

EDV und Prüfen

Themen: Ausgewählte Problemlösungen zu Prüfsoftware

ACL-Techniken für fortgeschrittene Anwender

Im Zusammenhang mit der kontinuierlichen ACL-Auswertung betrieblicher Massendaten sind uns eine Reihe von Fragen zu dieser Software zugegangen. Sie bezogen sich auf unterschiedlichste Themenbereiche. Einige Kolleginnen und Kollegen möchten Erfahrungen aus der IDEA-Anwendung auf ACL übertragen und sehen sich (bei vergleichbarer Oberfläche), einem unterschiedlichen Softwarekonzept gegenüber. Andere sind an der Einbindung externer Programme (Excel, Mail) interessiert. Das Prüfen und Umstellen von Feldformaten (Unicode, divergierende Dezimalstellen, Umlaute etc.) bereitet ebenfalls häufig Probleme. Erfahrenere Anwender möchten Skripte in Unterprogramme strukturieren und benutzerspezifische Funktionen gestalten.

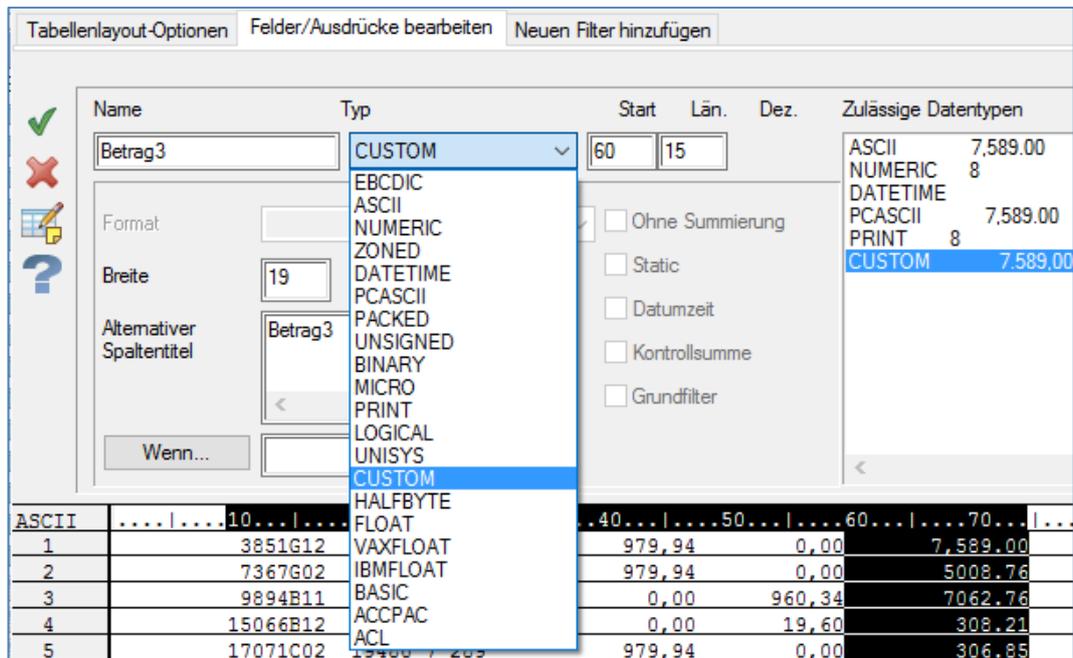
Nachfolgend finden Sie einige Hinweise, die zu eigenständigen Weiterentwicklungen anregen möchten:

- **Der Austausch von Zeichen (Dez.Stellen, Tausendertrennung, Sonderzeichen)**

Für diese Aufgabe stehen unterschiedliche Möglichkeiten zur Verfügung:

- **Generell** – Feldformat „Custom“ und Datei „Custom.dat“

Der Austausch von Zeichen begrenzt sich in der Regel auf einzelne Datenfelder. Für individuelle Konvertierungen sieht ACL innerhalb des Tabellenlayouts das Feldformat „Custom“ vor.



Dieses stützt sich auf eine Textdatei „Custom.dat“, die der Nutzer mit Hilfe eines Editors eigenständig erstellen muss.

■ **Ungültige Feldinhalte finden (Numerische Felder, Datumsfelder etc.)**

An dieser Stelle können ebenfalls verschiedene Wege verfolgt werden:

- **Funktion „Daten prüfen“**

Über die Menüfolge [Daten | Prüfen] werden die Inhalte ausgewählter Datenfelder mit dem zugehörigen Feldformat abgeglichen.

Am: 29.03.2016 13:36:45
 Befehl: [VERIFY FIELDS BELEGNR BLZ FAELLIGEIT KTONUMMER](#)
 Tabelle: Lieferantenrechnungen_3

32 30 30 35 2D 31 36 2D 30 39 20 30 30 3A 30 30 3A 30 30
Ungültige Felddaten in Datensatz 3 (Feld RECHNUNGSD)
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
Ungültige Felddaten in Datensatz 61 (Feld FAELLIGEIT)
35 37 31 34 2D 30 20
Ungültige Felddaten in Datensatz 249 (Feld TELEFONNUM)
35 37 31 34 2D 30 20
Ungültige Felddaten in Datensatz 250 (Feld TELEFONNUM)
35 37 31 34 2D 30 20
Ungültige Felddaten in Datensatz 251 (Feld TELEFONNUM)
35 37 31 34 2D 30 20
Ungültige Felddaten in Datensatz 252 (Feld TELEFONNUM)
35 37 31 34 2D 30 20
Ungültige Felddaten in Datensatz 253 (Feld TELEFONNUM)
35 37 31 34 2D 30 20
Ungültige Felddaten in Datensatz 254 (Feld TELEFONNUM)
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
Ungültige Felddaten in Datensatz 277 (Feld RECHNUNGSD)
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
Ungültige Felddaten in Datensatz 327 (Feld RECHNUNGSD)

10 Fehler bei der Gültigkeit der Daten festgestellt

Begrenzt auf ein vorgegebenes Fehlerlimit werden Gültigkeitsfehler (z.B. unrichtiger oder fehlender Datumseintrag in einem Datumsfeld) in einem Protokoll ausgewiesen. Das zugehörige Fehlerlimit kann innerhalb der ACL-Optionen eingestellt werden.

- **Angaben zur Feldstatistik**

Die Feldstatistik vermittelt (nur) für Datumsfelder ebenfalls eine entsprechende Übersicht:

Am: 29.03.2016 13:44:51
 Befehl: [STATISTICS ON RECHNUNGSD TO SCREEN NUMBER 5](#)
 Tabelle: Lieferantenrechnungen_3

RECHNUNGSD

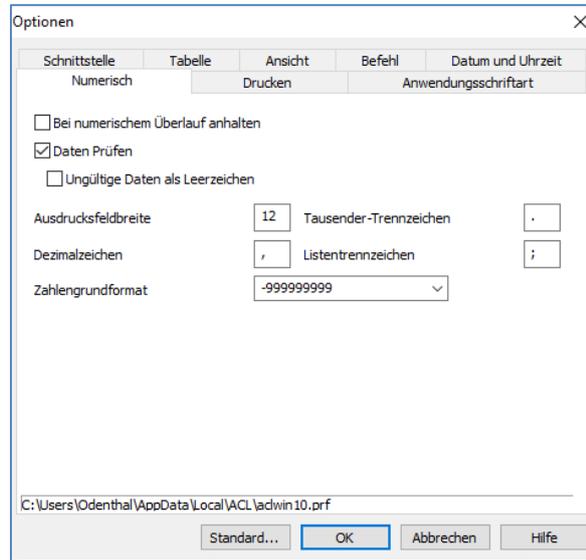
	Anzahl	Summe	Durchschnitt
Bereich	-	38.743	«Leere Datum»
Positiv	6.993	267.453.684	17.09.2004
Nullen	153	-	«Leere Datum»

Höchste	Niedrigste
27.01.2006	«Leere Datum»
27.12.2005	«Leere Datum»
21.12.2005	«Leere Datum»
21.12.2005	«Leere Datum»
21.12.2005	«Leere Datum»

Ungültige und leere Einträge werden als „Nullen“ oder „Leeres Datum“ ausgewiesen.

– Während der **Verarbeitung**

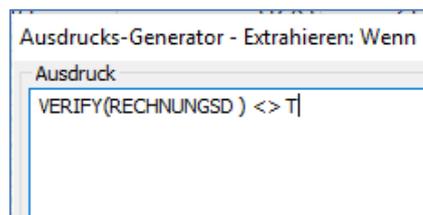
Eine Prüfung auf ungültige Feldinhalte kann in die laufende Verarbeitung eines Datenbestandes einbezogen werden. Die zugehörige Einstellung findet sich unter [Extras | Optionen | Numerisch | Daten Prüfen].



Deren Aktivierung führt dazu, dass jeder Analysevorgang mit einer zugehörigen Nachricht abgebrochen wird, sobald ein (beliebiger) ungültiger Feldinhalt innerhalb des Datensatzes festgestellt wird. Innerhalb von automatisierten Skript-Analysen ist ein solches Verhalten oft unerwünscht. In einem Skript kann diese Funktion mit dem Kommando „SET VERIFY OFF“ abgeschaltet werden.

– **Selektion** ungültiger Werte (Inhalte)

Die gezielte Selektion ungültiger Werte erfordert den Einsatz einer geeigneten Selektionsfunktion. Hierfür steht „Verify()“ zur Verfügung. Getestet werden Inhalte von numerischen Feldern sowie Zeichen- und Datumsfeldern, wie das nachfolgende Beispiel zeigt:



Datensätze mit fehlerhaften Einträgen werden in einer gesonderten Tabelle ausgegeben. Eine Verwendung dieser Funktion in Rechengleichungen und ACL-Skripten ist ebenfalls möglich.

– **Selektion** ungültiger *Feldformate*

Vielfach erfordern automatisierte Analysevorgänge vorgegebene Feldformate (Numerische Felder, Zeichenfelder, Datumsfelder). Ob diese vorliegen, kann mit der Funktion „FTYPE()“ getestet werden. Da FTYPE() auch Variable berücksichtigt und für weitere ACL-Elemente (Tabellen, Skripte) eingesetzt werden kann, eröffnen sich umfassende Prüfmöglichkeiten, die insbesondere für das Abfangen von Fehlern in der Skriptprogrammierung sinnvoll eingesetzt werden können.

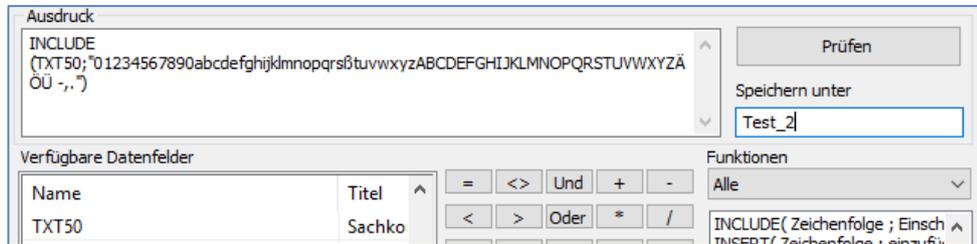
Ein kombinierter Einsatz von VERIFY und FTYPE bietet die beste Gewähr dafür, ungültige Feldformate und -inhalte bei automatisierten Verarbeitungsvorgängen zu erkennen.

■ **Feldinhalte bereinigen und normalisieren**

An dieser Stelle können ebenfalls verschiedene Wege verfolgt werden:

- **Erlaubte Inhalte** – Funktionen Include() und Exclude()

Aus Textfeldern können unerwünschte Sonderzeichen eliminiert werden.



Sachkontenlangtext	Test_2
Gebäude - Büro- & Wohngebäude	Gebude - Bro- Wohngebude
Gebäude - Büro- & Wohngebäude - Parallel 1	Gebude - Bro-Wohngebude - Parallel 1
Büro- & Wohngebäude - Geldwertkorrektur	Bro-Wohngebude - Geldwertkorrektur
Büro- & Wohngebäude - Geldwertkorrektur Parallel 1	Bro-Wohngebude - Geldwertkorrektur Parallel 1
Gebäude - Verkaufsbüros und Lager	Gebude - Verkaufsbros und Lager
Gebäude - Verkaufsbüros und Lager - Parallel 1	Gebude - Verkaufsbros und Lager - Parallel 1

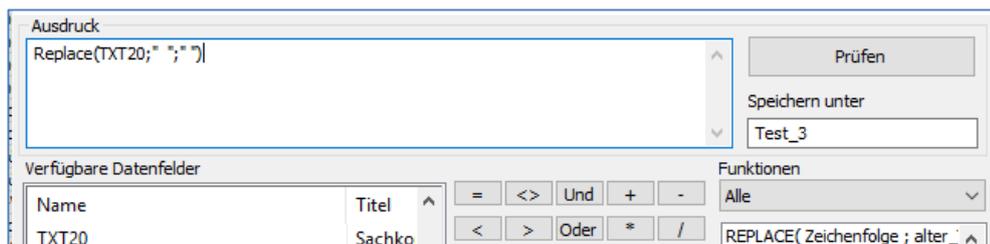
Falls unterschiedliche Schreibweisen bei Umlauten berücksichtigt werden sollen, sind diese *vorab* über die Funktion „Replace()“ zu harmonisieren:

(z.B. Replace(TXT50;“ä“;ae“)

Gleiches gilt für Groß- und Kleinschreibung mittels „UPPER()“ oder „LOWER()“. Hier sind beliebige Kombinationen möglich.

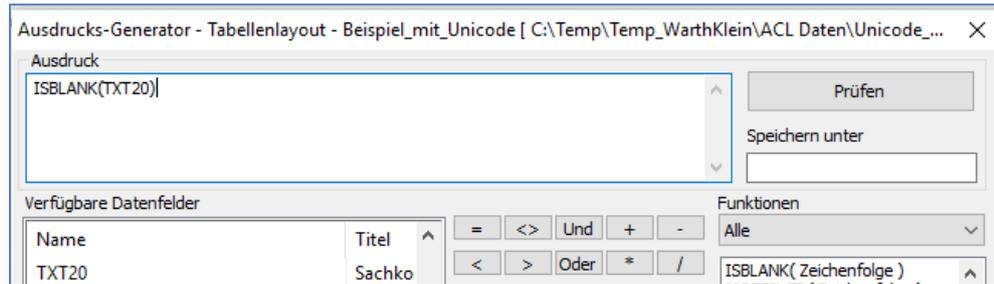
- **Unerlaubte doppelte Leerzeichen** – Funktionen Replace()

Doppelte Leerzeichen können in einfache Leerzeichen umgestellt werden:



Filter:	Sachkontenkurztext	Test_3
3	Land- Geldwert.-	Land- Geldwert.-

Der Test zu einem vollkommen „leeren“ Zeichenfeld (keine Sonderzeichen, Steuerzeichen, Nullzeichen) kann mit der Funktion „ISBLANK()“, z.B. bei Selektionsvorgängen unterstützt werden:

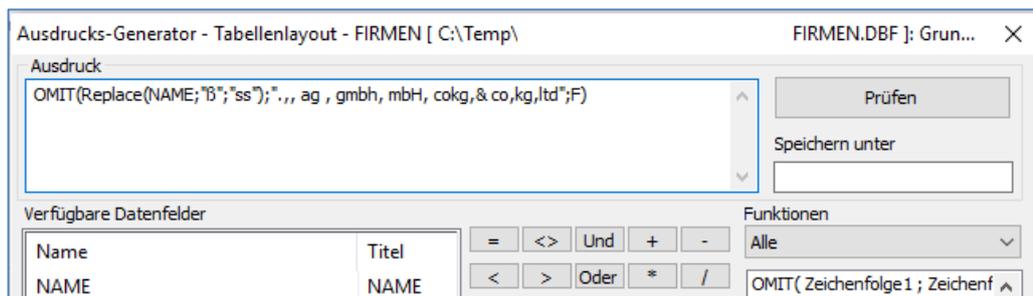


Filter:		
	Sachkontenkurztext	Test4
433	Sonst. VG/Sonst. For	F
434		T
435	Son.Ford-Gen.Vorzahl	F

Der Rückgabewert lautet F (False) oder T (True). Eine Verwendung in Rechengleichungen (z.B. bedingtes Rechnen) ist ebenfalls möglich.

– **Unerwünschte Textbestandteile – Funktionen OMIT()**

Zu einer Entfernung ganzer Textsequenzen, z.B. von Gesellschaftsbezeichnungen, oder Füllwörtern, eignet sich die Funktion „OMIT()“. Da die zu eliminierenden Textbestandteile (z.B. AG, KG etc.) aus allen Teilen des Suchfeldes entfernt werden, ist eine vorsichtige Konfektionierung geboten!



Die Bereinigung erfolgt hierarchisch von links nach rechts. Bereits bereinigte Zeichen können somit in weiter rechtsstehenden Ausdrücken nicht mehr berücksichtigt werden. Detaillierte, längere Angaben sollten daher vor Abkürzungen platziert werden. Die zusätzliche Berücksichtigung von Leerzeichen im Sinne von „AG“ verbessert das Ergebnis.

Filter:		
	NAME	test5
1	A. A. A. A. A. AAAQUAManfroid GmbH	A A A A A AAAQUAManfroid
2	A. A. A. A. A. AABA Abflußmeister-, Abflußreinigungs- u. Wz	A A A A A AABA Abflussmeister-, Abflussreinu
3	A. A. A. AABA Abflußreinigungs- und Wartungs GmbH	A A A AABA Abflussreinigungs- und Wartungs
4	A. A. AABALO Abflussreinigungs- und Wartungs GmbH	A A AABALO Abflussreinigungs- und Wartungs
5	A A - WIS Sicherungstechnik GmbH & Co KG	A A - WIS Sicherungstechnik
6	A. AADEBO Abflußdienst GmbH	A AADEBO Abflussdienst

Eine alternative Eingabe der Suchbegriffe in Groß- oder Kleinschreibung ist ausreichend, wenn der Test auf genaue Schreibweise (letzter Parameter „F“) deaktiviert wird. In der Regel ist eine Kombination mit weiteren Normalisierungsstrategien sinnvoll.

■ **Automatisierte Suche - Listen mit Suchbegriffen für Textfelder**

Die Kontrolle kritischer Inhalte von Textfeldern (z.B. Buchungstexten) wird häufig als Problem bezeichnet, wenn viele divergierende Suchbegriffe berücksichtigt werden sollen. Typische Beispiele sind das Journal Entry Testing (JET) oder die Ermittlung von Positionen, die in Embargo-Listen aufgeführt sind.

Die eleganteste Lösung besteht in einer getrennten Führung der Suchbegriffe (Suchlisten) und deren flexibler Abbildung auf unterschiedliche Datenbestände (z.B. Journaldateien). Hierzu bedarf es einer ACL-Funktionserweiterung auf der Basis von ACL-Skript, welche wir nachfolgend vorstellen:

– **Führung und Pflege der Suchlisten**

Diese Aufgabe kann durch den Anwender in Excel erfolgen:

	A
1	Suchbegriffe
2	test
3	fehler
4	storno
5	excotic
6	ipad
7	Sanitär
8	Consulting
9	bank

– **Übernahme der Suchliste nach ACL**

Die von dem Anwender erstellte und gepflegte Suchliste wird manuell nach ACL importiert und steht anschließend als ACL-Tabelle für beliebigen Einsatz bereit:

	Suchbegriffe
1	test
2	fehler
3	storno
4	excotic
5	ipad
6	Sanitär
7	Consulting
8	bank
<< Dateiende >>	

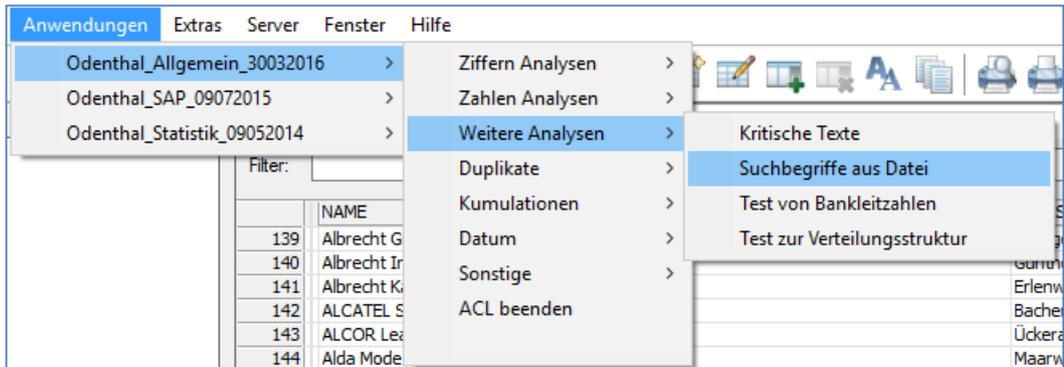
– **Import der Datei, in welcher gesucht werden soll**

Der Anwender importiert seine Dateien (Journaldateien, Rechnungen, Stammdaten etc.) in welchen er suchen möchte:

	NAME	STRASSE
139	Albrecht GmbH Bau- u. Projektentwicklung	Röntgenstr. 1
140	Albrecht Intern. Speditions GmbH	Gunther-Plüschow-Str. 14
141	Albrecht Kamera-Service GmbH	Erlenweg 13
142	ALCATEL SEL AG Bürokommunikation	Bachemer Landstr. 47
143	ALCOR Leasing GmbH	Ückerather Str. 5B
144	Alda Mode-Vertrieb GmbH	Maarweg 231

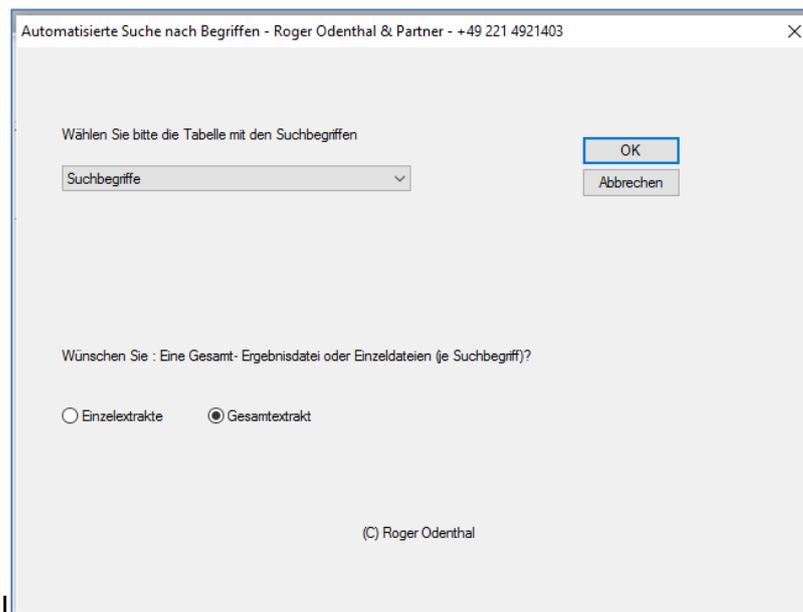
– **Aufruf unserer Funktionserweiterung**

Das hierfür vorgesehene Skript kann menübasiert aufgerufen werden, wenn die von uns bereitgestellten Menüdateien verwendet werden:

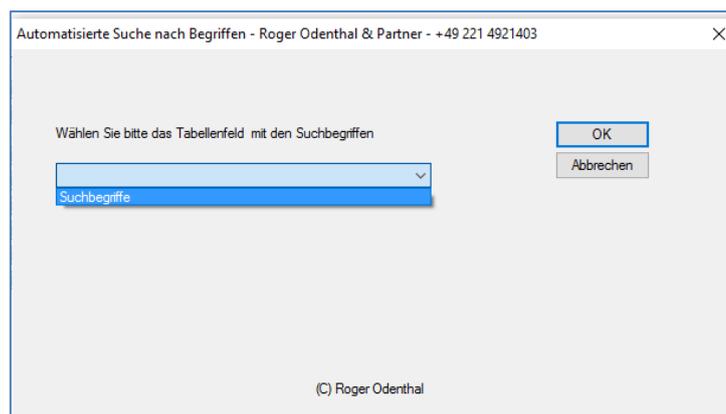


– **Auswahl der Tabellen (Suchbegriffe und zu suchende Daten)**

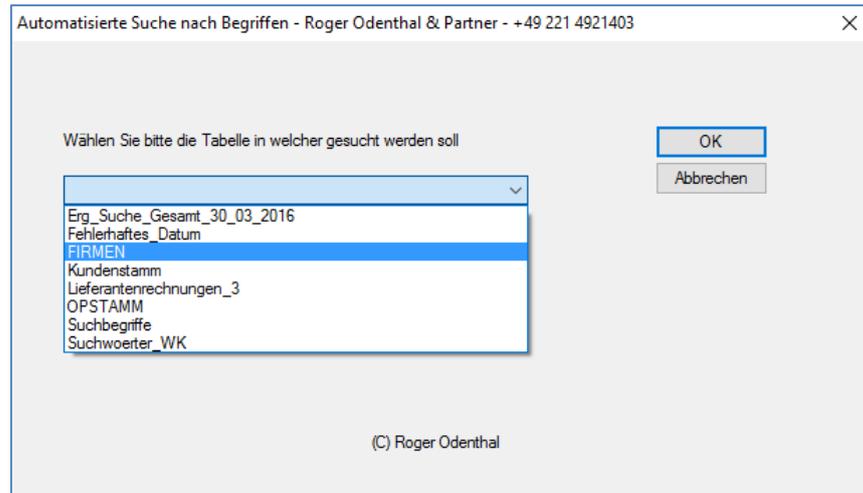
Die Suchergebnisse können wahlweise getrennt nach Begriffen in gesonderte Einzeldateien oder insgesamt in eine Tabelle mit allen Auffälligkeiten ausgegeben werden:



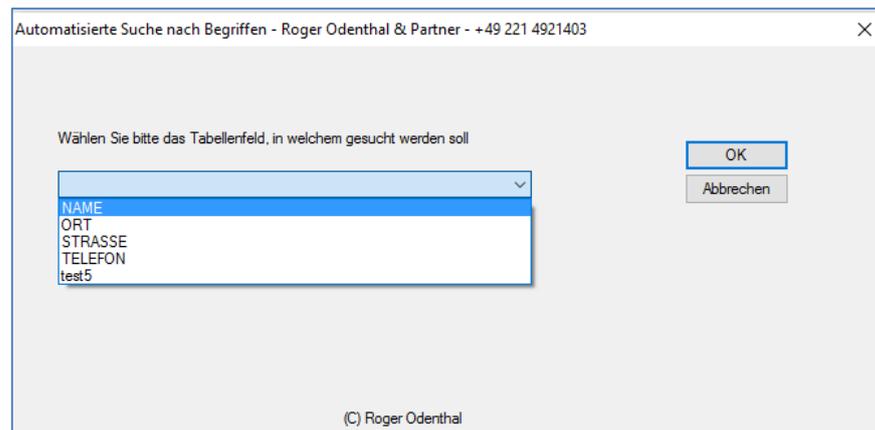
In einer Tabelle mit Suchbegriffen können ebenfalls verschiedene Felder mit unterschiedlichen Suchbegriffarten geführt und ausgewählt werden:



Dieses gilt ebenfalls für die Tabellen, in welchen gesucht werden soll:

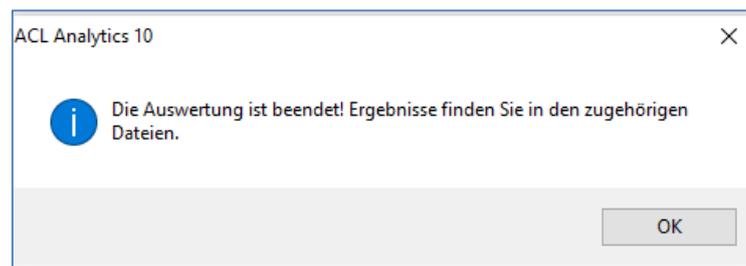


Sowie die zugehörigen Suchfelder:

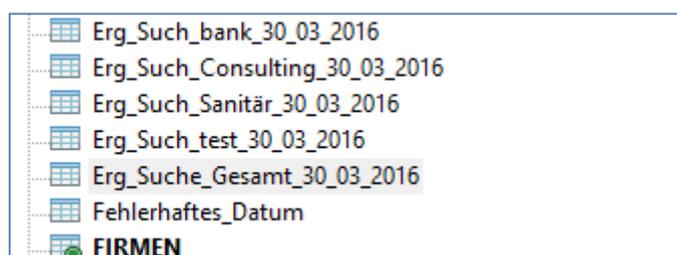


– **Abschluss der Verarbeitung und Ergebniskontrolle**

Nach abgeschlossener Analyse erhält der Anwender einen Hinweis:



Die einzelnen Ergebnisse findet er alternativ in der erzeugten Gesamtdatei oder den Einzeldateien mit den jeweiligen Treffern.



Die einzelnen Ergebnisse enthalten den Suchbegriff innerhalb der Bezeichnung der Ergebnistabelle. Es werden ausschließlich solche Ergebnistabellen erzeugt, zu welchen die Suche Treffer ermittelt hat:

NAME
1 A. S. I. Consulting GmbH & Co. KG
2 AC & S Analysis Consulting & Software GmbH
3 ADCO Advanced Consulting GmbH
4 Baucon Baubetreuung und Consulting GmbH
5 BBK Immobilienmanagement & Consulting GmbH
6 Bierther Consulting GmbH

Unsere aufgeführte Funktionserweiterung ermöglicht ACL-Anwendern eine flexible und automatisierte Suche mit einer Vielzahl unterschiedlichster Begriffe in großen Dateien.

Im Rahmen des kollegialen Gedankenaustausches stellen wir diese Lösung interessierten ACL-Nutzern zur Verfügung. Wir bitten in diesem Fall um eine entsprechende Nachricht.

■ Feldbezeichnungen und Feldformate vereinheitlichen

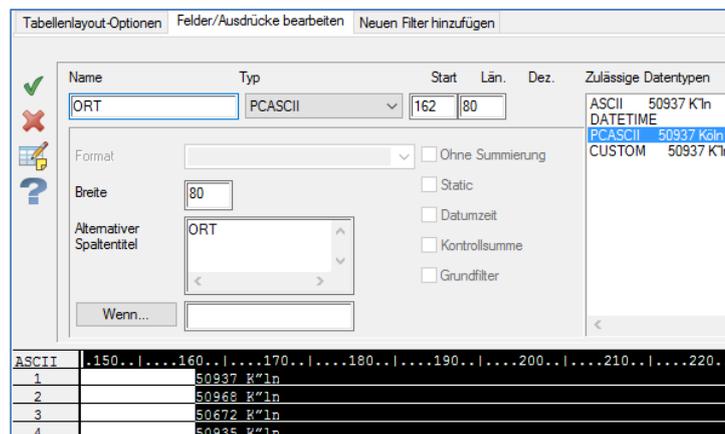
Automatisierte ACL-Auswertungsskripte beziehen sich regelmäßig auf festgelegte oder identifizierbare Felder und vorgegebene Feldformate. Probleme ergeben sich dann, wenn Feldbezeichnungen, Feldformate, Feldstrukturen (z.B. Feldlängen, Dezimalstellen, Datumsausprägungen etc.) oder sogar Feldinhalte (z.B. mit oder ohne führende Nullen, Leerstellen, Ausrichtung) nicht übereinstimmen. Zur Problemlösung stehen unterschiedliche Strategien bereit:

– Feldauswahl über Dialogmenü

Skripte können so entwickelt werden, dass sich feldbezogene Auswertungen jeweils auf ein *menüunterstützt ausgewähltes Tabellenfeld* beziehen. Die individuelle Feldbezeichnung wird in diesem Fall in einer Variable gespeichert. Die Variable, wird für die weitere Skriptprogrammierung verwendet und neutralisiert so unterschiedliche Bezeichnungen. Das Problem eines fehlerhaften Feldformates stellt sich insoweit nicht, als die Menüauswahl a priori auf Felder mit geeignetem Format eingegrenzt werden kann. Tabellenfelder mit abweichendem Format würden innerhalb des Auswahlmenüs nicht angezeigt und müssten gegebenenfalls vorab manuell innerhalb des Tabellenlayouts angepasst werden.

– Skriptbasierte Änderung des Original-Tabellenlayouts

Feldbezeichnungen und Feldformate können innerhalb des Tabellenlayouts manuell geändert werden.



Die Änderungen betreffen Bezeichnungen, Formate und Strukturen. Sie sind

ebenfalls mit den Befehlsfolgen [Delete Field / Define Field] skriptfähig. Die Feldänderung beinhaltet somit immer ein Löschen und neues Einfügen eines Feldes in ein bestehendes Tabellenlayout. Hierzu wären allerdings die zugehörigen Anfangsadressen und Feldlängen zu berücksichtigen, die sich nicht ohne weiteres automatisiert ermitteln lassen. Zwar haben wir auch hierfür komplexere Routinen entwickelt, die jedoch an dieser Stelle nicht behandelt werden. Somit ist die Änderung eines originalen Tabellenlayouts über ACL-Skript zwar möglich, aber schwierig.

– **Anpassung über Rechenfelder**

Mittels des Skriptbefehls [Define Field Computed] können Rechenfelder mit beliebiger Bezeichnung, wählbarem Format und vorgegebener Struktur in ein Tabellenlayout eingefügt werden. Da hierbei auch Formatänderungen berücksichtigt und Inhalte vorhandener Felder angeglichen werden können, handelt es sich um die einfachste Option auf dem Weg zur erfolgreichen Anpassung. Er sollte bevorzugt beschritten werden. Einige hierbei zu Verfügung stehenden Möglichkeiten werden nachfolgend gesondert dargestellt.

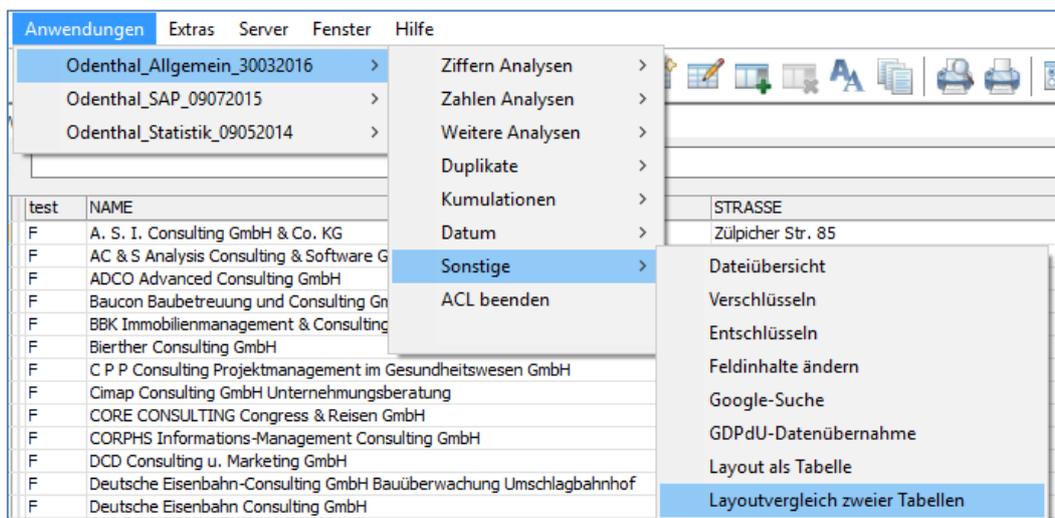
– **Tipps!**

Skriptbasierte Anpassungen von Tabellenfeldern können einfacher erfolgen, wenn diese während des *ACL-Datenimports möglichst als Textfelder* deklariert wurden.

ACL-Arbeitsbereiche eröffnen mit vorkonfektionierten Rechenfeldern eine besonders einfache und effiziente Option, um in einem Schritt alle erforderlichen Harmonisierungen vorzunehmen.

Wenn, bei gestückelten Datenexporten, Tabellen über die Funktion [Mischen] wieder zusammengefügt werden müssen, sollten in einem ersten Schritt nach dem Datenimport gleiche Felder mittels Rechnen erzeugt und (nur) diese in einem zweiten Schritt mit vorgegebener Feldreihenfolge in eine neue Tabelle extrahiert werden. Anschließend werden die Tabellen für das nachfolgende Mischen über das jeweils identische Tabellenlayout verfügen.

Ergeben sich trotz der aufgeführten Handhabung Probleme infolge eines nicht absolut identischen Tabellenlayouts, so kann für die Fehleranalyse auf unsere Programmerweiterung zum Vergleich zweier Tabellenlayouts zurückgegriffen werden:



Ansonsten nur schwer ermittelbare Unterschiede werden hier exakt angezeigt.

■ **Skriptbasierte Einfügung von Rechenfeldern – Bedingtes Rechnen**

Die nachfolgenden Hinweise hängen eng mit dem vorstehend aufgeführten Problembereich zusammen. Bedingtes Rechnen ist eine unabdingbare Voraussetzung für viele kaufmännische Verprobungen, aber auch für Formatumwandlungen und logische Testverfahren. Hierzu finden sich nachfolgend einige Beispiele:

– **Umwandlung eines Feldformates**

In diesem Fall ist die Bedingung Teil des ACL-Befehls (und nicht der Rechenformel)! Zunächst wird der vorhandene Feldtyp abgefragt und anschließend das Rechenfeld (häufig als einfache Formel) eingefügt.

Hierzu das nachfolgende Beispiel (Umwandlung eines Zeichenfeldes in Datum):

```
IF FTYPE("RECHNUNGSD") = "D" Define Field Rechnungsdatum COMPUTED RECHNUNGSD
IF FTYPE("RECHNUNGSD") = "C" Define Field Rechnungsdatum COMPUTED CTOD(RECHNUNGSD;"JJJJ-MM-TT")
IF FTYPE("RECHNUNGSD") = "N" Define Field Rechnungsdatum COMPUTED CTOD(STRING(RECHNUNGSD;10);"JJJJ-MM-TT")
```

Das zugehörige Ergebnis sieht wie folgt aus:

RECHNUNGSD	Rechnungsdatum
2003-10-31 00:00:00	31.10.2003
2005-04-16 00:00:00	16.04.2005
2005-16-09 00:00:00	
2005-04-27 00:00:00	27.04.2005
2005-04-16 00:00:00	16.04.2005

Das Kommando der zweiten Befehlszeile wird durchgeführt.

– **Ausgabe einer Anzeige (Richtig / Falsch) zu einer Verprobung**

Im Unterschied zur vorherigen Aufgabe werden hier verschiedene *Bedingungen innerhalb eines Rechenfeldes* abgefragt und berücksichtigt.

Die Syntax für bedingtes Rechnen ist dabei genau festgelegt:

1. Zeile Anweisung, Einleitung und Feldbezeichnung
2. Zeile Leer oder Ansichts-Formatangaben
3. Zeile Bedingung und Rechenformel
- n. Zeile und Folgezeilen weitere Bedingungen und Rechenformeln
- letzte Zeile Ausgangsformel (ohne Bedingung)

Beispiel (Suche eines Textfragmentes in einem Textfeld):

```
3
4 Define Field Kennzeichen COMPUTED
5 AS "Kennzeichen für Treffer"
6 "Consulting" If Find("Cons";Name)
7 "Service" If Find("Serv";Name)
8 "Test" If FIND("Test";Name)
9 "Kein Treffer"
10
```

Das zugehörige Ergebnis ist nachfolgend dargestellt:

Filter: <input type="text"/>		
	NAME	Kennzeichen für Treffer
13	A. K. Automobile GmbH	Kein Treffer
14	A K B PRIVAT- UND HANDELSBANK AG	Kein Treffer
15	A. S. Auto-Service GmbH	Service
16	A. S. I. Consulting GmbH & Co. KG	Consulting
17	A. S. Lochbleche GmbH	Kein Treffer
18	A-Z Medienservice GmbH	Service
19	AA Abfluß-AS GmbH Arbeiten an allen Abflußanlagen	Kein Treffer
20	AAA - ABC Vertriebs GmbH, Bremen	Kein Treffer

– **Tipps:**

Da ACL auf jeder Ebene einer Rechenoperation mit Blick auf die erforderliche Rechengenauigkeit rundet, kann es bei kombinierten Rechenoperationen (Division / Multiplikation), insbesondere, wenn Werte zwischendurch sehr klein werden, zu problematischen Ergebnisfehlern (Mitführung einer fehlerhaften Rundung) kommen. Daher ist für Rechenoperationen die Parametrisierung der Rechengenauigkeit erforderlich. Hierfür stehen verschiedene Optionen zur Verfügung:

SET MATH Anweisung zu Beginn eines Skriptes

DEC() Funktion innerhalb einzelner Rechenoperationen

Multiplikation eines Operanden mit 1,000 (gewünschte Anzahl von Dezimalstellen) innerhalb einer Rechengleichung.

Alle Optionen beeinflussen die Rechengenauigkeit in vergleichbarer Weise. Bei kombinierten Rechenoperationen sollte ferner die *Multiplikation immer vor der Division* erfolgen.

Bei der Erfassung von Skriptcode für Rechenfelder innerhalb des Skripteditors sind ferner einige Funktionstasten für Menüunterstützungen hilfreich:

F2 Einfügen von Feldern einer gerade geöffneten Tabelle

F8 Einfügen von Datumsangaben aus Kalender

F4 Einfügen von Tabellenbezeichnungen, Skriptnamen etc.

Die Gefahr unbeabsichtigter Syntaxfehler reduziert sich so erheblich.

■ **Datensatzübergreifende Auswertungen**

Die Auswertungslogik von ACL-Kommandos orientiert sich in der Regel an Tabellen und deren Felder. Datensatzübergreifende Auswertungen erfordern demgegenüber den Zugriff auf einzelne Tabellenzeilen sowie den Einsatz von Speichervariablen. In diesem Zusammenhang muss auf das *GROUP Kommando* in ACL-Skript hingewiesen werden, welches den Zugang zu einzelnen Datensätzen eröffnet. Diese können mehrfach, strukturiert oder bedingt abgearbeitet werden. Hiermit sind unter anderem nachfolgende Optionen verbunden:

– **Steigerung der Arbeitsgeschwindigkeit**

Alle Auswertungen, die innerhalb eines GROUP-Kommandos stehen werden je Datensatz abgearbeitet.

```
3 GROUP
4
5 EXTRACT RECORD IF RECHNUNGSB >= 10000 TO "Beträge Hoch"
6 EXTRACT RECORD IF RECHNUNGSB > 0 TO "Beträge Debitorisch"
7 EXTRACT RECORD IF Find("Köln";ORT) TO "Beträge Köln"
8 SUMMARIZE ON LIEFERNR TO "LieferantenSumme"
9
10 END
```

Nach Erreichen des letzten Datensatzes einer Tabelle sind sämtliche Aufgaben erledigt.

– **Differenzierte Verarbeitung unterschiedlicher Datensätze**

Die unterschiedliche Verarbeitung einzelner Datensätze kann innerhalb eines GROUP-Kommandos an Bedingungen gebunden werden. Schleifenverarbeitung ist ebenfalls möglich.

– **Gruppenorientierte Verarbeitung**

Arbeitsergebnisse (Kumulationen, Gruppensummen, etc.) können innerhalb der Gruppenverarbeitung erzeugt, in Speichervariable abgelegt oder als Ergebnis (Tabellenfelder) ausgegeben werden.

– **Anlegen neuer und leerer Tabellen mit beliebigen Tabellenlayouts**

Die Gruppenverarbeitung ermöglicht das Erstellen neuer Tabellen mit beliebigen Layouts. Hiermit können Analyseergebnisse in unterschiedlichster Form z.B. hierarchisch, kommentiert oder formatiert innerhalb von ACL-Tabellen ausgegeben werden.

Mit der Gruppenverarbeitung sind weitere Möglichkeiten verbunden, die hier über die aufgeführten Anregungen hinaus nicht dargestellt werden können. Ergänzende Informationen können der ACL-Hilfe entnommen werden.

■ **Der Einsatz von Unterprogrammen / Prozeduren**

Erfahrene Anwender objektorientierter Programmiersprachen werden bei ACL die Kapselung von Programmteilen in Substrukturen, Funktionen oder Prozeduren sowie den Einsatz von Arrays vermissen. Die kommandoorientierte ACL-Skriptsprache bietet keine vergleichbaren Möglichkeiten. Da der datensatzorientierte GROUP-Befehl ebenfalls auf ausgewählte Kommandos beschränkt ist und seinerseits keinen Aufruf von Subprozeduren ermöglicht, ist auch dieser Weg versperrt. Dieses ist der Nachteil, welcher der schnellen, hoch aggregierten und ergebnisorientierten Kommandobasis entgegensteht.

Einen kleinen Ausweg eröffnet der Aufruf (Do Script) von Skripten aus Skripten. Mit Erfahrung, Phantasie, Schleifen und Bedingungen lassen sich prozedurähnliche Strukturen gestalten, die das übersichtliche Programmieren erleichtern.

```
105 Comment Unterprogramm Verarbeitung zur Extrakterzeugung je nach Auswertungsalternative
106 Do Script Suchliste_2x if VARn_05 = 2
107
108 Comment unterprogramm Verarbeitung zur separaten Extrakterzeugung (Option 1) einleiten
109 Do Script Suchliste_3x while VARn_05 = 1 AND VARn_11 <= VARn_07
110
111
112 Comment Abschlusshinweis
113 Pause "Die Auswertung ist beendet! Ergebnisse finden Sie in den zugehörigen Dateien."
```

Weitere Funktionen zur Unterstützung der Prüfungsarbeit stellen wir Ihnen gerne in einem nächsten Newsletter vor. Zusätzliche Informationen zu den dargestellten Prüfhilfen finden Sie auf unserer Internet-Seite

www.roger-odenthal.de

Alle aufgeführten Programme erhalten Sie bei Übermittlung einer kurzen Nachricht kostenfrei über unsere Kontakt-Seite. Für Anregungen, Rückfragen und weitere Hinweise erreichen Sie uns gerne unter den angegebenen Kontaktdaten. Wir freuen uns, von Ihnen zu hören.

Mit freundlichen Grüßen


Roger Odenthal