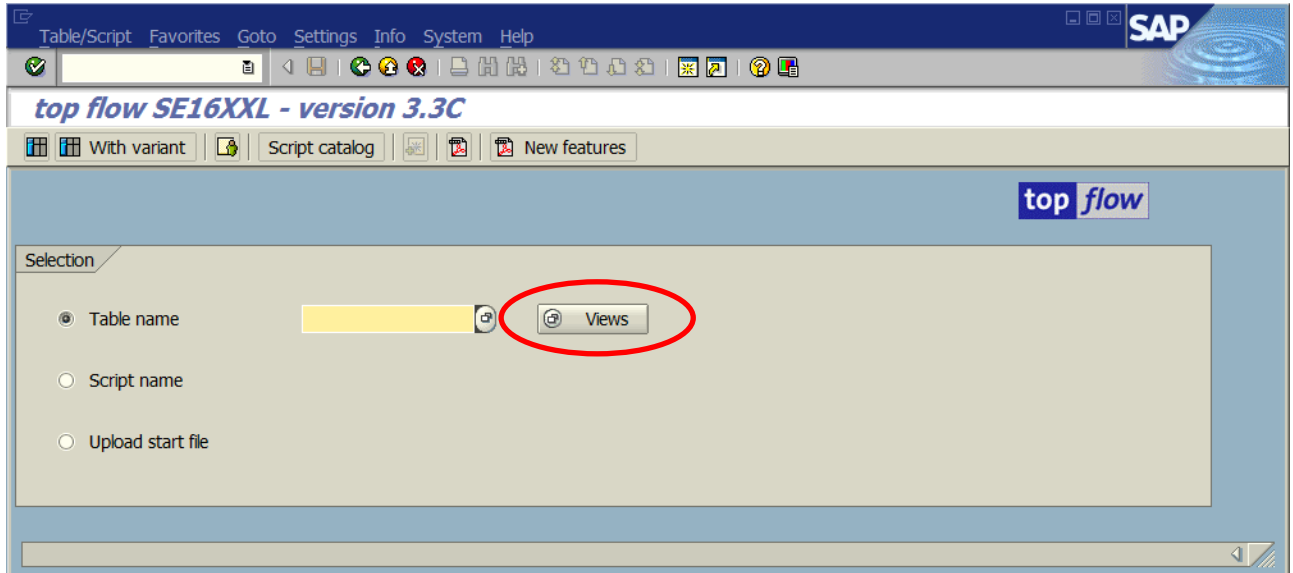
View **ESJISLSORDERIC1** is based on view **ESJISDSALESORDER** which is based on view **ESJISDSALESDOC** which is finally based on database table **VBAK**.

In SE16XXL the Table & Field Permissions are **based on real database tables** and not on views. In case of a view the necessary checks are carried out on underlying database tables. In order to find these database tables a **recursive search** is now performed for each view.

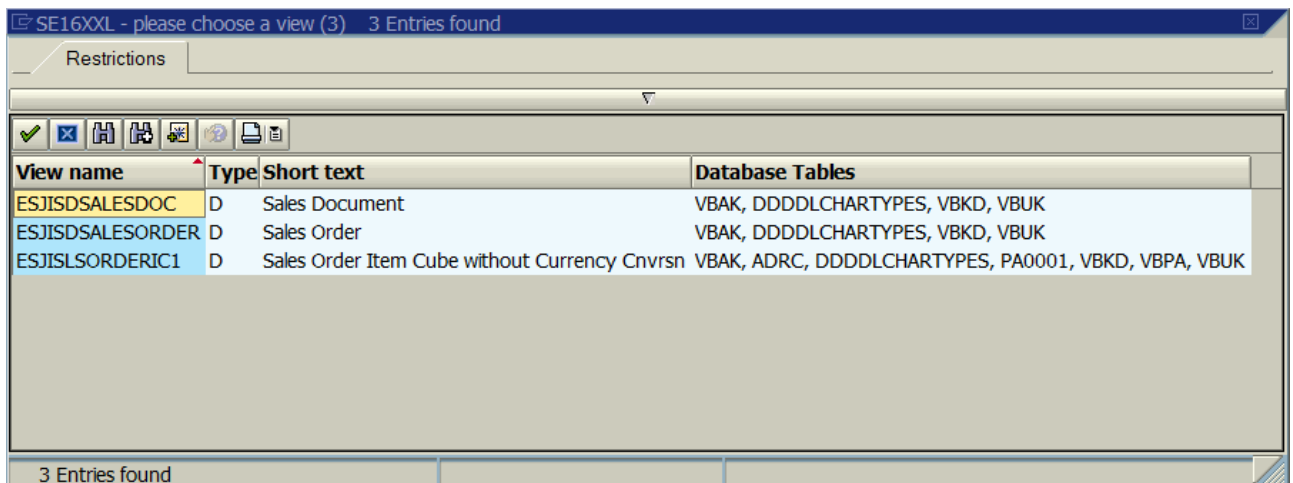
The same is true with the SE16XXL Authorization Checks at Record Level. They are also based on fields of real database tables. As for the permissions, the underlying database table fields are determined by means of a recursive search and then used for the authorization checks.

## F4 Help for Views

On the first screen of SE16XXL there is a special button for the F4 help for views:



The result of a search shows the views together with the underlying database tables. In our example we would get a result like the following:



The screenshot shows the SE16XXL search results window. The title bar reads 'SE16XXL - please choose a view (3) 3 Entries found'. The window has a 'Restrictions' field and a toolbar. The main area contains a table with the following data:

View name	Type	Short text	Database Tables
ESJISDSALESDOC	D	Sales Document	VBAK, DDDLCHARTYPES, VBKD, VBUK
ESJISDSALESORDER	D	Sales Order	VBAK, DDDLCHARTYPES, VBKD, VBUK
ESJISLSORDERIC1	D	Sales Order Item Cube without Currency Cnvrns	VBAK, ADRC, DDDLCHARTYPES, PA0001, VBKD, VBPA, VBUK

The status bar at the bottom of the window shows '3 Entries found'.

The intermediate views of a view do not show up in the column "Database Tables". As in the case of the permissions and authorizations, the underlying database tables are determined by means of a recursive search.

[Back to the top](#)

## New Pseudo Tables \$JEST and \$JESTC

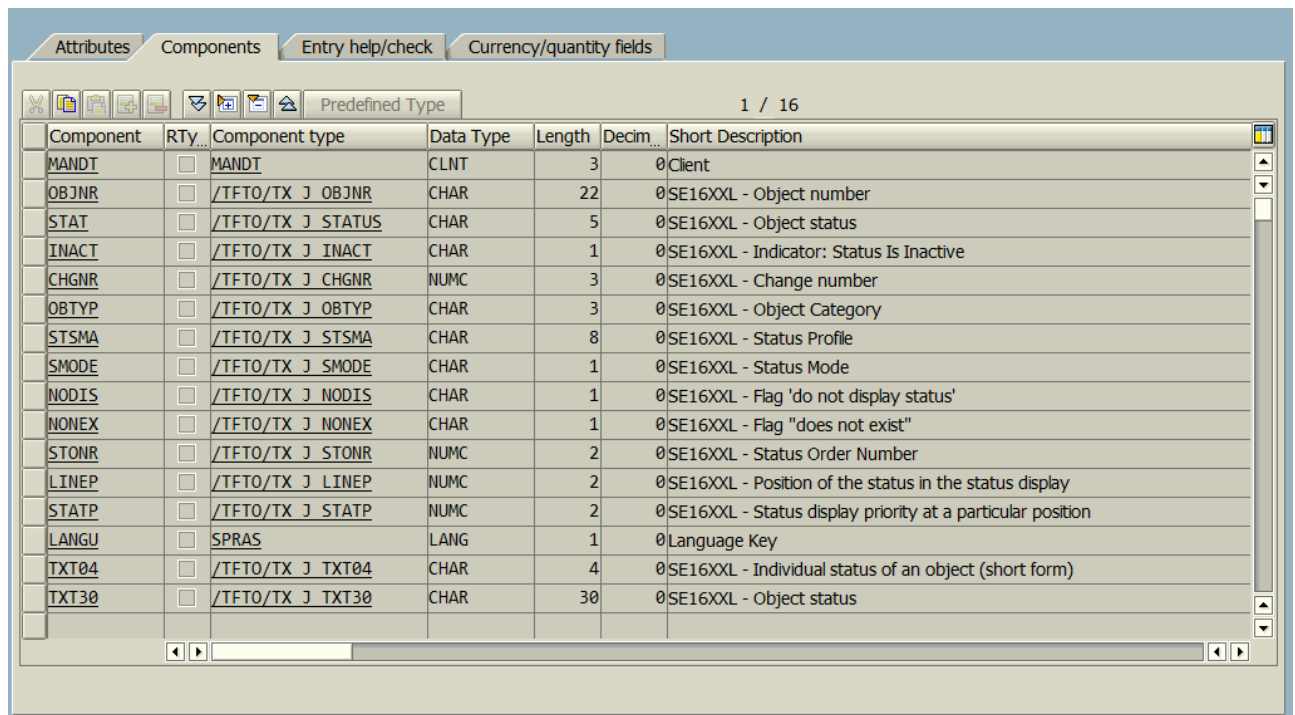
Table **JEST** (Individual Object Status) is used in SAP to keep track of the various states reached by a production order – among others. This table is involved in a complex logic and is never displayed as such in the standard transactions. Rather its contents are processed with the help of other associated tables, like JSTO, TJ02, TJ02T etc. to create a complicated, language dependent, virtual display.

Up to now table JEST could indeed be read with SE16XXL, but it was certainly not easy to match the entries with the information available in the standard transactions. For this reason two pseudo tables (**\$JEST** and **\$JESTC**) have been implemented.

### Pseudo Table \$JEST

The entries of this pseudo table correspond one-to-one to the original JEST entries, but are provided with additional information.

The structure of the entries is as follows:



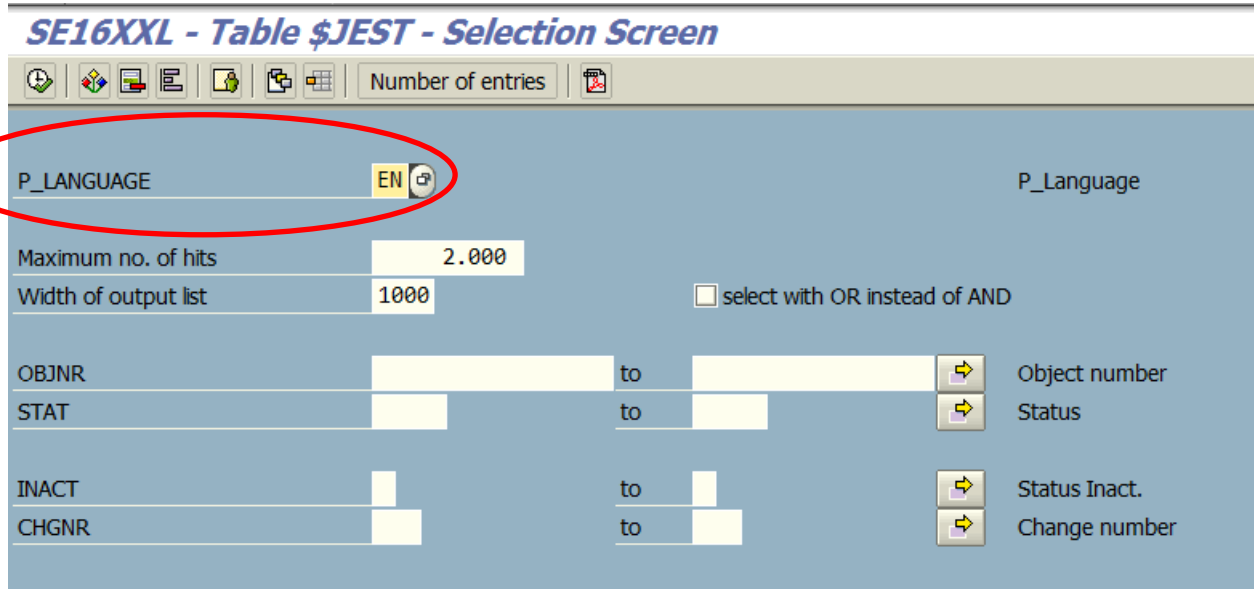
Component	RTY...	Component type	Data Type	Length	Decim...	Short Description
MANDT	<input type="checkbox"/>	MANDT	CLNT	3	0	Client
OBJNR	<input type="checkbox"/>	/TFTO/TX J OBJNR	CHAR	22	0	SE16XXL - Object number
STAT	<input type="checkbox"/>	/TFTO/TX J STATUS	CHAR	5	0	SE16XXL - Object status
INACT	<input type="checkbox"/>	/TFTO/TX J INACT	CHAR	1	0	SE16XXL - Indicator: Status Is Inactive
CHGMR	<input type="checkbox"/>	/TFTO/TX J CHGMR	NUMC	3	0	SE16XXL - Change number
OBTYP	<input type="checkbox"/>	/TFTO/TX J OBTYP	CHAR	3	0	SE16XXL - Object Category
STSMA	<input type="checkbox"/>	/TFTO/TX J STSMA	CHAR	8	0	SE16XXL - Status Profile
SMODE	<input type="checkbox"/>	/TFTO/TX J SMODE	CHAR	1	0	SE16XXL - Status Mode
NODIS	<input type="checkbox"/>	/TFTO/TX J NODIS	CHAR	1	0	SE16XXL - Flag 'do not display status'
NONEX	<input type="checkbox"/>	/TFTO/TX J NONEX	CHAR	1	0	SE16XXL - Flag "does not exist"
STONR	<input type="checkbox"/>	/TFTO/TX J STONR	NUMC	2	0	SE16XXL - Status Order Number
LINEP	<input type="checkbox"/>	/TFTO/TX J LINEP	NUMC	2	0	SE16XXL - Position of the status in the status display
STATP	<input type="checkbox"/>	/TFTO/TX J STATP	NUMC	2	0	SE16XXL - Status display priority at a particular position
LANGU	<input type="checkbox"/>	SPRAS	LANG	1	0	Language Key
TXT04	<input type="checkbox"/>	/TFTO/TX J TXT04	CHAR	4	0	SE16XXL - Individual status of an object (short form)
TXT30	<input type="checkbox"/>	/TFTO/TX J TXT30	CHAR	30	0	SE16XXL - Object status

The last two fields (TXT04 and TXT30) are language dependent.

Only the original fields of table JEST are available on the selection screen. The rest can only be selected on the result list by means of a filter.

### Selection Screen of \$JEST

The selection screen is as follows (with all possible criteria):



Notice the parameter P\_LANGUAGE. This parameter (in imitation of a CDS view) allows the user to specify the language for the language dependent information. If the parameter is left blank the current logon language is used.

### Result List of \$JEST

A typical result list could be as follows:

*SE16XXL - Table \$JEST - 2000 entries selected*

Table \$JEST - JEST with additional information

MANDT	OBJNR	STAT	INACT	CHGNR	OBJTYP	STMSA	SMDDE	NODIS	NONEX	STONR	LINEP	STATP	LANGU	TXT04	TXT30
<input type="checkbox"/>	800	CD500000000005	I0001		1	CDM	I				1	1	E	CRTD	Created
<input type="checkbox"/>	800	CD500000000005	I0251		1	CDM	I				2	3	E	CECR	Check ECR
<input type="checkbox"/>	800	CD500000000006	I0268		1	CDM	I				2	2	E	CCPO	All objects checked
<input type="checkbox"/>	800	CD500000000006	I0254		1	CDM	I				4	3	E	ECOP	ECO to be processed
<input type="checkbox"/>	800	CD500000000006	I0253		1	CDM	I				3	1	E	ECRA	ECR Approved
<input type="checkbox"/>	800	CD500000000006	I0252		1	CDM	I				2	1	E	CHKD	ECR Checked
<input type="checkbox"/>	800	CD500000000006	I0251	X	2	CDM	I				2	3	E	CECR	Check ECR
<input type="checkbox"/>	800	CD500000000006	I0001	X	2	CDM	I				1	1	E	CRTD	Created
<input type="checkbox"/>	800	CD50000000000600000001	I0001		1	CDO	I				1	1	E	CRTD	Created
<input type="checkbox"/>	800	CD50000000000600000001	I0259	X	2	CDO	I				2	4	E	ECIR	Object: insp. req. for ECR
<input type="checkbox"/>	800	CD50000000000600000001	I0260		3	CDO	I				2	3	E	CCO1	Object: change possible
<input type="checkbox"/>	800	CD50000000000600000001	I0261	X	2	CDO	I				2	2	E	CCO2	Object: change not necessary
<input type="checkbox"/>	800	CD50000000000600000001	I0263		1	CDO	I				3	2	E	CCOB	Object: to be processed
<input type="checkbox"/>	800	CD50000000000600000002	I0263		1	CDO	I				3	2	E	CCOB	Object: to be processed

If another language had been specified, both TXT04 and TXT30 would have values differing from the ones listed above.

## Pseudo Table \$JESTC

This table represents a condensed version of JEST. The result corresponds to one entry for each separate OBJNR value.

### Selection Screen of \$JESTC

*SE16XXL - Table \$JESTC - Selection Screen*

Number of entries

P\_LANGUAGE  P\_Language

Maximum no. of hits

Width of output list   select with OR instead of AND

OBJNR  to  Object number

STAT  to  Status

INACT  to  Status Inact.

The parameter P\_LANGUAGE is also present in this case.

The two selection criteria **STAT** and **INACT** can be used to restrict the selection of the underlying JEST entries. A STAT value beginning with 'T' represents a system status, one beginning with 'E' a user status. These two fields are **only available** on the selection screen. They are **not present** on the result list.

### Result List of \$JESTC

A typical result list might look like the following:

*SE16XXL - Table \$JESTC - 6 entries selected*

Table \$JESTC - JEST condensed

MANDT	OBJNR	LANGU	SYST_LINE	USER_LINE	SYST_STRG	USER_STRG
<input type="checkbox"/> 800	NP000000903679	E	ACAS CNF CRTD MANC NMAT NTUP PRC REL	Quot	ACAS CNF CRTD MANC NMAT NTUP PRC REL	Quot
<input type="checkbox"/> 800	NP000000903859	E	ACAS CNC CRTD MANC NMAT NTUP PCNF PRC *	Quot	ACAS CNC CRTD MANC NMAT NTUP PCNF PRC REL TECO	Quot
<input type="checkbox"/> 800	NP000000903879	E	ACAS CNC CNF CRTD MACM MANC NMAT NTUP*	Quot	ACAS CNC CNF CRTD MACM MANC NMAT NTUP PCNF PRC REL TECO	Quot
<input type="checkbox"/> 800	NP000000903899	E	ACAS CNF CRTD MANC NMAT NTUP PCNF PRC *	Quot	ACAS CNF CRTD MANC NMAT NTUP PCNF PRC REL	Quot
<input type="checkbox"/> 800	OR000000000006	E	REL CLSD CNC CSER MACM NTUP PRC		REL CLSD CNC CSER MACM NTUP PRC	
<input type="checkbox"/> 800	OR000000000007	E	REL CLSD CNC CSER NTUP PRC		REL CLSD CNC CSER NTUP PRC	

The field SYST\_LINE is limited to 40 characters. SYST\_STRG is of type STRING. The same can be said for USER\_LINE and USER\_STRG.

[Back to the top](#)

## Program for Scheduling a Script in Background



Up to now it was not possible to schedule an SE16XXL script in background from a background job. But there are companies that make use of third party job scheduling tools, which are only able to schedule a program plus variant in background.

In order to close this gap a new program has been implemented, which accepts all the necessary parameters and options through the selection screen and then immediately starts a script in background.


The program is called `/TFTO/TX_SCHED_SCRIPT_SF`.

It offers the following selection screen (partially filled as an example):

**SE16XXL - Schedule a script in background --> Server File**

**Script Parameters**

Script name:    with reduced memory usage

Variant:

Maximum no. of hits:

Language for additional texts:  Secondary language:

**Job Parameters**

non-standard job name:

Job class:  Exec.Target:


**Server File Parameters**



To change the path please press the input help button or the F4 key.



Path:


Filename:

overwrite an existing file  append to existing file


File type:  


Separator:   Decimal point:  

Date format:   Time format:  

use conversion exits Encoding:  

insert title lines  with byte order mark

insert field name headings Replacement character:  

insert field text headings Line feed:  

take reference fields of currency fields into account

eliminate first double quote from text fields

As may be seen, the parameters and options correspond to the ones available when scheduling a script in background directly from a SE16XXL session.

There are, however, some restrictions. Only the following options are available:

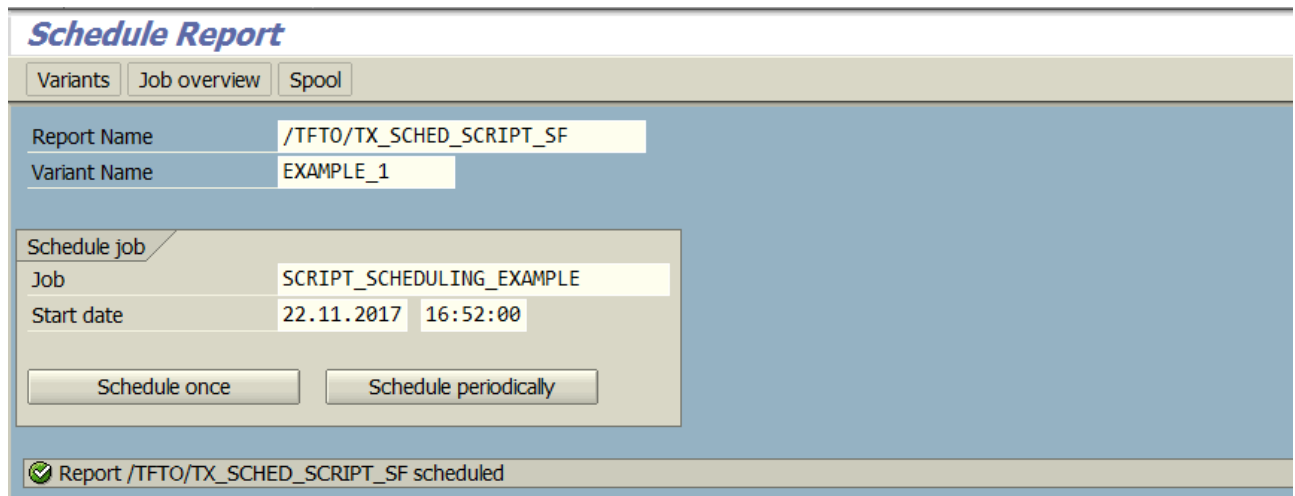
- with referenced script
- immediate start
- the result is written to a server file

The above listed options correspond to scheduling a script in background from the first screen of SE16XXL with immediate start and producing a server file as result.

With regard to "**immediate start**" an explanation is necessary.

It only means that the start parameters and options, like **start date**, **start time**, **scheduling period** etc. are associated with the outer scheduling program, i.e. /TFTO/TX\_SCHED\_SCRIPT\_SF, and not with the script. Each time the program runs, it starts immediately the script in background. But it is up to the user to define when the program should run. So there is no restriction in regard to the start.

Since all necessary parameters and options can be specified via selection screen, they can be saved in a program variant. The variant is selected when the program is scheduled in background:



In the SAP job overview two jobs will be found:

Job	Ln	Job CreatedB	Status	Start date	Start time	Duration(sec.)	Delay (sec.)
<input type="checkbox"/> SCRIPT_SCHEDULING_EXAMPLE		TOPFLOW	Finished	22.11.2017	16:52:54	1	54
<input type="checkbox"/> SCRIPT_\$CUSTOMER_ORDERS		TOPFLOW	Finished	22.11.2017	16:52:55	1	0
*Summary						2	54

The first one is the one we have scheduled. The second one is the job scheduled by the program, i.e. the one that processes the script.



If we examine the job log of the first job we will find:

**Job Log Entries for SCRIPT\_SCHEDULING\_EXAMPLE / 16503200**

Job log overview for job: SCRIPT\_SCHEDULING\_EXAMPLE / 16503200

Date	Time	Message text	Message class	Message no.	Message type
22.11.2017	16:52:54	Job started	00	516	S
22.11.2017	16:52:54	Step 001 started (program /TFTO/TX_SCHED_SCRIPT_SF, variant EXAMPLE_1, user ID TOPFLOW)	00	550	S
22.11.2017	16:52:55	OK - script '\$CUSTOMER_ORDERS' started in background	/TFTO/TX	270	S
22.11.2017	16:52:55	Job finished	00	517	S

The second job shows the details of the script run:

**Job Log Entries for SCRIPT\_\$CUSTOMER\_ORDERS / 16525400**

Job log overview for job: SCRIPT\_\$CUSTOMER\_ORDERS / 16525400

Date	Time	Message text	Message class	Message no.	Message type
22.11.2017	16:52:55	Job started	00	516	S
22.11.2017	16:52:55	Step 001 started (program /TFTO/TX_BATCH_SCRIPT_X, variant &0000000000565, user ID TOPFLOW)	00	550	S
22.11.2017	16:52:56	Target server file path is \usr\sap\tmp	/TFTO/TX2	026	S
22.11.2017	16:52:56	Performing global script \$CUSTOMER_ORDERS	/TFTO/TX	750	S
22.11.2017	16:52:56	Option "with reduced memory usage" is active	/TFTO/TX	874	S
22.11.2017	16:52:56	File \usr\sap\tmp\Customer_Orders.txt written	/TFTO/TX	621	S
22.11.2017	16:52:56	Job finished	00	517	S

In the SE16XXL Background Overview this second job shows up as follows:

**1 SE16XXL background jobs selected**

Job	Plnd date	Plnd time	Status	Period.	User	S.Type	Script	Glob	Priv	RMU	Script user	1st table	R.Data	R.Sub.	P
<input type="checkbox"/> SCRIPT_\$CUSTOMER_ORDERS	22.11.2017	16:52:55	Finished		TOPFLOW										

No request data are available because they are deleted when the server file has been written. This would be true even if the scheduling program had been started periodically. From the point of view of the script each run would be a one-time run.

[Back to the top](#)

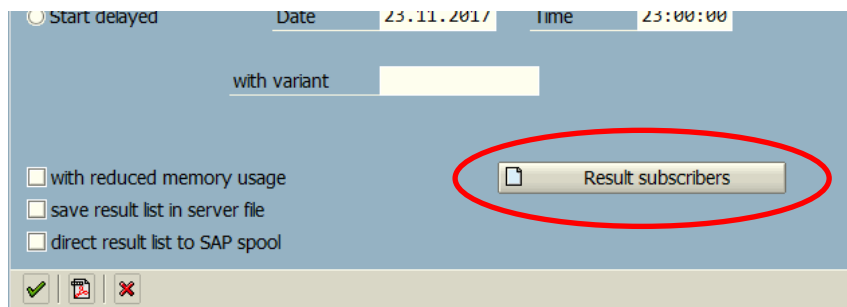
## New Option "one for all" for Result Subscribers

In a previous version of SE16XXL the possibility of sharing the result of a script in background with other chosen users ("**subscribers**") was introduced.

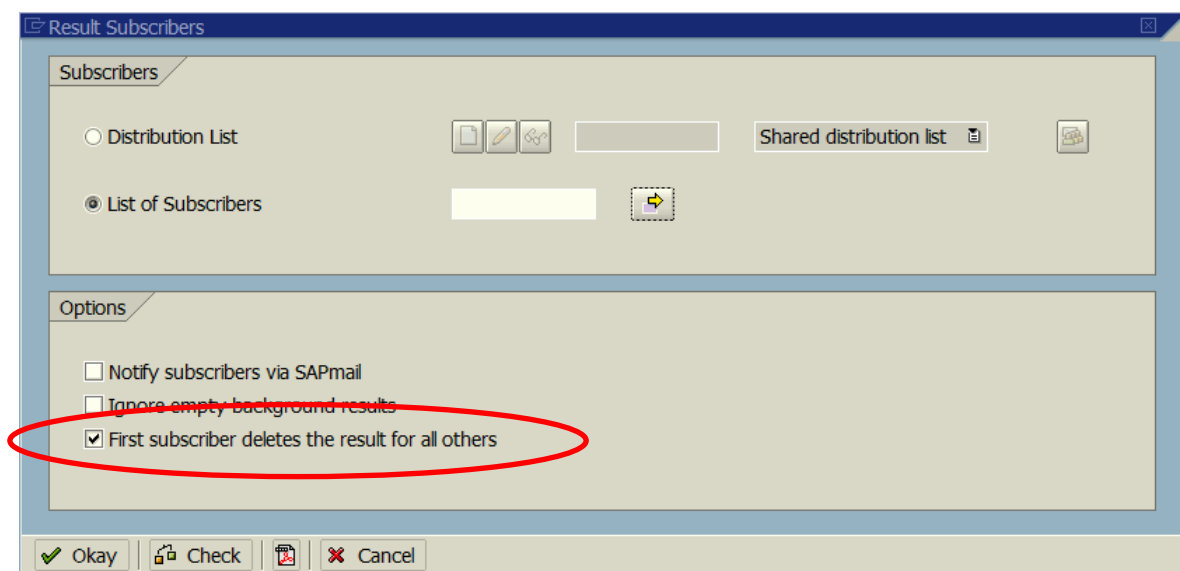
The original idea was to give access to the result to each user independently. The result was available to all users until all of them had assented, either implicitly or explicitly (depending on the user settings), to its deletion.

However, in some situations this logic is not the best one, the objective being to inform **just one person** out of a chosen group. As soon as **one of these persons** has seen the script result, the objective has been reached and the result can be deleted.

For this reason a **new option** has been added to the above described functionality, which is available on the popup window for background scheduling:



After pressing  the following dialog window shows up:



[Back to the top](#)