

## SE16XXL - Neue Funktionen - Historie

- 1			
Dezember 2017	Neue Version	<b>3.3C</b> mit folgende	en Hunktionen:
DCLCIIIUCI 2017	Ticuc Version	. J.JC IIIIt IOIZCIIG	on i unknomen.

		_	
	-	CDS-Views mit Parametern unterstützt	Mehr
	-	Sehr lange Feldnamen (bis zu 30 Stellen) unterstützt	Mehr
	-	Mehrstufige Views (View von Views) unterstützt	Mehr
	-	Zwei neue Pseudo-Tabellen (\$JEST und \$JESTC)	Mehr
	-	Spezialprogram für ein Script im Hintergrund	Mehr
	-	Neue Option "einer für alle" für Script-Teilnehmer	Mehr
Mai 2017		Neue Version <b>3.3B</b> mit folgenden Funktionen:	
	-	Funktionsgruppe zum Ausführen eines Scripts	Mehr
	-	Sprung zu einem Script mit Pseudo-Set/Get-Parametern	Mehr
	-	Neue Formate unterstützt beim Erstellen einer Datei	Mehr
	-	Verbesserte Erstellung einer Server-Datei i. Hintergrund	Mehr
	-	Neue Titel-Einstellungen für Parameter-Transaktionen	Mehr
	-	Eingebettete Tabellenkalkulation mit Einstellungen	Mehr
	-	Neue Ersetzungsvariablen für Titelzeilen	Mehr
Januar 2017		Neue Version <b>3.3A</b> mit folgenden Funktionen:	
	-	Mehrsprachige Script-Kurztexte	Mehr
	-	Neue Funktion "Tabellenkalkulation mit Vorlage"	Mehr
	-	Optionale Ausgabe der Sel.Kriterien beim Drucken	Mehr
	-	Leere Hintergrund-Ergebnisse wahlweise ignoriert	Mehr
	-	Einige neue Funktionen für Administratoren	Mehr
Juli 2016		Neue Version <b>3.3</b> mit folgenden Funktionen:	
	-	Neue Funktionen zum Hinzufügen von Spalten	Mehr
	-	Die Referenzfelder sind nun sichtbar	Mehr

## SE16XXL – Neue Funktionen - Historie



-	Teilnehmer für die Hintergrund-Ergebnisse	Mehr
-	Hintergrund-Server-Datei als E-Mail-Anhang	Mehr
-	Default-ALV-Layouts für Scripts	Mehr
-	Liste von in Scripts verwendeten Tabellen/Views	Mehr
-	Neue Transaktion /TFTO/SE16	Mehr
-	Verbesserung der Definition einer Vorgabedatei	Mehr
-	Verbesserung der Definition der Überschriften	Mehr
Oktober 2015	Neue Version 3.2E mit folgenden Funktionen:	
-	Script-Favoriten	Mehr
-	Verbesserungen des Script-Katalogs	Mehr
-	Internet-Pfad (http://) als Dokumentations-Pfad	Mehr
-	Sammelrollen für Scripts	Mehr
-	Spezielle Berechtigungs-Prüfungen	Mehr
April 2015	Neue Version <b>3.2D</b> mit folgenden Funktionen:	
April 2015	Neue Version <b>3.2D</b> mit folgenden Funktionen: Erläuternde Zusatztexte	Mehr
-	_	Mehr Mehr
-	Erläuternde Zusatztexte	
- -	Erläuternde Zusatztexte Option für doppelte Anführungszeichen	
- -	Erläuternde Zusatztexte Option für doppelte Anführungszeichen Neue Version 3.2C mit folgenden Funktionen:	Mehr
- August 2014	Erläuternde Zusatztexte Option für doppelte Anführungszeichen Neue Version <b>3.2</b> C mit folgenden Funktionen: Table Wizard in <i>SE16XXL</i> umbenannt	<u>Mehr</u>
- August 2014 -	Erläuternde Zusatztexte Option für doppelte Anführungszeichen Neue Version <b>3.2</b> C mit folgenden Funktionen: Table Wizard in <i>SE16XXL</i> umbenannt Hintergrund-Ergebnisliste zum SAP Spool leiten	<u>Mehr</u> <u>Mehr</u> <u>Mehr</u>
- August 2014	Erläuternde Zusatztexte Option für doppelte Anführungszeichen Neue Version <b>3.2</b> C mit folgenden Funktionen: Table Wizard in <i>SE16XXL</i> umbenannt Hintergrund-Ergebnisliste zum SAP Spool leiten Verbesserungen des Formeleditors	Mehr  Mehr  Mehr  Mehr
- August 2014	Erläuternde Zusatztexte Option für doppelte Anführungszeichen Neue Version 3.2C mit folgenden Funktionen: Table Wizard in SE16XXL umbenannt Hintergrund-Ergebnisliste zum SAP Spool leiten Verbesserungen des Formeleditors Verbesserungen des Script-Katalogs	Mehr           Mehr           Mehr           Mehr           Mehr
- August 2014	Erläuternde Zusatztexte Option für doppelte Anführungszeichen Neue Version 3.2C mit folgenden Funktionen: Table Wizard in SE16XXL umbenannt Hintergrund-Ergebnisliste zum SAP Spool leiten Verbesserungen des Formeleditors Verbesserungen des Script-Katalogs Verbesserungen der Speziellen Selektionsmaske	Mehr           Mehr           Mehr           Mehr           Mehr           Mehr



			_	
Oktober 20	13	Neue Version 3.2 mit folgenden Funktionen:		
	-	Vereinheitlichte Auswahl der Listenfelder		Mehr
	-	Spalten beliebig angeordnet im Std-Anzeigemodu	S	Mehr
	-	Funktion "Spalten fixieren" in allen Modi		Mehr
	-	Neue Option beim Filter Setzen		Mehr
	-	Virtuelle Joins		Mehr
	-	Full-Join		Mehr
	-	Join-Kriterien mit Intervall-Selektion		Mehr
	-	Join-Kriterien mit Präfix-Selektion		Mehr
	-	Select-Options mit Tausenden von Einträgen		Mehr
	-	Globale Parameter		Mehr
	-	Scripts "mit reduziertem Speicherbedarf" ausgefü	hrt	Mehr
	-	Neue Operation "Sortierte Sätze zählen"		Mehr
	-	Neue Operation "Formelfelder fixieren"		Mehr
	-	Neue Benutzer-Einstellungen verfügbar		Mehr
	-	Neue Funktion "Download zum Frontend"		Mehr
	-	Spezielle Kalenderfunktionen für Formeln		Mehr
	-	Script-Editor: neue Funktion "SELECT-Tabelle erse	etzen"	Mehr
	-	Weitere Verbesserungen		Mehr
April 2012	Neu	e Version <b>3.1A</b> mit folgenden Funktionen:		
-	Zugi	riffsrechte und SAP-Berechtigungs-Rollen		Mehr
-	Prob	oleme mit Tabellenkalkulation und Kyrillisch gelöst	t	Mehr
-	F4-F	Hilfe für Views		Mehr

Direktaufruf des Script-Katalogs beim Sichern eines Scripts

# Weitere historische Einträge ...

Formel nach einem Join beibehalten

Dokumentation umstrukturiert

Mehr ...

Mehr ...

Mehr ...

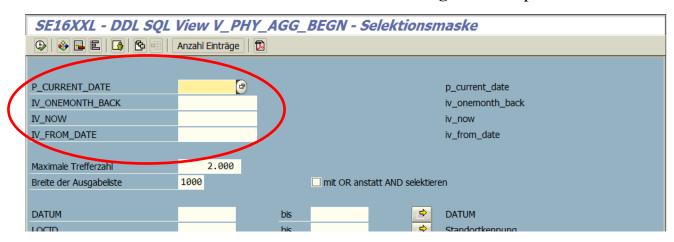


#### **CDS-Views mit Parametern**

Die sogenannten **CDS-Views** – die im Data Dictionary als *DDL SQL Views* dargestellt werden – sind in einer der neuesten SAP-Versionen eingeführt worden. Die meisten von ihnen konnten wie normale Datenbank-Views mithilfe von SE16XXL gelesen werden. Nur die mit **Parametern** weigerten sich, behandelt zu werden, denn diese Art von CDS-View benötigt eine spezielle SQL-Syntax.

Mit Version **3.3**C von SE16XXL ist diese Lücke geschlossen worden. Nun werden CDS-Views mit bis zu **30** Parametern unterstützt. Die Obergrenze von 30 sollte mehr als ausreichend sein (meistens haben CDS-Views **nur einen oder zwei** Parameter).

Auf der Selektionsmaske werden die Parameter immer ganz oben positioniert:



Damit soll ausgedrückt werden, dass die Parameterwerte von der Option "mit OR anstatt AND selektieren" **unabhängig sind**.

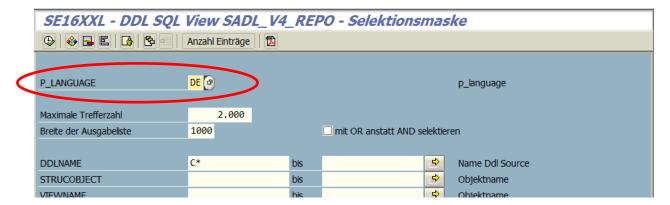
#### **ANMERKUNGEN:**

- 1) Es scheint kein Standardverhalten für Parameter zu geben. Manche sind implizit obligatorisch, andere können leer gelassen werden. Anscheinend hat jeder CDS-View eine eigene Logik. Da muss der Anwender experimentieren.
- Falls ein CDS-View einen **Mandanten**-Parameter hat, wird dieser Parameter mit dem aktuellen Mandanten (**SY-MANDT**) gefüllt und nicht angezeigt.
- Sämtliche Parameter (mit Ausnahme des Mandanten) werden auf der Standard-Selektionsmaske positioniert und angezeigt.
   Nur bei der Definition einer Script-Selektionsmaske ist es möglich, die sichtbaren Parameter auszuwählen.

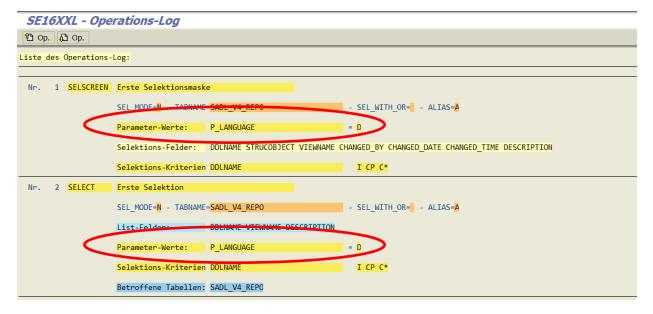


Anhand des CDS-Views **SADL\_V4\_REPO** werden wir nun die verschiedenen Situationen darstellen, in denen Selektions-Parameter auftreten können.

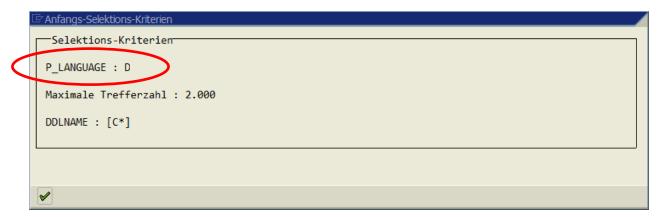
#### Standard-Selektionsmaske



## **Operations-Log (Springen** → **Operations-Log anzeigen)**



## **Selektions-Kriterien (Liste → Selektions-Kriterien anzeigen)**





### **Definition einer Script-Selektionsmaske**

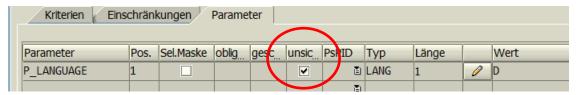


Die Parameter-Definition sieht im Detail wie folgt aus:



Falls der Parameter auf der Maske sichtbar ist (Option "Sel.Maske"), kann er auf "obligatorisch" oder "geschützt" gesetzt werden, wie bei normalen Selektions-Kriterien. Der Wert kann entweder mithilfe der Schaltfläche oder auf der simulierten Selektionsmaske festgelegt werden. Falls Parameter auf der Selektionsmaske erscheinen verringert sich die maximale Anzahl Blocke auf 8.

Ein Parameter kann auch auf "unsichtbar" gesetzt werden:



In diesem Fall kann der Wert nur mithilfe von festgelegt werden.

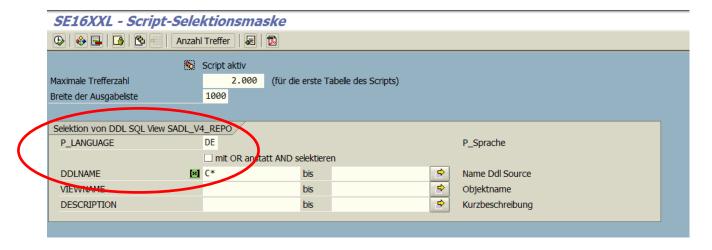
Falls keine der obigen Optionen gesetzt wird, wird der Parameter auf der Selektionsmaske fehlen und es wird ihm **kein Wert** beim SELECT zugeteilt:



Der Originalwert wird in diesem Fall in Klammern dargestellt. Es soll damit zum Ausdruck gebracht werden, dass er unberücksichtigt bleiben wird.



### Script-Selektionsmaske



Der Parameter befindet sich **ganz oben** auf dem Selektions-Block – die Option "mit OR anstatt AND selektieren" befindet sich unterhalb des Parameters.

### **Definition einer Script-Variante**



Wie bei normalen Selektions-Kriterien ist es möglich, dem Parameter eine Variable zuzuordnen ().



## Sehr lange Feldnamen unterstützt

Bis vor kurzem konnten nur DDIC-Strukturen bis zu 30 Stellen lange Feldnamen enthalten – für Felder von transparenten Tabellen und Views was die maximale Länge **16 Stellen** – m.E. mehr als genügend.

Diese Einschränkung ist mit einer der neuesten SAP-Versionen entfallen – es ist nun möglich, transparente Tabellen und Datenbank-Views mit **bis zu 30 stelligen Feldnamen** zu definieren.

Ein paar Beispiele sollen diese Neuerung veranschaulichen:

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

Es handelt sich um **echte** Feldnamen, definiert für den View **ISDSALESDOC**.

An den obigen Beispielen ist leicht zu erkennen, dass es, trotz der größeren Länge, zu Abkürzungen kommt (in den letzten zwei Namen steht STS für STATUS). Es ist auch ersichtlich, dass eine Verkettung von mehreren Worten ohne Trennzeichen der Lesbarkeit nicht besonders dienlich ist.

Diese ist nunmal die neue Situation mit transparenten Tabellen und Views in SAP.

**"Wo liegt das Problem?"** könnte man fragen – das Datenelement FIELDNAME ist bereits 30 Stellen lang und somit dürften die Namen ohne weiteres hineinpassen.

Das stimmt. Leider befasst sich SE16XXL auch mit Joins. In einer Join-Liste wird jede Tabelle von einem Alias repräsentiert, und ihre Felder werden mit dem Alias ergänzt, wie in **A~MATNR**. Intern wird dies als **A-MATNR** geführt, d.h. Feld MATNR von Unterstruktur A. Bis dato war eine solche Verkettung maximal 18 Stellen lang. Ist jedoch das Feld 30 Stellen lang, benötigt die Verkettung **32 Stellen**.

"Kein Problem, man ersetzt Datenelement FIELDNAME mit einem 32-stelligen Datenelement und alles wird gut" könnte eine mögliche Lösung sein.



**Leider nicht**. Es gibt einige Einschränkungen, die nicht überwunden werden können. Zum Beispiel ist der Feldname im **ALV-Feldkatalog** nur 30 Stellen lang und darf – es handelt sich hier um eine offizielle Schnittstelle – nicht angepasst werden.

### Wie das Problem gelöst wurde

Das Problem ist wie folgt gelöst worden:

Sämtliche Informationen in Bezug auf Feldnamen (Scripts, Script-Varianten, Hintergrund-Ergebnisse usw.) werden in Original-Länge, d.h. bis 32 Stellen, in die Datenbank geschrieben.

Wenn ein Script o.ä. geladen wird, werden die Feldnamen, die 28 Stellen übersteigen, dynamisch verkürzt. Die notwendigen Umsetzungstabellen werden im virtuellen Speicher gehalten. Die verküzten Feldnamen werden intern verwendet, wo dies notwendig ist, wie z.B. an der ALV-Schnittstelle.

Nach außen merkt der Anwender nichts von diesem komplizierten Spiel. Es gibt nur eine Situation, bei der die verkürzten Namen sichtbar werden, nämlich bei einem **Filter-Dialogfenster** mit einer Join-Liste. Ein solches Dialogfenster bedient sich einer Standard-SAP-Funktionalität, bei der die Felder nur 30 Stellen lang sind. In allen anderen Fällen werden die Original-Feldnamen angezeigt.

## Zusätzliche Anpassungen

Abgesehen von den internen Umsetzungen sind weitere Anpassungen wegen der extralangen Feldnamen notwendig (oder zumindest ratsam) geworden, nachdem bis dato alle Dialoge auf eine maximale Länge von 18 ausgerichtet waren. Wo es möglich war wurde die sichtbare Länge erhöht, entweder grundsätzlich oder in Abhängigkeit von den aktuell vorhandenen Feldnamen.

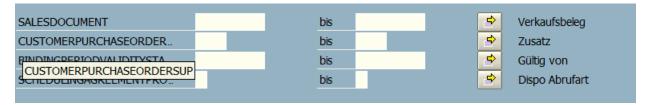
In den nachfolgenden Seiten werden einige Beispiele dieser sichtbaren Anpassungen präsentiert.



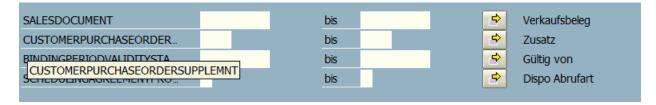
#### Standard-Selektionsmaske

Aus technischen Gründen gibt es nicht genug Platz auf der Selektionsmaske, um die Feldnamen in voller Länge anzuzeigen. Wenn aber der Mauszeiger über ein bestimmtes Feld bewegt wird, erscheint jetzt der vollständige Name:

Alte Version (der Name ist abgeschnitten):



Neue Version (der Name erscheint in voller Länge):



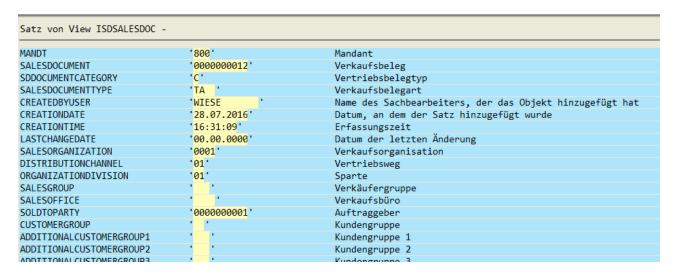
### **Detaillierte Sicht eines Listen-Eintrags**

#### Alte Version:

```
Satz von View ISDSALESDOC -
MANDT
               '800'
                                            Mandant
SALESDOCUMENT '0000000012'
                                            Verkaufsbeleg
SDDOCUMENTCATEGORY 'C
                                            SD-Belegtyp (lang)
SALESDOCUMENTTYPE 'TA
                                            Verkaufsbelegart
SALESDOCUMENTPROCESSINGTYPE
                                            Vertriebsbelegkennzeichnung
CREATEDBYUSER 'WIESE
CREATIONDATE '04.05.2016'
CREATIONTIME '17:33:22'
                                            Name des Sachbearbeiters, der das Objekt hinzugefügt hat
                                            Datum, an dem der Satz hinzugefügt wurde
                                            Erfassungszeit
LASTCHANGEDATE '00.00.0000'
                                            Datum der letzten Änderung
LASTCHANGEDATETIME '20.160.822.144.938,0240000' UTC-Zeitstempel in Langform (JJJJMMTThhmmssmmmuuun)
SALESORGANIZATION '0001'
                                            Verkaufsorganisation
DISTRIBUTIONCHANNEL '01'
                                            Vertriebsweg
ORGANIZATIONDIVISION '01'
                                            Sparte
SALESGROUP
                                            Verkäufergruppe
SALESOFFICE
                                            Verkaufsbüro
             '00000000002'
SOLDTOPARTY
                                            Auftraggeber
CUSTOMERGROUP '
                                            Kundengruppe
ADDITIONALCUSTOMERGROUP1 '
                                            Kundengruppe 1
ADDITIONALCUSTOMERGROUP2 '
                                            Kundengruppe 2
ADDITIONAL CLISTOMERGROUPS '
                                            Kundengrunne
```



#### Neue Version:

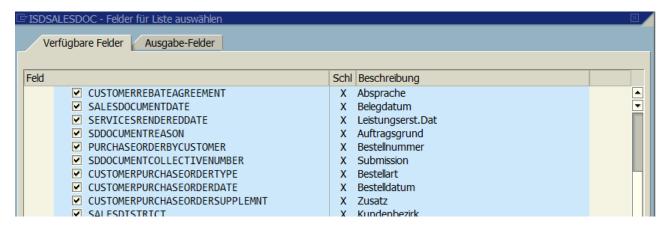


#### Auswahl der Listen-Felder

Alte Version (Die Spalte für den Feldnamen ist nicht breit genug):



#### Neue Version:





#### **Definition der Script-Selectionsmaske**

Alte Version (Spalten zu schmal):



#### Neue Version:



Diese Beispiele dürften genügen, um eine Vorstellung der durchgeführten Anpassungen zu vermitteln.

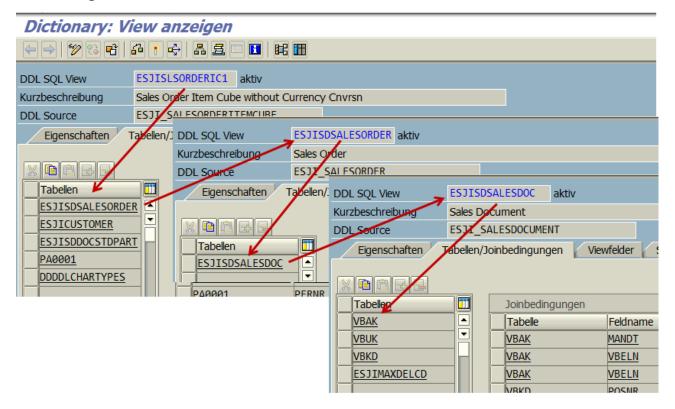


## Mehrstufige Views (Views von Views) unterstützt

Bis vor kurzem konnte ein **Datenbank-View** nur auf einer Tabelle mit Tabellenart **transparent, pool oder cluster** basieren, aber nicht auf einem anderen View.

In den neuesten SAP-Versionen ist diese Regel **gelockert worden**. In diesen SAP-Systemen existieren Views, die rekursiv **auf anderen Views basieren**. In manchen Fällen befindet sich die echte Datenbanktabelle nach **25 Stufen** von Views, oder sogar tiefer.

#### Zum Beispiel:



View **ESJISDSALESORDER**(1 basiert auf View **ESJISDSALESORDER**, der auf View **ESJISDSALESDOC** basiert, der endlich auf der DB-Tabelle **VBAK** basiert.

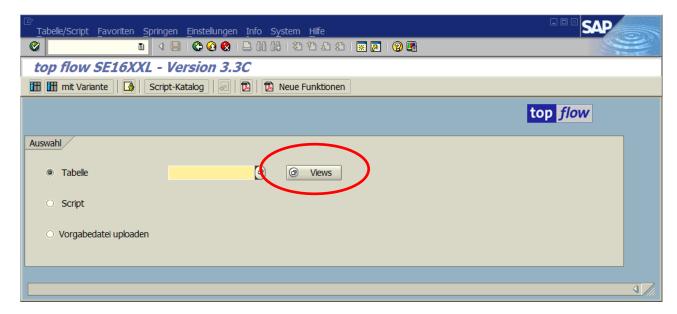
Die Zugriffsrechte für Tabellen und Felder basieren In SE16XXL **auf echten Tabellen** und nicht auf Views. Im Falle eines Views werden die notwendigen Prüfungen auf die zugrunde liegenden DB-Tabellen durchgeführt. Um diese DB-Tabellen zu ermitteln wird nun jeder View einer **rekursiven Suche** unterzogen.

Dasselbe gilt für die SE16XXL Berechtigungen auf Satzebene. Sie basieren ebenfalls auf Feldern von echten DB-Tabellen. Wie im Falle der Zugriffsrechte werden die zugrunde liegenden Tabellenfelder mithilfe einer rekursiven Suche ermittelt und dann für die Berechtigungs-Prüfungen verwendet.

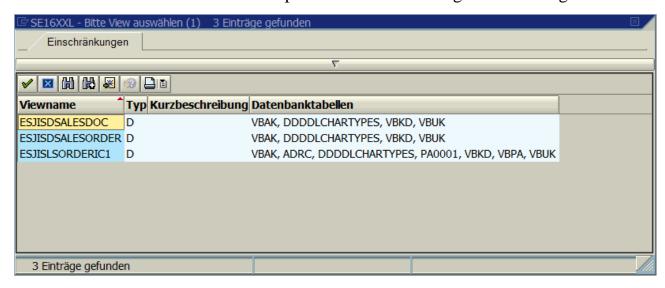


### F4-Hilfe für Views

Auf der Einstiegsmaske von SE16XXL befindet sich eine spezielle Schaltfläche für die F4-Hilfe für Views:



Das Resultat einer Suche zeigt die Views zusammen mit den zugrunde liegenden Datenbanktabellen. In unserem Beispiel würden wir ein Ergebnis wie folgt erhalten:



Die Zwischenstufen eines Views erscheinen nicht in der Spalte "Datenbanktabellen". Wie im Falle der Zugrifsrechte und Berechtigungen werden die zugrunde liegenden Datenbanktabellen mittels rekursive Suche ermittelt.



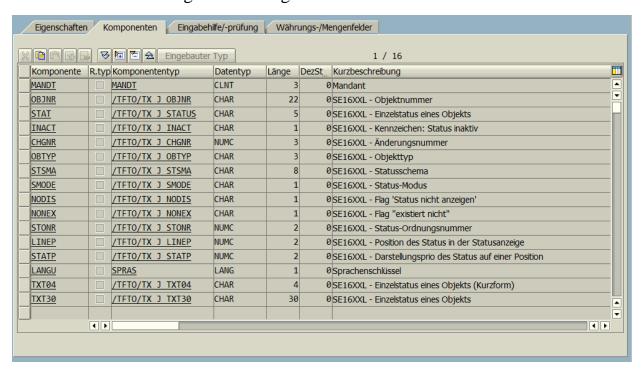
## Neue Pseudo-Tabellen \$JEST und \$JESTC

Die Tabelle **JEST** (Einzelstatus pro Objekt) wird u.a. in SAP dazu verwendet, die verschiedenen Zustände festzuhalten, die ein Fertigungsauftrag erreicht hat. Diese Tabelle unterliegt einer komplexen Logik und wird nie ohne Aufbereitung in den Standard-Transaktionen angezeigt. Vielmehr wird ihr Inhalt mithilfe verwandter Tabellen (wie JSTO, TJ02, TJ02T usw.) prozessiert, um daraus eine komplizierte sprachabhängige virtuelle Darstellung hervorzubringen.

Bis dato konnte JEST ohne weiteres mit SE16XXL gelesen werden. Es war aber sicherlich nicht einfach, den Zusammenhang mit den Informationen aus den Standard-Transaktionen herzustellen. Aus diesem Grund sind zwei Pseudo-Tabellen (\$JEST und \$JESTC) implemetiert worden.

#### Pseudo-Tabelle \$JEST

Die Einträge dieser Pseudo-Tabelle stehen in einer eins-zu-eins Beziehung mit den Original-JEST-Einträgen, sind jedoch mit zusätzlichen Informationen ausgestattet. Die Struktur der Einträge ist wie folgt:



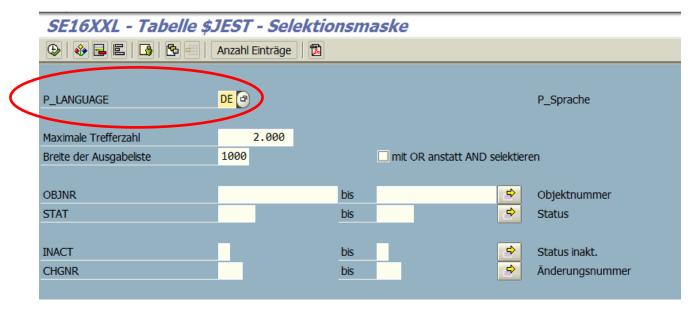
Die letzten zwei Felder (TXT04 und TXT30) sind sprachabhängig.

Nur die Originalfelder aus JEST stehen auf der Selektionsmaske zur Verfügung. Die übrigen Felder können in der Ergebnisliste nur mithilfe eines Filters selektiert werden.



#### Selektionsmaske von \$JEST

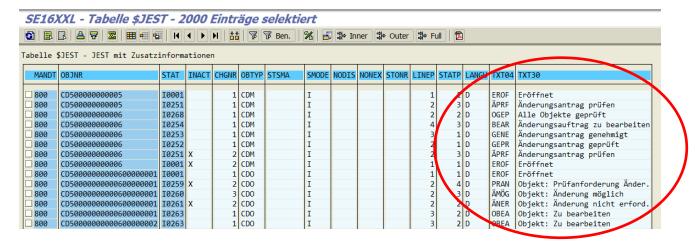
Die Selektionsmaske (mit den möglichen Kriterien) sieht wie folgt aus:



In Anlehnung an einen CDS-View ist der Parameter P\_LANGUAGE eingeführt worden, um die Sprache für die sprachabhängigen Informationen festzulegen. Wenn der Parameter leer gelassen wird, wird die aktuelle Logon-Sprache verwendet.

#### **Ergebnisliste von \$JEST**

Eine typische Ergebnisliste könnte wie folgt aussehen:



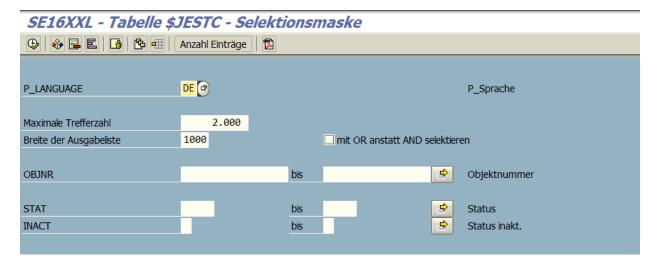
Wäre eine andere Sprache angegeben worden, so hätten sowohl TXT04 als auch TXT30 andere Werte als hier abgebildet.



#### Pseudo-Tabelle \$JESTC

Diese Tabelle repräsentiert eine kondensierte Version von JEST. Das Ergebnis ist ein Eintrag pro OBJNR-Wert.

#### Selektionsmaske von \$JESTC

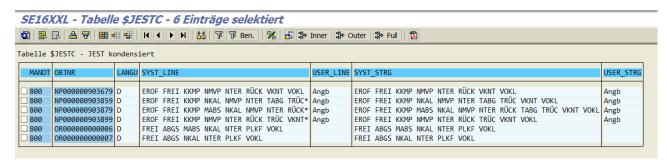


Der Parameter P\_LANGUAGE ist auch in diesem Fall präsent.

Die zwei Kriterien **STAT** und **INACT** können verwendet werden, um die Selektion der zugrunde liegenden JEST-Einträge einzuschränken. Ein STAT-Wert beginnend mit '**I**' stellt einen Systemstatus dar, einer mit '**E**' einen Anwenderstatus. Diese zwei Felder sind **nur** auf der Selektionsmaske vorhanden. In der Ergebnisliste treten sie **nicht** in Erscheinung.

#### **Ergebnisliste von \$JESTC**

Eine typische Ergebnisliste könnte wie folgt aussehen:



Das Feld SYST\_LINE ist auf 40 Stellen begrenzt. SYST\_STRG ist ein STRING. Ähnliches trifft für USER\_LINE und USER\_STRG zu.

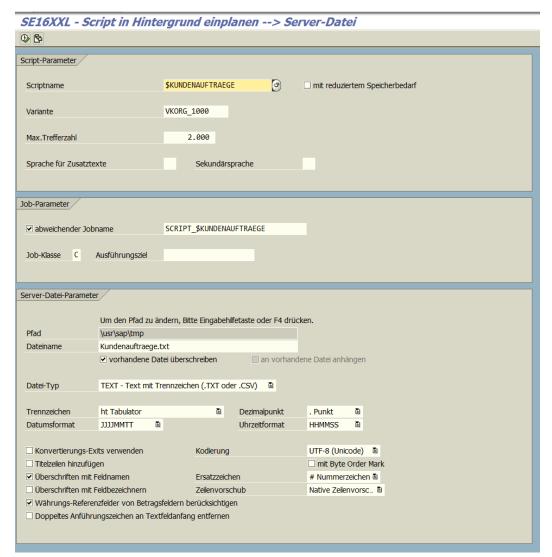


## Programm zum Einplanen eines Scripts im Hintergrund

Bis dato war es nicht möglich, ein SE16XXL Script aus einem Hintergrund-Job ebenfalls im Hintergrund einzuplanen. Es gibt jedoch Unternehmen, die Tools von Drittanbietern für die Jobeinplanung verwenden. Solche Tools können nur ein Programm plus Variante einplanen.

Um diese Lücke zu schließen ist ein neues Programm implementiert worden, das alle notwendigen Parameter und Optionen via Selektionsmaske annimmt und sofort ein Script im Hintergrund startet.

Das Programm heißt /TFTO/TX\_SCHED\_SCRIPT\_SF.
Es bietet folgende Selektionsmaske (als Beispiel zum Teil gefüllt) an:



Wie man sehen kann, entsprechen die Parameter und Optionen denen, die bei der Einplanung während einer SE16XXL-Sitzung zur Verfügung stehen.



Es gibt jedoch einige Einschränkungen. Nur folgende Optionen sind verfügbar:

- mit referenziertem Script
- Start sofort
- das Ergebnis wird in eine Server-Datei geschrieben

Obige Optionen entsprechen dem Einplanen eines Scripts im Hintergrund aus der Einstiegsmaske von SE16XXL mit sofortigem Start und mit Server-Datei als Ziel.

In Bezug auf "Start sofort" ist eine Erklärung notwendig.

Es bedeutet lediglich, dass Start-Parameter und -Optionen, wie **Startdatum**, **Startzeit**, **Periode** usw. sich auf den Job mit dem Einplanungs-Programm beziehen, d.h. /TFTO/TX\_SCHED\_SCRIPT\_SF, und nicht auf das Script. Bei jedem Lauf des Programms wird das Script im Hintergrund sofort gestartet. Es liegt aber an dem Anwender, den Zeitpunkt für dieses Programm festzulegen. Somit gibt es keinerlei Einschränkung bezüglich Start.

Nachdem sämtliche Parameter und Optionen mittels Selektionsmaske festgelegt werden können, können sie in eine Programm-Variante gespeichert werden. Die Variante wird bei der Einplanung des Programms angegeben:



In der SAP Job-Übersicht sind zwei Jobs zu sehen:

Jobname	Sp	Job-Erstelle	Status	Startdatum	Startzeit	Dauer(sec.)	Verzögerung(sec.)
☐ SCRIPT_EINPLANUNGS_BEISPIEL☐ SCRIPT_\$KUNDENAUFTRAEGE				27.11.2017 27.11.2017		1	45 0
*Zusammenfassung						2	45

Der erste ist derjenige, den wir eingeplant haben. Der zweite ist vom Programm eingeplant worden – der Job mit dem Script.



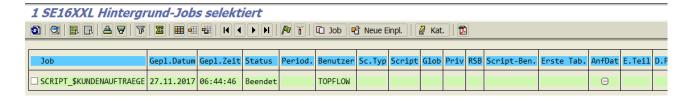
Der Job-Log des ersten Jobs enthält folgende Meldungen:



Der zweite Job zeigt die Details der Script-Ausführung:



In der SE16XXL-Hintergrund-Übersicht taucht der zweite Job wie folgt auf:



Es gibt keine Anforderungs-Daten mehr, weil sie nach der Erstellung der Server-Datei gelöscht wurden. Auch im Falle einer periodischen Einplanung des Programms wäre die Situation ähnlich. Aus der Sicht des Scripts wäre jeder Lauf eine Einmal-Einplanung.



## Neue Option "einer für alle" für Ergebnis-Teilnehmer

In einer früheren Version von SE16XXL wurde die Möglichkeit geschaffen, das Ergebnis eines Scripts im Hintergrund mit anderen "**Teilnehmern**" zu teilen.

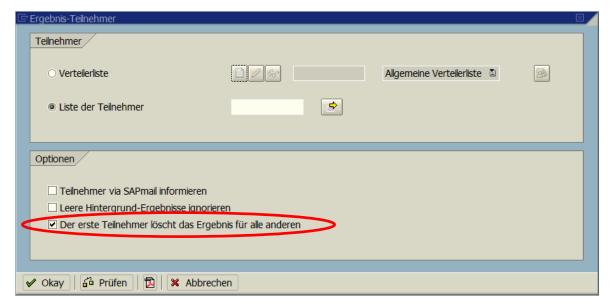
Die ursprüngliche Idee war, das Ergebnis für jeden Teilnehmer verfügbar zu machen. Jeder Teilnehmer musste implizit oder explizit (abhängig von den Einstellungen) seine Einwilligung zur Löschung geben.

Wenn jedoch das Ziel darin besteht, **nur eine Person** aus einer Gruppe zu informieren, ist die obige Logik unpassend. Sobald **eine dieser Personen** das Script-Ergebnis gesehen hat, ist das Ziel erreicht und das Ergebnis kann gelöscht werden.

Aus diesem Grund ist eine **neue Option** zur obenbeschriebenen Funktionalität hinzugefügt worden:



Nach Betätigen von Ergebnis-Teilnehmer erscheint folgendes Dialog-Fenster:





## Funktionsgruppe zum Ausführen eines SE16XXL Scripts

Dieses Thema dürfte mehr Programmierer interessieren als normale Anwender.

Bis dato konnte ein SE16XXL-Script nur mithilfe der Transaktion /TFTO/SE16SCRIPT von einem ABAP-Programm ausgeführt werden. Dadurch wurde zwar das Script ausgeführt, das aufrufende Programm hatte aber keine Möglichkeit auf die Ergebnisse zuzugreifen, die dann wie üblich dem Benutzer angezeigt wurden.

Um diese Probleme zu überwinden wurde eine **neue Funktionsgruppe** implementiert, die es einem ABAP-Programm ermöglicht, ein Script auszuführen und auf das Ergebnis anhand einer **Referenz-Variable** zuzugreifen. Dadurch bekommt das Programm die volle Kontrolle über die Ergebnisliste – es können unerwünschte Sätze entfernt werden, der Inhalt anderer ergänzt und zum Schluß angezeigt werden, nicht unbedingt mithilfe von ALV. Das Resultat muss nicht einmal gezeigt werden – stattdessen könnte das Programm daraus eine eigene Liste zusammenbauen.

Die neue Funktionsgruppe heißt /TFTO/TX\_SFMI (SFMI = Script Function Module Interface). Der wichtigste Funktionsbaustein ist /TFTO/TX\_SFMI\_CALL\_SCRIPT.

Drei **Beispielprogramme** sind erstellt worden, um dem Programmierer eine Vorstellung zu geben, wie die Funktionsgruppe zu verwenden ist:

- /TFTO/SFMI\_CALL\_SIMPLE
- /TFTO/SFMI CALL EXAMPLE
- /TFTO/SFMI\_CALL\_EXTRA

Anhand des Codings der Beispielprogramme und der einzelnen Funktionsbausteine müsste ein ABAP-Programmierer in der Lage sein, diese neue Funktionalität zu verwenden.

Nachdem jedes Programm ein Script aufruft, das verständlicherweise nicht auf dem Entwicklungssystem des Programmierers vorhanden sein kann, empfiehlt es sich, eine **Kopie** von jedem Beispielprogramm zu erstellen und den Namen des Scripts durch ein eigenes zu ersetzen. Der Programmierer kann dann – mithilfe des ABAP-Debuggers – Tests zur Erkundung der Funktionen durchführen.

Für weitere Informationen lesen Sie bitte Funktionsgruppe SFMI.



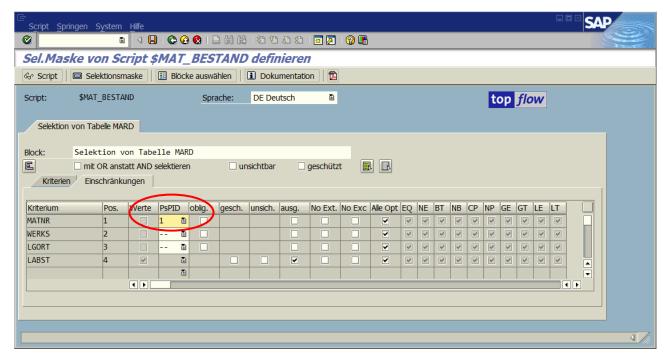
## Sprung zur Transaktion /TFTO/SE16SCRIPT mit Pseudo-Set/Get-Parametern

Bis dato war es möglich, einen Sprung zur Transaktion /TFTO/SE16SCRIPT zu definieren, wobei der Name des auszuführenden Scripts mithilfe des Set/Get-Parameters /TFTO/TX\_SCRIPT spezifiziert wurde. Nach einem Doppelklick auf die beteiligte Spalte kam zwar die Selektionsmaske des Scripts zum Vorschein, jedoch nur mit **Standard**-Selektionskriterien. Es gab **keine** Möglichkeit, kontextbezogene Werte aus der angeklickten Zeile an das Script weiterzuleiten. Das war ein gravierendes Manko. Diese Art von Sprung war damit so gut wie unbrauchbar.

Um Abhilfe zu schaffen wurde ein Mechanismus implementiert, der auf sogenannte **Pseudo-Set/Get-Parameter** basiert. Er bietet die Möglichkeit, einem Script kontextbezogene Werte zu überreichen, damit ein sinnvoller Sprung aus der Ergebnisliste zu einem Script möglich und praktikabel wird. Einzige Bedingung ist, dass das Script mit einer speziellen Selektionsmaske ausgestattet ist.

Ein einfaches Beispiel wird die Vorgehensweise veranschaulichen.

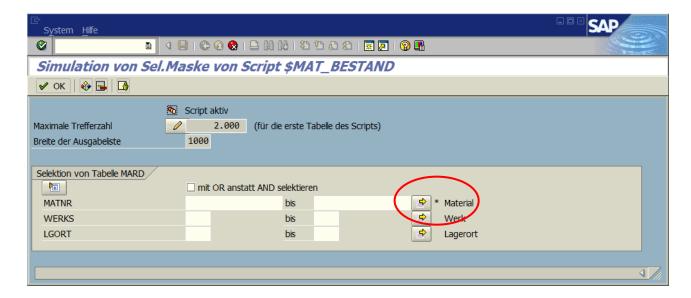
Wir beginnen mit dem aufzurufenden Script. Es liest die Material-Bestandsdaten (Tabelle MARD) und zeigt die verfügbaren Mengen. Wir gehen davon aus, dass das Script bereits eine spezielle Selektionsmaske besitzt. Ansonsten müsste sie definiert werden. Der nächste Schritt besteht darin, ein Pseudo-Set/Get-Parameter der Materialnummer zuzuordnen. Das geschieht in der Definition der speziellen Selektionsmaske (auf der Anwendungsleiste des Script-Katalogs) unter dem Reiter "Einschränkungen":





Zu diesem Zweck wurde eine neue Spalte mit der Überschrift **PsPID** hinzugefügt. In diesem Beispiel weisen wir der MATNR die Kurzform "1" zu, die dem Pseudo-Set/Get-Parameter /TFTO/TX\_CRIT1 entspricht. Ein Blick auf die F4-Hilfe zeigt Werte von 1 bis 9, die ausreichend sein dürften – beim Aufruf eines Scripts (in einer Sprung-Definition) können bis zu 9 Kontext-Werte mitgegeben werden.

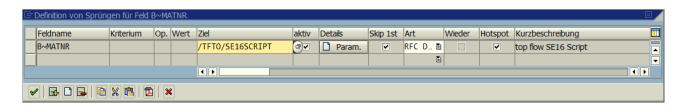
Auf der **simulierten** Selektionsmaske (Selektionsmaske) erscheinen die Kriterien mit einem Pseudo-Set/Get-Parameter mit einem kleinen **Stern** auf der rechten Seite:



Auf diese Weise kann der Anwender schnell sehen, ob solche Zuweisungen festgelegt wurden (der Stern erscheint nicht im echten Dialogbetrieb).

Die **Zielseite** ist fertig. Wir können unsere Aufmerksamkeit der **aufrufenden** Seite widmen, d.h. der Sprung-Definition. In unserem Beispiel betrachten wir ein Script, das Kundenaufträge liest. Jede Position enthält eine Materialnummer. Wenn der Anwender auf diesen Hotspot klickt, soll ein Sprung zum obenbeschriebenen Script stattfinden, wodurch der verfügbare Bestand des beteiligten Materials gezeigt wird.

Wir definieren den Sprung auf die übliche Weise:



Nach der Angabe der Zieltransaktion (/TFTO/SE16SCRIPT) und Betätigen von **ENTER** wechseln wir zu den Set/Get-Parametern (Taste Param.).

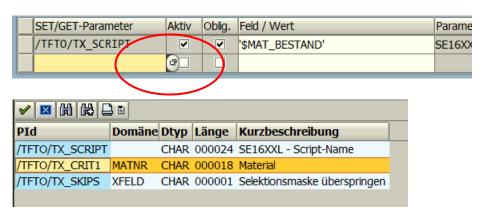


Hier geben wir den Namen des Ziel-Scripts als Literal ein:

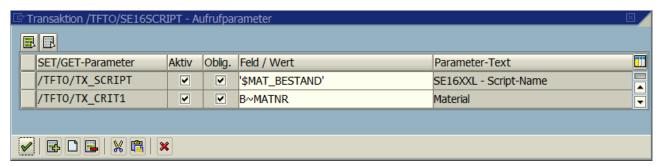


Das ist in diesem Zusammenhang **unbedingt** notwendig, ansonsten hat das Programm keine Möglichkeit, das beteiligte Script zu identifizieren – man muss bedenken, dass jedes Script eigene Pseudo-Set/Get-Parameter haben kann, die meistens völlig andere Bedeutungen besitzen, als die von anderen Scripts.

Jetzt können wir anhand der F4-Hilfe die gewünschten Pseudo-Set/Get-Parameter auswählen:

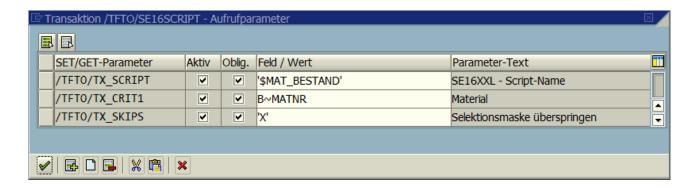


Wir wählen /TFTO/TX\_CRIT1 aus und ordnen ihm die Materialnummer zu. Das Ergebnis sieht dann wie folgt aus:

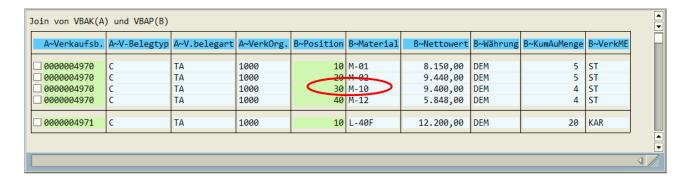


In der obigen Liste der F4-Hilfe ist der letzte Eintrag /TFTO/TX\_SKIPS, ein spezieller Pseudo-Set/Get-Parameter, der zur Verfügung steht, um die Selektionsmaske gänzlich zu **überspringen**. Wir wählen diesen Parameter ebenfalls aus – damit steht das Endergebnis fest:

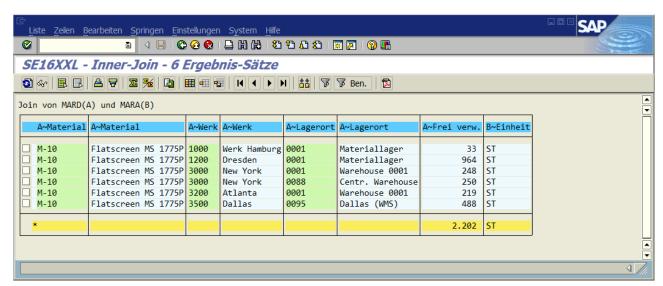




Wir sichern die Sprung-Definition und führen anschließend unser Script aus. Auf der Ergebnisliste klicken wir auf eine Materialnummer um zu sehen, ob unser Sprung wie gewünscht funktioniert:



#### Das Resultat ist:



Wie erwartet, es funktioniert.



### Neue Datei-Formate (XLSX etc.) beim Erstellen einer Datei

Bis dato waren nur Text-Dateien mit Trennzeichen möglich – vom Typ .**TXT** oder .**CSV** – beim Download zum Frontend oder beim Sichern in eine Server-Datei. Gleiches galt für die Ergebnisse eines Scripts im Hintergrund beim Schreiben in eine Server-Datei.

Diese Einschränkung ist etwas gelockert worden. Folgende Datei-Formate sind jetzt zusätzlich verfügbar, sogar für die Hintergrund-Verarbeitung:

- **XLSX** Office Open XML-Arbeitsmappe

- XML XML-Kalkulationstabelle

- **MHTML** Einzelnes Webarchiv

Diese Datei-Formate können von einem Tabellenkalkulations-Programm direkt gelesen werden. Dateien dieser Art, die von einem Hintergrund-Script erstellt wurden, können auch als E-Mail-Anhang versendet werden, genauso wie die üblichen Text-Dateien.

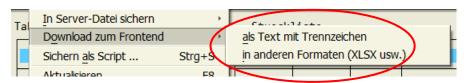
**ANMERKUNG**: Das erste Format (XLSX) ist nicht auf allen SAP-Versionen verfügbar, für die SE16XXL freigegeben ist.

### **Dialog-Modus**

Um diese neuen Formate zu unterstützen, ist die Menü-Struktur des Dialogs angepasst worden. Die bisherigen Funktionen sind gesplittet worden:



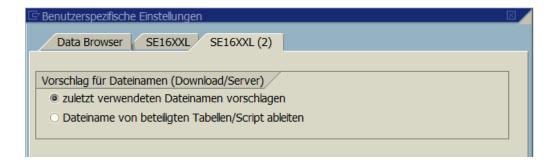
#### und



Die erste Unterfunktion entspricht der bisherigen Funktion. Die zweite Unterfunktion ist für die neuen Datei-Formate vorgesehen.



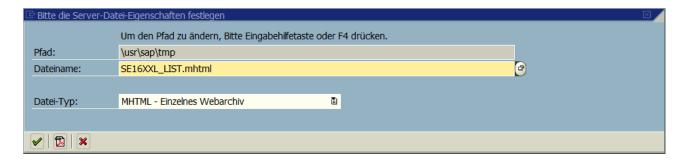
In diesem Zusammenhang ist eine **neue Option** den **Benutzer-Einstellungen** hinzugefügt worden:



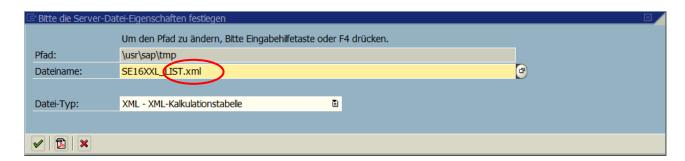
Die Standard-Einstellung verhält sich wie bisher, d.h. es wird der zuletzt verwendete Dateiname vorgeschlagen. Falls die zweite Alternative aktiviert wird, generiert das Programm den Vorschlag entweder anhand der beteiligtren Tabellen, oder, im Falle eines Scripts, anhand des Scriptnamens.

#### Ergebnisliste in eine Server-Datei sichern

Wenn die zweite Unterfunktion gewählt wird, erscheint folgendes Dialogfenster:



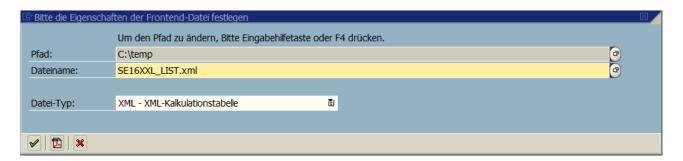
Wie man sehen kann, fehlen alle Zusatzoptionen, die sonst bei Text-Dateien mit Trennzeichen vorhanden sind. Das liegt daran, dass das Format (Datei-Typ) alles festlegt. Falls der Anwender ein anderes Format aus der Dropdown-Liste auswählt (das zuletzt verwendete wird jeweils vorgeschlagen), wird die **Dateiendung** automatisch angepasst:





#### Download der Ergebnisliste zum Frontend

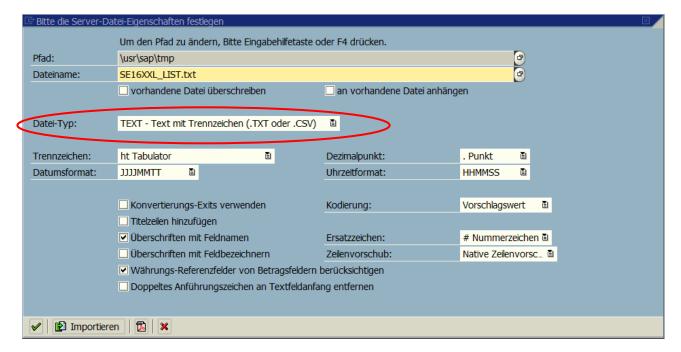
Diese Unterfunktion verhält sich ähnlich. Ein Dialogfenster wie folgt erscheint:



**ANMERKUNG:** Anders als beim Download einer Text-Datei, erscheint in diesem Fall kein Standard-Download-Popup. Der Grund dafür ist, dass keine explizite Codierung möglich ist, denn diese neuen Datei-Formate werden intern als **Binärdateien** behandelt. Somit kann der Download auf unsichtbare Weise durchgeführt werden.

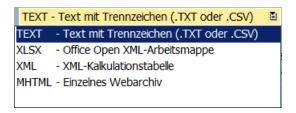
#### Hintergrund-Ergebnisse in Server-Datei speichern

Diese Option, verfügbar bei der Einplanung eines Scripts im Hintergrund, bietet nun auch die Möglichkeit, die Server-Datei in einem der neuen Formate anzulegen. Hierzu ist eine **Dropdown-Liste** zum entsprechenden Dialogfenster hinzugefügt worden:





Der Anwender kann nun einen anderen Datei-Typ als "Text mit Trennzeichen" auswählen:



In diesem Fall verschwinden die restlichen Optionen aus dem Dialogfenster und die Dateiendung wird entsprechend angepasst:



**ANMERKUNG**: Es ist nicht möglich, eine Datei mit einem der neuen Formate an eine bereits vorhandene Datei anzuhängen, weil das Ergebnis inkonsistent wäre. Deswegen steht nur die Option "überschreiben" zur Verfügung.

Die Option "ALV-Formatierung verwenden" betrifft Felder vom Typ NUMC. ALV zeigt solche Felder stets **ohne führende Nullen** an. Sollte die Datei führende Nullen enthalten, darf diese Option nicht aktiviert werden.



## Erstellung einer Server-Datei im Hintergrund ohne Verwendung des ABAP-Memory

Wenn ein Script im Hintergrund eine Server-Datei erstellt hat, wurde bisher die Ergebnisliste via **ABAP-Memory** (Export to Memory / Import from Memory) zur notwendigen Software übergeben. Dieser Mechanismus funktioniert bestens für moderate Datenmengen, jedoch, wie es meistens passiert, kümmert sich die Realität nicht um Annahmen, mit dem Resultat, dass ab und zu die ganze Prozedur vom System abnormal beendet wurde.

Um diesen Problemen entgegenzuwirken, wird die Erstellung einer Server-Datei im Hintergrund jetzt direkt durchgeführt, ohne grössere Datenmengen über das ABAP-Memory zu transferieren. Es wird trotzdem eine Obergrenze für die Daten geben, aber nicht so eng wie bisher.

### **Zum Anfang**

## Neue Titel-Einstellungen für /TFTO/SE16PARAM-basierte Transaktionen

Zwei neue Einstellungen stehen für Parameter-Transaktionen (basierend auf die Transaktion /TFTO/SE16PARAM) zur Verfügung, beide einem geeigneten Parameter zugeordnet:

Parameter	Beschreibung
SP_STTYP	Steuert den Titel der Selektionsmaske
SP_RTTYP	Steuert den Titel der Ergebnisliste

Beide Parameter können einen der folgenden Werte annehmen:

Wert	Bedeutung
space	Standardtitel – wie bisher
SN	Script-Name
SX	Script-Kurztext
TC	Transaktionscode
TX	Transaktionstext
TT	Transaktionscode + Text

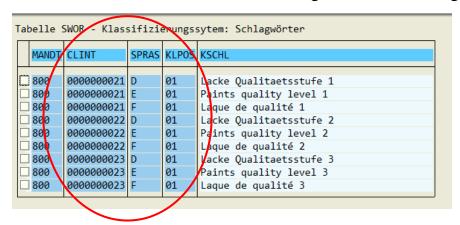


## Eingebettete Tabellenkalkulation berücksichtigt Einstellungen

Bis dato wurden von der *Tabellenkalkulation* (*eingebettet*) die Benutzer-Einstellungen in Bezug auf die Ausgabe-Formatierung ignoriert, mit Ausnahme von der Option "Feldname / Feldbezeichner" für die Spaltenüberschriften. Die Feldinhalte wurden immer auf die gleiche Weise ausgegeben.

Das hat sich jetzt geändert. Die Einstellungen in Bezug auf die Konvertierungsexits und auf die Anzeige (Standard / ALV) werden nun berücksichtigt. Ein Beispiel mit Tabelle SWOR, in der das Feld CLINT vom Typ NUMC ist, und Feld SPRAS mit einer Konvertierungsexit versehen ist (ISOLA), wird das Thema veranschaulichen.

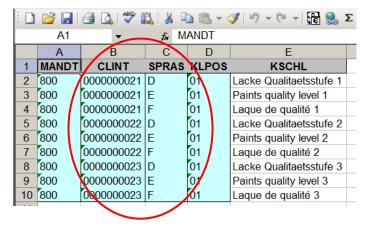
Eine kleine Auswahl von SWOR-Einträgen könnte wie folgt aussehen:



Die Benutzer-Einstellungen sind wie folgt belegt:



Das Ergebnis mit Tabellenkalkulation (eingebettet) ist wie folgt:

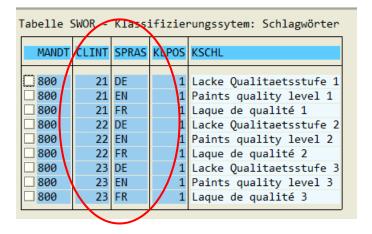




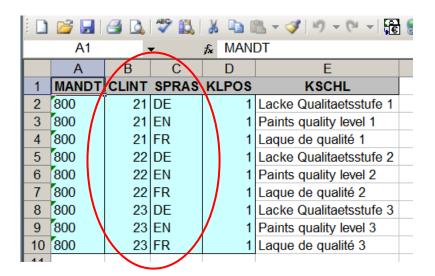
Nun schalten wir auf ALV mit Konvertierungsexits um:



Die ALV-Ausgabe sieht dann wie folgt aus:



Die eingebettete Tabellenkalkulation passt sich entsprechend an:



**ANMERKUNG**: Die führenden Nullen von Feld CLINT werden nicht wegen einer Konvertierungsexit unterdrückt (es gibt keine), sondern wegen des Typs NUMC, der von ALV immer auf diese Weise behandelt wird.



## Neue Ersetzungsvariablen für Titelzeilen

Drei neue Ersetzungsvariablen sind zu den bisherigen hinzugefügt worden, die für die Definition von Titelzeilen zur Verfügung stehen.

Diese neuen Variablen sind besonders für **Parameter-Trnsaktionen** interessant, die auf die Transaktion /TFTO/SE16PARAM basieren.

Variable	Beschreibung
&SCRNAME	Name des Scripts – erscheint auch im Falle einer Parameter- Transaktion basierend auf /TFTO/SE16PARAM. Die Variable &SCRIPT, die es weiterhin gibt, wird unter obigen Bedingungen unterdrückt.
&TCODE	Aktueller Transaktionscode
&TCTEXT	Kurztext des aktuellen Transaktionscodes

Bisher gab es nur die Variable &SCRIPT zum Anzeigen des Namens des aktuellen Scripts. Allerdings wurde deren Inhalt im Falle einer auf /TFTO/SE16PARAM basierenden Parameter-Transaktion gelöscht (....). Der Administrator kann nun stattdessen &SCRNAME verwenden, die nie gelöscht wird, und somit immer ausgegeben wird.

Darüber hinaus zeigen &TCODE und &TCTEXT Informationen in Bezug auf die aktuelle Transaktion. Damit wird es einfacher herauszufinden, wie eine bestimmte Ergebnisliste zustandegekommen ist.



## Mehrsprachige Script-Kurztexte

Bis dato konnte ein Script nur **einen** Kurztext in einer "undefinierten" Sprache haben, was ein Problem für größere Unternehmen mit Angestellten aus vielen Teilen der Welt darstellte. Diese Unzulänglichkeit ist nun behoben worden. Die Script-Kurztexte können jetzt in mehrere Sprachen übersetzt werden. In Abhängigkeit von der Logon-Sprache wird der passende Kurztext angezeigt. Nur wenn kein Kurztext für die aktuelle Sprache vorliegt, wird der Original-Kurztext (der bei der Erstellung vergebene) verwendet.

Die Übersetzung kann **implizit** oder **explizit** stattfinden.

## Implizite Übersetzung

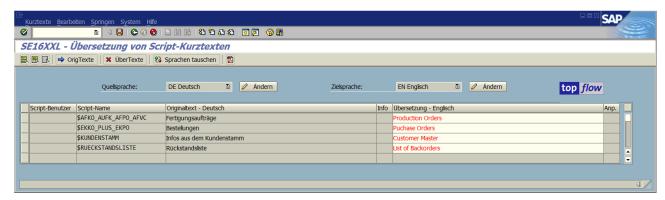
Eine implizite Übersetzung findet statt, wenn ein Script erneut gesichert wird, nachdem es erweitert wurde, z.B. durch Hinzufügen eines zusätzlichen Joins, und die aktuelle Logon-Sprache von der bei der Erstellung verwendeten abweicht. Eine ähnliche Situation ergibt sich, wenn Script-Attribute geändert werden, ebenfalls unter einer anderen Logon-Sprache. Allerdings ist eine solche implizite Übersetzung ein "Zufallsprozess", der nicht empfehlenswert ist, insbesondere wenn zahlreiche Scripts betroffen sind.

## **Explizite Übersetzung**

Der Script-Katalog bietet nun folgende Menü-Funktion:

Ausgewählte Scripts → Script-Kurztexte übersetzen

Daraufhin öffnet sich ein Dialog wie im folgenden Bild:



Für mehr Informationen lesen Sie bitte Übersetzung von Script-Kurztexten.

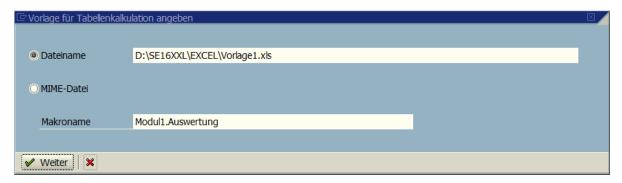


## **Tabellenkalkulation mit Vorlage**

Es ist nun möglich, eine **Vorlage-Datei** (und wahlweise einen **Makro**) beim Aufruf der Funktion "*Tabellenkalkulation* (*eingebettet*)" zu spezifizieren. Diese Funktion befindet sich im Menü

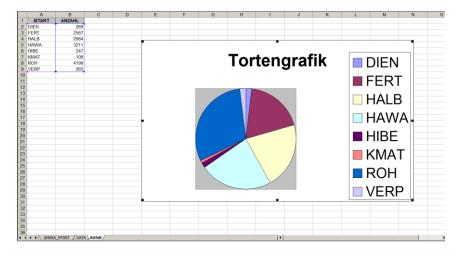
### Springen → Tabellenkalkulation mit Vorlage

Nachdem die Funktion ausgewählt wurde, ersscheint folgendes Dialogfenster:



Die Vorlage kann entweder eine Datei am Frontend sein oder im MIME-Repository (Transaktion SE80). Der Name eines auszuführenden Makros kann auch angegeben werden. Im obigen Beispiel sind die entsprechenden Namen eingetragen worden.

Die Vorlage ermöglicht es, die Tabellenkalkulation mit Zusatzinformationen zu ergänzen, in unserem Beispiel mit einer aus den selektierten Daten abgeleiteten Tortengrafik:

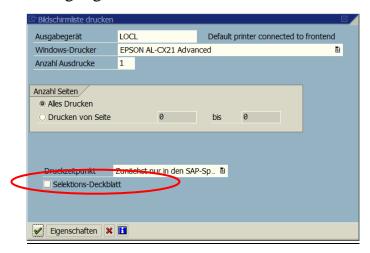


Der angegebene Makro muss so implementiert werden, dass er in der Lage ist, die Tortengrafik aus den von SE16XXL übergebenen Daten zu konstruieren.

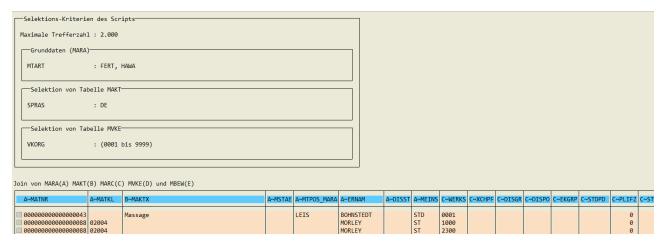


# Ausgabe von Selektions-Kriterien beim Drucken

Wenn die Ergebnisliste von SE16XXL gedruckt wird, d.h. zum SAP-Spool geleitet, entweder mithilfe der Funktion *Programm* → *Ausführen* + *Drucken* auf der Selektionsmaske, oder *Liste* → *Drucken* bei der Ergebnisliste, oder beim Leiten eines Hintergrund-Ergebnisses zum SAP-Spool, erscheint jedesmal ein Dialog-Fenster zur Festlegung der Druck-Parameter:



Durch Setzen der Option Selektions-Deckblatt kann verlangt werden, dass die benutzten Selektions-Kriterien zur Spool-Ausgabe hinzugefügt werden. Die Spool-Liste könnte dann wie folgt aussehen:



Diese zusätzliche Information kann für Empfänger der Liste hilfreich sein, da sie dadurch besser verstehen können, welche Art von Daten selektiert wurde.

Die Selektions-Kriterien können auch bei der Ergebnisliste angezeigt werden. Hierzu gibt es die Menü-Funktion *Liste*  $\rightarrow$  *Selektions-Kriterien anzeigen*.



# Leere Hintergrund-Ergebnisse wahlweise ignoriert

In manchen Fällen liefert ein Script keine Ergebnisse, d.h. es endet mit folgender Meldung:

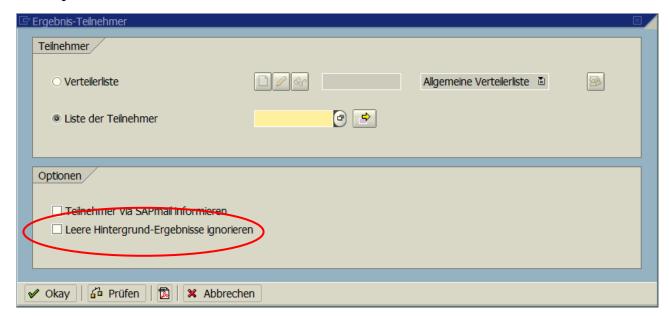


Dies kann auch bei einem Lauf im Hintergrund passieren. Normalerweise stellt das kein Problem dar.

Falls allerdings die Hintergrund-Resultate für eine Serie von Teilnehmern **veröffentlicht**, oder als **E-Mail-Anhang** an verschiedene Empfänger versendet werden sollten, kann das Fehlen eines echten Ergebnisses von den Teilnehmern/Empfängern als lästig empfunden werden. Um diese Unzulänglichkeit zu beseitigen, ist eine **spezielle Option** für beide Fälle eingeführt worden.

### Erster Fall – Ergebnis-Teilnehmer

Die Option befindet sich bei Ergebnis-Teilnehmer



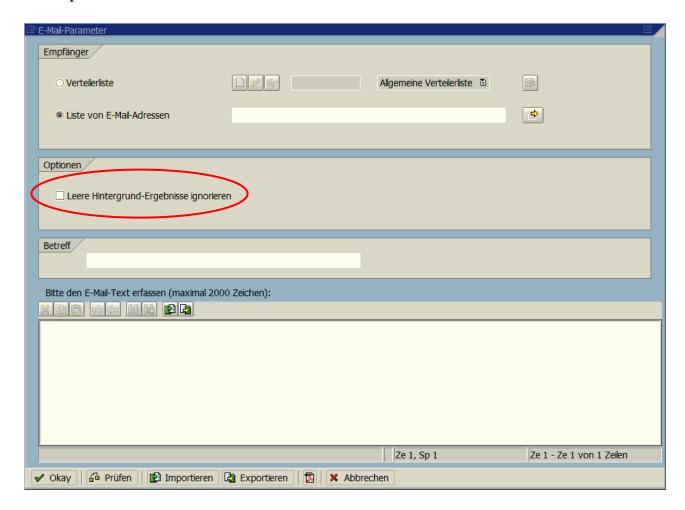
Falls diese Option aktiviert wird, wird ein leeres Ergebnis nicht veröffentlicht:

- Es wird kein Link auf das leere Ergebnis für die Teilnehmer erstellt.
- Die Teilnehmer werden nicht mittels SAPmail informiert.



# **Zweiter Fall – E-Mail-Anhang**

Die Option befindet sich bei E-Mail-Parameter



Falls diese Option aktiviert wird, wird ein leeres Hintergrund-Ergebnis ignoriert. Das bedeutet, dass keine E-Mail mit Anhang an die festgelegten Empfänger versendet wird.

**ANMERKUNG:** Obiges Verhalten ist optional, da es manchmal notwendig sein kann, die Teilnehmer/Empfänger über das Fehlen eines echten Resultats zu informieren.



# Einige neue Funktionen für Administratoren

Die nachfolgenden Themen sind nur für Administratoren relevant. Sie werden an dieser Stelle nur erwähnt um normale SE16XXL-Anwender über mögliche, für sie relevante, Begleiterscheinungen zu informieren.

### Einschränkungen für Funktionscodes

Der SE16XXL-Administrator kann nun die Verfügbarkeit bestimmter Funktionscodes einschränken – für sämtliche Benutzer, für gewisse Benutzer-gruppen oder für einzelne Benutzer. Anhand eines Beispiels kann das Ganze veranschaulicht werden.

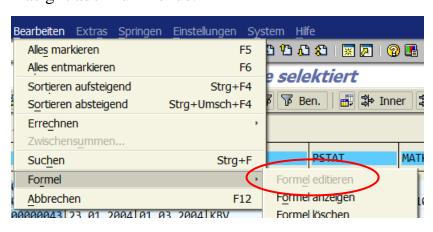
Die Anwendungsleiste der Ergebnisliste sieht normalerweise aus wie folgt:

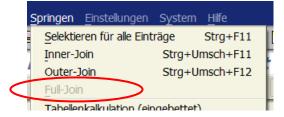


Falls die Funktionscodes für "*Formel editieren*" und "*Full-Join*" für den aktuellen Anwender unzulässig wären, würde sie stattdessen wie folgt erscheinen:



#### Das gilt auch für Menüs:







### Berechtigungsobjekte anstelle von Rollen

In früheren SAP-Versionen war es nicht gestattet, Berechtigungsobjekte in einem Kunden-Namensraum zu definieren. Deshalb wurden Berechtigungs-Prüfungen in SE16XXL mithilfe von leeren, den Anwendern zugeordneten, Rollen durchgeführt. Dieser Mechanismus wurde von SAP als Ersatz für eine "normale" Berechtigungs-Prüfung empfohlen (siehe SAP-Hinweis 104010).

Diese Einschränkung ist in neueren SAP-Versionen aufgehoben worden. Es ist damit möglich, eine Serie von Berechtigungsobjekten im Namensraum /TFTO/ zu definieren. Diese können nun in Alternative zu, oder in Kombination mit den bisherigen Rollen verwendet werden. Der normale Anwender merkt davon nichts. Nur falls die eigene Benutzer-Definition (Transaktion SU01) betrachtet wird, kann es verwunderlich sein, dass trotz Fehlen von /TFTO/-Rollen, die SE16XXL-Funktionalität dennoch zur Verfügung steht.

#### /TFTO/SE16PARAM-Transaktionen ohne Selektionsmaske

Parameter-Transaktionen basierend auf der Transaktion /**TFTO/SE16PARAM** sind schon seit langem möglich. Sie sind jeweils mit einem SE16XXL globalen Script verbunden. Wenn eine solche Transaktion aufgerufen wird, erscheint zunächst die Selektionsmaske des betroffenen Scripts, gefolgt von der Ergebnisliste, falls es Treffer gibt. Dies ist das normale Verhalten.

Auf Wunsch einiger Unternehmen ist es nun möglich, eine Parameter-Transaktion zu definieren, die die Selektionsmaske unterbindet – die Ergebnisliste erscheint dann unmittelbar nach dem Transaktions-Aufruf.



# Neue Funktionen zum Einfügen von Sonder-Spalten

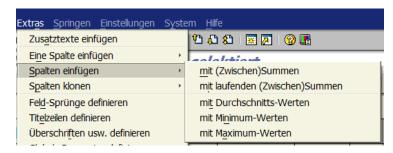
In der SE16XXL-Ergebnisliste stehen im Menü *Extras* neue Funktionen zur Verfügung. Sie ermöglichen das Einfügen von neuen Spalten ohne Verwendung einer Formel, die für Anwender ohne ABAP-Kenntnisse ein Problem darstellt. Sie sind in drei Gruppen unterteilt worden:

### Eine Spalte Einfügen ...



Die erste Funktion ("... *mit Anzahl sortierter Sätze*") war bereits in der vorigen Version unter dem Namen "*Sortierte Sätze zählen*" vorhanden und ist lediglich umbenannt worden. Die restlichen zwei Funktionen sind hingegen neu. Siehe Laufende Nummer Einfügen und Prozentsatz Einfügen.

# Spalten Einfügen ...



Für weitere Informationen siehe **Spalten Einfügen**.

# Spalten Klonen ...

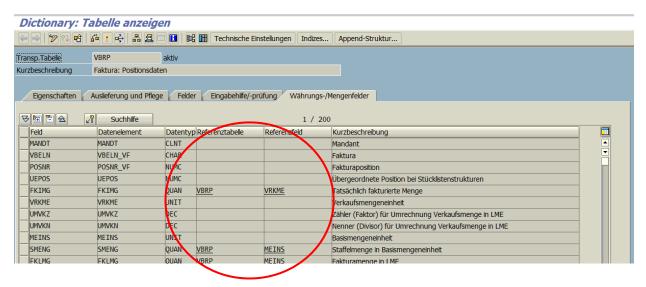


Für weitere Informationen siehe Spalten Klonen.



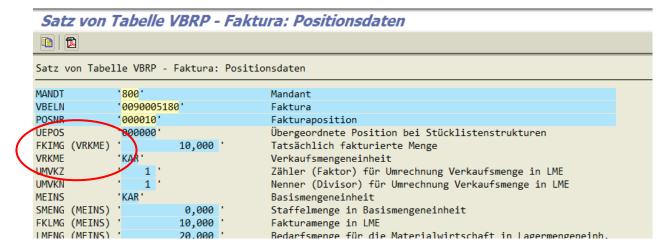
#### Referenzfelder Sichtbar

Ein Großteil der im vorhergehenden Thema erwähnten Funktionen haben mit **Beträgen** (einem Währungsfeld zugeordnet) oder mit **Mengen** (mit Maßeinheitsfeld) zu tun. Solche zugeordneten Felder werden "**Referenzfelder**" genannt und sind in der DDIC-Definition einer Tabelle unter "Währungs-/Mengenfelder" zu sehen:



Dis dato war die Beziehung zwischen einem Betrags-/Mengenfeld und seinem Referenzfeld im SE16XXL-Umfeld nicht sofort ersichtlich. Sie wurde nur indirekt nach dem Hinzufügen von ALV-(Zwischen)Summen in der Liste angedeutet.

Dieses Manko ist nun beseitigt worden – die Referenzfelder sind jetzt an vielen Stellen in SE16XXL direkt sichtbar, z.B. in der Detailanzeige einer Zeile:



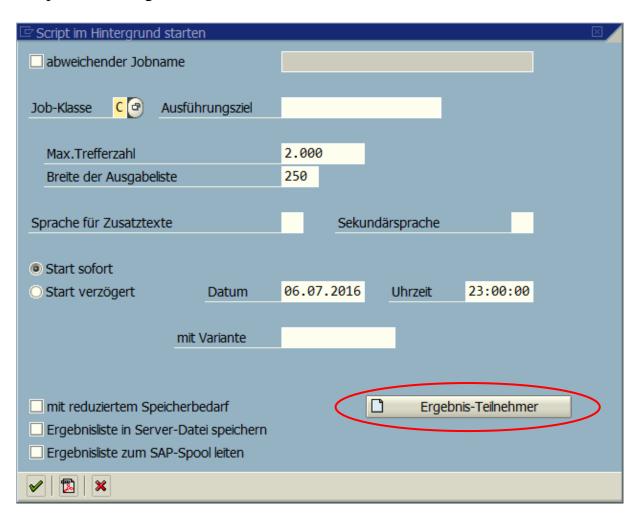
Für weitere Informationen siehe Referenzfelder.



# Teilnehmer an Hintergrund-Ergebnissen

Bis dato stand das Ergebnis eines **globalen** Hintergrund-Scripts nur jener Person zur Verfügung, die es eingeplant hatte, es sei denn, es wurde in eine Server-Datei gesichert oder zum SAPspool geleitet. Dieses Manko ist nun beseitigt worden. Das Ergebnis kann jetzt für andere Anwender, "**Teilnehmer**" genannt, verfügbar gemacht werden. Die entsprechende Funktionalität muss allerdings vom Administrator **aktiviert** werden. Ferner benötigt der Benutzer, der seine Ergebnisse teilen will, eine spezielle Berechtigungs-Rolle.

Sind diese Voraussetzungen erfüllt, bietet das Dialog-Fenster zum Einplanen eines Scripts im Hintergrund eine zusätzliche Schaltfläche an:



Für weitere Informationen lesen Sie bitte Teilnehmer an Hintergrund-Ergebnissen.

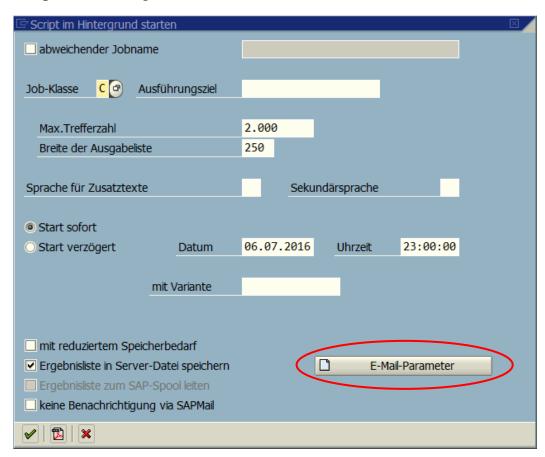


# Hintergrund-Server-Datei als E-Mail-Anhang

Es ist seit langem möglich, das Ergebnis eines Hintergrund-Scripts in eine Datei auf dem Applikations-Server zu sichern ("Server-Datei"). Die Idee war, dass durch die Freigabe des Laufwerks die restlichen Anwender Zugriff auf die Datei erlangen würden. In nachhinein stellte sich jedoch heraus, dass in zahlreichen Firmen etliche Benutzer keinen Zugriff auf die Server-Datei bekommen. Dennoch benötigen Sie die aus Hintergrund-Scripts stammenden Informationen.

Um diese Unzulänglichkeit zu überwinden, ist es nun möglich, eine Hintergrund-Server-Datei als **E-Mail-Anhang** zu **versenden**. Diese Funktionalität ist allerdings nur verfügbar, wenn sie vom Administrator **aktiviert** wurde. Ferner muss der Einplaner mit einer entsprechenden Berechtigungs-Rolle ausgestattet sein. Die E-Mail-Adressen der Empfänger müssen ebenfalls mehrere Bedingungen erfüllen.

Sind die Voraussetzungen erfüllt, bietet das Dialog-Fenster zum Einplanen eines Scripts im Hintergrund eine zusätzliche Schaltfläche an:



Für weitere Informationen lesen Sie bitte Server-Datei als E-Mail-Anhang.



# **Default ALV-Layouts für Scripts**

Die Ergebnisliste eines Scripts kann auf verschiedene Weise modifiziert werden – durch die Auswahl anderer Ausgabe-Felder, durch Neuanordnung der Spalten und auch, in ALV (List und Grid), durch das Laden eines bestimmten ALV-Layouts. Bis dato musste dies manuell geschehen, beim Ausführen des Scripts. Klarerweise kann das Scripts so ausgelegt werden, dass das "Standard"-Layout automatisch eingestellt wird. Das setzt aber voraus, dass alle Anwender sich mit demselben Layout zufrieden geben, was nicht immer der Fall ist.

Um noch mehr Flexibilität anbieten zu können, kann deshalb nun jeder Benutzer für jedes Script ein **Default-ALV-Layout** festlegen – die Zuordnung ist **individuell**, d.h. für **dasselbe** globale Script können zwei Anwender **zwei unterschiedliche** ALV-Layouts festlegen. Eine Zuordnung ist auch für benutzerspezifische Scripts möglich.

Als Konsequenz wird bei jeder Ausführung eines Scripts das Default-Layout, falls definiert, automatisch geladen. Selbstverständlich kann die Zuordnung jederzeit geändert oder zurückgenommen werden. Es muss aber erwähnt werden, nachdem ALV-Layouts nicht von SE16XXL sondern von ALV selber verwaltet werden, dass weder die eigentlichen Layouts, noch die Default-Festlegungen mit den zugehörigen Scripts downgeloaded oder transportiert werden können.

ANMERKUNG: Es gibt keine direkte Beziehung zwischen ALV-Layouts und Scripts. ALV betrachtet Layouts in Verbindung zu Reports, d.h. jeder Report hat seine eigenen Layouts. Diese Logik funktioniert prima mit "normalen" Reports die eine oder mehrere ALV-Listen erstellen. SE16XXL ist jedoch kein "normaler" Report – vielmehr handelt es sich um ein Programm, das alle Arten von Listen mit allen möglichen Strukturen produziert. Sollten die ALV-Layouts mit diesem einen Programm in Verbindung stehen, würde der Anwender unter allen definierten Layouts (für alle möglichen Tabellen) das Richtige aussuchen müssen. Aus diesem Grund sind in SE16XXL die Layouts nicht mit dem Programm, sondern mit der Kombination von DDIC-Tabellen, die die Ergebnisliste ausmachen. Zum Beispiel teilen sich alle Scripts, die eine Join-Liste von VBAK und VBAP erzeugen, dieselben ALV-Layouts. Wenn der Benutzer eine derartige Liste erweitert, indem er einen Join mit VBUK durchführt, ist die resultierende Liste implizit mit den Layouts verbunden, die einen Join von VBAK, VBAP und VBUK betreffen.

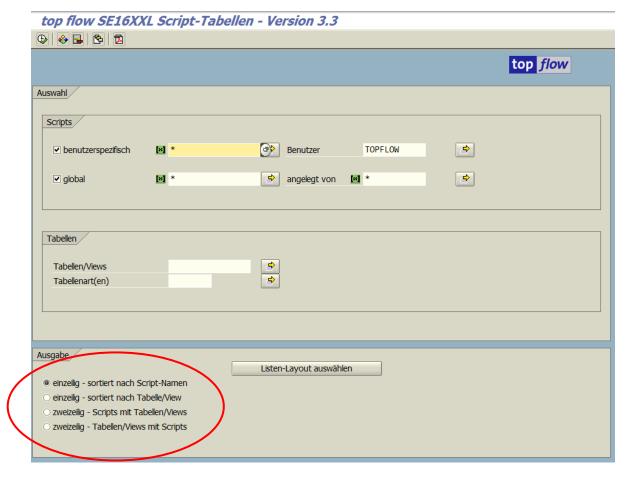
Für weitere Informationen über diese Thema lesen Sie bitte Default-Layouts.



# **Liste von DDIC-Tabellen/Views in Scripts**

Gelegentlich haben sich sowohl Anwender als auch Administratoren gefragt, welche Datenbank-Tabellen und Views werden wohl von den vielen SE16XXL-Scripts eines Mandanten verwendet. Eine konkrete Antwort auf diese und ähnliche Fragen war bis dato sehr schwierig zu finden, angesichts der Tatsache, dass im Script-Katalog nur die erste Tabelle eines Scripts explizit zu sehen ist. Die intern verwendeten Tabellen sind nur in der Detailanzeige der Script-Operationen sichtbar, die für jedes Script getrennt aufzurufen ist. Eine langwierige und fehleranfällige Beschäftigung.

Diese Unzulänglichkeit ist nun beseitigt worden. Ein Tool ist jetzt verfügbar, das vier unterschiedliche Listen produziert – alle in Bezug auf dieses eine Thema. Die Selektionsmaske sieht wie folgt aus:



Für detailliertere Informationen lesen Sie bitte Liste von Script-Tabellen.



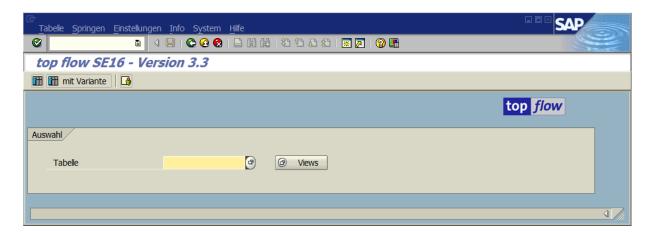
### **Neue Transaktion /TFTO/SE16**

Eine der Gründe warum einige Firmen von Standard-SE16 auf SE16XXL umsteigen, sind die umfassenden Berechtigungsprüfungen der letzteren Transaktion.

Das bedeutet jedoch, dass die beteiligten Benutzer Zusatzfunktionen zur Verfügung gestellt bekommen, die aus der Perspektive der Firma nicht unbedingt wünschenswert sind.

Deshalb ist eine "abgespeckte" Version von SE16XXL, /TFTO/SE16 genannt, zur Verfügung gestellt worden. Sie bietet sämtliche Berechtigungsprüfungen wie SE16XXL an, jedoch ohne Zusatzfunktionen, d.h. keine Scripts, keine Joins, keine Formeln usw.

Die Selektionsmaske ist wie folgt:



In der Ergebnisliste steht als "komplexe" Operation nur die Menü-Funktion

# Extras → Zusatztexte einfügen

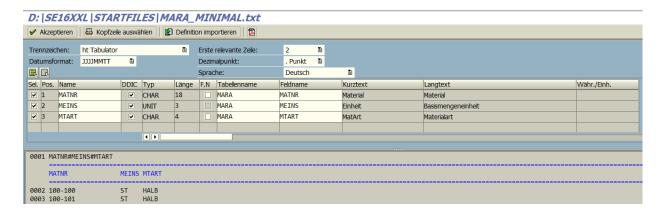
zur Verfügung, die erläuternde Texte zu den Originalspalten hinzufügt.



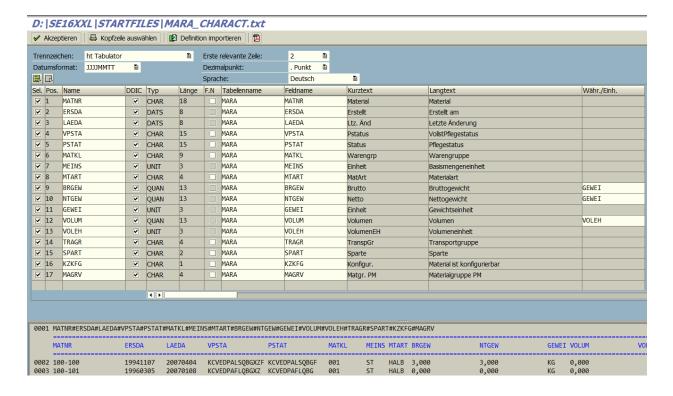
# Verbesserung der Definition einer Vorgabedatei

Bis dato war bei der Definition einer Vorgabedatei der obere Teil der Maske fix. Wenn die Vorgabedatei mehr als 9 Spalten aufwies, musste der Anwender rauf und runter blättern, um alle zu sehen. Diese Einschränkung ist jetzt aufgehoben worden. Die zwei Teile der Maske sind nun variabel, die Trennlinie kann bewegt werden. Standardmäßig versucht das Programm soviele Spalten wie möglich zu zeigen, ohne dabei den unteren Teil zu sehr zu benachteiligen.

#### Beispiel 1 – Drei Spalten:



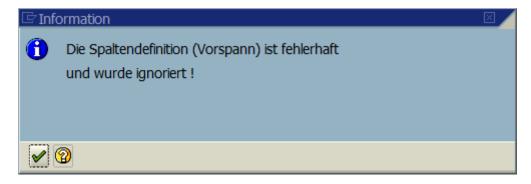
#### Beispiel 2 – Siebzehn Spalten:





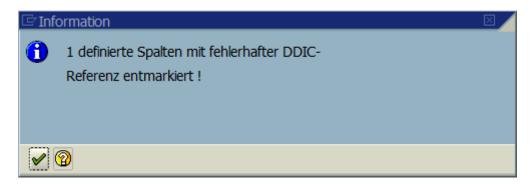
### **Zweite Verbesserung**

Bis dato, falls der Datei-Vorspann mit den Spaltendefinitionen fehlerhaft war, aufgrund von falschen DDIC-Zuordnungen (z.B. beim Upload einer Vorgabedatei, die auf einem anderen System erstellt worden war), wurden sämtliche Definitionen ignoriert:

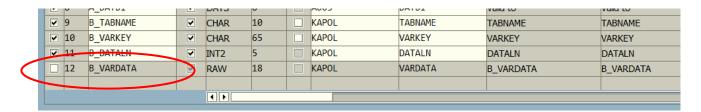


Diese Reaktion hat sich als zu hart für den Benutzer herausgestellt, besonders wenn die Vorgabedatei sehr viele und möglicherweise komplizierte Spalten aufweist.

Deshalb beschränkt sich nun das Programm auf die Deaktivierung der fehlerhaften Spalten, und bietet damit dem Anwender die Möglichkeit, das Problem auf vernünftige Weise zu lösen:



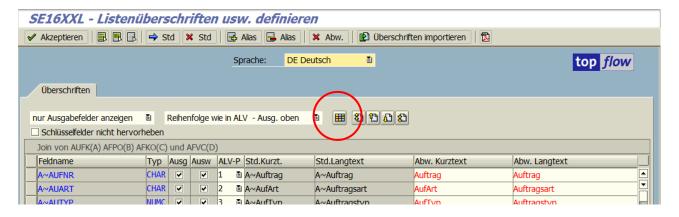
Die Spalten-Definition zeigt die Details:



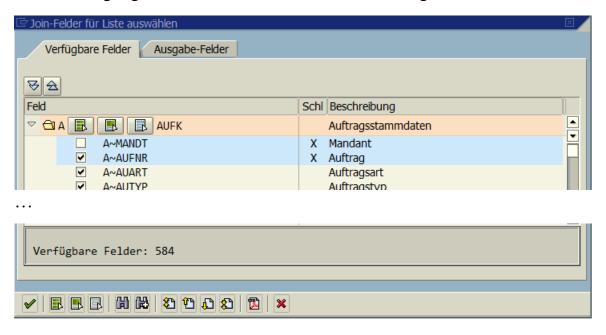


# Verbesserung der Definition der Überschriften

U. a. bietet die Maske zur Definition von Listenüberschriften die Möglichkeit, die Anordnung der Ausgabe-Spalten zu ändern. Bis dato konnte dies auf zwei Arten durchgeführt werden, entweder durch Änderung der ALV-Position mithilfe der entsprechenden Drop-Down-Liste, oder durch Drag & Drop im unteren Bereich der Maske. Beide Methoden sind relativ umständlich. Deshalb wurde jetzt eine dritte Art hinzugefügt, um die Operation zu erleichtern. Die Schaltfläche ist nun auf der Maske verfügbar – sie ruft denselben Dialog auf, der zum selben Zweck in der Ergebnisliste verwendet wird:



Nach Betätigung von erscheint der Auswahl-Dialog:

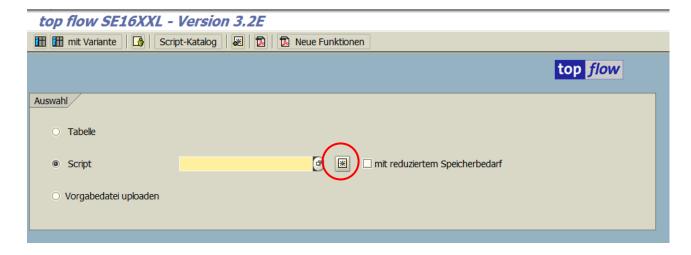


Der Rest dürfte selbsterklärend sein.



# **Script-Favoriten**

**Script-Favoriten** oder nur **Favorites** sind eingeführt worden, um die Anwender zu befähigen, den am häufigsten verwendeten Scripts eine gewisse **Struktur** zu verleihen. Favoriten können anstelle der üblichen F4-Hilfe zur Auswahl eines Scripts benutzt werden. Eine entsprechende Ikonen-Schaltfläche ist zu diesem Zweck auf der Einstiegsmaske von SE16XXL zur Verfügung gestellt worden:



Nach Betätigung erscheinen die Favoriten des Anwenders:



Eine Darstellung in Listenform ist auch verfügbar:





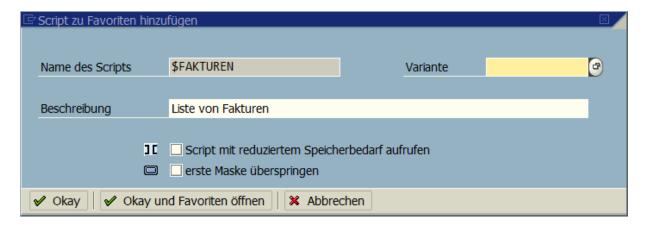
Obigen Bildern kann entnommen werden, dass Ordner und Unterordner in den Favoriten angelegt werden können, die dadurch die Scripts nach verschiedenen Kriterien gruppieren. Sogenannte **Globale Favoriten** können ebenfalls inkludiert werden. Es handelt sich um Sammlungen von globalen Scripts, die separat von den Verantwortlichen für globale Scripts aufgebaut und gepflegt werden. Inkludierte globale Favoriten sind lediglich eine **Referenz** auf den echten Ordner. Werden die originalen globalen Favoriten geändert, spiegeln sich diese Änderungen sofort in allen benutzerspezifischen Favoriten wider, die sie inkludieren.

### **Favoriten-Pflege**

Man kann auf verschiedene Weisen ein Scripts zu den eigenen Favoriten hinzufügen. Die einfachste ist aus der SE16XXL-Einstiegsmaske:



Wenn betätigt wird, erscheint ein geeignetes Dialog-Fenster:



Für mehr Informationen zu diesem Thema lesen Sie bitte Script-Favoriten.

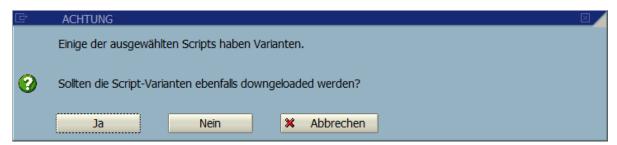


# Verbesserungen des Script-Katalogs

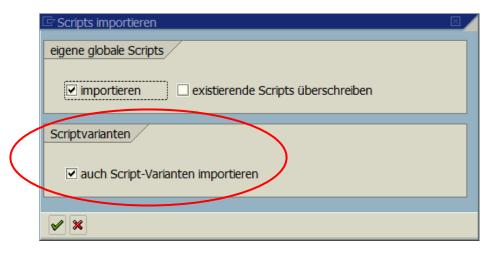
### **Down- und Upload von Scripts samt Varianten**

Bis dato mussten Script-Varianten getrennt von ihren Scripts down- und upgeloaded werden. Diese Funktionalität ist weiterhin verfügbar. Es ist jedoch ab jetzt möglich, beim Download von Scripts im Script-Katalog, die zugehörigen Script-Varianten ebenfalls downzuloaden.

Das wird erreicht, indem folgende Aufforderung mit beantwortet wird:



Auf ähnliche Weise können die zugehörigen Varianten beim Importieren von Scripts aus einer zuvor erstellen Download-Datei mit importiert werden:



Existierende Script-Varianten eines Ziel-Scripts werden **überschrieben** (auch die geschützten). Vorhandene Varianten ohne Upload-Gegenstück bleiben unverändert. Wir die Option **nicht** angekreuzt, bleiben Script-Varianten von existierenden Ziel-Scripts unverändert.

WICHTIG: machen Sie von dieser Option keinen Gebrauch, falls geschützte Varianten von anderen Benutzern unverändert bleiben sollten. Verwenden Sie bitte stattdessen das bisherige Verfahren (*Springen → Script-Varianten* ... auf der Einstiegsmaske von SE16XXL).



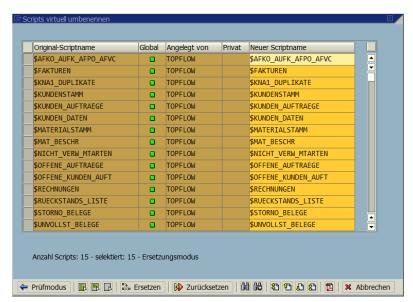
### Scripts beim Upload virtuell Umbenennen

Wenn eine Sicherungs-Datei upgeloaded wird, ist es nun möglich, die Namen der enthaltenen Scripts auf virtuelle Art abzuändern. Der Anwender kann entweder die Namen einzeln überschreiben, oder **globale Veränderungen** durchführen, wie einen **Präfix** oder **Suffix** hinzufügen, oder einen String durch einen anderen **ersetzen**. Danach verhält sich der Upload so, als ob die in der Datei enthaltenen Scripts von vornherein die modifizierten Namen gehabt hätten. Die Datei selber bleibt unverändert. Die Änderungen finden nur im virtuellem Speicher statt.

Um diese Funktionalität in Anspruch nehmen zu können, sollte folgende Option beim Upload im Script-Katalog angekreuzt werden:



Es erscheint daraufhin folgendes Dialog-Fenster:



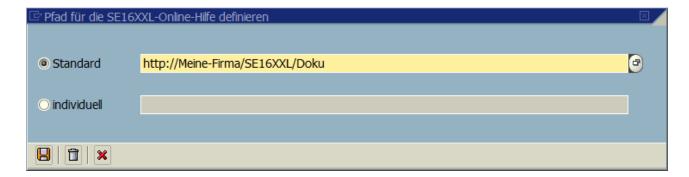
Mit Hilfe der Funktion können unterschiedliche globale Veränderungen herbeigeführt werden.

Für mehr Details lesen Sie bitte Scripts Virtuell Umbenennen beim Upload.



# **Internet-Pfad** (http://) als **Dokumentations-Pfad**

Wenn ein Pfad für die Dokumentation festgelegt wird (mithilfe der menü-Funktion *Springen → Pfad für Dokumentation*), ist es nun möglich, einen Internet-Pfad in der Form http://... bzw. https://... zu spezifizieren:



#### **ANMERKUNGEN:**

- Es wird für solche Pfade keine Prüfung auf Existenz vorgenommen. Falls die URL ungültig ist, meldet der Internet-Browser einen Fehler, wenn die Dokumentation abgerufen wird.
- Diese Art von Pfad eignet sich nur für MS Internet Explorer® oder Mozilla Firefox® als Dokumentations-Tools und sollte in Verbindung mit Adobe Reader® nicht verwendet werden.
- Es kann u.U. notwendig sein, in Mozilla Firefox® ein PDF-Plug-In zu installieren, damit die externen Links in der Doku korrekt funktionieren.

Diese Erweiterung sollte großen Firmen das Anbieten eines zentralisierten Pfades für alle Anwender erleichtern, auch wenn sie auf unterschiedlichen Servern und evtl. in verschiedenen Ländern arbeiten.



# Sammelrollen für Scripts

Um den Zugriff auf globale Scripts einzuschränken, können jedem globalen Script eine **Ausführungs**- und eine **Pflege**-Rolle zugeordnet werden. Nur Anwender, die mit der entsprechenden Rolle ausgestattet sind, dürfen das jeweilige Script ausführen bzw. pflegen. Diese Funktionalität ist seit langem verfügbar, allerdings durften bisher nur Einzelrollen für diese Zwecke eingesetzt werden. Ab Version 3.2E dürfen auch **Sammelrollen** verwendet werden.

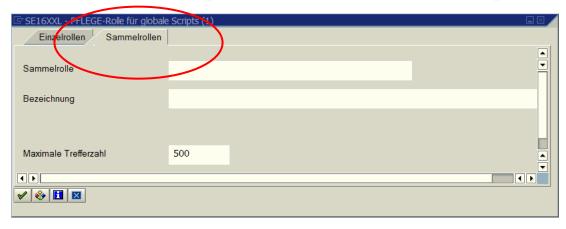
Die entsprechende F4-Hilfe ist diesbezüglich erweitert worden:



kennzeichnet eine Einzelrolle

kennzeichnet eine Sammelrolle.

Die Einstellungen für Script-Berechtigungen sind ebenfalls angepasst worden:





# Spezielle Berechtigungs-Prüfungen

Das ist eigentlich mehr Sache des Administrators als des normalen Anwenders. Es kann aber für normale Benutzer nicht schaden, einen kleinen Blick hinter die Kulissen zu werfen.

Bisher konnten in SE16XXL die selektierten Sätze "normalen" Berechtigungs-Prüfungen unterzogen werden, d.h. bestimmte Felder – in den Einstellungen unter "Berechtigungen auf Satzebene" hinterlegt – wurden gegen entsprechende Berechtigungs-Objekte geprüft. Jeder Satz wurde unabhängig von den anderen inspiziert, die beteiligten Kriterien aus dem Satz selber hergeholt. Das hat bisher gut funktioniert, und wird es auch in Zukunft tun.

Für manche Firmen sind solche Prüfungen auf Satzebene nicht ausreichend – sie benötigen komplexere Berechtigungs-Prüfungen, die **mehr als eine Tabelle** und **mehr als einen Satz** betreffen. Wie man sich leicht vorstellen kann, wäre es fast unmöglich, ein tabellengesteuertes Customizing aufzustellen, das sämtliche Kombinationen von Tabellen und Feldern berücksichtigen könnte. Derart allmächtige Konstrukte sind meistens in der Lage, alle Situationen zu bewältigen, mit Ausnahme derer, die gerade anstehen. Aus diesem Grund ist ein praxisnaherer Ansatz gewählt worden – für jede **Spezielle Berechtigungs-Prüfung** wird eine individuell kodierte Routine zur Verfügung gestellt. Die Tabellen/Feld Kombinationen, die für die Routine aktiviert werden sollen, werden dann in einem separatem Customizing festgelegt.

Fürs Erste sind zwei Prüfungen implementiert worden und stehen zur Verfügung:

- 1) Ein bestimmter Satz mit einer Kundennummer (**KUNNR**) soll nur dann angezeigt werden, wenn der Benutzer die Berechtigung für mindestens einen der Buchungskreise (**BUKRS**) besitzt, die in den zugeordneten **KNB1**-Sätze enthalten sind.
- 2) Eine ähnliche Prüfung in Bezug auf die Lieferantennummer (**LIFNR**) und auf die Buchungskreise der zugeordneten **LFB1**-Sätze.

Für normale Anwender verlaufen solche Prüfungen unbemerkt. Die entsprechenden Meldungen verraten nicht, welche Art von Berechtigungs-Prüfung durchgeführt wurde:

81 Sätze wegen fehlender Berechtigung ignoriert

Für mehr Informationen lesen Sie die Dokumentation (für Administratoren).



# Erläuternde Zusatztexte

Es ist nun möglich, die Ergebnisliste mit zusätzlichen Texten zu ergänzen. Nehmen wir z.B. an, dass die Ergebnisliste Einträge von Tabelle VBAK (Verkaufsbeleg: Kopfdaten) enthält. Betrachten wir unter den vielen Feldern dieser Tabelle VBTYP, AUART und VKORG. Der Inhalt dieser Felder besteht meistens aus einer Kombination von Buchstaben und/oder Zahlen, die nur für den technisch versierten Anwender verständlich sind. Ein kleiner Ausschnitt dieser Liste könnte wie folgt aussehen:



Es wäre für Normalanwender viel besser, wenn diese kryptischen Felder von ihren erläuternden Texten begleitet wären, wie die nachfolgende Liste veranschaulicht:



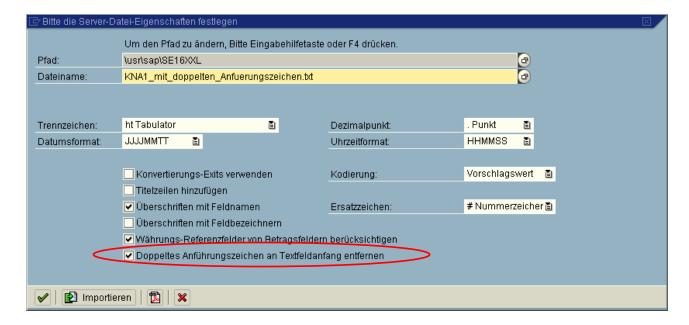
Diese Funktionalität ist nun unter *Extras*  $\rightarrow$  *Zusatztexte einfügen* verfügbar.

Für weitere Informationen lesen Sie bitte Erläuternde Zusatztexte.

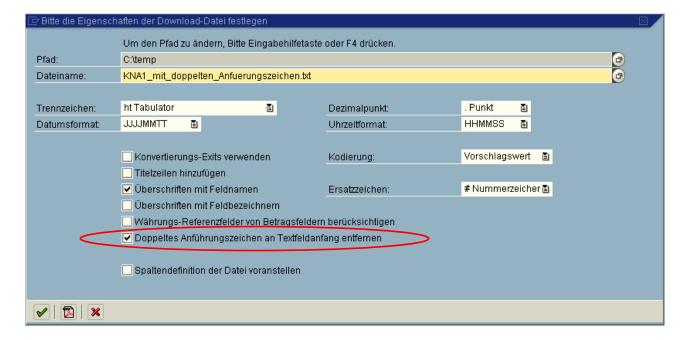


# Option für Doppelte Anführungszeichen

Textfelder beginnennd mit einem doppelten Anführungszeichen (") können in Microsoft® Excel zu Problemen führen. Um solche Probleme zu vermeiden ist eine neue Option beim Sichern der Ergebnisliste in eine Server-Datei verfügbar:



Die gleiche Option ist auch verfügbar wenn eine Ergebnisliste zum Frontend downgeloaded wird (mit der Menüfunktion *Liste* → *Download zum Frontend*):





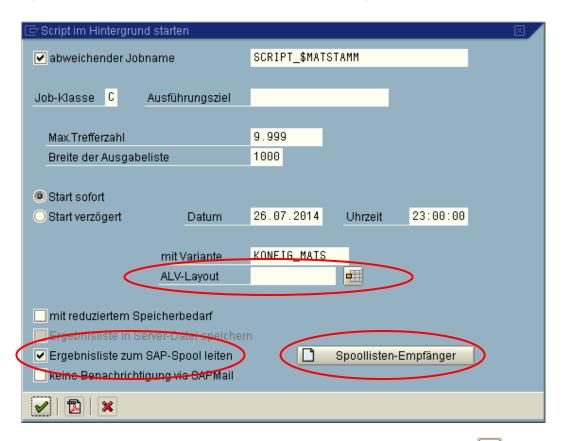
### Table Wizard in SE16XXL umbenannt

Der Table Wizard wurde schon immer von den meisten Anwendern SE16XXL genannt, wohl in Anlehnung an die Haupt-Transaktion /TFTO/SE16XXL. Aus diesem Grund ist das Programm nun offiziell in "**SE16XXL**" umbenannt worden. Auf allen Masken, Listen und in der Dukumentation ist diese Umbenennung durchgeführt worden. Nur die "Historie der neuen Funktionen" ist unverändert geblieben.



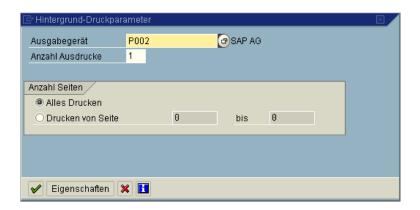
# Ergebnisliste eines Hintergrund-Scripts zum SAP Spool

Ab jetzt ist es möglich, die Ergebnisliste eines Scripts im Hintergrund zum SAP-Spool zu leiten. Das Dialogfenster wurde entsprechend erweitert:



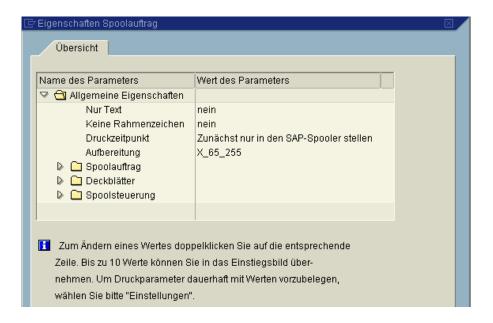
In diesem Zusammenhang können sowohl ein ALV-Layout ( ) als auch ein Spoollisten-Empfänger ( Spoollisten-Empfänger ) spezifiziert werden. Letzterer funktioniert genauso wie bei der Standard-SAP-Jobeinplanung.

Wenn (Weiter) betätigt wird, erscheint das Standard-Dialogfenster für die Druckparameter:



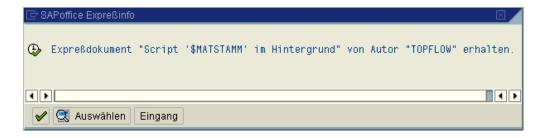


Mithilfe von Eigenschaften können die Parameter nach Wunsch gesetzt werden:



WICHTIG: mehr als 255 Spalten können nicht gedruckt werden.

Falls die Benachrichtigung via SAPMail nicht deaktiviert wurde, wird am Ende des Scripts ein Popup auf dem Bildschirm erscheinen:



Ein Klick auf und das SAPMail öffnet sich sofort:



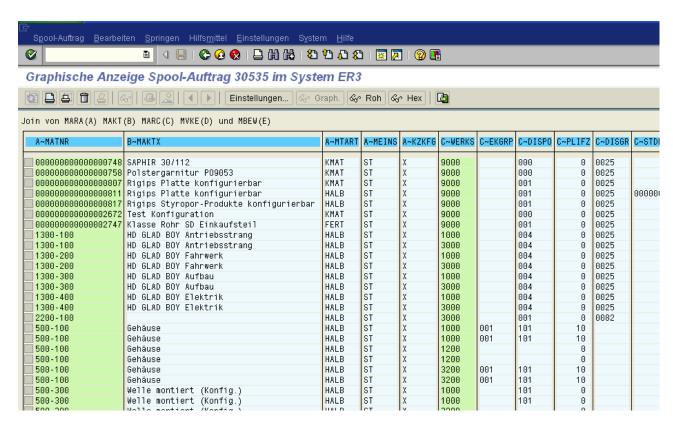


Der Spool-Auftrag kann entweder mithilfe der Menü-Funktion **Dokument**  $\rightarrow$  **Ausführen** (im SAPMail) oder mit der Standard-Transaktion **SP01** (Ausgabesteuerung) angezeigt werden:



ANMERKUNG: Falls "Sofort ausgeben" gewählt wurde, könnte der Spool-Auftrag bereits gelöscht worden sein.

Die Schaltfläche auf der Anwendungsleiste dient der Anzeige der Ergebnisliste:



ANMERKUNG: keine der üblichen SE16XXL-Funktionen sind verfügbar.



# Verbesserungen des Formeleditors

#### **Suchfunktion für ROW-Felder**

Die ROW-Felder der Ergebnisliste waren schon immer auf dem Baum auf der rechten Seite des Editors verfügbar:



Allerdings konnte sich die Suche nach einem bestimmten Feld als schwierig erweisen, wenn die beteiligte(n) Tabelle(n) zahlreiche Felder beinhaltete(n). Aus diesem Grund ist eine Suchfunktion implementiert worden.

# ANMERKUNG: Die Suchfunktion wirkt nur, wenn der Knoten "ROW-Felder" expandiert worden ist.

Nach Betätigung von ROW-Feld erscheint folgendes Dialogfenster:

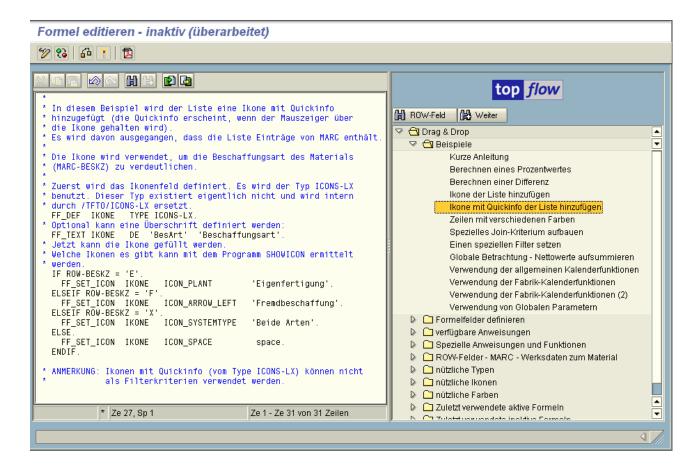


Es kann ein kompletter Feldname (wie MTART) oder nur ein Teil davon eingegeben werden. Wildcard-Zeichen (\*+) sind erlaubt aber nicht notwendig. Z.B. "MAT" wird MATNR, MATKL, PMATA, BMATN, MATFI und RMATP nacheinander finden (nach jedem Treffer muss Weiter betätigt werden).

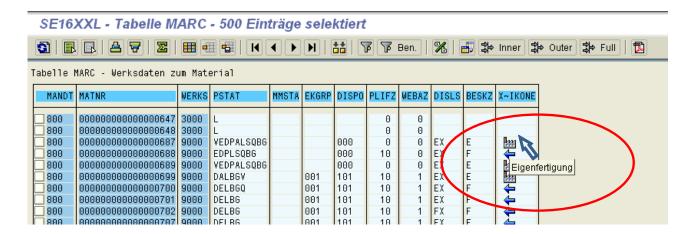


# Ikonen mit Quickinfo unterstützt

Es ist ab jetzt möglich, Ikonen mit Quickinfo zu definieren. Ein Beispiel wird zeigen, wie es funktioniert:



Das Ergebnis könnte wie folgt aussehen (Quickinfo beachten):



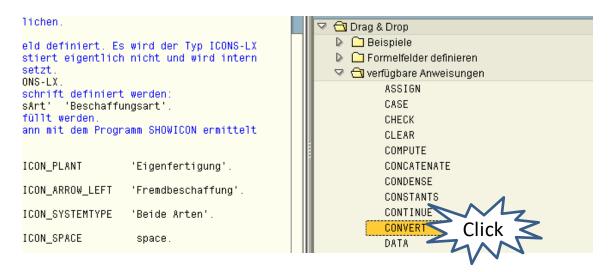


### **CONVERT-Anweisung erlaubt**

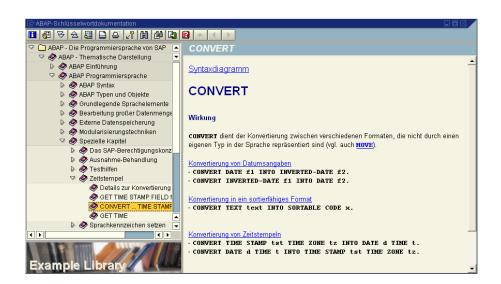
Die ABAP-Anweisung *CONVERT* darf ab jetzt in einer SE16XXL-Formel verwendet werden. Diese Anweisung ist besonders nützlich für die Konvertierung zwischen Datum + Uhrzeit und UTC-Zeitstempel und umgekehrt (siehe nächster Punkt).

### Standard-Hilfe für ABAP-Anweisungen

Die Standard-ABAP-Hilfe aus SE38 ist nun auch im Formeleditor verfügbar. Ein **Doppelklick** auf die Anweisung ist alles, was dazu benötigt wird:



#### Die ABAP-Hilfe kommt dann zum Vorschein:

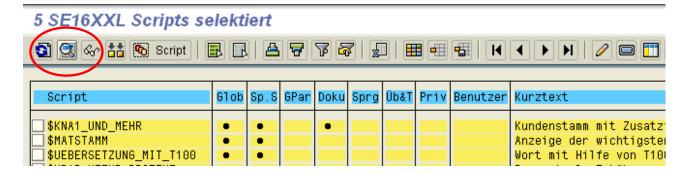




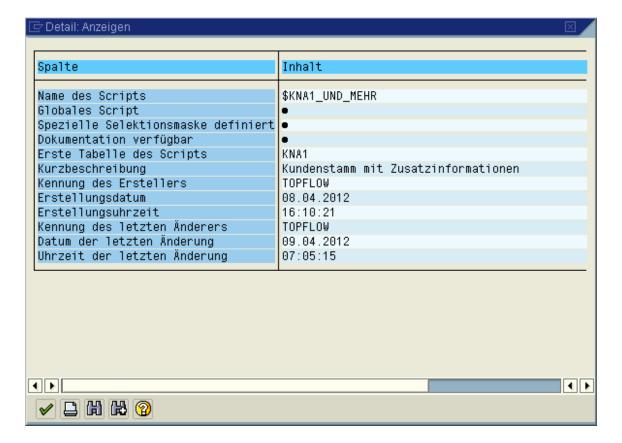
# Verbesserungen des Script-Katalogs

Jede Zeile der Script-Katalog-Liste stellt ein Script dar. Mit jeder neuen Version von SE16XXL sind mehr Attribute den Scripts zugeordnet worden – es wird immer schwieriger, sie alle auf einmal auf der Liste darzustellen.

Als Abhilfe ist die Standard-ALV-Funktion (Details) für die Anwender zur Verfügung gestellt worden:



Eine typische Ausgabe für ein Script könnte wie folgt aussehen:



ANMERKUNG: Initialwerte werden nicht angezeigt.



# **Parametertransaktionen eines Scripts**

Seit langem ist es möglich, einen Transaktionscode einem Script zuzuordnen. Jetzt kann diese Art von Zuordnung im Script-Katalog sichtbar gemacht werden.

Zwei neue Spalten sind zu diesem Zweck eingeführt worden:

- **TCode** für den ersten oder einzigen Transaktionscode;
- **Mehr** um anzudeuten, dass mehr als ein Transaktionscode zugeordnet ist.



Ein Doppelklick auf den Transaktionscode zeigt dessen Definition in Detail:



Ein Doppelklick auf — listet alle zugeordneten Transaktionscodes auf:



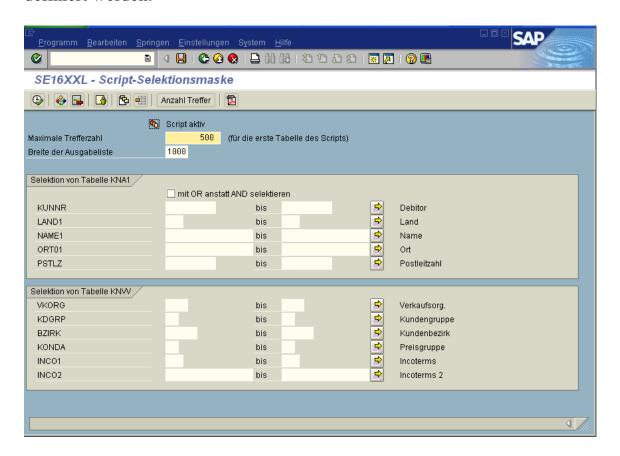


# Verbesserungen der Speziellen Selektionsmaske

### Selektionskriterien eines Blocks in beliebiger Reihenfolge

Bei der Definition einer speziellen Selektionsmaske für ein Script ist es nun möglich, die Kriterien eines Blocks in einer beliebigen Reihenfolge zu setzen. Die wichtigeren Kriterien können z.B. im oberen Bereich des Blocks platziert werden. Ein Beispiel soll zeigen, wie das bewerksteligt werden kann.

Das Script in unserem Beispiel besteht aus KNA1 (Kundestamm) gejoined mit KNVV (Kunden-Vertriebsdaten). Eine entsprechende Selektionsmaske ist bereits definiert worden:



Der Endanwender des Scripts hat allerdings die wichtigsten Selektionskriterien signalisiert: *Land* und *Postleitzahl* für KNA1, *Preisgruppe* und *Kundengruppe* für KNVV. Dementsprechend werden wir die Kriterien auf den zwei Blöcken neu anordnen.

Als erstes rufen wir den Script-Katalog auf und wählen die Funktion auf der Anwendungsleiste. Der Dialog zum Definieren der Selektionsmaske erscheint:





Als nächstes betätigen wir die Schaltfläche unterhalb des KNA1-Reiters – das Dialogfenster für die Auswahl der Selektionskriterien kommt zum Vorschein:



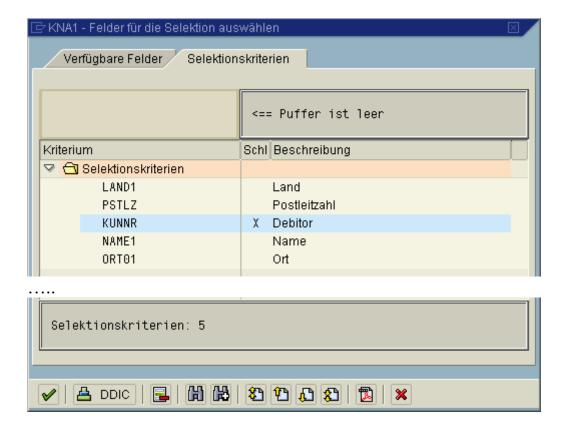
Wir möchten nur die Reihenfolge der Kriterien ändern – deshalb wählen wir den zweiten Reiter (Selektionskriterien ausgewählten Kriterien zu zeigen:



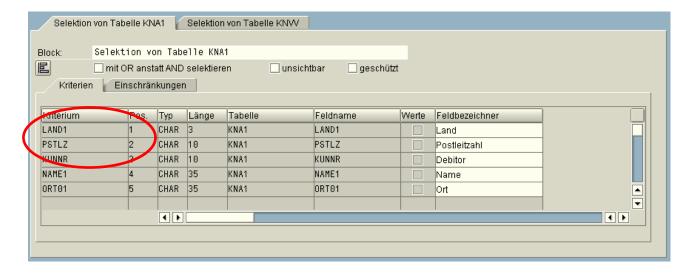
Dieses Dialogfenster ist dem ähnlich, das zum Anordnen der Spalten der Ergebnisliste vorgesehen ist. Und es funktioniert auch genauso.



Mithilfe von Drag & Drop bewegen wir LAND1 und PSTLZ nach oben:



Bei der Rückkehr zur Definition der Selektionsmaske erscheinen die Kriterien in der neuen Reihenfolge:

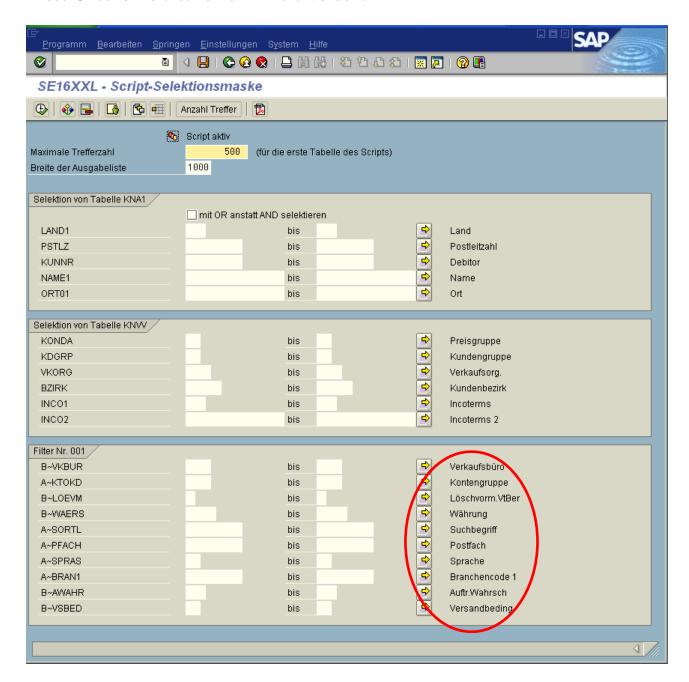


Mit dem KNVV-Block wird auf ähnliche Weise vorgegangen. Am Ende sind die Kriterien so angeordnet wie sie der Endanwender haben möchte.



#### Texte ohne Alias auf Filter-Blöcken

Die Texte auf einem Block mit Filterkriterien waren bis dato mit Alias ausgestattet. Diese Unschönheit ist nun eliminiert worden:



Beachten Sie auch die Anordnung der Filterkriterien.

## **Zum Anfang**



## Standard-Scriptvarianten

Script-Varianten sind praktisch: einerseits bieten sie dem Anwender eine Auswahl von vordefinierten Selektionskriterien bei der Ausführung eines bestimmten Scripts, andererseits unterstützen sie **Datums-**, **Zeit-** und **TVARVC-**Variablen.

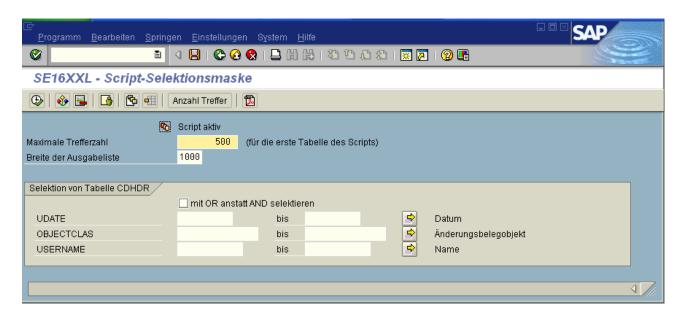
Es ist nun möglich, für ein Script eine *Standard-Variante* festzulegen, die automatisch geladen wird, sobald das Script ausgeführt wird. Auf diese Weise werden die Variablen der speziellen Selektionsmaske automatisch auf die gewünschten Werte gesetzt.

Bei der Ausführung des Scripts muss der Anwender sich nicht mit dem Laden der "richtigen" Variante befassen. Falls gewünscht ist es aber weiterhin möglich, jede andere Variante zu laden,. Ein Beispiel wird all dies veranschaulichen.

Wir beginnen mit dem Script-Katalog und einem Script, das die Änderungsbelege (Tabellen CDHDR und CDPOS) liest:



Wenn das Script ausgeführt wird ( script ), erscheint die spezielle Selektionsmaske:



Es muss die Script-Variante erstellt werden, die dann als Standard festgelegt wird.

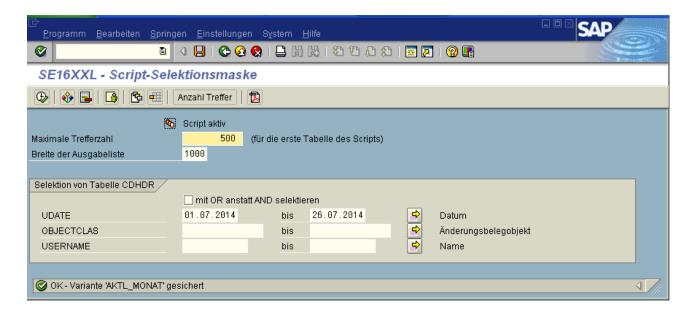
Das geschieht wie übich durch Betätigen von auf der Systemfunktionsleiste.

Wir kreieren eine Variante namens **AKTL\_MONAT**, die eine dynamische Datumsvariable für das Feld **UDATE** (Datum der Änderung) verwendet:

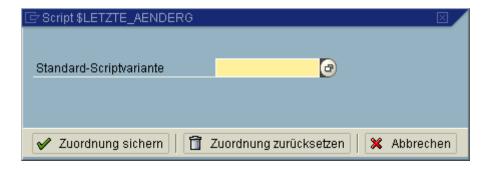


#### SE16XXL - Script-Variante definieren <u>□</u> \$LETZTE\_AENDERG Variante AKTL\_MONAT geschützt Beschreibung Änderungen vom Beginn des aktuellen Monats Selektion von Tabelle CDHDR oblig, gesch, ausgb UDATE Von Monatsanfang bis heute OBJECTCLAS USERNAME

Nach dem Sichern erscheint erneut die Selektionsmaske:

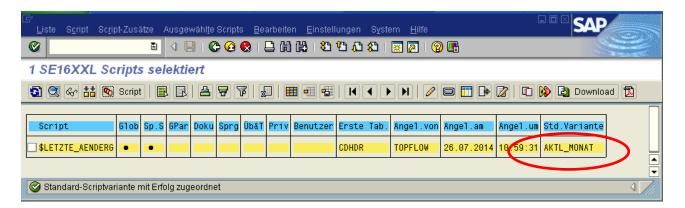


Wir kehren zum Script-Katalog zurück und wählen (nach dem Setzen des Cursors auf unser Script) *Script-Zusätze*  $\rightarrow$  *Std.Scriptvariante setzen/zurücksetzen* im Menü. Folgendes Dialogfenster erscheint:

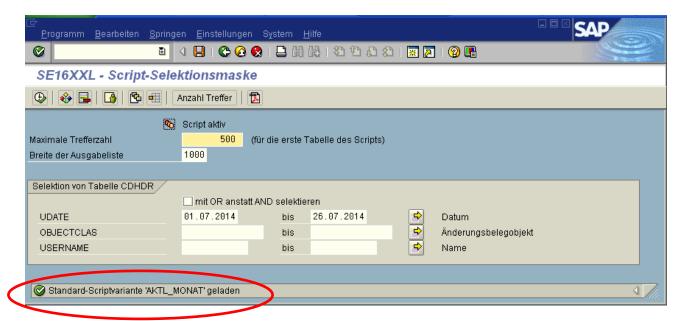




Wir bedienen uns der F4-Hilfe um unsere Variante auszuwählen und beenden die Operation mit <a href="#">V Zuordnung sichern</a>. Das Ergebnis ist auf der Katalog-Liste zu sehen:



Um die Zuordnung zu kontrollieren, führen wir nun das Script erneut aus ( Script ):



Beachten Sie den Hinweis auf der Statusleiste.

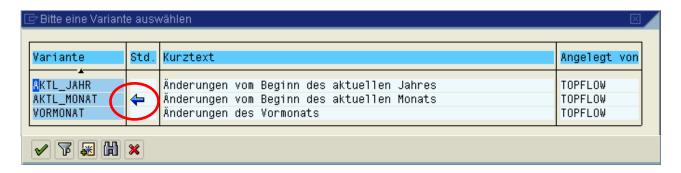
Ab diesem Moment wird die Standard-Variante wie im obigen Beispiel bei jeder Ausführung des Scripts **automatisch** geladen.

Die Standard-Zuordnung kann jederzeit geändert oder zurückgesetzt werden, indem die Menü-Funktion *Script-Zusätze*  $\rightarrow$  *Std.Scriptvariante setzen/zurücksetzen* erneut aufgerufen wird.

Es ist wichtig, dass sich die Anwender der Standard-Zuordnungen bewusst sind. Zu diesem Zweck sind einige Ergänzungen implementiert worden. Sie werden nun kurz erläutert.



Wenn auf der Selektionsmaske eine Variante gewählt wird ( auf der Anwendungsleiste), erscheint eine Liste der verfügbaren Script-Varianten:

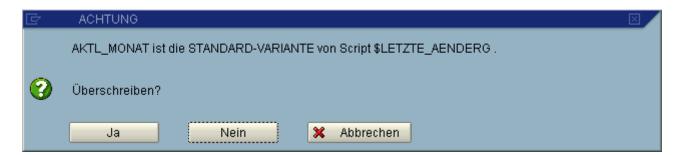


Die Ikone = zeigt die Standard-Variante an, sofern es eine gibt.

Dieselbe Ikone wird vom Tool für Download, Upload und Löschung von Script.-Varianten verwendet, die aus der **Einstiegsmaske** von SE16XXL mittels Menü-Funktion "*Springen* → *Script-Varianten* ..." zu erreichen ist:



Beim Sichern der Standard-Variante eines Scripts nach einer Änderung wird folgende Warnung ausgegeben:



Ähnliche Warnungen erscheinen, wenn Standard-Varianten gelöscht werden sollen.

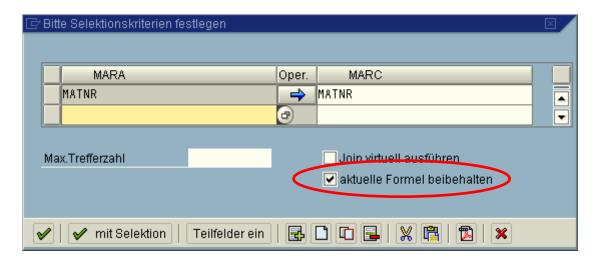
### **Zum Anfang**



## Zusätzliche Verbesserungen

### Script-Editor – letzte Formel verfügbar nach SELECT-Operation

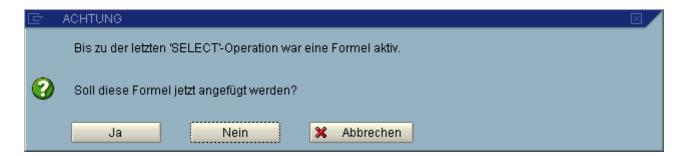
Die Geltung einer Formel wird in einem Script von der nachfolgenden SELECT-Operation begrenzt. In Dialog, bei der Durchführung eines Joins, kann der Anwender festlegen, dass die aktive Formel beibehalten wird:



Im Script-Editor hingegen, aufgrund der Komplexität der Situation stand bis dato diese Option **nicht** zur Verfügung.

### Dieses Manko ist nun auf folgende Art eliminiert worden:

Wird eine **Formel** nach einer solchen SELECT-Operation ein- oder angefügt, stellt der Script-Editor fest, dass bis zu diesem SELECT eine Formel aktiv war und gibt infolgedessen folgendes Popup aus:

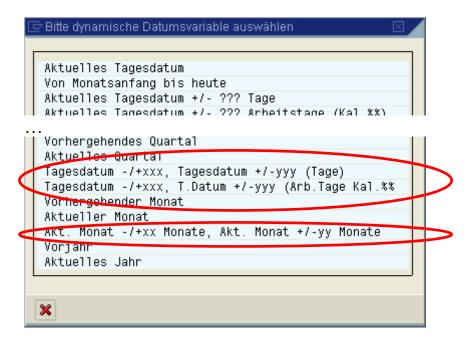


Antwortet der Anwender mit Ja , wird die **letzte aktive Formel** ein- bzw. angefügt, möglicherweise in Bezug auf die Alias angepasst .



## Script-Varianten – Datums-Variablen mit Plus und Minus

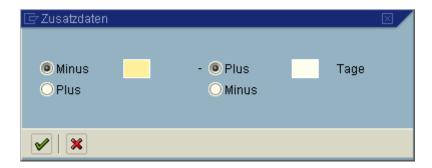
Einige der Datums-Variablen für Script-Varianten umfassen einen Zeitraum in Bezug auf das **aktuelle Datum** bzw. den **aktuellen Monat**:



Bis dato musste dieser Zeitraum immer das aktuelle Datum (oder Monat) beinhalten. Es war also möglich, einen Zeitraum von X Tagen in der Vergangenheit bis Y Tage in die Zukunft zu definieren, nicht aber von X Tagen bis Y Tagen, beide in der Vergangenheit (oder beide in der Zukunft).

#### Dieses Manko ist nun eliminiert worden.

Wird beispielsweise die Variable "**Tagesdatum -/+xxx, Tagesdatum +/- yyy** (**Tage**)" gewählt, erscheint folgendes Dialogfenster:

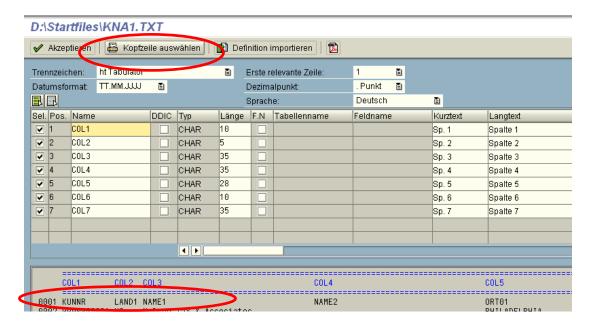


Damit ist es möglich, **jede sinnvolle Kombination** von Plus und/oder Minus auszuwählen.



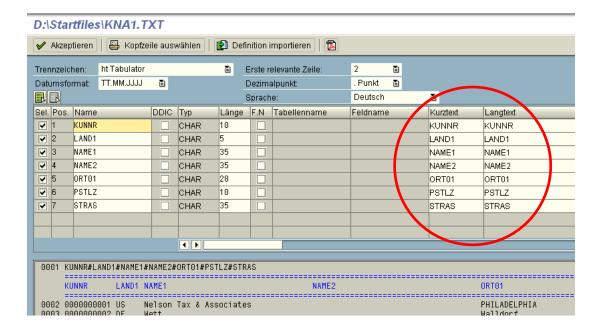
## Definition einer Vorgabedatei – Auswahl der Kopfzeile

Wenn die Struktur einer Vorgabedatei definiert wird, ist es möglich, eine **Kopfzeile** mit den Namen der Spalten auszuwählen, indem man von der Kopfzeile auswählen Schaltfläche auf der Anwendungsleiste Gebrauch macht:



Bis dato wurden nur die Spaltennamen der Zeile entnommen, die Kurz- und Langtexte hingegen blieben unverändert.

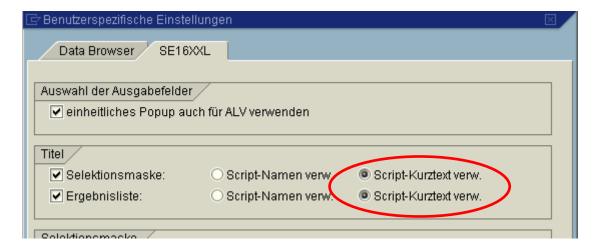
Mit der neuen Version werden sowohl die **Kurz-** als auch die **Langtexte** überschrieben:



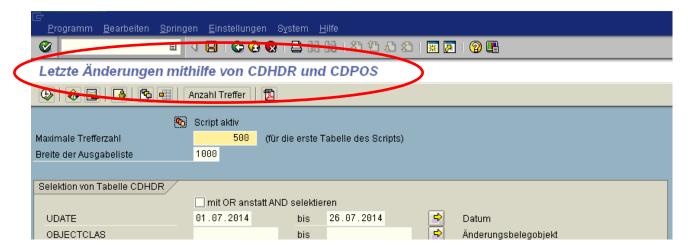


## **Script-Kurztext als Titel (Benutzer-Einstellung)**

In den Benutzer-Einstellungen kann nun festgelegt werden, dass der Script-Kurztext als Titel für die Selektionsmaske und/oder die Ergebnisliste verwendet wird:



#### Beispiel einer Selektionsmaske:



### Beispiel einer Ergebnisliste:





### Verbesserte Unterstützung von ALV-Layouts

Wenn bis dato ein ALV-Layout auf der Ergebnisliste ausgewählt wurde, wurden die Spaltenbreiten **automatisch** optimiert. **Dieses Manko ist nun eliminiert worden**. Es ist dadurch möglich geworden, ein ALV-Layout zu definieren, bei dem bestimmte Spalten entweder **breiter** oder **schmäler** als normal sind. Die Breite einer Spalte kann vergrößert werden um sie hervorzuheben, die einer anderen mit sehr langen Texten kann verkleinert werden.

Ein Beispiel wird diese Effekte veranschaulichen.

Wir beginnen mit einer Liste von KNA1-Einträgen:



Die Spaltenbreiten sind automatisch von ALV optimiert worden. Wir laden nun ein Layout, das Spalte KTOKD **breiter** und STRAS **schmäler** macht:



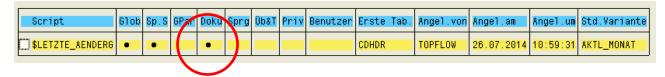
Das funktioniert ebenfalls, wenn das Layout vorweg auf der Selektionsmaske ausgewählt wird ( ).

Eine weitere Verbesserung betrifft die Funktion "*Ausführen + Drucken*" auf der Selektionsmaske. Bis dato wurde ein ausgewähltes ALV-Layout ignoriert. Ab jetzt wird es berücksichtigt.

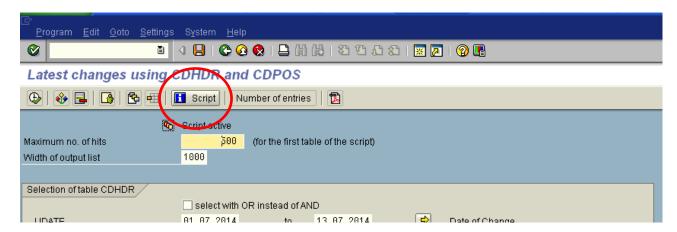


## Script-Dokumentation verfügbar in der Ergebnisliste

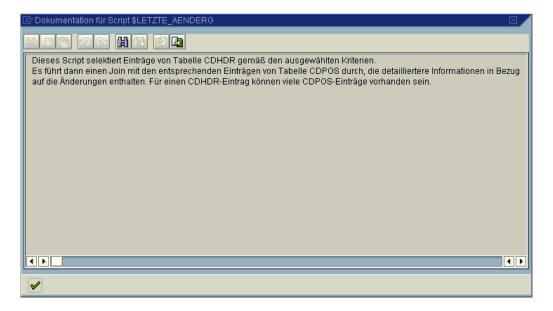
Die Dokumentaion eines Scripts war bis dato nur im Script-Katalog verfügbar (mittels **Doppelklick** auf den entsprechenden Punkt anzeigbar):



Oder auf der Selektionsmaske mithilfe von script auf der Anwendungsleiste:



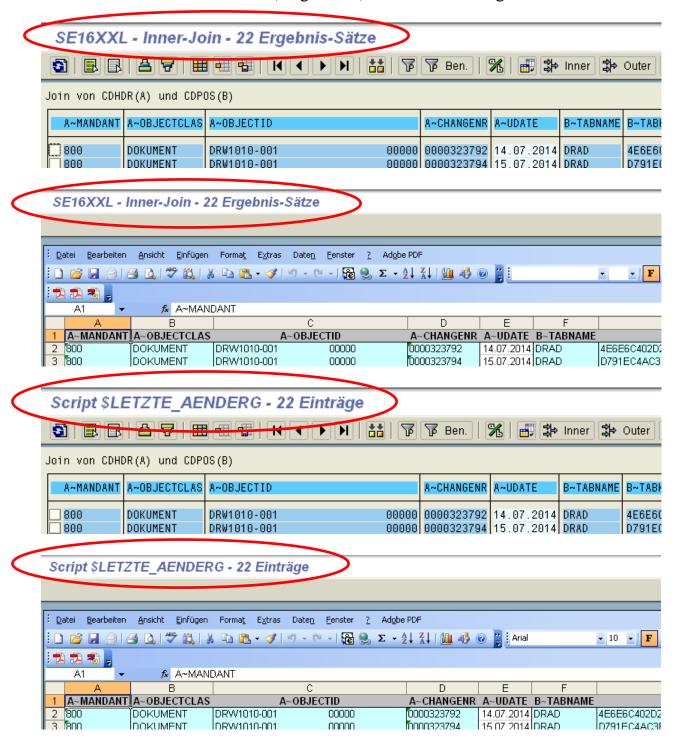
Ab jetzt kann die Script-Dokumentation auch aus der Ergebnisliste mit Hilfe der Menü-Funktion *Springen* → *Script-Dokumentation* erreicht werden. In allen Fällen wird die gleiche Information ausgegeben – wie in nachfolgendem Beispiel:





### Tabellenkalkulation (eingebettet) – Titel wie die Ergebnisliste

Bis dato war der Titel der "eingebetteten Tabellenkalkulation" unabhängig von den Benutzer-Einstellungen. Dieses Manko ist beseitigt worden. Nun widerspiegelt der Titel der "Tabellenkalkulation (eingebettet)" den Titel der Ergebnisliste:



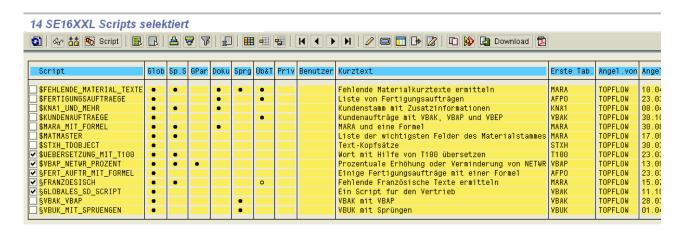
#### **Zum Anfang**



## Transport von Scripts

Es ist jetzt möglich, Scripts (und wahlweise die zugehörigen Varianten) von einem System zum anderen mit Hilfe von regulären Transportaufträgen zu transportieren. Die Funktionalität ist in der Hauptliste des Script-Katalogs verfügbar. Sowohl globale als auch benutzerspezifische Scripts können transportiert werden. Beim Transport gelten dieselben Berechtigungen die nötig sind, um ein Script zu ändern. Nur der Administrator hat mehr Rechte.

Die Operation ist unkompliziert. Auf einer Liste von Scripts werden diejenigen markiert, die es zu transportieren gilt:



Dann wird die Menüfunktion "Ausgewählte Scripts → Einem Transportauftrag hinzufügen" gewählt. Als Reaktion wird die Nummer eines Workbench-Auftrags verlangt:



Nach der Wahl eines bestehenden Auftrags, oder der Anlage eines neuen, bekommt man folgende Bestätigung:



Weitere Informationen befinden sich unter Transport von Scripts.

### **Zum Anfang**



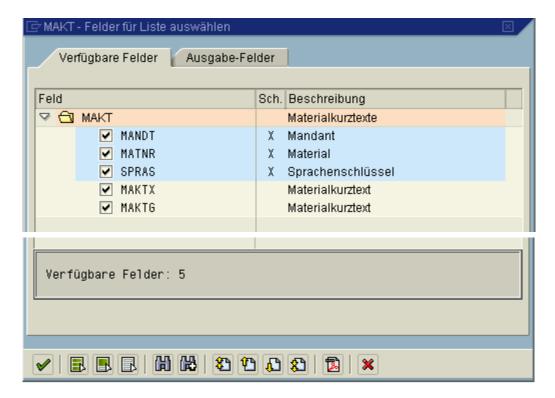
### Vereinheitlichte Feldauswahl

Für alle drei Anzeigemodi (SE16-Standard, ALV-List und ALV-Grid) ist es jetzt möglich ein und dasselbe Dialogfenster zur Auswahl der in der Ergebnisliste auszugebenden Felder zu verwenden. Dieses neue Dialogfenster ersetzt das alte Fenster für den SE16-Standardmodus.

Im Falle von ALV liegt die Entscheidung beim Anwender – Das neue Dialogfenster kann mithilfe der Benutzer-Einstellungen ( ) deaktiviert werden:



Das neue Dialogfenster erscheint, wenn die E Drucktaste betätigt wird:



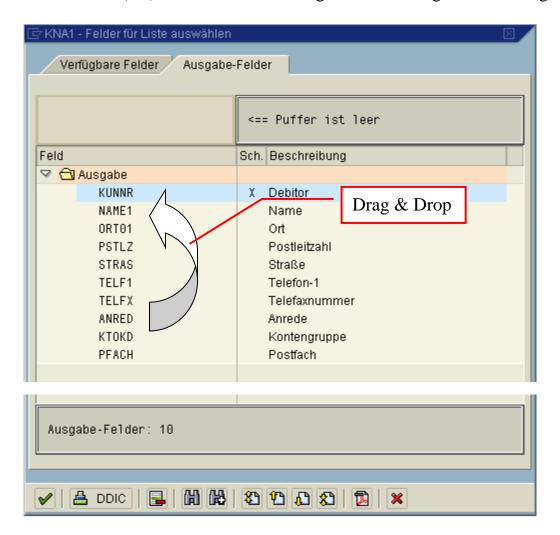
Für weitere Informationen siehe Vereinheitlichte Feldauswahl.



## Spalten beliebig angeordnet im Std-Anzeigemodus

Bis dato wurden in SE16-Standardmodus die Spalten in der Reihenfolge des Data Dictionary angezeigt. Mit der neuen Version ist eine beliebige Anordnung möglich, wie bei beiden ALV-Anzeigemodi.

Machen Sie von **Drag & Drop** auf dem zweiten Bereich des Dialogfensters für die Feldauswahl (H) Gebrauch, um die Ergebnisliste wie gewünscht zu gestalten:



Mithilfe der DDIC Drucktaste können Sie die Data Dictionary Reihenfolge wiederherstellen.

Weitere Informationen sind unter Vereinheitlichte Feldauswahl zu finden.



## Funktion "Spalte fixieren" in allen Anzeigemodi

Bis dato standen im SE16-Standardmodus nur die zwei Funktionen "Schlüsselspalten fixieren" und "Schlüsselspalten entfixieren" zur Verfügung.

Diese Beschränkung ist mit der neuen Version aufgehoben worden.

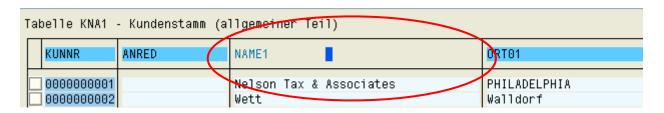
Im Standard-Anzeigemodus ist die Funktion im Menü erreichbar unter

#### *Einstellungen* → *Spalte fixieren*

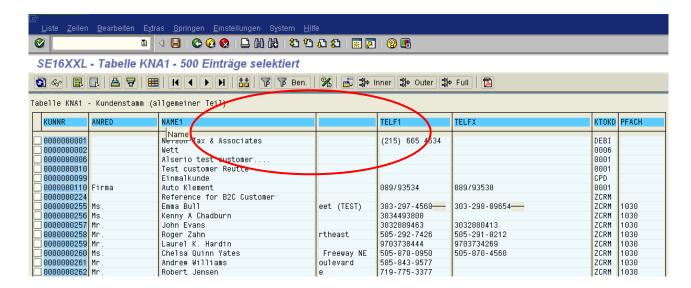
In beiden ALV-Anzeigemodi ist der Menü-Pfad:

#### Einstellungen → Spalten → Spalte fixieren

Vor dem Aufruf der Funktion sollte die entsprechende Spalte markiert werden:



Als Resultat bleiben die ersten drei Spalten der Liste stehen, wenn nach rechts gescrollt wird:





## **Neue Filter-Option**

Bis dato wurden bei einem Filter immer die Kriterien der zu behaltenden Sätze spezifiziert. Es war damit sehr einfach eine positive Auswahl zu treffen, z.B. alle Kunden eines bestimmten Ortes, oder alle Buchhaltungsbelege eines bestimmten Jahres. Auch kompliziertere Kriterien waren kein Problem, wie z.B. alle Materialien mit bestimmten Materialarten, Maßeinheiten, Änderungsdatum usw. zu behalten.

Schwieriger war es eine negative Auswahl zu treffen, z.B. das Ausblenden aus der Liste der Materialien passend zu den oben erwähnten Kriterien. Für ein einzelnes Kriterium war die Sache noch zu meistern. Mit mehreren Kriterien konnte die Spezifikation leicht zu Fehlinterpretationen und fehlerhaften Kriterien führen.

Um diese Schwierigkeiten aus der Praxis zu eliminieren, ist das Filter-Dialogfenster erweitert worden. Die Kriterien können nunmehr immer als positive Auswahl formuliert werden – anschließend werden sie entweder zum **Behalten** oder zum Ausblenden der passenden Sätze verwendet:



Passende Sätze ausblenden stellt die neue Option dar.



#### Virtuelle Joins

Es passiert von Zeit zu Zeit, dass ein bestimmter Join interessant wäre, der aber, mangels eines geeigneten Sekundärindex, in der Praxis so gut wie unmöglich erscheint. In ähnlichen Situationen, aufgrund der geringen Anzahl Einträge der zu joinenden Tabelle, kann die Operation durchaus vollzogen werden, aber auf imperformante Weise.

Zur Überwindung solcher Schwierigkeiten sind "virtuelle" Joins eingeführt worden.

Nach außen verhält sich ein virtueller Join wie ein gewöhnlicher: die Resultate sind identisch. Intern ist aber die Arbeitsweise eine andere:

Ein **gewöhnlicher Join** macht von der SQL-Klausel "SELECT ... FOR ALL ENTRIES" Gebrauch, die alle notwendigen Einträge paketweise von der Datenbank holt. Dieses Prozedere funktioniert ausgezeichnet, falls ein geeigneter Index (primär oder sekundär) zum Besorgen der notwendigen Daten existiert.

Ein **virtueller Join** verwendet hingegen zum Lesen der Einträge der zu joinenden Tabelle nur die Kriterien, die auf der **Zwischen-Selektionsmaske** (die <u>Pflicht</u> ist) angegeben werden. In einer solchen Situation kann nur ein kleiner Anteil der selektierten Einträge für die Durchführung des Joins notwendig sein – das geht aber wesentlich zügiger als wenn man versucht, nur die relevanten Sätze zu lesen. Stehen einmal die Sätze zur Verfügung, wird der Join selber wie gewohnt im Speicher durchgeführt.

Mit virtuellen Joins öffnen sich für den interessierten Anwender neue Perspektiven. Joins, die bisher undenkbar waren, stellen nun eine gangbare Alternative dar. Es könnte sogar von Vorteil sein auch dann einen virtuellen Join anzuwenden, wenn ein passender Sekundärindex wohl existiert – letztendlich hängt die Gesamtperformance eines bestimmten Joins von vielen Faktoren ab und nur anhand von Versuchen kann die optimale Lösung ermittelt werden.

Normalerweise ist die Option "virtuell" wirklich optional. Im Falle eines <u>Full-Joins</u> ist sie Pflicht und kann nicht deaktiviert werden.

**ANMERKUNG**: Wir haben bisher nur von Joins gesprochen – das Obenbesagte gilt aber ebenfalls für die Operation "Selektiere für alle Einträge".

Wir werden jetzt ein Beispiel eines virtuellen Joins zeigen um das Ganze zu verdeutlichen.

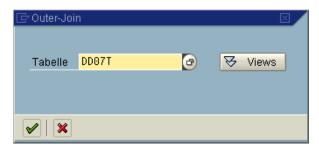


**Beispiel**: Nehmen wir eine Liste von Einträgen von Tabelle VBUK an, die u.a. das Feld VBTYP (Vertriebsbelegtyp) enthalten. Das Feld VBTYP ist mit der Domäne VBTYP verbunden, die im Data Dictionary mit einer Serie von Festwerten definiert ist. Die Tabelle DD07T beinhaltet die Kurztexte der Domänenfestwerte. Es existiert jedoch kein Sekundärindex für das Feld DOMVALUE\_L, das für einen Join verwendet werden könnte: ein idealer Kandidat also für einen virtuellen Join.

Ausgangspunkt – eine Liste von VBUK-Einträgen:



Wir markieren den Spaltenkopf von VBTYP und geben DD07T als zu joinende Tabelle ein:

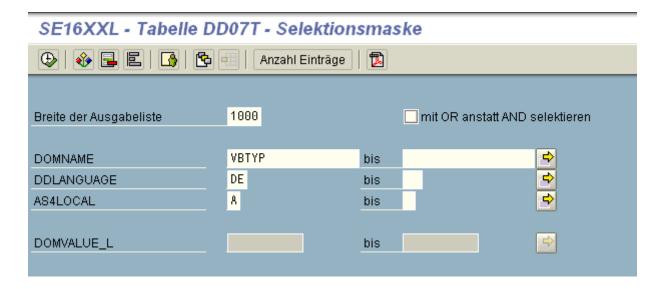


Auf dem darauffolgenden Dialogfenster aktivieren wir die Option "Join virtuell ausführen":

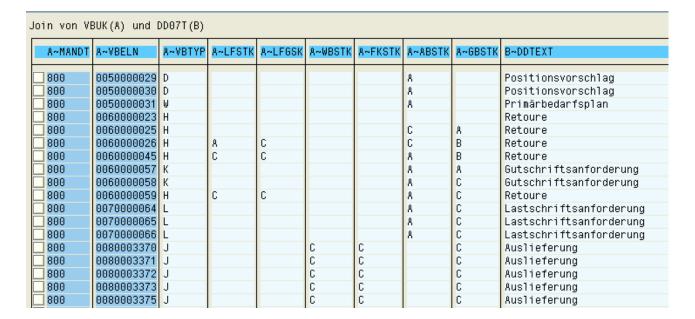




Wir fahren fort, imdem wir die ✓ mit Selektion Schaltfäche betätigen – die Zwischenselektionsmaske erscheint:



Alle DD07T-Einträge, die die Selektionskriterien erfüllen, werden nun von der Datenbank gelesen. Nach der Durchführung des Joins liegt die Ergebnisliste vor:





### Full-Join

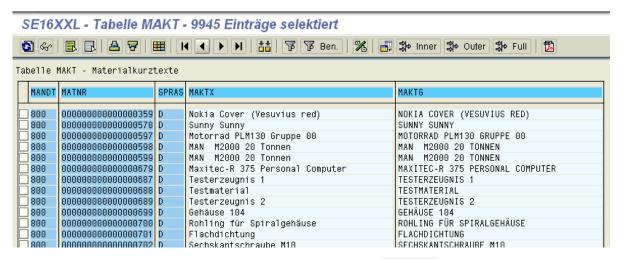
Bis dato waren zwei Arten von Joins verfügbar, Inner und Outer. Beide haben eine asymmetrische Wirkung indem sie von einer Liste von Einträgen ausgehen, die dann mit passenden Einträgen einer anderen Datenbanktabelle oder View verbunden werden. Auf der rechten Seite des Resultats sind nur Einträge zu finden, die mit einem Eintrag der Original-Liste in Verbindung stehen. Für die meisten Zwecke sind diese Joins ideal, denn normalerweise basiert das anvisierte Ergebnis auf einer bestimmten Tabelle, deren Einträge Schritt für Schritt mit Zusatzinformationen angereichert werden.

Wenn aber das Ziel darin besteht, fehlende Einträge auf beiden Seiten zu ermitteln, wird eine neue Art von Join benötigt, "Full-Join" genannt. Ein Full-Join besitzt eine symmetrische Natur – er verbindet die Einträge auf der linken Seite mit denen auf der rechten, und umgekehrt. Bisher war die linke Hälfte immer vollbesetzt, nur auf der rechten Seite (im Falles eines Outer-Joins) konnten Einträge fehlen. Mit dem Full-Join können Einträge sowohl links als auch rechts fehlen.

Dieses Verhalten wird am besten anhand eines Beispiels verdeutlicht. Wir werden Materialkurztexte (MAKT) einer Sprache mit denen einer weiteren Sprache verbinden. Ziel dabei ist es, die fehlenden Texte zu ermitteln, egal auf welcher Seite. Dieser Join verwendet dieselbe Tabelle auf beiden Seiten. Es muss aber klar sein. dass ein Full-Join auch zwischen unterschiedlichen Datenbanktabellen oder Views durchgeführt werden kann.

**ANMERKUNG:** Im Falle eines Full-Joins sollten alle beteiligten Einträge gelesen werden, sonst werden Einträge als fehlend wahrgenommen, die einfach nicht selektiert worden sind.

Wir beginnen mit allen Materialkurztexten der Sprache "D":

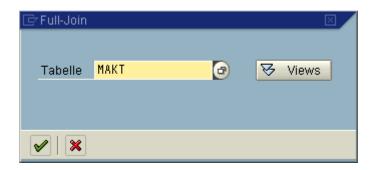


Wir markieren die Spalte MATNR und betätigen auf der Anwendungsleiste.





Das Popup für die Angabe der neuen Tabelle erscheint:



Danach ist das Dialogfenster für die Join-Kriterien an der Reihe:



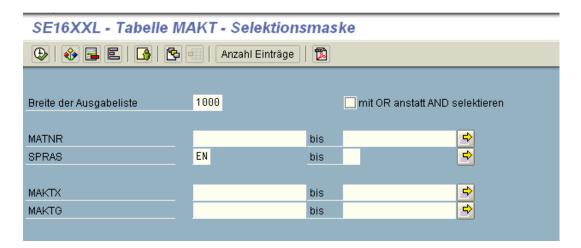
Es fällt auf, dass die Option gesetzt ist, und nicht deaktiviert werden kann. Ferner ist die "Weiter" Drucktaste ( ) deaktiviert. Man kann nur fortfahren, indem man von der Schaltfläche der erwarteten Selektionsmaske erscheint aber folgendes Dialogfenster:



Hier tragen wir die Namen der <u>gemeinsamen Felder</u> ein (in diesem Fall nur eines), die zur Ergebnisliste hinzuzufügen sind, um auf jeden Fall eine komplett gefüllte Spalte zu haben, unabhängig vom Vorhandensein eines Eintrags rechts oder links. Ohne gemeinsame Felder wäre eine sinnvolle Sortierung der Liste nicht möglich.

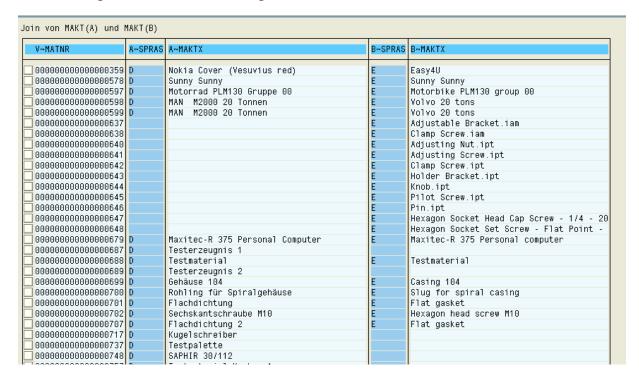


Die erwartete Zwischenselektionsmaske kommt jetzt zumVorschein – wir geben Sprache "EN" als einziges Kriterium ein:



Zu bemerken ist, dass sogar das Feld MATNR zur Verfügung steht, das normalerweise bei einem Join geschützt wäre – im Falle eines Full-Joins werden aber nur die Kriterien aus der Zwischenselektionsmaske für den Datenbakzugriff herangezogen.

Das Endergebnis könnte wie folgt aussehen:



Das gemeinsame Feld befindet auf der linken Seite. Es ist immer gefüllt, auch wenn links oder rechts der entsprechende Eintrag fehlt.

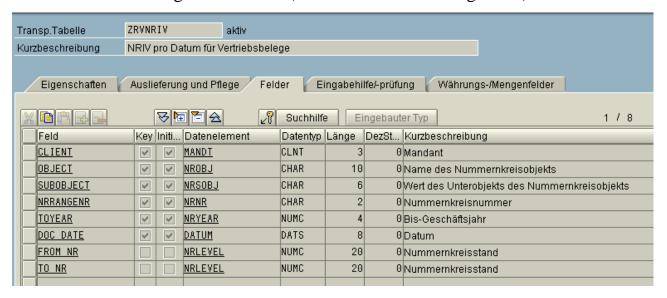


### Join-Kriterien mit Intervall-Selektion

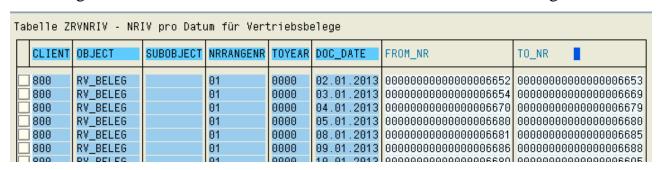
In manchen Firmen werden täglich spezielle Datenbank-Tabellen ähnlich wie NRIV aktualisiert. Solche Tabellen enthalten **pro Nummernkreis und Datum** die **niediegste** und **höchste** Nummer der beteiligten Belege, meistens Vertriebs- bzw. Buchhaltungsbelege. Auf diese Weise ist es per Programm relativ einfach, die Belege zu selektieren die während einer bestimmten Zeitspanne entstanden sind. Bis dato konnten solche Tabellen mit Intervall-Informationen mit SE16XXL kaum auf eine vernünftige Art genutzt werden. Diese Lücke ist mit der neuen Version geschlossen worden.

Um zu verdeutlichen, wie eine Intervall-Selektion arbeitet, werden wir ein Beispiel anhand einer angedachten Kundentabelle namens **ZRVNRIV** durchziehen, welche die Nummernintervalle von Vertriebsbelegen pro Kreis und Datum enthält. Die Standard-Tabelle **VBUK** (Vertriebsbeleg: Kopfstatus und Verwaltungsdaten) wird anschließend mit ZRVNRIV mit einem Join verbunden.

ZRVNRIV besitzt folgende Struktur (teilweise von NRIV abgeleitet):

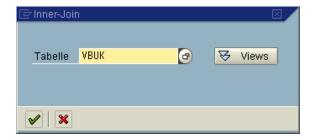


Wir selektieren zunächst die Einträge mit **DOC\_DATE** zwischen Januar und Juni 2013. In der Ergebnisliste markieren wir die untere und obere Grenze der Belegnummern:





Ziel ist es, die passenden Einträge von Tabelle VBUK zu selektieren. Es ist anzunehmen, dass solche Einträge vorhanden sind – wir können also ohne weiteres einen Inner-Join durchführen ( ):



Nach Angabe der Tabelle erscheint das Dialogfenster für die Join-Kriterien:



SE16XXL ahnt nicht, was wir vorhaben – folglich sind die Felder auf der rechten Seite leer. Wir tragen VBELN in beide ein, betätigen dann die erste Drucktaste, um "Untere Grenze (inklusive)" als Operator auszuwählen:



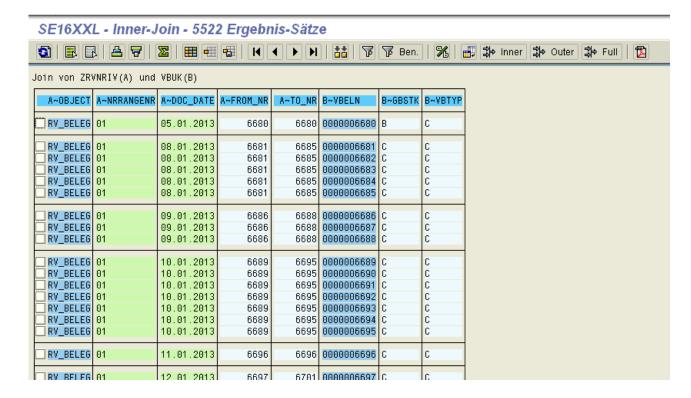
Ähnlich verfahren wir mit der zweiten Drucktaste, aber mit "Obere Grenze".



Nachdem die Intervall-Felder von ZRVNRIV doppel so lang sind, wie das VBELN-Feld von VBUK, schalten wir auch die Teilfelder ein, um den passenden Offset spezifizieren zu können. Das Dialogfenster sieht nun wie folgt aus:



Das Endergebnis ist unten zu sehen – der Anzeigemodus ist auf **ALV-List** mit Trennlinien gesetzt worden, um die Intervalle besser aufzuzeigen:



**ANMERKUNG**: Es darf nur <u>ein</u> Intervall pro Join angegeben werden. Zusätzlich müssen beide Grenzen – untere und obere – verwendet werden.



# Join-Kriterien mit Präfix-Selektion

Wenn ein Join durchgeführt wird, werden normalerweise die kompletten Inhalte der beteiligten Felder als Selektions-Kriterien verwendet. Gelegentlich wäre es aber von Vorteil, wenn man die Werte als Präfix verwenden könnte.

Nachfolgendes Beispiel ist gut geeignet, eine solche Situation zu verdeutlichen. Nehmen wir an, es sind Bestellungen (EKKO) selektiert worden. Es kommt bei der Kundenauftragserfassung vor, dass manche Anwender vom Feld **VBKD-BSTKD\_M** (*Bestellnummer des Kunden als Matchcodefeld*) Gebrauch machen, das 35 Zeichen lang ist. Darin tragen sie die Nummer der beteiligten Bestellung ein, gefolgt von einem Vermerk, wie z.B.

"4509876352 – so früh wie möglich"

Ein Sekundärindex ist dem Feld BSTKD\_M zugeordnet – es wäre aber mit den bisher verfügbaren Mitteln nicht möglich, einen Join durchzuführen.

Aus diesem Grund ist der neue Join-Operator "als Präfix von" eingeführt worden:



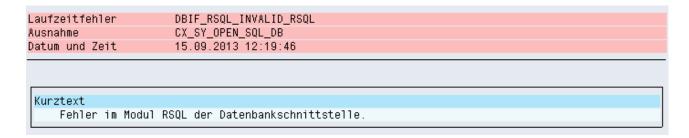
Die Join-Kriterien wären in unserem Beispiel wie folgt:





## Select-Options mit Tausenden von Einträgen

Es konnte bis dato zu einem Kurzdump kommen, falls man versucht hat, SE16XXL mit einer Select-Option mit **Tausenden** von Einträgen zu starten – was leicht vonstatten geht, indem eine Spalte aus der Tabellenkalkulation in die Zwischenablage kopiert und dann in die Select-Option hochgeladen wird:



Ursache ist die maximale Größe, die eine SQL-Anweisung haben darf.

Um diese Schwierigkeit zu umgehen, ist eine **Spezialbehandlung** in SE16XXL implementiert worden: wenn eine Select-Option eine große Anzahl Einträge enthält, wird die passende SELECT-Klausel auf andere Art abgesetzt, damit die vorgeschriebenen Grenzen nicht überschritten werden. Es gelten aber folgende Einschränkungen:

- Funktioniert nur mit Haupt-Selektionsmasken, nicht bei Joins;
- Nur <u>eine</u> Select-Option darf sehr viele Einträge enthalten:
- Die Einträge müssen Einzelwerte mit dem <a> Operator sein:</a>



Falls diese Bedingungen nicht erfüllt sind, wird die SELECT-Operation wie gewöhnlich durchgeführt und das Ergebnis wird ein Kurzdump sein.



### **Globale Parameter**

Manchmal enthalten Scripts Formeln die auf einen konstanten Wert oder auf eine Serie von Konstanten basieren, um z.B. eine Berechnung durchzuführen oder die Zeilen in Kategorien aufzuteilen. Bis dato musste das Script jedesmal angepasst oder mehrfach kopiert werden, um Ergebnisse von unterschiedlichen Konstanten zu erhalten. Eine aufwendige und nicht besonders elegante Vorgehensweise.

Globale Parameter sind eingeführt worden um solche Probleme zu beseitigen. Sie werden während des Aufbaus eines Scripts definiert und stehen, sobald sie deklariert worden sind, allen Formeln des Scripts zur Verfügung, unabhängig von der jeweiligen Join-Ebene.

Zwei Arten von Globalen Parametern gibt es: echte **Parameter** mit nur **einem Wert** und **Select-Options** die eine **Serie von Werten** annehmen können, genau wie echte Select-Options es tun.

Die definierten globalen Parameter sind im Formel-Editor auf dem Drag&Drop-Baum auf der rechten Seite unter dem Titel "Globale Parameter" verfügbar. Im Coding der Formel wird ein globaler Parameter zwischen zwei Fragezeichen wie in **?PARAM?** und **?SELOPT?**[] angesprochen – unter den Annahme dass PARAM und SELOPT die entsprechenden Namen sind.

Einmal in einer Formel verwendet, kann ein globaler Parameter **weder geändert noch gelöscht** werden – dies würde die Formel ungültig machen. Mithilfe des Script-Editors können die in einem Script definierten globalen Parameter sowohl geändert als auch gelöscht werden. Die beteiligten Formeln müssen aber danach manuell entsprechend angepasst werden.

Der letzte Schritt besteht darin, eine **spezielle Selektionsmaske** für das Script zu definieren. Etwaige globale Parameter des Scripts werden auf dem **zweiten Block** der Maske zur Verfügung stehen. Bis zu **zehn** Parameter und/or Select-Option haben auf diesem Block Platz.

Bei der Ausführung des Scripts trägt der Anwender die gewünschten Werte auf der speziellen Selektionsmaske ein, diese werden direkt an die verschiedenen Formeln des Script weitergereicht. Auf diese Weise kann **ein einziges Script** für eine breite Palette von Resultaten verwendet werden.

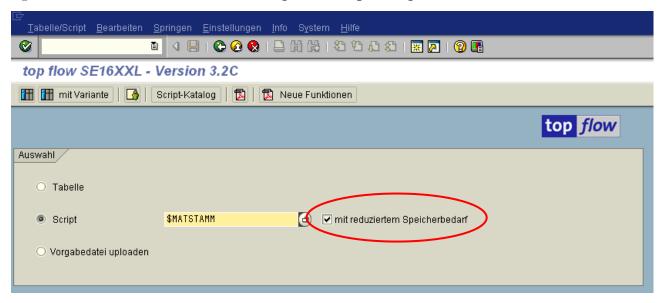
Für ausführlichere Informationen siehe Globale Parameter Definieren.



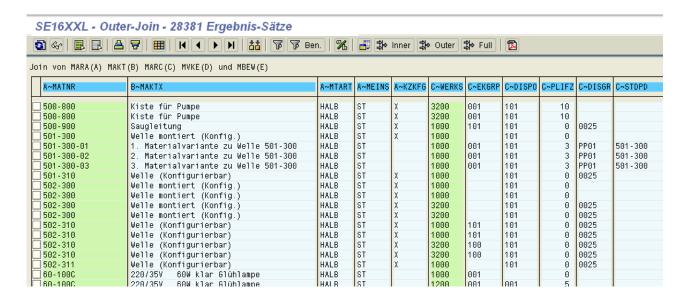
## Scripts "mit reduziertem Speicherbedarf" ausführen

Bis dato wurden bei der Ausführung eines Scripts **sämtliche Felder** der selektierten Einträge der beteiligten Datenbank-Tabellen bzw. Views gelesen. Aufgrund dessen endete die SE16XXL Sitzung gelegentlich wegen **Speichermangels** mit einem **Kurzdump**. Dieses Verhalten erschwerte die Analyse von großen Datenmengen, denn zahlreiche Datenbank-Tabellen weisen sehr viele Felder auf, die gelesen wurden, obwohl sie auf keinste Weise zu den Script-Ergebnissen beitrugen.

Um diese negativen Effekte zu verringern, ist die Option "**mit reduziertem Speicherbedarf**" für die Ausführung von Scripts eingeführt worden:

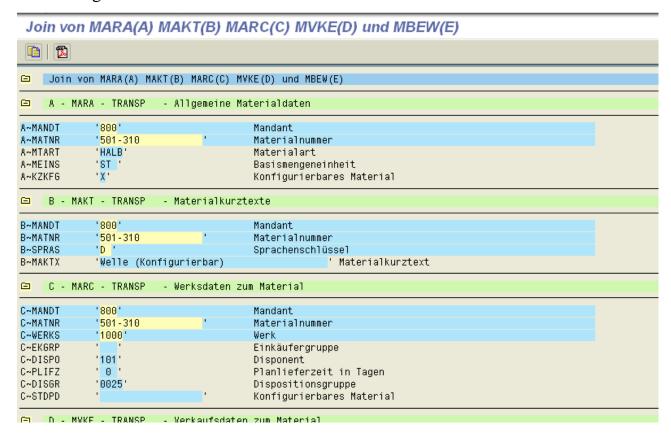


Oberflächlich betrachtet sieht alles wie sonst aus, wenn von dieser Option Gebrauch gemacht wird. Die Ergebnisliste bietet das erwartete Erscheinungsbild:





Ein Blick auf die Details einer Listenzeile verrät aber, dass nur die relevanten Felder der beteiligten Tabellen vorhanden sind:



Wenn die Option "**mit reduziertem Speicherbedarf**" aktiviert wird, werden nur die Schlüsselfelder sowie diejenigen Felder aus der Datenbank selektiert, die für das Resultat des Scripts benötigt werden. Der Speicherverbrauch kann somit erheblich gesenkt werden.

Als **Kehrseite** sind – wie man oben gesehen hat – auch nur solche Felder in der Ergebnisliste vorhanden. **Dies stellt aber kein echtes Problem dar**. Wenn ein bestimmtes Feld auf der Liste erscheinen sollte, muss es lediglich auf der Selektionsmaske mit Hilfe der Menüfunktion *Einstellungen → Felder für Liste auswählen* in die Auswahl übernommen werden: **alle Felder** der Ergebnisliste werden aus der Datenbank mitselektiert.

ANMERKUNG: Die Option ist auch für die Ausführung im Hintergrund möglich:





### Neue Operation "Sortierte Sätze zählen"

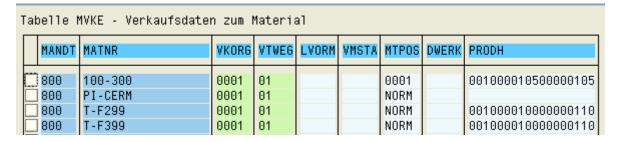
Bis dato konnte das Zählen der Sätze einer Ergebnisliste gemäß bestimmten Kriterien nur mit Hilfe einer Formel mit globaler Betrachtung (mit STATICS Variablen) bewerkstelligt werden – eine relativ komplizierte und fehleranfällige Prozedur.

Um diese Funktionalität allen Anwendern anbieten zu können, unabhängig von ihren ABAP-Kenntnissen, ist die neue Menü-Funktion *Extras* → *Sortierte Sätze zählen* implementiert worden. Sie ist einfach zu bedienen und kann sogar in derselben Ergebnisliste mehrfach hintereinander mit unterschiedlichen Kriterien angewendet werden, ganz ohne komplizierte Formeln.

Vor dem Aufruf der Funktion **muss** die Liste **sortiert** werden. Ansonsten erscheint folgende Meldung:



Wir werden die Anwendung dieser Funktion anhand eines Beispiels (basierend auf Tabelle MVKE – Verkaufsdaten zum Material) erläutern. Ziel ist es die Materialien zu zählen, die pro Kombination von Verkaufsorganisation und Vertriebsweg definiert sind. Zu Beginn sortieren wir die Liste nach VKORG und VTWEG:



Nach Aufruf der Menü-Funktion *Extras* → *Sortierte Sätze zählen* erscheint folgendes Dialogfenster. Es wird der Name des V-Feldes verlangt, der als Zähler fungieren wird:



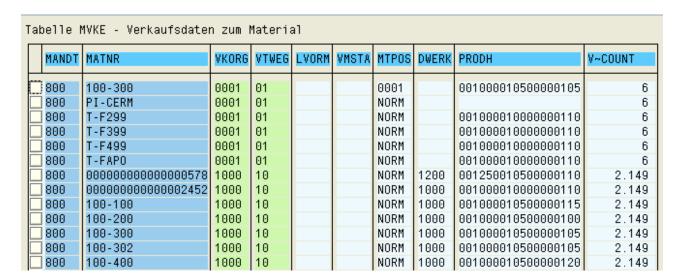
V-Felder (V für "verschiedenartig") sind eine neue Art von Feldern, ähnlich wie X-Felder (Formelfelder), die für alle möglichen Zwecke verwendet werden.



Nach der Festlegung des Zählfeldes erscheint folgendes Dialogfenster, dasselbe das für "Sätze mit Duplikaten ausblenden" und ähnliche Funktionen verwendet wird:



Wir könnten das VTWEG Feld entmarkieren, falls wir lediglich die Materialien pro VkOrg zählen möchten. In diesem Beispiel akzeptieren wir die Kriterien wie sie sind. Die Ergebnisliste kommt erneut zum Vorschein, ergänzt auf der rechten Seite durch eine neue Spalte mit den Zählerwerten:



Dem Zählfeld ist das Pseudo-Alias V vorangestellt, so wie Formelfelder ein X vorne tragen.

Alle Materialien die zu derselben Kombination von Verkaufsorganisation und Vetriebsweg angehören, weisen den gleichen Zählerwert auf.

Im vorliegenden Beispiel interessieren uns die einzelnen Materialien nicht – wir möchten nur die Zählerstände sehen. Wir eliminieren also benachbarte Duplikate aus der Liste (Funktion – *Benachbarte Duplikate ausblenden*).

Unsere Statistik sieht dann wie folgt aus (nach Reduzierung auf die relevanten Spalten):



Tak	Tabelle MVKE - Verkaufsdaten zum Material							
	VKORG	VTWEG	V~COUNT					
	0001	01	6					
	1000	10	2.149					
	1000	12	553					
	1000	14	23					
	1000	16	36					
	1020	20	51					
	1020	22	16					
	1020	24	96					
	1030	24	16					
	2000	10	22					
	2000	12	8					
	2000	30	12					
	2200	10	37					
	2300	10	385					

Die Liste kann nach V~COUNT absteigend umsortiert werden, um die Kombinationen mit den meisten Materialien im oberen Bereich zu erhalten:

Tabelle MVKE - Verkaufsdaten zum Material							
VKORG	VTWEG	V~COUNT					
1000	10	2.149					
9000	10	1.556					
3020	30	793					
1000	12	553					
2300	10	385					
3000	10	327					
3020	14	281					
2400	10	270					
2500	10	268					
3020	10	240					
3020	12	204					
1020	24	96					
R300	R1	81					
R300	R5	80					

WICHTIG: falls Sie ein zuverlässiges Resultat anstreben, müssen Sie sicherstellen, dass Sie dazu berechtigt sind und dass alle relevanten Sätze gelesen werden. Ansonsten wird die Statistik fehlerhaft sein.

**ANMERKUNG:** Sollte die Tabelle sehr viele Einträge enthalten empfielt es sich, zunächst mit wenig Sätzen ein entspechendes Script zu erstellen und es dann ohne Begrenzung der Treffer auszuführen.



# Neue Operation "Formelfelder fixieren"

Bis dato lebten Formelfelder nur solange, wie die entsprechende Formel. Aufgrund dieses Verhaltens war es schwierig, neue Werte auf Basis von bereits definierten zu ermitteln.

Um den Anwendern mehr Flexibilität bei der Gestaltung von Scripts anzubieten, ist die neue Operation "Formelfelder fixieren" eingeführt worden.

Die ausgewählten Formelfelder werden in ähnliche V-Felder konvertiert, möglicherweise mit anderen Namen, aber mit denselben Eigenschaften wie die Originale. Die Werte werden zum Zeitpunkt der Ausführung übernommen und bleiben danach unverändert. Die Formel und ihre Felder verschwinden. V-Felder können nach einem Join beibehalten werden. Sie verschwinden automatisch falls eine "Selektieren für alle Einträge" Operation durchgeführt wird.

**ANMERKUNG:** Nur die Formelfelder von sichtbaren Listenzeilen werden in die entsprechenden V-Felder übernommen – **ausgeblendete Sätze werden ignoriert** und erhalten Initialwerte, sobald sie wieder sichtbar gemacht werden.

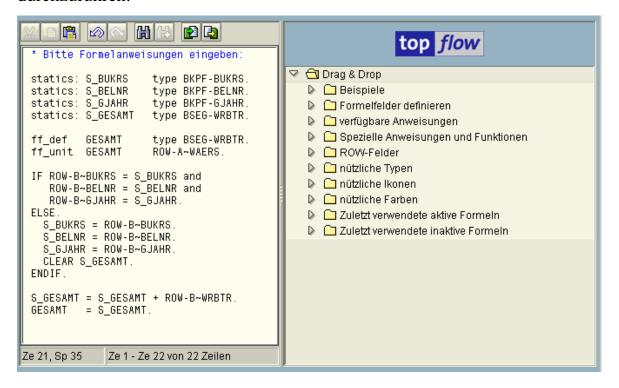
All das klingt relativ abstrakt – ein Beispiel wird verdeutlichen, wie diese Operation angewendet werden könnte.

Wir werden eine Liste von Segmenten von Buchhaltungsbelegen erstellen, jede Zeile mit dem Betrag und dem Prozentwert in Bezug auf den Gesamtbetrag des Beleges. Das Betragsfeld (BSEG-WRBTR) referenziert das Währungsfeld BKPF-WAERS. Aus diesem Grund werden wir das Beispiel mit den Kopfsätzen (mit EUR als Währung) beginnen und diese mit den entsprechenden Segmenten verbinden:

Join von Bl	Join von BKPF(A) und BSEG(B)						
A~BUKRS	A~BELNR	A~GJAHR	A~WAERS	B~BUZEI	B∼WRBTR		
9000	94000000000	2013	EUR	1	51,10		
9000	94000000000	2013	EUR	2	51,10		
9000	94000000000	2013	EUR	3	20,40		
9000	94000000000	2013	EUR	4	20,40		
9000	94000000001	2013	EUR	1	44,88		
9000	94000000001	2013	EUR	2	44,88		
9000	94000000002	2013	EUR	1	66,43		
9000	94000000002	2013	EUR	2	66,43		
9000	94000000002	2013	EUR	3	53,04		
9000	94000000002	2013	EUR	4	53,04		
0000	0400000000	2012	CIID	1	20 66		



Nachdem wir den Gesamtbetrag für jeden Beleg benötigen, sortieren wir zunächst die Liste nach A~BUKRS, A~BELNR, A~GJAHR und B~BUZEI und definieren dann eine Formel mit globaler Betrachtung, um die notwendige Berechnung durchzuführen:



Nach Aktivierung der Formel sieht das Ergebnis wie folgt aus:

S	SE16XXL - Inner-Join - 668 Ergebnis-Sätze									
5	🛐   🖪 📳   🔠   🔀   🖽 🖷 🖷									
Joi	Join von BKPF(A) und BSEG(B)									
	A~BUKRS	A~BELNR	A~GJAHR	A~WAERS	B~BUZEI	B~WRBTR	X~GESAMT			
	9000	94000000000	2013	EUR	1	51,10	51,10			
	9000	9400000000	2013	EUR	2	51,10	102,20			
	9000	9400000000	2013	EUR	3	20,40	122,60			
	9000	9400000000	2013	EUR	4	20,40	143,00			
	9000	94000000001	2013	EUR	1	44,88	44,88			
	9000	94000000001	2013	EUR	2	44,88	89,76			
	9000	94000000002	2013	EUR	1	66,43	66,43			
	9000	94000000002	2013	EUR	2	66,43	132,86			
	9000	94000000002	2013	EUR	3	53,04	185,90			
	9000	94000000002	2013	EUR	4	53,04	238,94			
	0000	0400000000	2012	CIID	1	20 66	20 66			

Das letzte Segment eines jeden Belegs enthält den Gesamtbetrag. Um die anderen Sätze zu entfernen, ändern wir die Sortierung von B~BUZEI auf "**absteigend**" und blenden dann die benachbarten Duplikate mithilfe von aus.

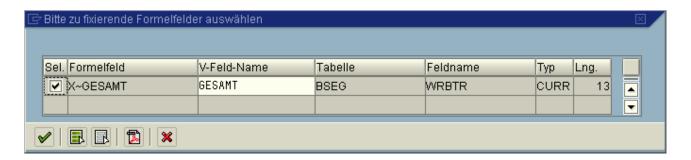


Die resultierende Liste wird von den nicht benötigten Spalten befreit:

Join von BKPF(A) und BSEG(B)						
A~BUKRS	A~BELNR	A~GJAHR	A~WAERS	X~GESAMT		
9000	9400000000	2013	EUR	143,00		
9000	94000000001	2013	EUR	89,76		
9000	94000000002	2013	EUR	238,94		
9000	94000000003	2013	EUR	110,28		
9000	94000000004	2013	EUR	16,32		

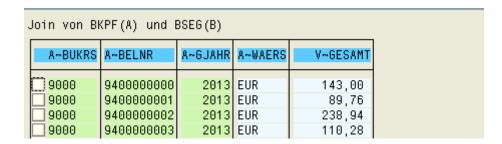
An dieser Stelle haben wir ein kleines Problem: wir brauchen das Formelfeld X~GESAMT, um die Prozentwerte der einzelnen Segmenten zu berechnen – d.h. eine neue Formel wird benötigt. Unglücklicherweise kann zu einem Zeitpunkt nur eine Formel existieren. Wenn wir aber die aktuelle Formel löschen, verschwindet das Feld X~GESAMT ebenfalls. Das neue Coding kann nicht zur aktuellen Formel hinzugefügt werden, weil sie eine Sortierfolge voraussetzt die nicht mehr gegeben ist. Was tun?

Die Lösung liegt in der neuen Operation "Formelfelder fixieren", die im Menü über *Bearbeiten → Formel → Formelfelder fixieren* zu erreichen ist. Folgendes Dialogfenster kommt zum Vorschein:



Wir lassen den Namen unverändert und betätigen die ✓ Taste. Die Liste enthält nun eine Spalte V~GESAMT anstatt von X~GESAMT.

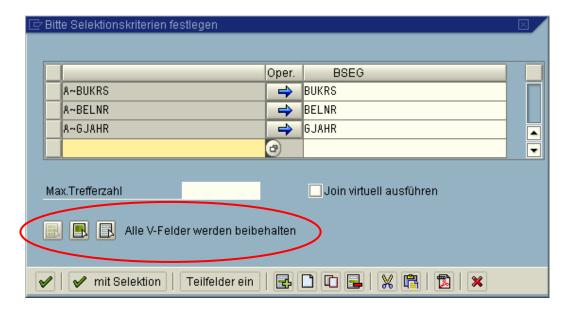
Die Formel selber ist nicht mehr da:



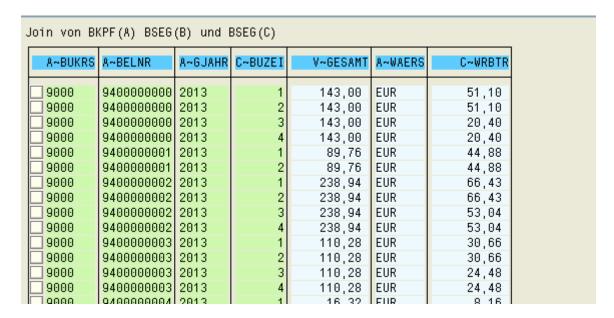
Um unnötige Daten loszuwerden, verwerfen wir die ausgeblendeten Sätze.



Nun, nachdem die bisherige Formel nicht mehr da ist und wir trotzdem den Gesamtbetrag haben, können wir ohne Probleme eine neue Formel zur Berechnung der Prozentwerte definieren. Zuvor müssen wir aber die entsprechenden BSEG-Einträge erneut selektieren, denn die bisherigen sind teilweise abhanden gekommen. Der Join wird also erneut durchgeführt. Das Dialogfenster für die Join-Kriterien weist dieses Mal zusätzliche Optionen in Bezug auf die V-Felder auf:



Nachdem wir das Feld V~GESAMT benötigen, lassen wir die Option unverändert. Die Ergebnisliste sieht nach Umsortierung und Anpassung wie folgt aus:



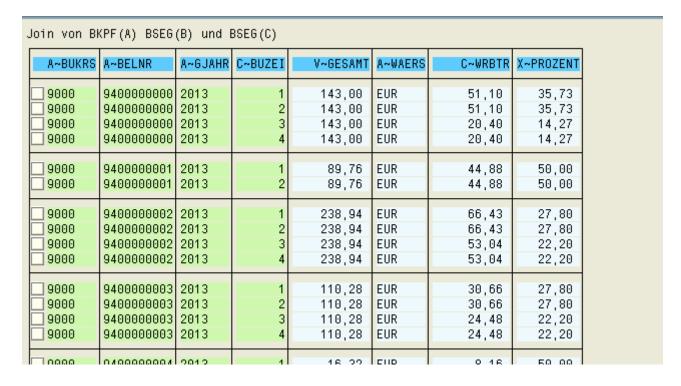
Jede Zeile enthält sowohl den Gesamtbetrag als auch den Segmentbetrag. Anhand dieser beiden Werten ist eine Berechnung des Prozentwertes sehr einfach.



Wir gestalten die Formel wie folgt:



Das Endergebnis (mit Trennlinien zwischen den einzelnen Belegen) ist:



Es ist nicht zwingend für das Resultat aber wir könnten erneut von der Funktion "Formelfelder fixieren" Gebrauch machen und X~PROZENT in V~PROZENT umwandeln.

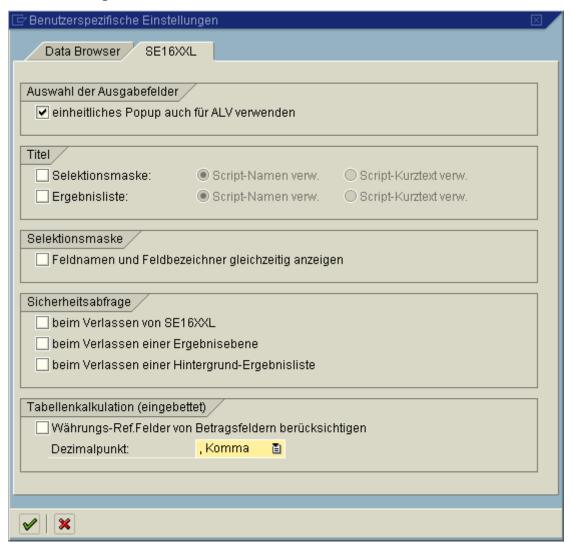


# **Neue Benutzer-Einstellungen**

Es ist nicht einfach, zwei Personen zu finden, die bei allen Aspekten eines bestimmten Themas die gleiche Meinung teilen. Gleiches gilt für die Art, wie sie eine bestimmte Aufgabe erledigen, oder ein bestimmtes Werkzeug anwenden. Aus diesem Grund sind neue benutzerspezifische Einstellungen für SE16XXL eingeführt worden, damit jeder die Art und Weise, in der das Programm unter bestimmten Umständen reagiert, individuell gestalten kann.

Die Benutzer-Einstellungen sind mithilfe der Drucktaste auf der Anwendungsleiste der Einstiegsmaske (sowie der Selektionsmaske) zu erreichen, ober als Menüfunktion *Einstellungen → Benutzerparameter* ... in der Ergebnisliste. Das Dialogfenster trägt nun zwei Reiter – der erste davon (**Data Browser**) ist unverändert geblieben.

Die neuen Optionen sind unter dem zweiten Reiter (SE16XXL) zu finden:



Die verschiedenen Einstellungen werden jetzt kurz erläutert.



#### Einheitliches Popup auch für ALV verwenden

Diese Option bewirkt, dass die neue, vereinheitlichte Feldauswahl für alle Anzeigemodi (ALV-List und -Grid inklusive) angewendet wird. Bei deaktivierter Option kommen die Standard-Dialogfenster von ALV wie gewohnt zum Einsatz.

#### Scriptname als Titel der Selektionsmaske verwenden

Bis dato verriet der Titel einer Selektionsmake nur etwas in Bezug auf die betroffene Datenbank-Tabelle oder View, ungeachtet eines evtl. ausgeführten Scripts. Mit der neuen Option wird der Name des aktuell ausgeführten Scripts verwendet. Die Einstellung wirkt nicht bei Zwischen-Selektionsmasken, denn in einem solchen Fall wird das Script "erweitert" und ist somit nicht mehr aktiv.

#### Scriptname als Titel der Ergebnisliste verwenden

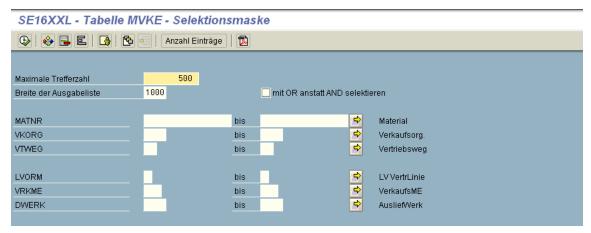
Auf ähnliche Weise kann der Titel der Ergebnisliste eines Scripts so gestaltet werden, dass darin der Name des Scripts angezeigt wird, anstelle der beteiligten Datenbank-Tabellen und/oder Views.

#### Selektionsmaske – Feldname und Feldbezeichner gleichzeitig anzeigen

Bis dato waren entweder die Namen oder die Bezeichner der Felder auf einer Selektionsmaske zu einem Zeitpunkt zu sehen – in Abhängigkeit von der primären Einstellung unter dem Data Browser Reiter:



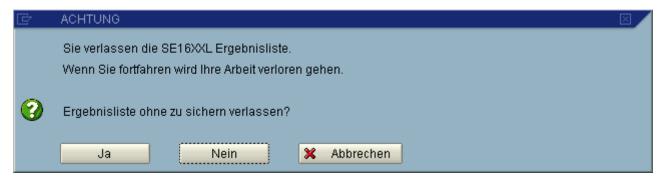
Mit der neuen Option sind beide Werte gleichzeitig auf allen Selektionsmasken sichtbar (die Anordnung hängt von der primären Einstellung ab):





#### Sicherheitsabfrage beim Verlassen von SE16XXL

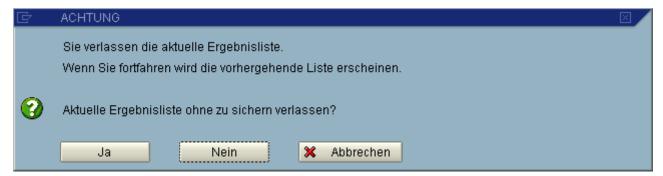
Bis dato wurde keine Warnung beim Verlassen der SE16XXL Sitzung ausgegeben, ungeachtet der Join-Ebene oder anderer komplizierten Operationen, die bis zu diesem Zeitpunkt durchgeführt worden waren. Dies konnte eine Menge Arbeit bedeuten, die es zu wiederholen galt, wenn der Anwender nichts gesichert hatte und z.B. versehentlich die falsche Taste betätigte. Man kann nun diese Option aktivieren, um solche Unannehmlichkeiten zu verhindern. Wenn die Sitzung verlassen wird, bewusst oder versehentlich, gibt SE16XXL folgenden Hinweis aus, wenn Operationen durchgeführt wurden, die nicht als Script gesichert wurden:



Die voreingestellte Antwort ist **Nein** – wenn man ENTER drückt, wird die Sitzung nicht verlassen.

#### Sicherheitsabfrage beim Verlassen einer Ergebnisebene

Diese Option ähnelt der oben beschriebenen mit dem Unterschied, dass der Hinweis auch beim Verlassen einer Ebene (durch Betätigung von Die Meldung in diesem Fall lautet:



Diese Einstellung ist aus Gründen der Vollständigkeit eingeführt worden, könnte aber meiner Meinung nach auf Dauer richtig nerven. Die Praxis wird es zeigen.

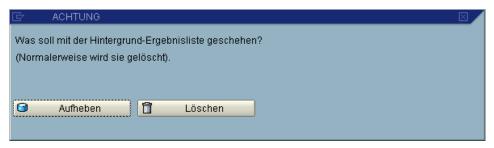
**ANMERKUNG**: Beide Optionen wirken weder mit Transaktion /TFTO/SE16SCRIPT noch mit von /TFTO/SE16PARAM abgeleiteten Parametertransaktionen.



#### Sicherheitsabfrage beim Verlassen einer Hintergrund-Ergebnisliste

Bis dato hatte das Verlassen einer im Hintergrund erstellten Ergebnisliste deren **Löschung** zur Folge. Dieses Verhalten war damit begründet, einer Überlastung des Systems durch unnötigen Datenmüll entgegenzuwirken. Die Kehrseite aber war, dass zwischendurch wichtige Resultate gelöscht wurden bevor sie vollständig überprüft worden waren.

Mit der Aktivierung dieser Option können derartige unangenehme Situationen vermieden werden. Anstelle einer lautlosen Löschung der betroffenen Hintergrundliste wird von SE16XXL folgender Hinweis ausgegeben:



Dadurch ist es Sache des Anwenders zu bestimmen, was danach geschieht.

**ANMERKUNG**: diese Option gilt auch für Transaktion /TFTO/SE16SCRIPT.

#### Tabellenkalkulation – Währungs-Ref.Felder von Betragsfeldern berücksichtigen

Währungsbeträge (Typ CURR) werden in SAP als gepackte Zahlen gespeichert. Die Anzahl Dezimalstellen ist im Data Dictionary definiert und nicht im Feld hinterlegt. Darüber hinaus sind solche Felder Währungsfeldern (Typ CUKY) zugeordnet. Im Customizing kann für jede Währung eine abweichende Anzahl Dezimalstellen festgelegt werden.

Der japanische Yen (JPY) z.B. ist eine Währung, die für gewöhnlich mit Null Dezimalstellen gecustomized wird – die meisten Währungen haben hingegen zwei. Wenn die eingebettete Tabellenkalkulation aufgerufen wird, werden die Betragsfelder normalerweise gemäß der Data Dictionary Definition ausgegeben. Ist stattdessen diese Option aktiv, wird das referenzierte Währungsfeld berücksichtigt. Beispiel:

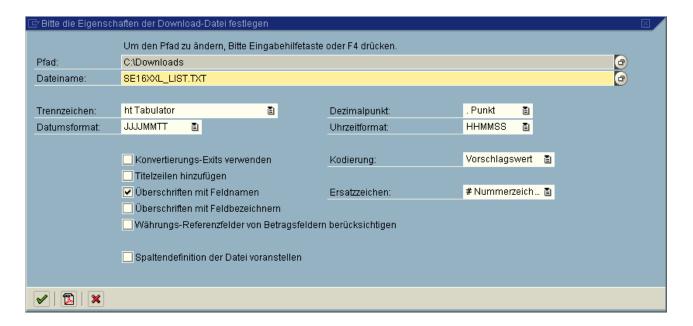
Angenommen, das Betragsfeld enthält den Wert **1234567**. Die normale Ausgabe wäre **12345,67**. Bei aktiver Option und Währung JPY wäre sie stattdessen **1234567**.



## **Download zum Frontend**

Es steht eine neue Funktion zum Herunterladen des Inhalts der Liste zur Verfügung. Sie ist über das Menü zu erreichen (*Liste*  $\rightarrow$  *Download zum Frontend*).

Das Haupt-Dialogfenster bietet ähnliche Optionen wie beim Sichern in eine Server-Datei:



Diese neue Download-Funktion ist flexibler als die Standard-Funktionen.

Eine Option ist besonders interessant:



Wenn diese Option aktiviert wird, wird der erzeugten Datei eine Definition der beteiligten Data Dictionary Felder vorangestellt. SE16XXL erkennt diese Spaltendefinition, wenn die Datei als Vorgabedatei hochgeladen wird. Eine Zuordnung der Spalten zu Dictionary Tabellenfeldern, wie dies bei normalen Dateien angebracht ist, erübrigt sich.

Für weitere Informationen siehe Download zum Frontend.



# Spezielle Kalenderfunktionen für Formeln

Aus Sicherheitsgründen dürfen Funktionsbausteine in Formeln nicht aufgerufen werden. Es gibt aber im SAP-System elementare Funktionen, die nur in Form von Funktionsbausteinen verfügbar sind. Einige davon betreffen den Kalender und die Werkskalender.

Mithilfe von ABAP-Anweisungen ist es möglich, Datumsberechnungen durchzuführen wie z.B. heute plus zehn Tage usw. Es ist aber nicht möglich, Informationen bezüglich der Woche eines Datums zu erhalten oder herauszufinden, ob das Datum ein Werktag in Verbindung mit einem bestimmten Werkskalender ist.

Um den Anwendern diese Funktionalität anbieten zu können ohne die Sicherheit zu beeinträchtigen, ist für Formeln eine Serie von speziellen Funktionen implementiert worden. Sie sind auf dem Drag&Drop-Baum auf der rechten Seite des Formeleditors zu finden:



Mehr Informationen in Bezug auf einzelne Funktionen können durch Doppelklick erhalten werden. Die angezeigte Dokumentation fasst jeweils die Funktionen einer Gruppe zusammen – diejenigen, die unabhängig von einem Werkskalender sind und die restlichen:



Um eine einfache Schnittstelle anzubieten, sind alle Komplikationen der beteiligten Funktionsbausteine der Gruppe SCAL, wie z.B. "DATE\_CONVERT\_TO\_FACTORYDATE" usw. entfernt worden. Im Fehlerfall liefern die Funktionen einen Initialwert zurück. Es ist also sehr zu empfehlen, bei Verwendung der Funktionen auf Initialwert abzufragen,

Die letzten Drei Funktionen benötigen einen Werkskalender. Zu diesem Zweck ist eine spezielle Anweisung (FF\_SET\_CALENDAR) eingeführt worden. Der Werkskalender kann als Literal, als Variable oder als globaler Parameter angegeben werden:

```
Beispiel 1 - als Literal:
                                                                           v
     FF_SET_CALENDAR '01'
   Beispiel 2 - als Variable:
     DATA: LL_KALID
                     TYPE SCAL-FCALID
     LL_KALID = '01'
     FF_SET_CALENDAR LL_KALID.
   Beispiel 3 - als Feld der aktuellen Zeile:
    FF_SET_CALENDAR ROW-B~FABKL. " --> T001W-FABKL
   Beispiel 4 - als globaler Parameter:
     FF_SET_CALENDAR ?KALID?
     Der globale Parameter KALID ist mit Hilfe der Menüfunktion
      Extras --> Globale Parameter definieren" separat definiert
                  Ze 1, Sp 1
                                                Ze 11 - Ze 33 von 35 Zeilen
\checkmark
```

Auf dem Drag&Drop-Baum rechts sind auch nützliche Typen verfügbar:

```
🔽 🔁 nützliche Typen
        TYPE
              SY-DATUM
                               Datum (JJJJMMTT)
        TYPE SY-UZEIT
                               Uhrzeit (HHMMSS)
        TYPE SY-INDEX
                               Integer
        TYPE SY-UNAME
                               Benutzername
        TYPE SCAL-DATE
                               Datum für Kalenderfunktionen
               SCAL-FACDATE
        TYPE
                               Fabrikdatum für Kalenderfunktionen
        TYPE
               SCAL-WEEK
                               Woche für Kalenderfunktionen
               SCAL-FCALID
        TYPE
                               Fabrikkalender
nützliche Ikonen
```

Mehr Infos zu diesem Thema sind in den Beispielen ( Deispiele ) zu finden.



# Script-Editor – Funktion "SELECT-Tabelle ersetzen"

Bisher war es nicht möglich, die Datenbanktabelle (oder View) einer bestimmten SELECT-Operation eines Scripts zu ersetzen. Es gibt aber Situationen bei denen eine solche Möglichkeit wünschenswert wäre.

Wenn z.B. eine Serie von Tabellen existiert, die entweder die gleiche Struktur oder ähnliche Strukturen aufweisen, und es ist ein kompliziertes Script für eine davon aufgebaut worden, könnte es praktisch sein, dieses eine Script zu kopieren und für jede der übrigen Tabellen der Serie anzupassen.

Ein weiteres Beispiel könnte das Ersetzen einer bestimmten Tabelle mit einem ihrer Projektions-Views sein, oder umgekehrt – aus welchem Grund auch immer.

Auf jeden Fall ist **diese Funktionalität ab jetzt im Script-Editor verfügbar**. Wie Sie aber annehmen dürfen, gibt es dazu einige Einschränkungen:

- Die Ersetzung ist immer möglich, wenn beide Tabellen die **exakt gleiche Struktur** aufweisen;
- Falls die beiden Strukturen abweichen, die Tabellen jedoch **identische Schlüsselfelder** besitzen, kann trotzdem die Ersetzung stattfinden, es sind allerdings weitere Anpassungen des Scripts nicht auszuschließen;
- Sind die Schlüsselfelder unterschiedlich, müssen alle "verwendeten" Schlüsselfelder (mit Ausnahme des Mandanten) in der Ersatztabelle vorhanden sein. "Verwendet" bedeutet, dass sie an irgendeiner Operation beteiligt sind, z.B. als Join-, Sortier- oder Filterkriterien. Die Eigenschaft als Ausgabe-Feld zu wirken (Operation CHOOSE\_LF), wird in diesem Zusammenhang ignoriert. Falls keines der Schlüsselfelder verwendet wird, muss zumindest das erste Schlüsselfeld übereinstimmen.
- In allen anderen Fällen ist eine Ersetzung nicht möglich.

Wenn ein bestimmtes Schlüsselfeld in der neuen Tabelle fehlt, aber irgendwo im Script verwendet wird, können Sie versuchen, diese Verwendung zu eliminieren. Im Erfolgsfall wird die Ersetzung möglich werden.

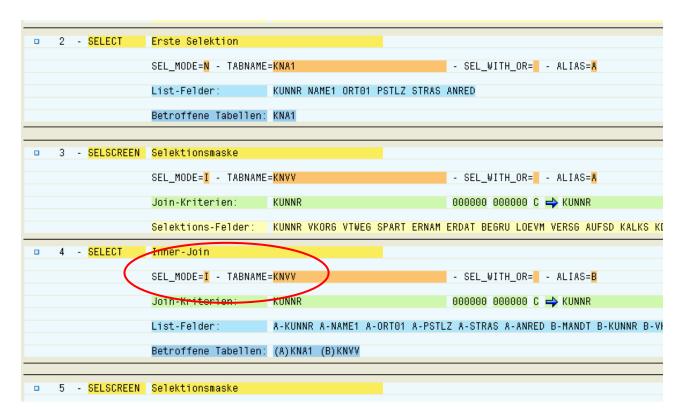
**ANMERKUNG:** Sogar die Tabelle der ersten SELECT-Operation ist ersetzbar, nicht aber im Falle einer Vorgabedatei.

Um das oben Gesagte zu verdeutlichen, werden wir ein paar Beispiele durchspielen.



#### Beispiel #1 Eine Tabelle ersetzen durch eine mit gleicher Struktur

Wir beginnen mit einem Script, das u.a. auch die Tabelle KNVV enthält:



Wir werden KNVV durch ZTXKNVV ersetzen, einer Tabelle mit exakt gleicher Struktur. Um dies zu bewerkstelligen, setzen wir den Cursor auf die entsprechende SELECT-Operation und verlangen mithilfe eines Rechtsklicks nach dem Kontextmenü:

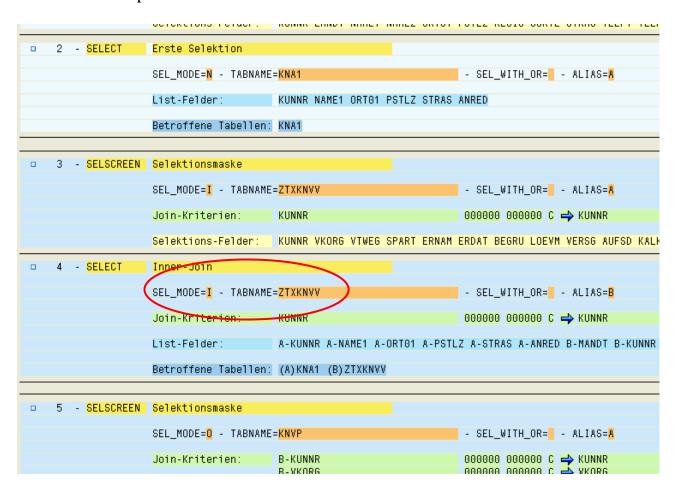




Der Script-Editor fragt uns daraufhin nach dem Namen der neuen Tabelle – wir geben ZTXKNVV ein:



Die SELECT-Operation weist nun die neue Tabelle auf – KNVV ist ersetzt worden:



Aufgrund der indentischen Struktur beider Tabellen liefert eine Konsistenzprüfung des Scripts ( ) folgende Meldung:



Wir haben nichts anderes erwartet.



#### Beispiel # 2 Unterschiedliche Strukturen, gleiche Schlüsselfelder

Wir werden nun KNA1 (die Tabelle der ersten SELECT-Operation) mit dem View W\_KNA1 ersetzen, der nur einige der Felder von KNA1 beinhaltet. Die Schlüsselfelder von beiden sind aber identisch.

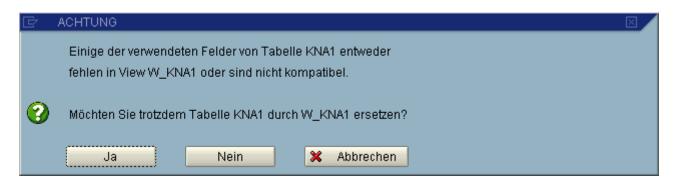
Wir gehen, wie beim ersten Beispiel, mit Hilfe des Kontextmenüs vor:



Als Ersatz spezifizieren wir W\_KNA1:



Nun konfrontiert uns der Script-Editor mit folgender Aufforderung:



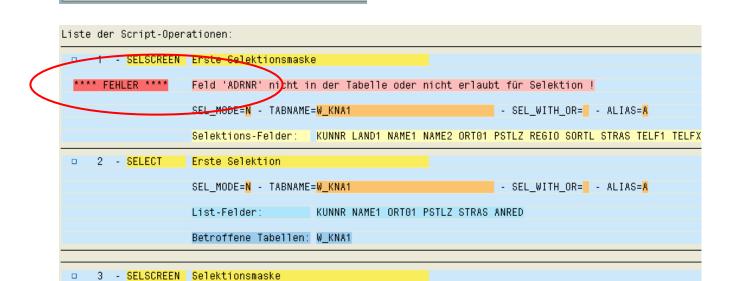
Wir lassen uns nicht einschüchtern und antworten mit



Zunächst erscheint das Ergebnis so wie wir es erwartet haben:

```
Liste der Script-Operationen:
     1 - SELSCREEN Erste Selektionsmaske
                       SEL MODE=N - TABNAME=W KNA1
                                                                            - SEL_WITH_OR= - ALIAS=A
                                            KUNNR LAND1 NAME1 NAME2 ORTO1 PSTLZ REGIO SORTL STRAS TELF1 TELFX XCPC
                       Selektions-Felder:
 □ 2 - SELECT
                       Erste Selektion
                       SEL_MODE=N - TABNAME=W_KNA1
                                                                            - SEL_WITH_OR=| - ALIAS=<mark>A</mark>
                       List-Felder:
                                             KUNNR NAME1 ORTO1 PSTLZ STRAS ANRED
                       Betroffene Tabellen: W_KNA1
     3 - SELSCREEN Selektionsmaske
                       SEL_MODE=I - TABNAME=KNVV
                                                                             - SEL_WITH_OR=<mark>|</mark> - ALIAS=<mark>A</mark>
                       Join-Kriterien: KUNNR
                                                                            000000 000000 C 📥 KUNNR
                       Selektions-Felder:
                                             KUNNR YKORG YTWEG SPART ERNAM ERDAT BEGRU LOEVM VERSG AUFSD KALKS KDGF
     4 - SELECT
                       Inner-Join
                       SEL_MODE=I - TABNAME=KNVV
                                                                             - SEL_WITH_OR=<mark>|</mark> - ALIAS=<mark>B</mark>
```

Jedoch müssen wir dieses Mal bei der Konsistenzprüfung ( ) feststellen, dass mehrere Anpassungen durchgeführt werden müssen:



Um ein funktionsfähiges Script zu erhalten, müssen wir sämtliche Inkonsistenzen eliminieren.

CEL MODE-T TARMAME-MNUV

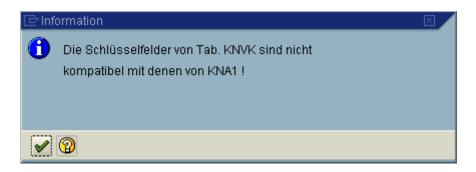
Note: Note: Die Script-Operationen sind NICHT konsistent!



## Ersetzung mit differierenden Schlüsselfeldern

Es können zahlreiche Situationen in diesem Zusammenhang auftreten. Wenn alle verwendeten Schlüsselfelder der bisherigen Tabelle in der Ersatztabelle vorkommen, dürfte die Ersetzung erfolgversprechend sein.

Werden hingegen keine davon verwendet, muss das erste Schlüsselfeld der neuen Tabelle (abgesehen vom Mandantenfeld) zum ersten der zu ersetzenden Tabelle passen. Ansonsten wird folgende Meldung ausgegeben:



#### **WICHTIG:**

Die Funktion "SELECT-Tabelle ersetzen" ist kompliziert. Eigentlich ist sie angedacht um Tabellen mit anderen gleicher Struktur zu ersetzen. Die übrigen Situationen, bei denen die Felder und / oder die Schlüsselfelder nicht übereinstimmen, sind mehr oder weniger als Zugabe zur Verfügung gestellt worden. Der aufmerksame Anwender sollte sich aber darüber im Klaren sein, dass in solchen Fällen weitere Anpassungen mit hoher Wahrscheilichkeit notwendig werden. Etwaige Zusatzmerkmale des Scripts, wie z.B. eine spezielle Selektionsmaske, Sprünge usw. werden demnach soweit wie möglich automatisch angepasst – bedürfen aber vielleicht noch einer manuellen Überarbeitung.

Fazit: Diese Funktionalität sollte mit Bedacht angewendet werden.



# Weitere Verbesserungen

Es sind mehrere kleinere Verbesserungen in Version 3.2 von SE16XXL entstanden. Sie werden in den nachfolgenden Seiten kurz erläutert.

#### Felder vom Typ STRING und XSTRING unterstützt

Bis dato waren Felder vom Type STRING und XSTRING weder in einer Ergebnisliste zu sehen, noch in der Detailanzeige einer Einzelzeile. Mit Version 3.2 stehen solche Felder als legitime Spalten einer Liste zur Verfügung. STRING-Felder können sogar als Filterkriterien verwendet werden. Als **Selektionskriterien** können sie weiterhin <u>nicht verwendet werden</u>, denn sie werden vom System nicht unterstützt (mit SAP\_BASIS Version 700 können PARAMETERS mit Typ STRING definiert werden, nicht aber SELECT-OPTIONS).

## Gepackte Felder (Typ P) beliebiger Länge auf Selektionsmasken

Bis dato konnten Felder vom Typ **P** (gepackt) mit einer internen Länge von mehr als **8 Bytes** nicht als Selektionskriterien verwendet werden. Grund waren gelegentliche Kurzdumps auf älteren Versionen des SAP-Systems. Dieses Problem scheint inzwischen gelöst zu sein – folglich können ab jetzt gepackte Felder beliebiger Länge auf Selektionsmasken erscheinen.

#### Neue Datumsvariablen für Script-Varianten

Folgende Datumsvariablen sind für Script-Varianten hinzugefügt worden:

```
Zweites Quartal ????
Drittes Quartal ????
Viertes Quartal ????
Vorhergehendes Quartal
Aktuelles Quartal
Iagesdatum - xxx, Tagesdatum + yyy (Tage)
Tagesdatum - xxx, Tagesdatum + yyy (Arbeitstage)
Vorhergehender Monat
Aktueller Monat
(Monatsanfang-xx Monate, Monatsende+yy Monate)
Vorjahr
Aktuelles Jahr
```

Darüber hinaus stehen ab jetzt manche Datumsvariablen auch für Felder vom Typ **ACCP** und **NUMC** (der Länge 6 oder 4) zur Verfügung.



#### ABAP-Anweisung "WRITE TO" verfügbar in Formeln

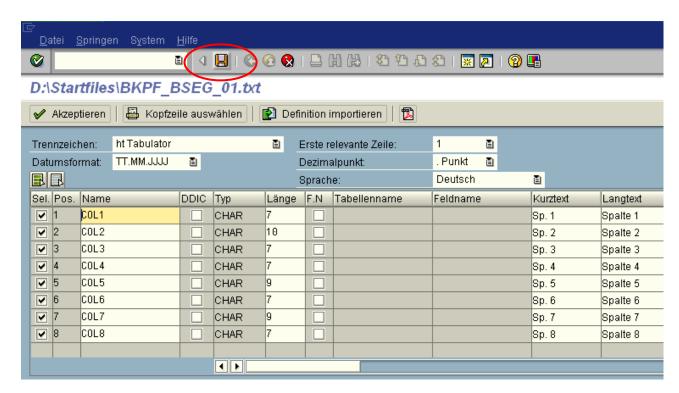
Bis dato war die ABAP-Anweisung "WRITE TO" in einer Formel nicht erlaubt. Diese Einschränkung ist nun aufgehoben worden.

Mit Ausnahme der Option "...**USING EDIT MASK** '==CONV' "bei der CONV eine Konvertierungs-Exit darstellt. Sie ist aus Sicherheitsgründen untersagt.

#### Spaltendefinition einer Vorgabedatei als Script gesichert

Bis dato war es beim Hochladen einer Vorgabedatei nicht möglich die Definition der Spalten zwischendurch zu sichern – man war gezwungen, bis zum Erscheinen der Ergebnisliste zu warten. Bei Vorgabedateien mit vielen Spalten musste z.B. im Falle eines Systemabsturzes die ganze Definitionsarbeit wiederholt werden – eine aufwendige und nicht besonders interessante Tätigkeit.

Die Definition kann nunmehr zu jeder Zeit mithilfe des Save-Buttons ( ) auf der Systemleiste gesichert werden:



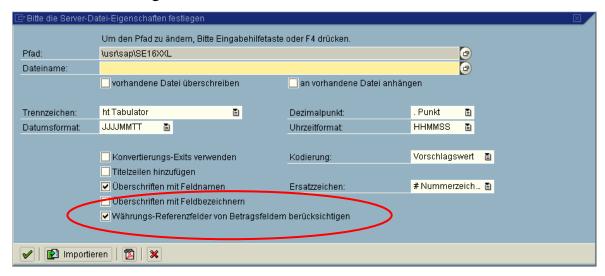
Daraufhin erscheint das bekannte Dialogfenster zum Sichern eines Scripts. Sollte aus irgendeinem Grund der Definitionsvorgang unterbrochen werden, kann er zu einem späteren Zeitpunkt mithilfe der Schaltfläche Definition importieren fortgesetzt werden.



#### Referenzfelder von Währungsbeträgen

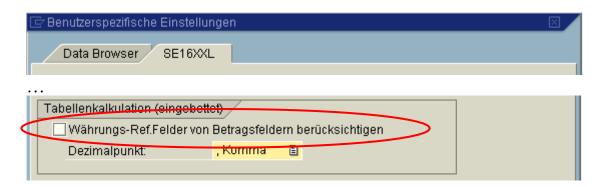
Bis dato wurden Betragsfelder (Typ CURR) beim Sichern in eine **Server-Datei** oder bei der eingebetteten Tabellenkalkulation immer mit den im Data Dictionary definierten Kommastellen ausgegeben, ungeachtet der assoziierten Währung. Das funktioniert problemlos mit den meisten Währungen, wie EUR oder USD, denn sie besitzen zwei Kommastellen wie die Mehrzahl der Betragsfelder. Wird aber eine Währung, wie z.B. der Japanische Yen (**JPY**), mit **Null** Dezimalstellen gecustomized, so ist das Ergebnis **100 Mal** kleiner als erwartet, indem **12345** Yen in **123,45** konvertiert werden. Diese Situation tritt selten auf – sogar bei manchen japanischen Firmen wird der Yen mit Zwei Dezimalstellen gecustomized, um solche Probleme zu vermeiden.

Ab jetzt ist eine entsprechende Option auf dem Dialogfenster zum Sichern in eine Server-Datei verfügbar:



Diese Option steht ebenfalls bei der neuen Funktion "**Download zum Frontend**" zur verfügung.

Im Falle der "Tabellenkalkulation (eingebettet)" ist die Option unter den "Benutzer-Parametern" zu finden:





#### Script-spezifische Maximale Trefferzahl

Es ist jetzt möglich, eine individuelle maximale Trefferzahl (als Defaultwert) einem Script zuzuweisen, vorausgesetzt es besitzt eine **spezielle Selektionsmaske**. Ein Beispiel wird die Vorgehensweise verdeutlichen.

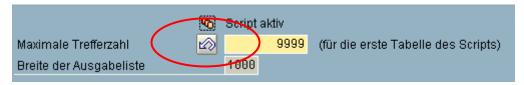
Im Script-Katalog wird die Drucktaste auf der Anwendungsleiste betätigt, nachdem der Cursor auf das gewünschte Script positioniert wurde. Daraufhin erscheint der Dialog zum Definieren einer speziellen Selektionsmaske:



Nun wird die Selektionsmaske Schaltfläche betätigt, um die Simulation der Selektionsmaske zu erhalten:



Um den individuellen Wert festzulegen, muss die Drucktaste auf der linken Seite des Feldes betätigt werden:



Es ist unschwer zu erkennen, dass die Drucktaste zum Zurücksetzen gedacht ist. Im Anzeigemodus deutet ein auf der linken Seite des Feldes darauf hin, dass der Wert individuell gesetzt wurde:





## Auswahl von ALV-Layouts bereits auf der Selektionsmaske

Bis dato konnten definierte ALV-Layouts nur in der Ergbnisliste – durch Betätigung der Drucktaste auf der Applikationsleiste – ausgewählt werden. Ab jetzt steht diese Funktion auch auf der Selektionsmaske zur Verfügung – vorausgesetzt, der Anwender befindet sich in einem der zwei ALV-Anzeigemodi. Es wird damit möglich, bereits im Vorfeld das gewünschte Layout zu selektieren.

Ein paar Anmerkungen sind zu diesem Thema jedoch angebracht:

- ALV-Layouts sind keinem Script zugeordnet. Sie sind nur mit der Struktur der Ergebnisliste gekoppelt d.h. mit den beteiligten Tabellen und/oder Views.
- Im Falle einer Änderung der Listenstruktur eines Scripts, z.B. durch Hinzufügen einer neuen Tabelle, können die bisherigen Layouts nicht mehr verwendet werden.
- ALV-Layouts können keine Filterinformationen enthalten, weil SE16XXL von der Filter-Funktionalität von ALV keinen Gebrauch macht.

Die Tatsache, dass ALV-Layouts von Scripts entkoppelt sind, hat technische Ursachen: ALV geht davon aus, dass ein Layout einem bestimmten Report zugeordnet ist – im Falle von SE16XXL ist der Report immer derselbe, unabhängig von der Struktur der Liste. Sollte diese Strategie verwendet werden, würden sämtliche Layouts bei der F4-Hilfe auftauchen, unabhängig von der Beschaffenheit der Ergebnisliste. Diese Schwierigkeit ist dadurch überwunden worden, indem jedes Layout einem Pseudo-Report zugeordnet wird, dessen Name von der beteiligten Tabelle (bzw. Tabellen) abgeleitet wird. Zum Glück prüft ALV nicht nach ob der Report wirklich existiert (andere Programme tun es).

Die Kehrseite dieses Mechanismus ist, dass ein einziger Report bei der F4-Hilfe angegeben werden kann. Wenn Layouts an Scripts gebunden wären, würden dann nur solche erscheinen, auch wenn viele andere passend zu den betroffenen Tabellen existieren würden.



# Zugriffsrechte mithilfe von SAP-Berechtigungs-Rollen zugeordnet

Dies ist der wichtigste Aspekt der neuen Version. Ist eigentlich interessanter für Administratoren als für normale Anwender.

Bis dato wurden die komplexen Zugriffsrechte für Tabellen und Felder meistens anhand von speziellen "Zugriffsrollen" definiert. Diese Rollen hatten keinerlei Bezug zu den SAP-Berechtigungs-Rollen, und somit musste die Zuordnung zu den Anwendern getrennt von der normalen Berechtigungspflege durchgeführt werden. Eine unbefriedigende Situation, speziell für große Firmen,.

Die neue Version macht es möglich, SE16XXL Zugriffsrollen mit normalen SAP-Berechtigungs-Rollen zu verbinden:



Indem der Administrator die relevanten SAP-Rollen den Benutzern zuordnet, ordnet er **indirekt** die dazugehörigen Zugriffsrollen zu.

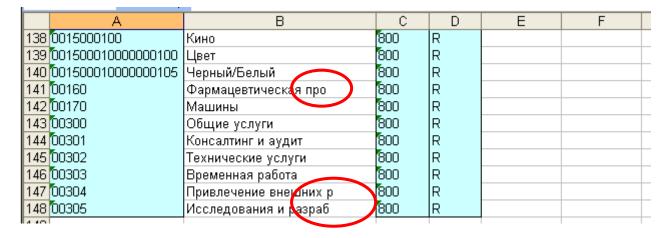
Ferner kann die Pflege der Zugriffsrechte gemäß dem "Vier-Augen-Prinzip" durchgeführt werden – d.h. eine Person definiert die Zugriffsrollen, eine zweite Person ordnet sie den Anwendern zu.



# Probleme mit der eing. Tabellenkalkulation und Kyrillisch gelöst

Die Funktion "Tabellenkalkulation (eingebettet)", die auf eine Standard-SAP-Funktionalität basiert, hat auf Unicode-Systemen ein Problem mit Multibyte-Codepages, wie z.B. Kyrillisch. Wenn ein Text in Kyrillisch in der eingebetteten Tabellenkalkulation angezeigt wird, werden längere Texte abgeschnitten:

PRODH	VTEXT	MANDT	SPRAS
☐ 0015000100 ☐ 0015000100000000100 ☐ 001500010000000105 ☐ 00160 ☐ 00170 ☐ 00300 ☐ 00301 ☐ 00302 ☐ 00303	Кино Цвет Черный/Белый Фармацевтическая промышленность Машины Общие услуги Консалтинг и аудит Технические услуги Временная работа	800 800 800 800 800 800 800 800	R R R R R R R
□ 00304 □ 00305	Привлечение внешних ресурсов Исследования и разработки	800 800	R R



Dieses Problem hat mit der Implementierung der Standard-SAP-Funktionalität zu tun. Um den Anwendern die Komplikationen von SAP-Hinweisen zu ersparen, ist SE16XXL ertüchtigt worden, das Problem an der Quelle zu eliminieren. Wenn nun auf einem Unicode-System die Funktion "Tabellenkalkulation (eingebettet)" aufgerufen wird, wird eine logische Kopie der zu exportierenden Zeilen angefertigt, in der jedes Feld vom Typ C bis zu fünfmal länger als im Original ist. Damit wird die Standard-Funktionalität überlistet und es findet keine Abschneidung statt.

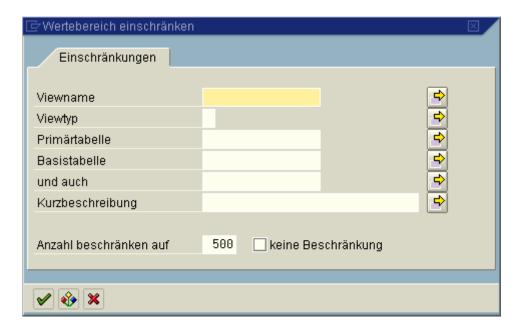


## **F4-Hilfe für Views**

SE16XXL hat seit Beginn Datenbank- und Projektions-Views unterstützt. Die Standard-F4-Hilfe für Tabellen auf der ersten Maske war aber (und ist weiterhin) auf Datenbanktabellen beschränkt. Dieses Manko ist jetzt beseitigt worden. Die erste Maske ist zu diesem Zweck aufgewertet worden, indem eine neue Schaltfläche rechts neben dem Tabellennamen hinzugefügt worden ist:



Wird diese Schaltfläche betätigt, erscheint folgendes F4-Dialogfenster:



Mehr erfahren Sie unter F4-Hilfe für Views.



# Formel nach einem Join beibehalten

Bis dato war der Geltungsbereich einer Formel auf die Ergebnisliste beschränkt, in der sie definiert worden war. Sobald ein "Inner-" oder "Outer-Join" durchgeführt wurde, wurde die aktive Formel verworfen und musste, falls nötig, erneut in der resultierenden Join-Liste definiert werden.

Es stimmt, dass alle zuletzt definierten Formeln im Formel-Editor unter dem Baum-Knoten (rechts) "Zuletzt verwendete aktive Formeln" zu finden sind, aber die meisten Anwender haben das vermutlich nicht bemerkt – darüber hinaus müssen Aliases den beteiligten Feldern hinzugefügt werden, wenn von einer simplen Liste zu einer Join-Liste übergegangen wird.

Diese Unannehmlichkeiten gehören der Vergangenheit an. Wird jetzt ein Join durchgeführt, erscheint eine entsprechende Option, falls eine Formel aktiv ist:



Aliases werden automatisch hinzugefügt, falls notwendig:

```
IF ROW-AENAM IS INITIAL.

USER_NAME = ROW ERNAM.

ELSE.

USER_NAME = ROW-A-ENAM.

ELSE.

USER_NAME = ROW-A-AENAM.

ENDIF.
```

Diese Option gilt nicht für Formeln mit **globaler Betrachtung**, denn sie benötigen eine Sortierfolge, die in der Ergebnisliste nicht vorhanden ist.



# **Dokumentation umstrukturiert**

Die <u>Haupt-Dokumentation</u> von SE16XXL ist teilweise neu geschrieben worden, um die Entwicklungen der letzten Jahre zu berücksichtigen. Darüber hinaus tragen jetzt viele Masken und Dialogfenster die Schaltfläche , um die spezifische Dokumentation anzuzeigen. Somit ist es für die Anwender einfacher, die passende Dokumentation zum richtigen Zeitpunkt zu bekommen.

Ferner sind sämtliche <u>Tutorials</u> jetzt komplett neu geschrieben worden. Auf diese Weise wird der Leser von den neueren Funktionen profitieren, wie z.B. "Sprünge" oder spezielle Listeüberschriften.



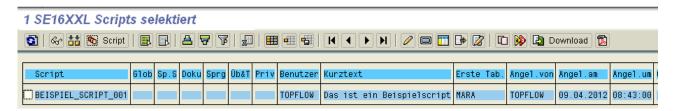
# Direktaufruf des Script-Katalogs beim Sichern eines Scripts

Beim Anlegen eines Scripts, ist es jetzt möglich, unmittelbar zum Script-Katalog zu springen, um zusätzliche Funktionalitäten hinzuzufügen, wie z.B. eine spezielle Selektionsmaske usw. Das Dialogfenster zum Sichern ist zu diesem Zweck entsprechend erweitert worden:



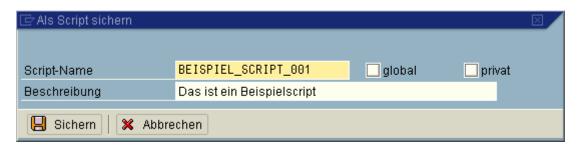
Wird nach Eingabe der notwendigen Parameter die Schaltfläche

Sichern und Script-Katalog aufrufen betätigt, so wird die aktuelle Sitzung von SE16XXL beendet, und der Script-Katalog für das aktuelle Script eröffnet:



Dem Anwender stehen alle Funktionen des <u>Script-Katalogs</u> zur Verfügung, z.B. der Script-Editor, die Definition von Überschriften, usw.

ANMERKUNG: wird das Script aus dem Katalog mithilfe von gestartet, wird die Schaltfläche Sichern und Script-Katalog aufrufen auf dem Dialogfenster fehlen, um eine Rekursion zu verhindern:



#### zum Anfang

Copyright © 2009 - 2018 - SY-TABIX GmbH