

## RFC-Selektion

Ab SE16XXL **Version 3.5** ist es möglich, eine **Join-Operation** (einschließlich “Selektion für alle Einträge”) mit einer Tabelle oder View aus einem **Remote-SAP-System** via RFC durchzuführen. **RFC** steht für **Remote Function Call** – es ist eine SAP-Funktionalität, die den Datenaustausch zwischen verschiedenen SAP-Systemen ermöglicht. Dadurch ist es beispielsweise möglich, die Einträge derselben Tabelle aus zwei verschiedenen Systemen zu vergleichen, ohne die Einträge auf einem System herunterladen und auf das andere hochladen zu müssen.

Bevor wir mit der Beschreibung dieser Funktionalität aus der Sicht der Benutzerschnittstelle fortfahren, kann es durchaus interessant sein, etwas über RFC und die Implementierung zu erzählen, die keineswegs unkompliziert ist.

### Allgemeine Informationen

Um Daten mit einem anderen SAP-System austauschen zu können, ist es notwendig, eine entsprechende **RFC-Verbindung** zu definieren. Dies erfolgt durch den Administrator mithilfe der Transaktion **SM59**. Für SE16XXL sind nur Verbindungen vom Typ “**3**” (Verbindung zu ABAP-System) relevant. Jede definierte Verbindung hat einen Namen (“**RFC-Destination**” genannt), der die RFC-Verbindung **eindeutig** identifiziert. Die Definition umfasst das Zielsystem und möglicherweise den Standard-Logon-Benutzer usw. Eine RFC-Verbindung kann auch als “**Trusted System**” definiert werden – in diesem Fall wird kein RFC-Anmeldebild angezeigt, wenn eine Verbindung mit dem Zielsystem hergestellt wird – der aktuelle Logon-Benutzer wird stattdessen implizit übernommen.

### Sicherheitsaspekte

Sicherheit ist wahrscheinlich der wichtigste Aspekt in Bezug auf RFC-Verbindungen. Aus diesem Grund beschränkt die SE16XXL-Implementierung die RFC-Selektion entweder auf **Trusted Systems** oder auf eine Verbindung, bei der der Remote-Logon-Benutzer mit dem lokalen Benutzer **identisch** ist. Die implizite Annahme ist, dass ein gegebener Logon-Benutzer die gleiche Person auf allen beteiligten Systemen darstellt – eine Annahme, die eigentlich als selbstverständlich angesehen werden kann. Trusted Systems werden dringend empfohlen, da in diesem Fall das RFC-Anmeldebild nicht angezeigt wird. Wenn eine “normale” RFC-Verbindung verwendet wird, wird das RFC-Anmeldebild während einer RFC-Selektion **mehrmals** angezeigt, denn eine vorhandene RFC-Verbindung wird automatisch geschlossen, wenn ein SUBMIT intern durchgeführt wird (und SE16XXL verwendet mehrere SUBMITs). Der Logon-Vorgang muss jedes Mal wiederholt werden – **mit genau den gleichen Parametern**. Ist dies nicht der Fall, weigert sich das Programm, die RFC-Selektion fortzusetzen.

## RFC-Destinationen

Auf einem einzelnen SAP-System können Dutzende oder sogar Hunderte von RFC-Destinationen definiert sein (in Transaktion SM59). Sie stehen **nicht automatisch** als Ziele für eine SE16XXL RFC-Selektion zur Verfügung. Vielmehr muss der Administrator in den **SE16XXL-Einstellungen** festlegen, welche Destinationen von wem verwendet werden dürfen (auf drei Ebenen: alle Benutzer, Benutzergruppen und Einzelbenutzer). In der F4-Hilfe für RFC-Destinationen werden nur die erlaubten angezeigt, zusammen mit “NONE”, die immer implizit erlaubt ist. “NONE” ist eine besondere Destination (mit dem aktuellen System, Mandanten und Benutzer als Ziel), die zum Testen der RFC-Selektion verwendet werden kann, falls andere Destinationen nicht verfügbar sind.

## Globale Einstellung

Um ein bestimmtes System vor RFC-Zugriffen durch andere Systeme zu schützen, gibt es eine **globale Einstellung** in den SE16XXL-Einstellungen, die vom Administrator aktiviert werden muss. Diese Einstellung ist standardmäßig (nach dem Upgrade auf Version 3.5) **deaktiviert**. Dies bedeutet, dass das System kein Ziel für RFC-Selektionen sein kann, auch wenn andere Systeme es als mögliche Destination deklariert haben.

**ANMERKUNG:** Die RFC-Destination “NONE” funktioniert auch dann, wenn die globale Einstellung deaktiviert ist.

## Implementierung

Man muss bedenken, dass ein Remote-System eine andere SAP-Version haben kann, als das lokale System. Dies bedeutet, dass im Data Dictionary Tabellen und Views enthalten sein können, die entweder auf dem lokalen System **unbekannt** sind oder deren **Strukturen sich** möglicherweise von den lokalen **unterscheiden**. Selbst lokal unbekannte Datentypen können in diesen Strukturen vorhanden sein (**INT8**, **DECFLOAT16** und **DECFLOAT34** sind nicht auf allen SAP-Systemen präsent).

Eine weitere Schwierigkeit wird durch **Konvertierungsexits** dargestellt. Dies sind Routinen, die ein internes Datenformat in ein externes konvertieren und umgekehrt. Sie werden an einige Domänen (elementare Datenelemente) angeheftet und werden vom System während der Dynpro-Verarbeitung und bei der Erstellen von Listen **automatisch** aufgerufen (WRITE-Anweisung). Das Problem bei Konvertierungsexits ist, dass ihr **Coding**, zumindest theoretisch, von System zu System **unterschiedlich** sein kann. Dies bedeutet, dass es nicht korrekt ist, den internen Wert eines Felds von einem Remote-System zu holen und in externen Wert auf dem lokalen System zu konvertieren.

Sobald die RFC-Destination bekannt ist, kontaktiert das Programm das Zielsystem und prüft, ob es die Remote-Selektion zulässt. Die SE16XXL-Version muss **mindestens 3.5** sein – ältere Versionen unterstützen diese Funktionalität nicht.

Das Programm stellt dann sicher, dass der Logon-Benutzer auf dem Remote-System mit dem auf der aufrufenden Seite identisch ist. Falls kein Trusted System verwendet wird, kann es nämlich passieren, dass sie unterschiedlich sind.

An dieser Stelle kann der Anwender die Datenbank-Tabelle oder View für die Selektion spezifizieren. Die **F4-Hilfe** wird ebenfalls **via RFC** auf dem Zielsystem ausgeführt. Sobald die Tabelle bekannt ist, ruft das Programm die Definition aus dem Remote-Data-Dictionary ab und überprüft gleichzeitig alle **Zugriffsrechte**, die auf dem Remote-System vorhanden sein können. Die Remote-Struktur der Tabelle wird dann mit der aus dem lokalen Data Dictionary verglichen. Alle Felder, die entweder nicht im lokalen System vorhanden sind oder sich in ihren Eigenschaften unterscheiden, werden durch ein geeignetes Feld auf lokaler Seite **ersetzt**.

Unbekannte Datentypen werden wie folgt ersetzt:

Original-Datentyp	Ersatz-Datentyp
INT8	DEC 19,0
DECFLOAT16	FLTP
DECFLOAT34	FLTP

DECFLOAT-Variablen (dezimaler Gleitkomma) können Werte annehmen, die größer sind als FLTP (Gleitkomma) darstellen kann. In solchen Fällen wird der ursprüngliche Wert durch einen sehr großen Wert bestehend aus lauter 9 ersetzt.

Nach diesen Präliminarien ruft das Programm die **Selektionsmaske** für die gewählte Datenbank-Tabelle/View auf. Dies erfolgt **via RFC** auf dem Remote-System. Diese Vorgehensweise hat den Vorteil, dass die Original-DDIC-Definition auf dem Original-System verwendet wird und somit nichts auf dem lokalen System simuliert wird.

Schließlich können die Tabelleneinträge aus der Remote-Datenbank selektiert werden. Das Programm übermittelt die notwendigen Informationen an das Remote-System, wo ein entsprechendes dynamisches Coding generiert und ausgeführt wird. Sämtliche **Berechtigungs-Prüfungen**, die normalerweise lokal angewendet werden, werden auch während der RFC-Selektion auf der Remote-Seite durchgeführt. Die selektierten Tabelleneinträge werden dann in ein geeignetes Format konvertiert und an den Aufrufer auf der lokalen Seite übermittelt, wo sie wieder in das normale Format konvertiert werden.

Felder mit einem **Konvertierungsexit** werden besonders behandelt. Für diese Felder werden **zwei Werte** übertragen, der **interne** und der **externe** Wert. Die Tabellenstruktur auf der lokalen Seite enthält für solche Felder **zwei separate Felder**, die alternativ beim Erstellen der Ergebnisliste verwendet werden, abhängig von den Einstellungen des Benutzers (mit oder ohne Konvertierungsexit).

Es gibt **drei Ausnahmen** von der oben beschriebenen Behandlung von Feldern mit Konvertierungsexit:

- 1) Bei Feldern mit Konvertierungsexit **ALPHA** wird nur der interne Wert übertragen und der Konvertierungsexit lokal aufgerufen. Dies geschieht unter der Annahme, dass dieser Konvertierungsexit auf allen Systemen auf die gleiche Weise funktioniert.
- 2) Felder vom Typ **LANG** (Sprachschlüssel) werden ähnlich behandelt – es wird nur der interne Wert an das lokale System übermittelt. Der Konvertierungsexit **ISOLA** wird dann lokal verwendet, um den externen Wert zu erzeugen. Der Grund für diesen Ansatz wird deutlich werden, wenn die spezielle Selektionsmaske von Scripts erläutert wird. Es ist zu beachten, dass die Einträge der zugrunde liegenden Tabelle (T002) von System zu System leicht abweichen können – es ist ein notwendiger Kompromiss.
- 3) Felder vom Typ **UNIT** (Einheitenschlüssel für QUAN-Felder) werden ebenfalls nur durch einen Wert, den internen, dargestellt. Der externe Wert wird über den Konvertierungsexit **CUNIT** erzeugt. Diese spezielle Handhabung ist notwendig, denn diese Felder werden von ALV benötigt, wenn Spalten aufsummiert werden, und ALV kann nicht dazu veranlasst werden, zwei unterschiedliche UNIT-Felder (interner / externer Wert) für diese Art von Operationen zu verwenden. Es ist darauf hinzuweisen, dass die Einträge von Tabelle T006A, auf die der Konvertierungsexit CUNIT basiert, ebenfalls von System zu System leicht abweichen können.

## Formelfelder und V-Felder

Formelfelder, die von Remote-Feldern abgeleitet werden, erben die “Remote“-Eigenschaft. Allerdings wird nur der **interne Wert** berücksichtigt, d.h. sie werden intern nicht durch zwei separate Felder (für internen und externen Wert) dargestellt. Dasselbe gilt für **V-Felder**, die direkt oder indirekt von Remote-Feldern abgeleitet werden.

## Script-Implementierung

Damit ein Script mit RFC-Selektionen mit **unterschiedlichen** RFC-Destinationen ausgeführt werden kann, gibt das Programm ein **Dialog-Fenster** aus, um dem Anwender die Möglichkeit zu bieten, die **Original-RFC-Destination(en)** des Scripts zu **ersetzen**. Selbstverständlich wird die Durchführbarkeit des Scripts **nach** dieser Ersetzung überprüft. Falls das Script mit den neuen RFC-Destinationen nicht ablauffähig ist, wird der Vorgang mit einer entsprechenden Fehlermeldung beendet. Die Original-RFC-Destinationen werden nur dann unverändert verwendet, wenn das Script erweitert wird, d. h. wenn Überschriften, Sprünge oder eine spezielle Selektionsmaske hinzugefügt oder geändert werden oder wenn das Script editiert wird.

## Spezielle Selektionsmaske

Die spezielle Selektionsmaske eines Scripts erfordert eine separate Erörterung. Der Grund dafür ist, dass einige der Blöcke auf eine **Remote-Tabelle** oder, im Falle von Filterblöcken, auf ein **Feld einer Remote-Tabelle** verweisen können. Wie wir bereits gesehen haben, wird die normale Selektionsmaske auf dem Remote-System ausgegeben, wodurch keine besondere Behandlung erforderlich ist, da alle Informationen “lokal” für das Remote-System sind.

Anders verhält es sich auf einer **speziellen Selektionsmaske**, denn in diesem Fall müssen die Select-Options für ein Remote-Feld auf dem lokalen System verarbeitet werden. Für normale Felder ist dies eine einfache Angelegenheit. Felder, die einem **Konvertierungsexit** zugeordnet sind, sind komplizierter. Normalerweise zeigen solche Felder auf einer Selektionsmaske ihre externen Werte, die auch in der F4-Hilfe angezeigt werden. Jedoch sollte bei Remote-Feldern der Konvertierungsexit nur auf dem Remote-System durchlaufen werden. Dies geschieht automatisch für die F4-Hilfe, die auf dem Remote-System stattfindet. Sobald jedoch ein Wert ausgewählt wurde, wird er an das lokale System übermittelt, wo der Konvertierungsexit nicht aufgerufen werden darf. Eigentlich könnte die F4-Hilfe sowohl interne als auch externe Werte übertragen. Allerdings könnte der Anwender einen Wert direkt eingeben, ohne die F4-Hilfe zu verwenden. In diesem Fall müsste das Programm den externen Wert in den internen Wert konvertieren, die notwendige Routine befindet sich jedoch auf dem Remote-System. Eine solche Vorgehensweise würde bedeuten, dass alle notwendigen Konvertierungsexits ständig via RFC aufgerufen würden, ein umständliches Verfahren.

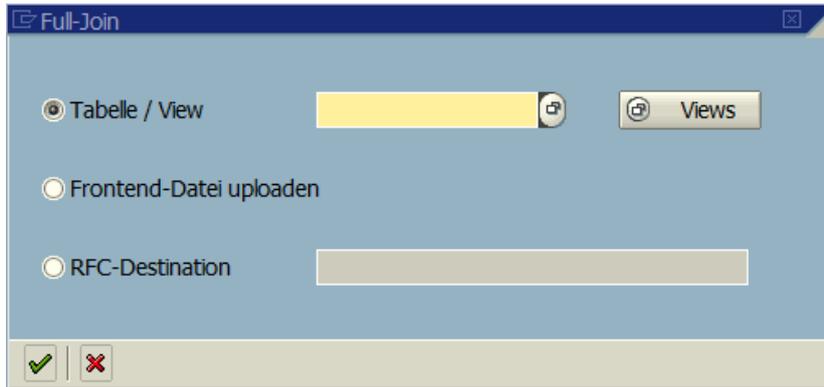
Um es kurz zu machen, wurde beschlossen, das Handling dieser Situationen zu vereinfachen. Demzufolge werden in den Select-Options der Remote-Felder immer die internen Werte angezeigt. Die oben beschriebenen Fälle (**ALPHA**, **LANG** und **UNIT**) bilden eine Ausnahme. Felder vom Typ **LANG** verwenden insbesondere den lokalen Konvertierungsexit **ISOLA**, um den externen Wert anzuzeigen, denn die internen Werte sind für normale Anwender völlig **unverständlich** (z.B. steht der interne Wert “4” für “**RO**” (Rumänisch)).

Nach diesen einleitenden Bemerkungen zur Implementierung können wir nun mit der Beschreibung der RFC-Selektion aus der Sicht der Benutzeroberfläche fortfahren. Auf den folgenden Seiten werden die verschiedenen Aspekte dieser Funktionalität näher erläutert. Die Beispiele werden sich unter anderem auf die spezielle Handhabung von Konvertierungsexits konzentrieren. Es sollte jedoch bedacht werden, dass nur etwa **5% aller Felder von transparenten Tabellen** mit einem **Konvertierungsexit** versehen sind, ein Prozentsatz, der die gesamte Situation relativieren dürfte.

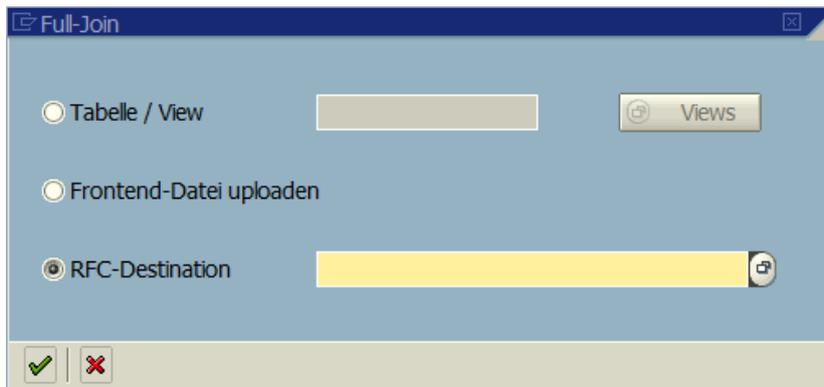
## Beschreibung einer RFC-Join-Operation

Wir möchten die lokalen Einträge der Tabelle **DD01L** (Domänen) vom Typ **STRING** mit denen auf einem Remote-System vergleichen. Nachdem wir DD01L wie üblich selektiert haben, beginnen wir die Full-Join-Operation durch Betätigen der  Full Schaltfläche auf der Applikationsleiste.

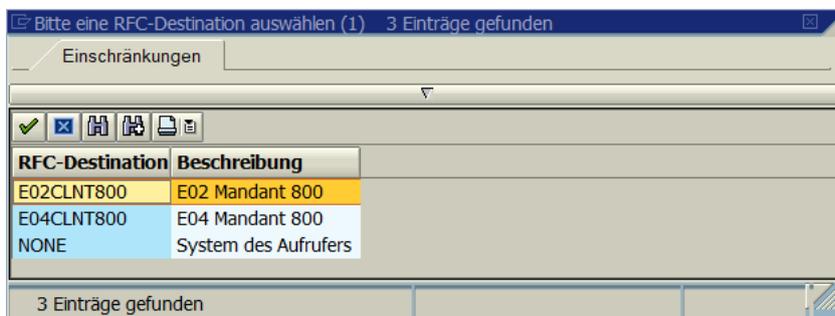
Als Reaktion gibt das Programm folgendes Dialog-Fenster aus:



Wir wählen die Option **“RFC-Destination”**:



Um eine Liste der **erlaubten Destinationen** zu erhalten, nutzen wir die F4-Hilfe:



Wir wählen den ersten Eintrag aus und machen weiter – im Dialogfenster können wir nun die Datenbank-Tabelle oder View auswählen. Wir geben DD01L direkt ein:

Was folgt, ist die Festlegung der Join-Kriterien:

	DD01L	Oper.	DD01L[E02]
	DOMNAME	→	DOMNAME
	DATATYPE	→	DATATYPE

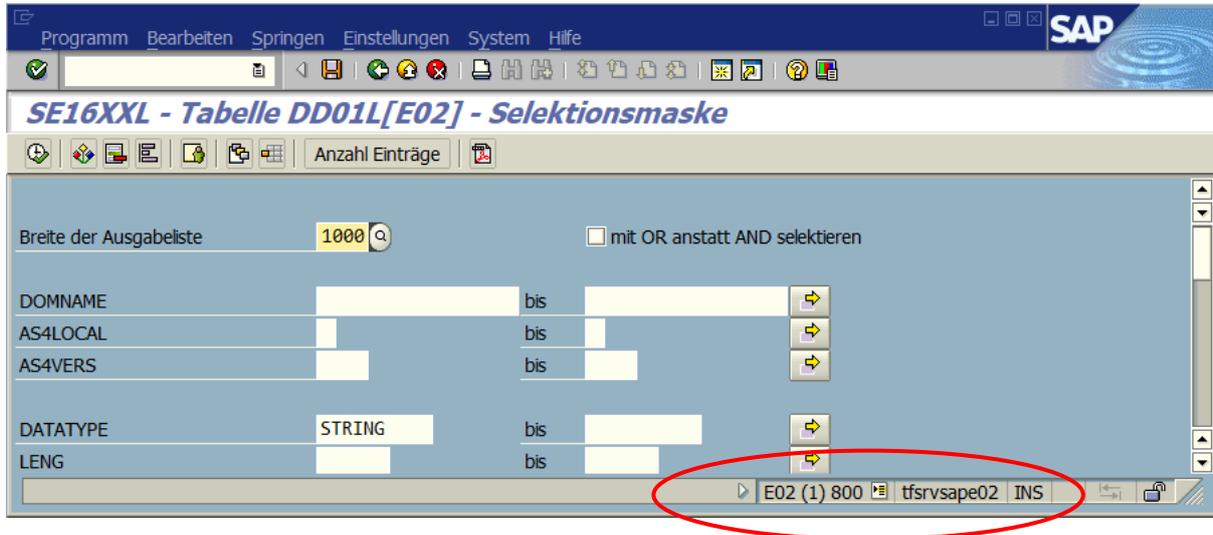
Beachten Sie, dass die neue Tabelle auf der rechten Seite **“DD01L[E02]”** heißt, um anzudeuten, dass es sich um eine Remote-Tabelle auf dem System **E02** handelt.

Nach Betätigung von **mit Selektion** erscheint das Dialogfenster für die gemeinsamen Felder des Full-Joins:

Tabelle	Feldname	Typ	Lng.	V-Feld
DD01L [E02]	DOMNAME	CHAR	30	DOMNAME
DD01L [E02]	DATATYPE	CHAR	4	DATATYPE

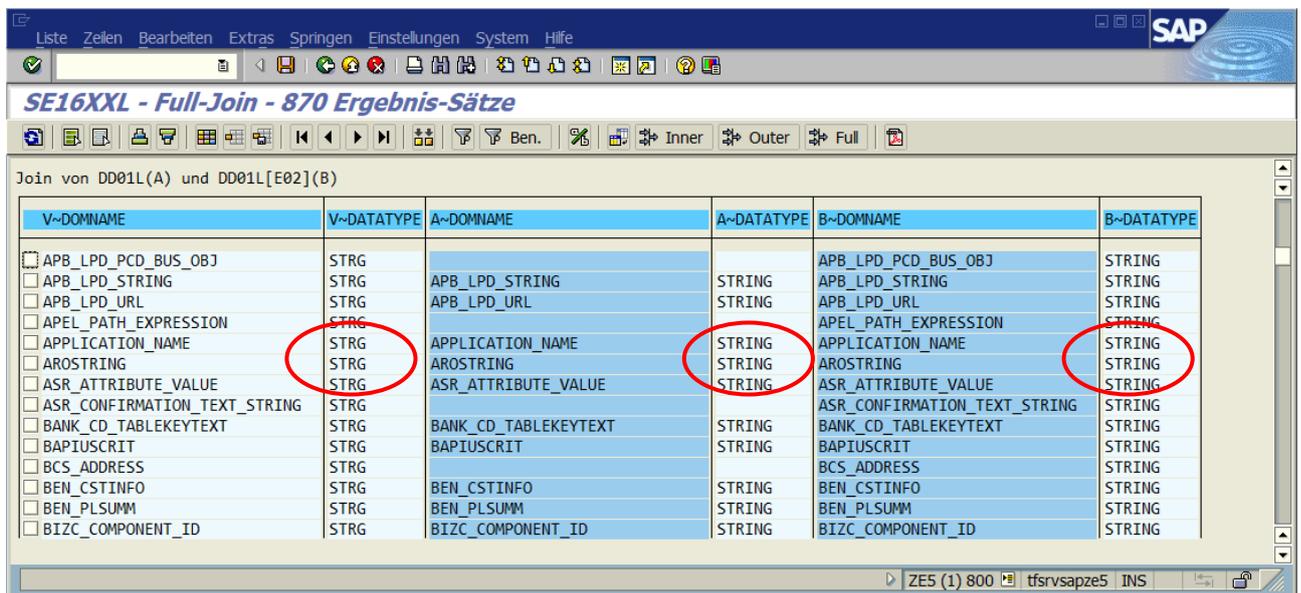
Auch hier heißt die Tabelle **“DD01L[E02]”**.

Nun kommt die Selektionsmaske zum Vorschein:



Die Informationen am unteren Rand zeigen uns, dass diese Selektionsmaske auf dem **Zielsystem** ausgegeben wurde. Wir geben **STRING** in die Select-Option für **DATATYPE** ein und betätigen anschließend  auf der Anwendungsleiste.

Die Ergebnisliste erscheint:



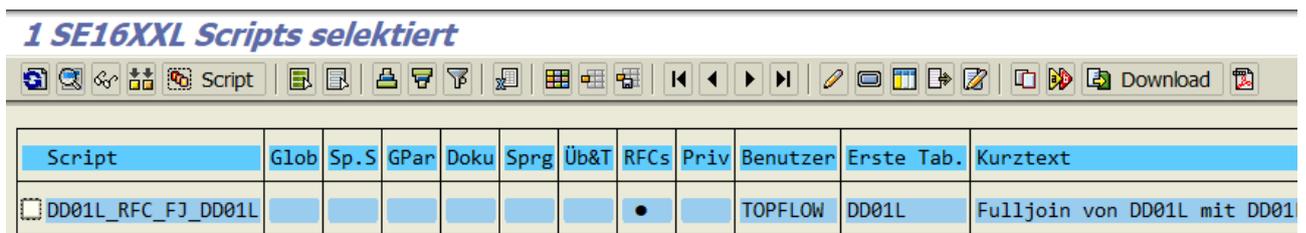
Das Feld **DATATYPE** (Datentyp) ist mit einem **Konvertierungsexit** ausgestattet. Der interne Wert **STRG** erscheint als **STRING** in den Spalten **A~DATATYPE** und **B~DATATYPE**, während er in **V~DATATYPE** unkonvertiert angezeigt wird, denn V-Felder, wie bereits zuvor erwähnt, erben den externen Wert nicht.

Nun, um zu zeigen wie sich ein Script mit RFC-Selektionen verhält, speichern wir unsere Arbeit als Script, indem wir auf  auf der Systemfunktionsleiste drücken:



Darüber hinaus statten wir das Script mit einer speziellen Selektionsmaske aus, um die Besonderheiten in Bezug auf Remote-Tabellen zeigen zu können. Daher betätigen wir , um direkt zum Script-Katalog zu springen, wo die spezielle Selektionsmaske definiert werden kann:

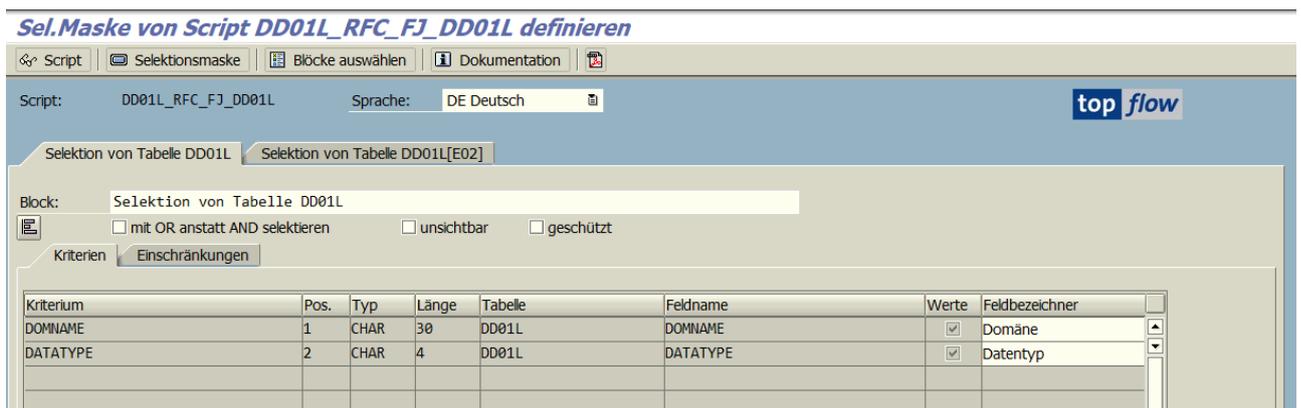
**1 SE16XXL Scripts selektiert**



Script	Glob	Sp.S	GPar	Doku	Sprg	Üb&T	RFCs	Priv	Benutzer	Erste Tab.	Kurztext
DD01L RFC FJ DD01L							•		TOPFLOW	DD01L	Fulljoin von DD01L mit DD01

Mithilfe von  auf der Anwendungsleiste gelangen wir zur Definition der speziellen Selektionsmaske:

**Sel.Maske von Script DD01L RFC FJ DD01L definieren**



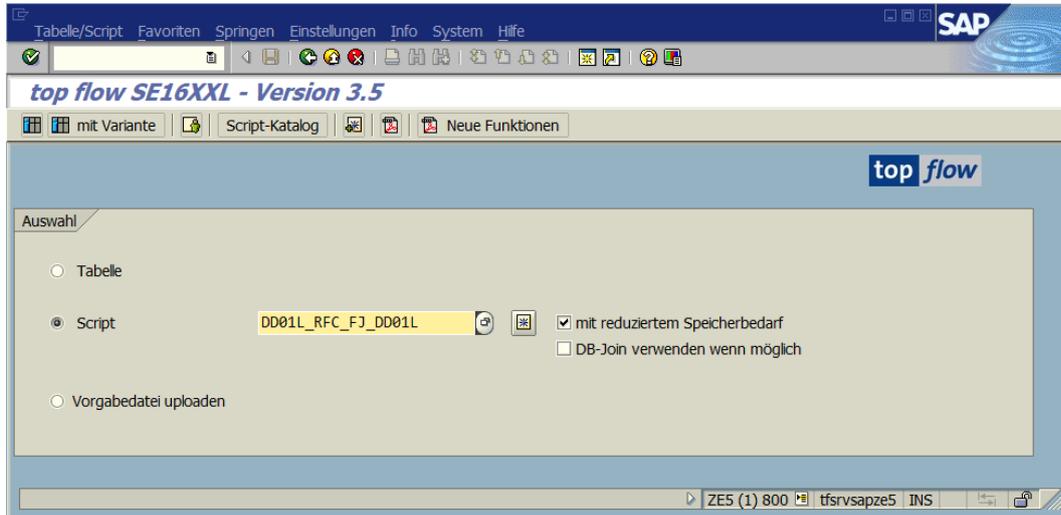
Kriterium	Pos.	Typ	Länge	Tabelle	Feldname	Werte	Feldbezeichner
DOMNAME	1	CHAR	30	DD01L	DOMNAME	<input checked="" type="checkbox"/>	Domäne
DATATYPE	2	CHAR	4	DD01L	DATATYPE	<input checked="" type="checkbox"/>	Datentyp

Wir wählen die notwendigen Kriterien aus und sichern die so aufgebaute spezielle Selektionsmaske.

Nun sind wir bereit zu zeigen, wie sich das Script bei der Ausführung verhält.

## Ausführung eines Scripts mit RFC-Selektionen

Das neu erstellte Script wird nun ausgeführt:



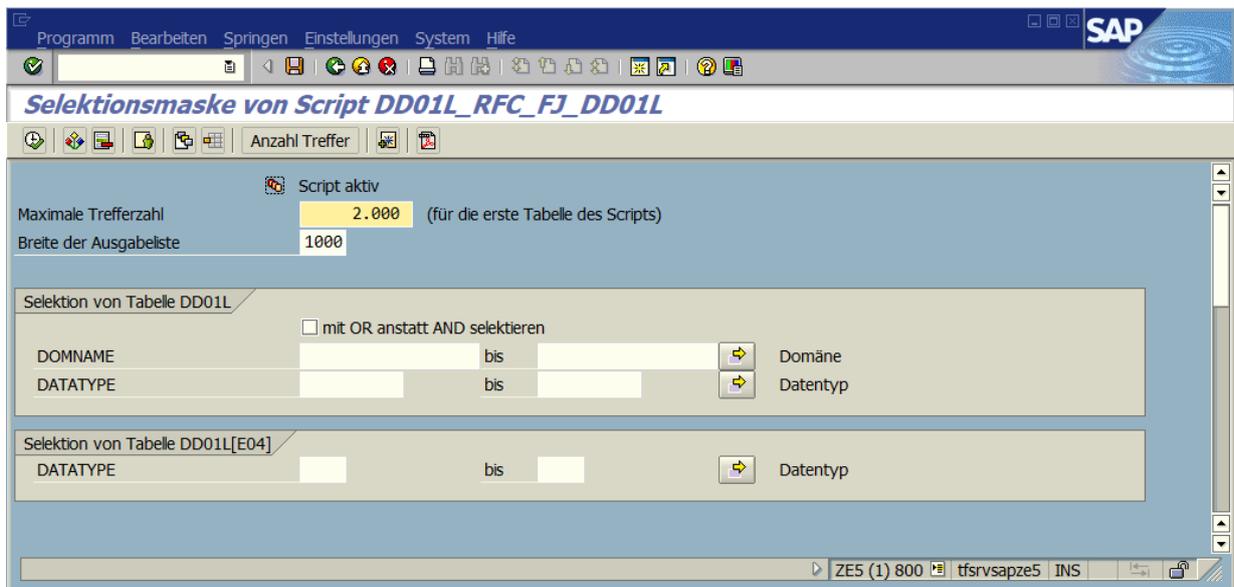
Nach Betätigung von  auf der Anwendungsleiste antwortet das Programm mit folgendem Dialogfenster:



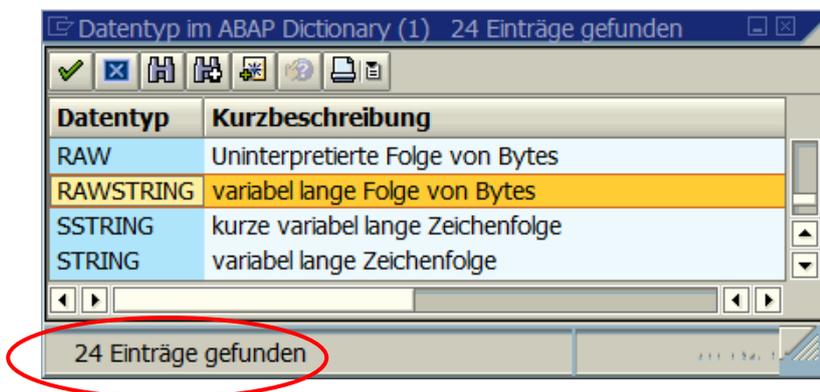
Dieses Dialogfenster bietet uns die Möglichkeit, die **Original-RFC-Destination** des Scripts durch eine andere zu **ersetzen**. Wir nutzen diese Gelegenheit und wählen über die F4-Hilfe **E04CLNT800**, die mit dem Remote-System **E04** verbunden ist.

**ANMERKUNG:** Unser Beispielscript verwendet nur eine **einzelne** RFC-Destination. Es ist jedoch möglich, RFC-Selektionen mit **verschiedenen Destinationen nacheinander** durchzuführen und dadurch Scripts mit **mehreren** RFC-Destinationen zu erstellen. In diesem Fall würde obiges Dialogfenster alle beteiligten Destinationen anzeigen, die auf Wunsch durch andere ersetzt werden könnten. Es ist jedoch **nicht gestattet, zwei oder mehr** Original-Destinationen durch **nur eine** zu ersetzen, da dies die **Logik des Scripts gefährden würde**, das davon ausgeht, dass verschiedene RFC-Destinationen tatsächlich unterschiedlich sind.

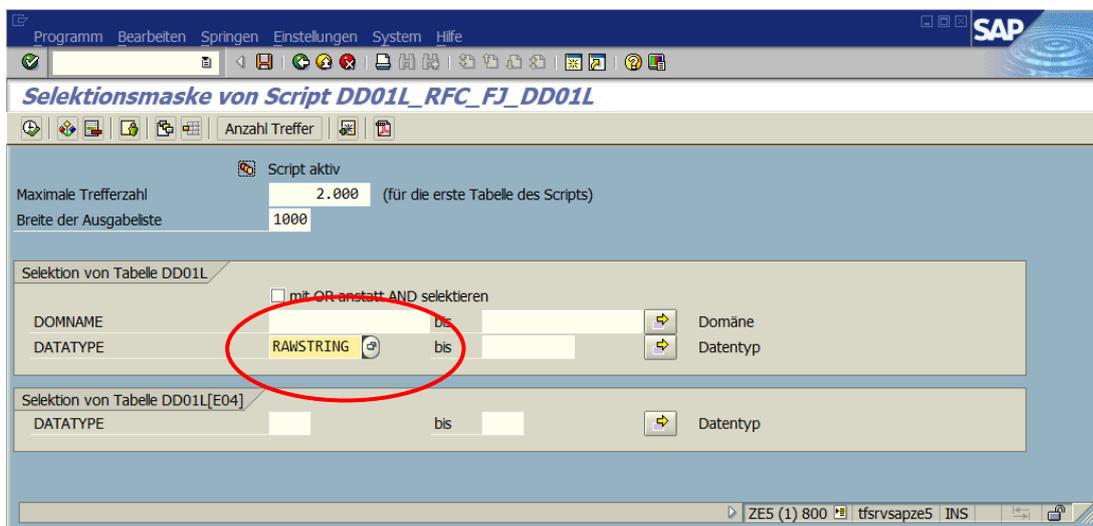
Nach Betätigung von  bekommen wir die spezielle Selektionsmaske:



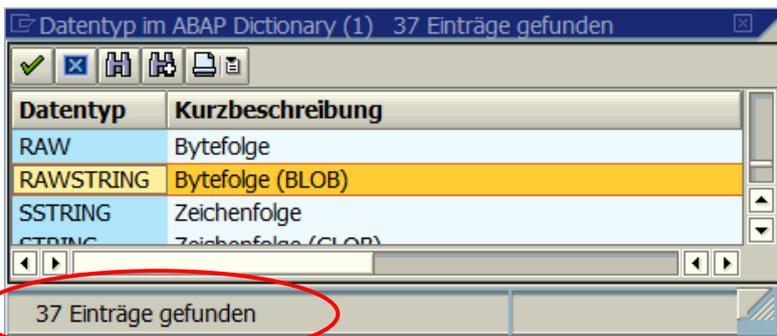
Wir zeigen nun das unterschiedliche Verhalten der F4-Hilfe für **lokale** und **remote** Kriterien. Wir beginnen mit der F4-Hilfe für die **lokale DATATYPE**-Select-Option:



Das Ergebnis auf der Selektionsmaske zeigt sich wie folgt:

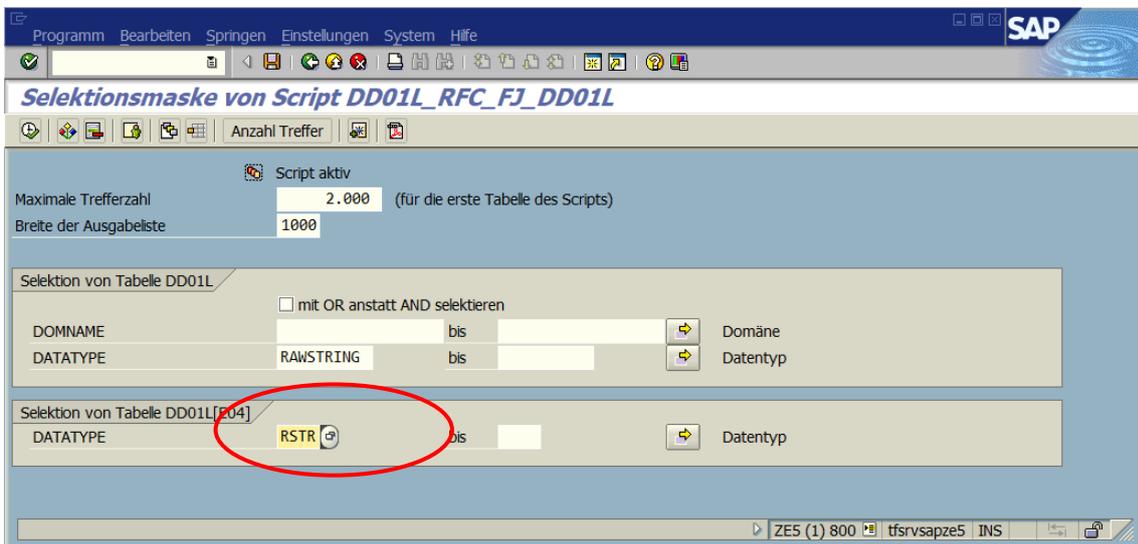


Nun wiederholen wir das Ganze für die **Remote-Select-Option DATATYPE**:

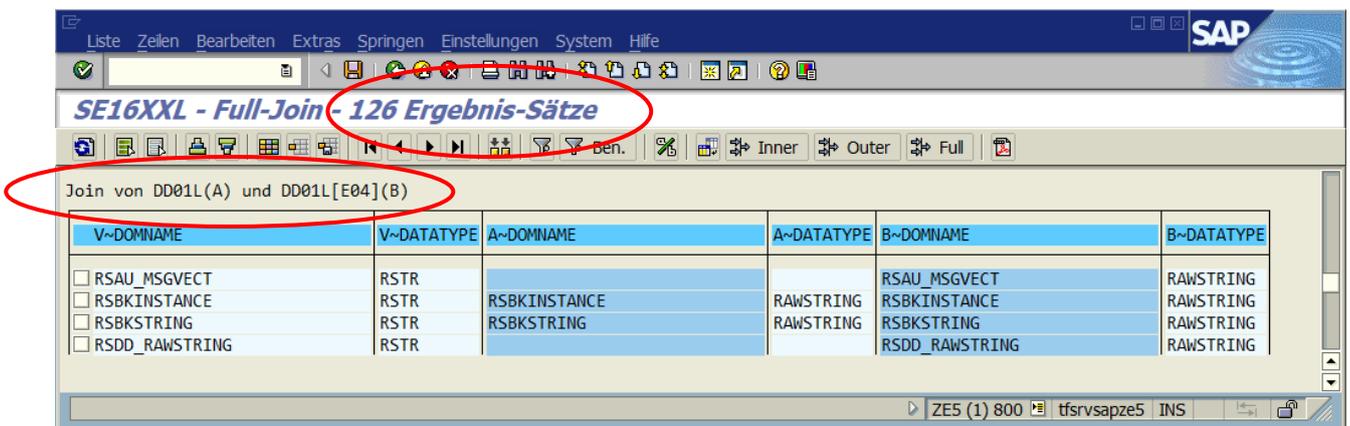


Beachten Sie, dass die Anzahl der Einträge unterschiedlich ist, was darauf hinweist, dass die F4-Liste auf einem anderen System ausgegeben wurde.

Das Ergebnis zeigt diesmal **den internen Wert**:



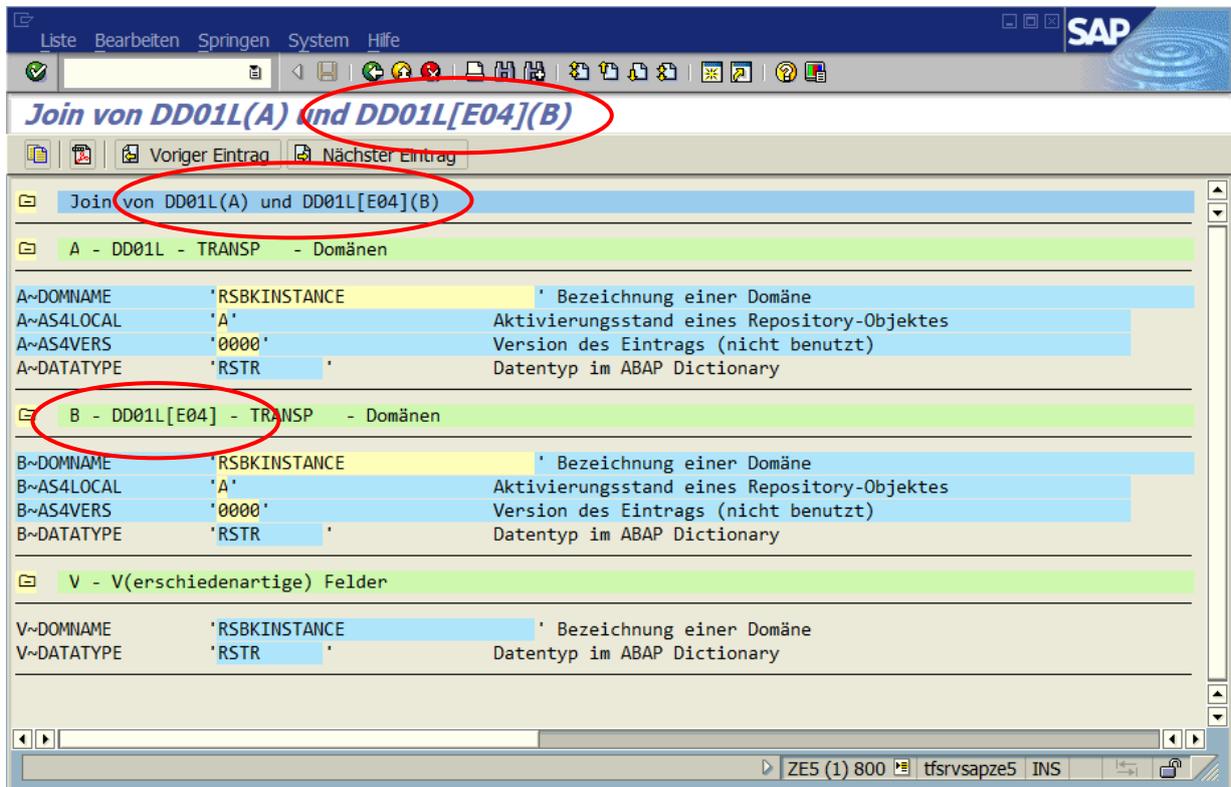
Es ist jetzt soweit. Wir können  betätigen um die Ergebnisliste zu erhalten:



Die Anzahl der Einträge unterscheidet sich von der ersten Ausführung mit System **E02**. Und die Überschrift sagt uns deutlich, dass die Remote-Einträge tatsächlich auf dem **E04** System selektiert wurden.

## Anzeige eines Ergebnis-Eintrags in Detail

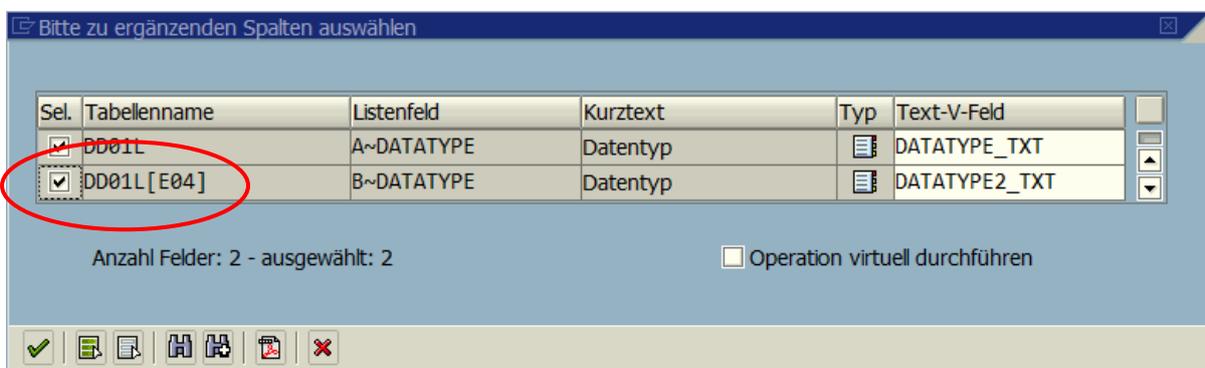
Durch Doppelklick auf eine Zeile der Ergebnisliste erhalten wir die Detailanzeige:



Auch hier erscheint die Remote-Tabelle immer als “DD01L[E04]”.

## Operation “Zusatztexte einfügen”

Die Operation *Extras* → *Zusatztexte einfügen*, die in der Ergebnisliste verfügbar ist, sucht nach den zugehörigen Texten im **lokalen** System für **lokale** Felder und auf dem entsprechenden **Remote**-System für **Remote**-Felder:



Die daraus resultierende Liste zeigt deutlich, dass die Texte aus verschiedenen Systemen stammen:

SE16XXL - Full-Join - 126 Ergebnis-Sätze

Join von DD01L(A) und DD01L[E04](B)

V~DOMNAME	V~DATATYPE	A~DOMNAME	A~DATATYPE	V~DATATYPE_TXT	B~DOMNAME	B~DATATYPE	V~DATATYPE2_TXT
<input type="checkbox"/> RSAU_MSGVECT	RSTR				RS AU_MSGVECT	RAWSTRING	Bytefolge (BLOB)
<input type="checkbox"/> RSBKINSTANCE	RSTR	RSBKINSTANCE	RAWSTRING	variabel lange Folge von Bytes	RSBKINSTANCE	RAWSTRING	Bytefolge (BLOB)
<input type="checkbox"/> RSBKSTRING	RSTR	RSBKSTRING	RAWSTRING	variabel lange Folge von Bytes	RSBKSTRING	RAWSTRING	Bytefolge (BLOB)
<input type="checkbox"/> RSDD_RAWSTRING	RSTR				RSDD_RAWSTRING	RAWSTRING	Bytefolge (BLOB)
<input type="checkbox"/> RSDRSTRING	RSTR	RSDRSTRING	RAWSTRING	variabel lange Folge von Bytes	RSDRSTRING	RAWSTRING	Bytefolge (BLOB)
<input type="checkbox"/> RSIS_XML	RSTR	RSIS_XML	RAWSTRING	variabel lange Folge von Bytes	RSIS_XML	RAWSTRING	Bytefolge (BLOB)
<input type="checkbox"/> RSO_RES_XML	RSTR				RSO_RES_XML	RAWSTRING	Bytefolge (BLOB)
<input type="checkbox"/> RSRD_RAWSTRING	RSTR	RSRD_RAWSTRING	RAWSTRING	variabel lange Folge von Bytes	RSRD_RAWSTRING	RAWSTRING	Bytefolge (BLOB)
<input type="checkbox"/> RSTT_BLOB	RSTR	RSTT_BLOB	RAWSTRING	variabel lange Folge von Bytes	RSTT_BLOB	RAWSTRING	Bytefolge (BLOB)
<input type="checkbox"/> RSWR_DATA_XSTRING	RSTR	RSWR_DATA_XSTRING	RAWSTRING	variabel lange Folge von Bytes	RSWR_DATA_XSTRING	RAWSTRING	Bytefolge (BLOB)
<input type="checkbox"/> SADL_RAP_OPXY_EDMX_XSTRING	RSTR				SADL_RAP_OPXY_EDMX_XSTRING	RAWSTRING	Bytefolge (BLOB)
<input type="checkbox"/> SALV_BS_BLOB	RSTR				SALV_BS_BLOB	RAWSTRING	Bytefolge (BLOB)
<input type="checkbox"/> SALV_WD_BLOB	RSTR				SALV_WD_BLOB	RAWSTRING	Bytefolge (BLOB)
<input type="checkbox"/> SAUNIT_SERIALIZED_DATA	RSTR	SAUNIT_SERIALIZED_DATA	RAWSTRING	variabel lange Folge von Bytes	SAUNIT_SERIALIZED_DATA	RAWSTRING	Bytefolge (BLOB)
<input type="checkbox"/> SBRN_KEY	RSTR	SBRN_KEY	RAWSTRING	variabel lange Folge von Bytes	SBRN_KEY	RAWSTRING	Bytefolge (BLOB)
<input type="checkbox"/> SCC_PARAMETERS	RSTR				SCC_PARAMETERS	RAWSTRING	Bytefolge (BLOB)
<input type="checkbox"/> SIGN_DOC	RSTR	SIGN_DOC	RAWSTRING	variabel lange Folge von Bytes	SIGN_DOC	RAWSTRING	Bytefolge (BLOB)
<input type="checkbox"/> SIM_DOM_TESTDATA	RSTR				SIM_DOM_TESTDATA	RAWSTRING	Bytefolge (BLOB)
<input type="checkbox"/> SMOJ_BIN	RSTR				SMOJ_BIN	RAWSTRING	Bytefolge (BLOB)

ZE5 (1) 800 tfsrvsapze5 INS

## Währungs- und Mengenfelder mit externen Referenzen

Normalerweise bezieht sich ein Feld vom Typ **CURR** oder **QUAN** auf ein Feld vom Typ **CUKY** (Währungsschlüssel) bzw. **UNIT** (Einheitenschlüssel), das zur **gleichen Tabelle** gehört. Zum Beispiel ist das Feld **VBAK-NETWR** (Nettowert des Auftrags in Belegwährung) dem Feld **VBAK-WAERK** (Währung des Vertriebsbelegs) zugeordnet. Es gibt jedoch auch Felder dieser Art, die sich auf ein Feld in einer anderen Tabelle beziehen. **BSEG-DMBTR** (Betrag in Hauswährung) ist ein Beispiel – es verweist auf das Feld **T001-WAERS** (Währungsschlüssel eines Buchungskreises).

Um (in der Ergebnisliste) solche Felder korrekt **aufzusummieren**, entweder in **ALV** oder mit einer Operation wie **Extras → Spalten einfügen → mit (Zwischen)Summen**, ist es notwendig, zuerst einen **Join** mit der Tabelle mit dem Währungs- bzw. Einheitenschlüssel durchzuführen. Im obigen Beispiel muss **T001** mit **BSEG** gejoint werden, bevor die Summe von **DMBTR** auf sinnvolle Weise durchgeführt werden kann – sonst würden wir Äpfel mit Birnen vergleichen.

Im Falle eines **Remote-Feldes** wird das entsprechende Referenz-Feld nur in einer **Remote-Tabelle** mit **derselben RFC-Destination** wie das Feld selbst gesucht. Wenn wir **BSEG[E04]-DMBTR** befriedigen möchten, müssen wir einen Join mit **T001[E04]** durchführen.

**ANMERKUNG:** [E04] repräsentiert eine **bestimmte RFC-Destination** – auch wenn **zwei RFC-Destinationen** auf **dasselbe Remote-System** verweisen, gelten sie trotzdem als **verschieden**.

Ein kurzes Beispiel wird veranschaulichen, was besprochen wurde.

Angenommen, es sind einige Einträge von BSEG von einem Remote-System selektiert worden. Die Detailanzeige eines Eintrags der Ergebnisliste wäre wie folgt:

Join von BSEG(A) und BSEG[E04](B)			
A - BSEG - CLUSTER - Belegsegment Buchhaltung			
B - BSEG[E04] - TRANSP - Belegsegment Buchhaltung			
B~MANDT	'800'		Mandant
B~BUKRS	'0001'		Buchungskreis
B~BELNR	'0090000001'		Belegnummer eines Buchhaltungsbeleges
B~GJAHR	'2017'		Geschäftsjahr
B~BUZEI	'002'		Nummer der Buchungszeile innerhalb des Buchhaltungsbelegs
...			
B~TXDAT_FROM	'00.00.0000'		Gültig-ab-Datum des Steuersatzes
B~QSSKZ	'		Quellensteuerkennzeichen
B~DMBTR [ ... ]		150,00	Betrag in Hauswährung
B~WRBTR [ . T001[E04]-WAERS ]		150,00	Betrag in Belegwährung
B~KZBTR [ ... ]		0,00	Ursprünglicher Kürzungsbetrag in Hauswährung
B~PSWBT (B~PSWSL)		150,00	Betrag für die Fortschreibung im Hauptbuch
B~PSWSL	'EUR'		Fortschreibewährung für Hauptbuchverkehrszahlen
B~TXBHW [ ... ]		0,00	Ursprünglicher Steuerbasisbetrag in Hauswährung
B~TXRFW [ ... ]		0,00	Ursprünglicher Steuerbasisbetrag in Belegwährung

Die externe Referenz von **B~DMBTR** verweist auf **T001[E04]-WAERS**.

Nun führen wir einen Join mit **T001** mit derselben RFC-Destination durch. Das Ergebnis, wie erwartet, ist wie folgt:

B~TXDAT_FROM	'00.00.0000'		Gültig-ab-Datum des Steuersatzes
B~QSSKZ	'		Quellensteuerkennzeichen
B~DMBTR (C~WAERS)		150,00	Betrag in Hauswährung
B~WRBTR [ . T001[E04]-WAERS ]		150,00	Betrag in Belegwährung
B~KZBTR [ ... ]		0,00	Ursprünglicher Kürzungsbetrag in Hauswährung
B~PSWBT (B~PSWSL)		150,00	Betrag für die Fortschreibung im Hauptbuch
B~PSWSL	'EUR'		Fortschreibewährung für Hauptbuchverkehrszahlen
B~TXBHW (C~WAERS)		0,00	Ursprünglicher Steuerbasisbetrag in Hauswährung
B~TXRFW [ ... ]		0,00	Ursprünglicher Steuerbasisbetrag in Belegwährung

Beachten Sie, dass jetzt die externe Referenz befriedigt wurde (**C-WAERS**).

## Formelfelder mit Remote-Charakter

Falls ein Formelfeld wie folgt deklariert wird:

**FF\_DEF REM\_FLD TYPE ROW-....**

wo **ROW-....** ein Remote-Feld ist, erbt es den **“Remote”**-Charakter des Original-Feldes. Ein mögliches **FF\_UNIT**-Referenz-Feld muss dieselbe RFC-Destination verwenden.

## V-Felder mit Remote-Charakter

V-Felder, die von einem Remote-Feld abgeleitet werden, erben ebenfalls den **“Remote”**-Charakter des Original-Feldes. Dies geschieht z.B. wenn auf einigen Remote-Feldern eine Summierungs-Operation wie *Extras* → *Spalten einfügen* → *mit (Zwischen)Summen* durchgeführt wird.

## Die RFC Destination(en) eines Scripts ändern

Es kann vorkommen, dass die in einem Script verwendeten RFC-Destinationen von der Systemadministration **geändert** oder **gelöscht** werden, sodass manche Operationen nicht mehr möglich sind. Wir werden diese Situation anhand eines Beispiels veranschaulichen.

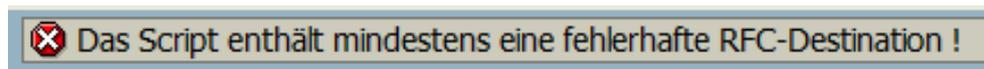
Angenommen, wir möchten Überschriften zu einem bestimmten Script hinzufügen. Um dies zu erreichen, rufen wir den Script-Katalog auf:

**1 SE16XXL Scripts selektiert**

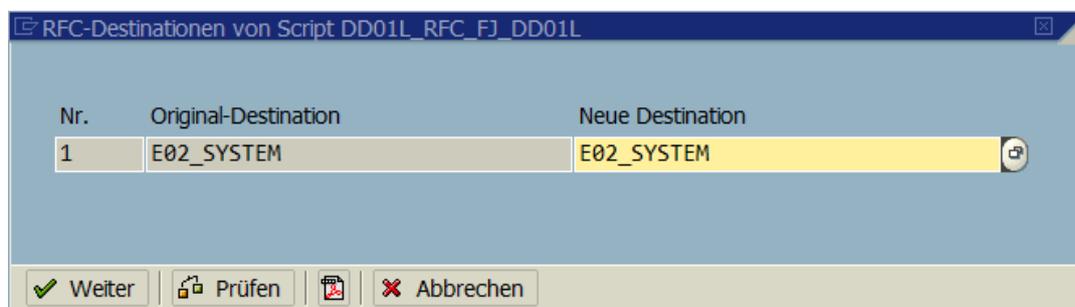


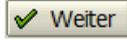
Script	Glob	Sp.S	GPar	Doku	Sprg	Üb&T	RFCs	Priv	Benutzer	Erste Tab.	Kurztext
<input type="checkbox"/> DD01L RFC FJ DD01L		•					•		TOPFLOW	DD01L	Fulljoin von DD01L mit DD01L

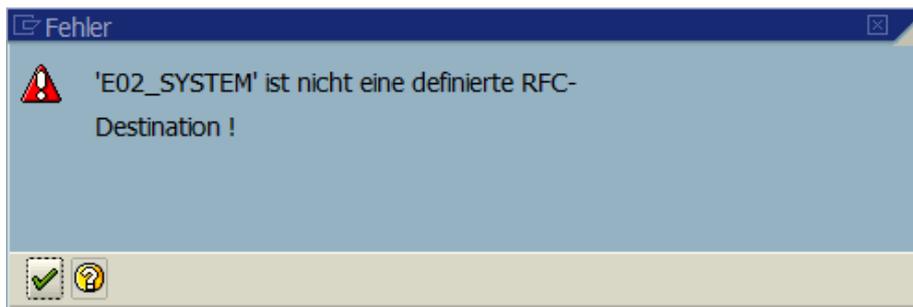
Als nächstes verwenden wir die  Schaltfläche auf der Anwendungsleiste, um die Überschriften zu definieren. Jedoch zu unserer Überraschung reagiert das Programm mit folgender Fehlermeldung:



Um herauszufinden, was vor sich geht, versuchen wir einfach, das Script auszuführen, indem wir die  Schaltfläche betätigen. Als Ergebnis wird das bekannte Dialogfenster angezeigt:



Wenn wir  betätigen, erscheint folgende Fehlermeldung:



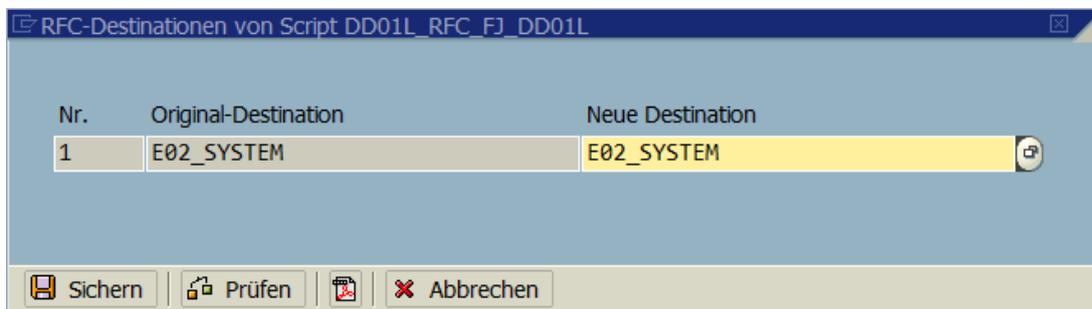
Anscheinend hat jemand diese RFC-Destination gelöscht.

Was kann getan werden? Wir könnten die verschwundene Destination durch eine vorhandene ersetzen, das Script ausführen und es erneut mit der neuen Destination speichern. Es gibt jedoch eine einfachere und schnellere Methode, um dies zu erreichen.

Wir verwenden folgende Menüfunktion:

***Script-Zusätze → RFC-Destination(en) ändern***

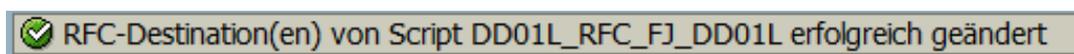
Das bekannte Dialogfenster erscheint, diesmal mit der Schaltfläche  **Sichern** :



Wir ändern die RFC-Destination:

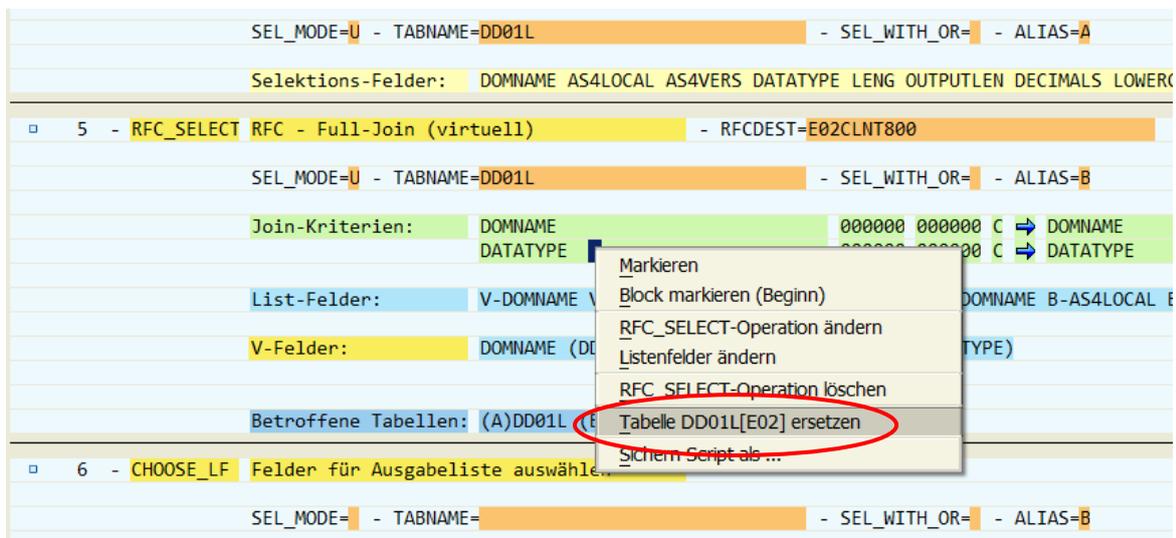


Nach dem Sichern wird folgende Meldung ausgegeben:



## Eine RFC-Destination mithilfe des Script-Editors ändern

Der Script-Editor bietet ebenfalls eine Möglichkeit, das RFC-Destination einer **RFC\_SELECT**-Operation zu ändern. Diese Funktionalität ist jedoch nicht sofort ersichtlich. Sie steht nicht zur Verfügung, wenn eine RFC\_SELECT-Operation geändert wird. Stattdessen ist sie über das **Kontextmenü** zu erreichen:



Wählen Sie auf dem Kontextmenü die Operation **“Tabelle DD01L[E02] ersetzen”**. Das Programm antwortet mit folgendem Dialogfenster:



Hier können Sie eine andere RFC-Destination mittels F4-Hilfe auswählen:



Indem Sie den Tabellennamen unverändert lassen, ändern Sie nur die RFC-Destination der RFC\_SELECT-Operation. Das Ergebnis ist:

4 - SELSCREEN Selektionsmaske

SEL\_MODE=U - TABNAME=DD01L - SEL\_WITH\_OR= - ALIAS=A

Selektions-Felder: DOMNAME AS4LOCAL AS4VERS DATATYPE LENG OUTPUTLEN DECIMALS LOWERC

5 - RFC\_SELECT RFC - Full-Join (virtuell) - RFCDEST=E04CLNT800

SEL\_MODE=U - TABNAME=DD01L - SEL\_WITH\_OR= - ALIAS=B

Join-Kriterien: DOMNAME 000000 000000 C → DOMNAME  
 DATATYPE 000000 000000 C → DATATYPE

List-Felder: V-DOMNAME V-DATATYPE A-DOMNAME A-DATATYPE B-DOMNAME B-AS4LOCAL B-

V-Felder: DOMNAME (DD01L-DOMNAME) DATATYPE (DD01L-DATATYPE)

Betroffene Tabellen: (A)DD01L (B)DD01L[...]

6 - CHOOSE\_LF Felder für Ausgabeliste auswählen

SEL\_MODE= - TABNAME= - SEL\_WITH\_OR= - ALIAS=B

### Strikte Join-Logik

Wie bereits in der Einleitung erwähnt, kann es vorkommen, dass die Eigenschaften eines bestimmten Tabellenfeldes von **System zu System unterschiedlich sind**. Ein detailliertes Beispiel wird zeigen, wie mit einer solchen Situation umzugehen ist. Um die Ergebnisliste klein zu halten, wählen wir nur Materialien aus, die mit ‘M’ beginnen.

Angenommen, wir sind auf einem der neuesten SAP-Systeme angemeldet, z.B. auf einem S/4-System. Wir selektieren Einträge von MARA (Materialstamm):

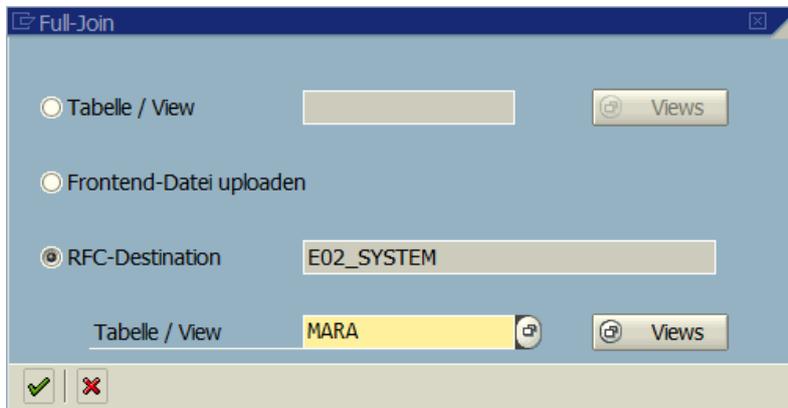
**SE16XXL - Tabelle MARA - 12 Einträge selektiert**

Tabelle MARA - Allgemeine Materialdaten

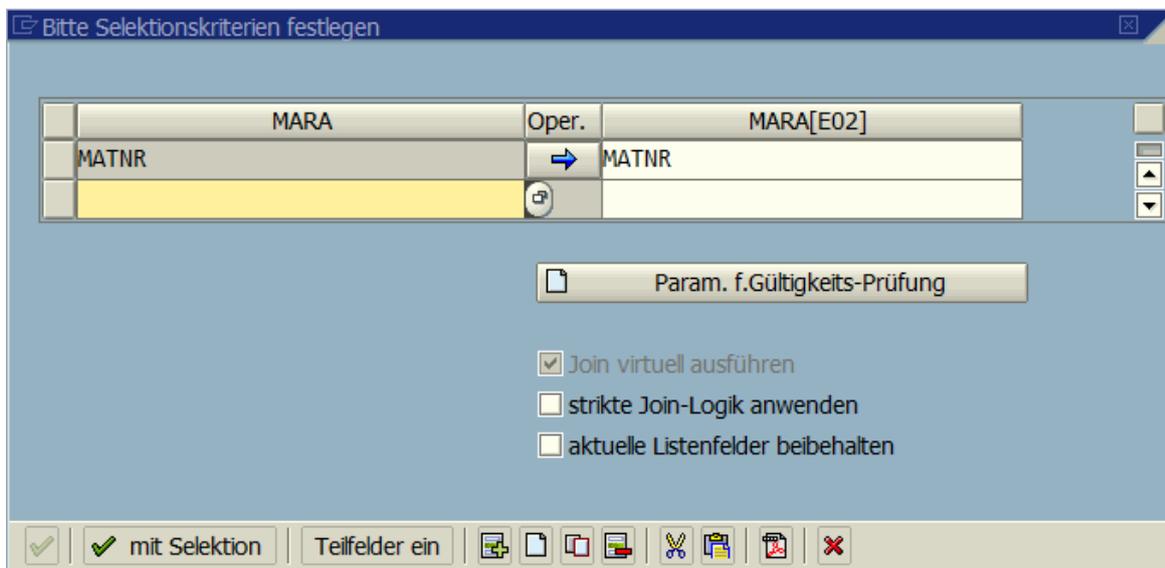
	MANDT	MATNR	VPSTA	MTART	MATKL	MEINS	GEWEI
<input type="checkbox"/>	800	M000-T00-KOMP-CHARGE-01	KVDLSQBX	HAWA	02	ST	KG
<input type="checkbox"/>	800	M000-T00-KOMP-CHARGE-02	KCVEDPLSQBX	HAWA	02	ST	KG
<input type="checkbox"/>	800	M000-TXX-MAX-123-456-789-01	KVDPAXCLSQBG	FERT	02	ST	G
<input type="checkbox"/>	800	M001-T01-MAX-LAENGE-123-456-789-01	KCVDPALSQBGX	FERT	01	ST	KG
<input type="checkbox"/>	800	M003-T03-MAX-CHARGE-123-456-789-02	KCVDPALX	FERT	01	ST	KG
<input type="checkbox"/>	800	M003-T03-MAX-LAENGE-123-456-789-01	KVDPALSQBGX	FERT	01	ST	KG
<input type="checkbox"/>	800	M005-T05-MAX-LAENGE-123-456-789-01	KVCDPALSQBGX	FERT	01	ST	KG
<input type="checkbox"/>	800	M020-T20-MAX-LAENGE-123-456-789-01	KCVDXPALSQBG	FERT	01	ST	KG
<input type="checkbox"/>	800	M020-T20-MAX-LAENGE-123-456-789-02	KCVDPALSQBGX	FERT	01	ST	KG
<input type="checkbox"/>	800	M03-001	KDALBVGZXQ	FERT	01	ST	KG
<input type="checkbox"/>	800	M03-001-123-123-123-123-123-123-1234	KDALBVGZX	FERT	01	ST	KG
<input type="checkbox"/>	800	M100-100	KVEDALBGZX	HALB	01	ST	KG

Auf diesem System ist die Materialnummer (MATNR) **40 Zeichen lang** statt der üblichen 18 auf älteren SAP-Versionen.

An dieser Stelle führen wir einen Full-Join mit Tabelle MARA auf einem Remote-System durch, wo MATNR **nur 18 Zeichen lang** ist:



Es erscheint nun das Dialogfenster für die Join-Kriterien:



Gefolgt vom Dialogfenster für das gemeinsame Feld:



Und dann die Selektionsmaske:

**SE16XXL - Tabelle MARA[E02] - Selektionsmaske**

Anzahl Einträge

Breite der Ausgabeliste: 1000  mit OR anstatt AND selektieren

MATNR: M\*

MTART:  bis

Zum Schluss kommt das Ergebnis des Full-Joins zum Vorschein:

**SE16XXL - Full-Join - 16 Ergebnis-Sätze**

Join von MARA(A) und MARA[E02](B)

V~MATNR	A~MANDT	A~MATNR	A~MTART	A~MATKL	A~MEINS	B~MANDT	B~MATNR	B~MTART	B~MATKL	B~MEINS
<input type="checkbox"/> M000-T00-KOMP-CHAR	800	M000-T00-KOMP-CHARGE-01	HAWA	02	ST					
<input type="checkbox"/> M000-T00-KOMP-CHAR	800	M000-T00-KOMP-CHARGE-02	HAWA	02	ST					
<input type="checkbox"/> M000-TXX-MAX-123-4	800	M000-TXX-MAX-123-456-789-01	FERT	02	ST					
<input type="checkbox"/> M001-T01-MAX-LAENG	800	M001-T01-MAX-LAENGE-123-456-789-01	FERT	01	ST					
<input type="checkbox"/> M003-T03-MAX-CHARG	800	M003-T03-MAX-CHARGE-123-456-789-02	FERT	01	ST					
<input type="checkbox"/> M003-T03-MAX-LAENG	800	M003-T03-MAX-LAENGE-123-456-789-01	FERT	01	ST					
<input type="checkbox"/> M005-T05-MAX-LAENG	800	M005-T05-MAX-LAENGE-123-456-789-01	FERT	01	ST					
<input type="checkbox"/> M01-001						800	M01-001	FERT	001	ST
<input type="checkbox"/> M020-T20-MAX-LAENG	800	M020-T20-MAX-LAENGE-123-456-789-01	FERT	01	ST					
<input type="checkbox"/> M020-T20-MAX-LAENG	800	M020-T20-MAX-LAENGE-123-456-789-02	FERT	01	ST					
<input type="checkbox"/> M03-001	800	M03-001	FERT	01	ST					
<input type="checkbox"/> M03-001-123-123-12	800	M03-001-123-123-123-123-123-123-1234	FERT	01	ST	800	M03-001-123-123-12	HALB	001	ST
<input type="checkbox"/> M100-100	800	M100-100	HALB	01	ST					
<input type="checkbox"/> M20-001						800	M20-001	FERT	001	ST
<input type="checkbox"/> M20-002						800	M20-002	FERT	001	ST
<input type="checkbox"/> MAT_24						800	MAT_24	FERT		L

Wie leicht erkennbar ist, **stimmt etwas nicht** – das gemeinsame Feld **V~MATNR** ist zu kurz, um die MATNRs unseres Systems vollständig aufzunehmen. Die langen Namen wurden **abgeschnitten**. Und, was schlimmer ist, es gibt ein Material (**M03-001-123-123-12**) auf dem Remote-System, das zufällig **genau dem Präfix** eines unserer Materialien **entspricht** und daher mit ihm **gejoint** wurde.

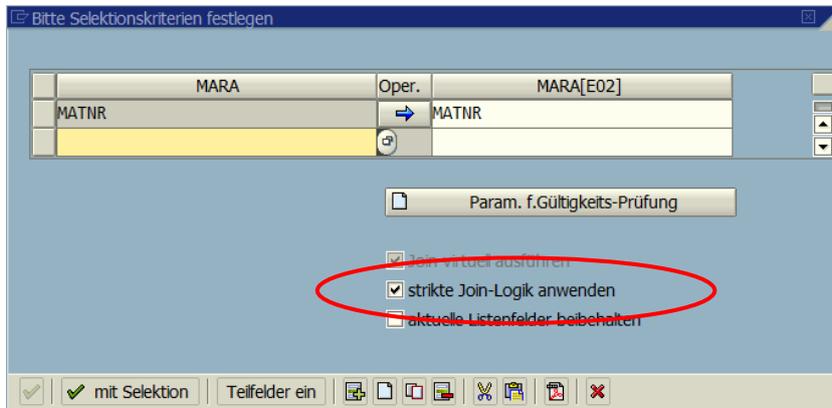
**ANMERKUNG:** Falls es sich beim Join nicht um einen Full-Join handelt, ist das gemeinsame Feld **V~MATNR** nicht vorhanden, das Problem mit dem Verbinden unterschiedlicher MATNRs bleibt jedoch bestehen.

Das ist offensichtlich keine zufriedenstellende Art, einen Join durchzuführen. Was kann getan werden, um die Sache in Ordnung zu bringen?

Für solche Probleme wurde eine neue Option implementiert: **“Strikte Join-Logik anwenden”**. Sie steht auf dem Dialogfenster für die Join-Kriterien zur Verfügung. Sie ist nicht auf Remote-Selektionen beschränkt, sondern kann auch in anderen Situationen angewendet werden.

Um zu zeigen, wie sie wirkt, werden wir unser Beispiel noch einmal durchgehen.

Nachdem die ersten Screenshots identisch sind, beginnen wir mit dem Dialogfenster zur Angabe der Join-Kriterien:



Dieses Mal aktivieren wir die Option **“Strikte Join-Logik anwenden”**. Auf dem nachfolgenden Dialogfenster stellen wir fest, dass das gemeinsame Feld nun **40 Zeichen lang** ist:



Wir überspringen den Screenshot mit der Selektionsmaske und gelangen zur Ergebnisliste:

SE16XXL - Full-Join - 17 Ergebnis-Sätze

Join von MARA(A) und MARA[E02](B)

V~MATNR	A~MANDT	A~MATNR	A~MTART	A~MATKL	A~MEINS	B~MANDT	B~MATNR	B~MTART	B~MATKL	B~MEINS
<input type="checkbox"/> M000-T00-KOMP-CHARGE-01	800	M000-T00-KOMP-CHARGE-01	HAWA	02	ST					
<input type="checkbox"/> M000-T00-KOMP-CHARGE-02	800	M000-T00-KOMP-CHARGE-02	HAWA	02	ST					
<input type="checkbox"/> M000-TXX-MAX-123-456-789-01	800	M000-TXX-MAX-123-456-789-01	FERT	02	ST					
<input type="checkbox"/> M001-T01-MAX-LAENGE-123-456-789-01	800	M001-T01-MAX-LAENGE-123-456-789-01	FERT	01	ST					
<input type="checkbox"/> M003-T03-MAX-CHARGE-123-456-789-02	800	M003-T03-MAX-CHARGE-123-456-789-02	FERT	01	ST					
<input type="checkbox"/> M003-T03-MAX-LAENGE-123-456-789-01	800	M003-T03-MAX-LAENGE-123-456-789-01	FERT	01	ST					
<input type="checkbox"/> M005-T05-MAX-LAENGE-123-456-789-01	800	M005-T05-MAX-LAENGE-123-456-789-01	FERT	01	ST					
<input type="checkbox"/> M01-001						800	M01-001	FERT	001	ST
<input type="checkbox"/> M020-T20-MAX-LAENGE-123-456-789-01	800	M020-T20-MAX-LAENGE-123-456-789-01	FERT	01	ST					
<input type="checkbox"/> M020-T20-MAX-LAENGE-123-456-789-02	800	M020-T20-MAX-LAENGE-123-456-789-02	FERT	01	ST					
<input type="checkbox"/> M03-001	800	M03-001	FERT	01	ST					
<input type="checkbox"/> M03-001-123-123-12						800	M03-001-123-123-12	HALB	001	ST
<input type="checkbox"/> M03-001-123-123-123-123-123-123-1234	800	M03-001-123-123-123-123-123-123-1234	FERT	01	ST					
<input type="checkbox"/> M100-100	800	M100-100	FERT	HALB	01					
<input type="checkbox"/> M20-001						800	M20-001	FERT	001	ST
<input type="checkbox"/> M20-002						800	M20-002	FERT	001	ST
<input type="checkbox"/> MAT_24						800	MAT_24	FERT		L

Wie zu sehen ist, kann das gemeinsame V-Feld jetzt alle MATNRs aufnehmen, ohne sie abzuschneiden, und die fehlerhafte Verknüpfung ist **verschwunden**.

Für mehr Informationen in Bezug auf die **“strikte Logik”** siehe [Strikte Join-Logik](#).

Copyright © 2020 SY-TABIX GmbH – Alle Rechte vorbehalten.