

Einleitung

Über komplexe Textübernahmeregeln hinaus kann der Funktionsumfang des ELO-Connectors noch weiter vergrößert werden.

Durch die Verwendung von Skripten können hier mannigfaltige, zusätzliche Aktionen durchgeführt werden, wie z.B.

- Änderung der Formate erfasster Daten.
- weiterführende Plausibilitätsprüfung ausgelesener Werte.
- Fehlerbehandlung bei der Erkennung.

Natürlich gibt es noch weitere denkbare Szenarien, in denen der Einsatz von Skripten sinnvoll sein kann; an dieser Stelle wollen wir aber besonders Einsteigern anhand einiger einfacher Beispiele zeigen, wie mit geringem Aufwand simple Skripte zur Connector-Erweiterung genutzt werden können.

Vorbereitung

Hinweis:

Anwender die noch nie Kontakt mit Skripten hatten, sollten als Vorbereitung zuerst ein wenig Recherche im Internet betreiben um Einblick in die Grundzüge von Skripting zu erhalten.

Damit Sie die aufgeführten Musterbeispiele problemlos nachvollziehen können, empfehlen wir, das diesem Beitrag angefügte Beispiel-Dokument herunterzuladen und zu verwenden. Alternativ können Sie natürlich auch eines Ihrer eigenen Dokumente verwenden und die folgende Beschreibung entsprechend anpassen/ableiten, letzteres Vorgehen ist aber nur fortgeschrittenen Anwendern mit Skripting-Erfahrung anzuraten.

Zur Vereinfachung des Ablaufs arbeiten wir im DEMO-Archiv, das bei jeder ELOoffice-Installation miterstellt werden kann.

WICHTIG: Beachten Sie, dass ein ELOoffice-Archiv durch Skripting stark beschädigt werden kann und die ELO Digital Office GmbH für dadurch beschädigte Archive keinen Support leisten kann.

Verwenden Sie also unbedingt ein Test-Archiv, wenn Sie nicht 100%ig wissen, wie Skripte im Allgemeinen und mit ELOoffice im speziellen funktionieren.

Durchführung

Im Folgenden werden einige einfachere Schritte übersprungen - sollten Sie der hier gelieferten Beschreibung nicht ganz folgen können, so ziehen Sie bitte das Handbuch und/oder das Musterbeispiel zu Textübernahme-Regeln (auch im ELOoffice-Blog unter <http://elooffice.elo.com/blog/aktuelles/der-elo-connector-textuebernahmeregeln.html> zu finden) zu Rate.

Schritt 1

Erstellen Sie einen neuen Dokumententyp und laden Sie das Musterbeispiel-Dokument als Musterdokument.

Erstellen Sie außerdem eine Zone mit Namen „Rechnungsbetrag“. Die Position und Textübernahme-Regel sollte so gewählt werden, dass der Betrag (623,54 €) übernommen wird, bspw. *'brutto'*L*

Stellen Sie als Verschlagwortungsmaske „Rechnung“ ein und weisen Sie der Zone das Feld Rechnungsbetrag zu.

Würde die Connector-Verarbeitung nun für das Dokument gestartet, würde lediglich das Feld Rechnungsbetrag mit „623,54 €“ gefüllt werden. Die fertige Konfiguration sieht so aus:

The screenshot displays the ELO Connector configuration window. On the left, a sample invoice from LDH Landhandel is shown. The invoice details include the company name, address, and contact information. The main table lists two items: 'Heckenschere Superscharf' and 'Pflanzendünger Blühmeister'. The total amount is 623,54 €.

On the right, the configuration settings for the 'Rechnung' document are shown. The 'Verschlagwortungsmaske' (Indexing Mask) is set to 'Rechnung'. The 'Parameter Zone' is configured with the 'Klassifizierungszone' (Classification Zone) set to 'Rechnungsbetrag'. The 'Textübernahmeregel' (Text Transfer Rule) is set to '*brutto'*L*.

RECHNUNG
 LDH Landhandel, Musterstraße 123, 12345 Musterstadt
 An
 Blackworld GmbH
 Musterweg 21
 12345 Musterstadt

Rechnungs-Nr.: LDH2012700
 Rechnungs-Datum: 24.10.2013
 Lieferdatum: 22.10.2013
 Verkäufer: Barbara Padovani
 E-Mail: b.padovani@ldh.de
 Internet: www.landhandel-ldh.de
 Ihr Sachbearbeiter: Tom Berg

Wir haben Ihnen die folgenden Artikel geliefert:

Position	Menge	Einheit	Bezeichnung	Einzelpreis	Gesamtpreis
1	10	Stck.	Heckenschere Superscharf Artikelnr. 5496	49,90 €	499,00 €
2	2	Kg	Pflanzendünger Blühmeister Artikelnr. 9796	12,49 €	24,98 €
Summe netto					523,98 €
+ 19% Umsatzsteuer					99,56 €
Rechnungsbetrag brutto					623,54 €

Die gelieferte Ware bleibt bis zur vollständigen Bezahlung Eigentum des Lieferanten.
 Zahlbar innerhalb 14 Tagen nach Rechnungsdatum. Bei Zahlung bis zum 1.11.2013 Skonto 2%.
 Bitte überweisen Sie den Rechnungsbetrag auf untenstehendes Konto.
 Vielen Dank für Ihren Auftrag!

Sitz: Stuttgart, Registergericht Stuttgart, HRB 15059, Steuernummer 1234567890001, USt-ID-Nr. DE 123456789

ELO Verschlagwortungsmaske
 Rechnung
 ELO Ablageskript bearbeiten
☒ Ablage in Postbox
☐ Archiv-Ablage nach Indexaufbau
☐ Ablage in ELO Ordner
☐ Automatisches Geraderücken (Deskew)

Parameter Zone
☐ Klassifizierungszone
 Erkannter Text
 Textübernahmeregel
 *brutto*L*
 Übernahme Textsegment 3
 Übernehmen in Indexzeile
 Rechnungsbetrag

Schritt 2

Unser erstes Skript soll den Rechnungsbetrag zusätzlich in die Kurzbezeichnung schreiben, diesen also hinter der Kurzbezeichnung einfügen. Das lässt sich mit einem sehr kurzen Skript bewerkstelligen.

Allgemeine Herangehensweise:

In einem ELO-Connector-Ablageskript sollten Sie grundsätzlich damit beginnen, ein ELO-Objekt zu erzeugen, über das Sie auf Eigenschaften wie die Indexfeld-Inhalte eines Dokumentes bzw. dessen Verschlagwortung zugreifen können.

Die erste Zeile sollte also so aussehen:

```
Set ELO = CreateObject("ELO.office")
```

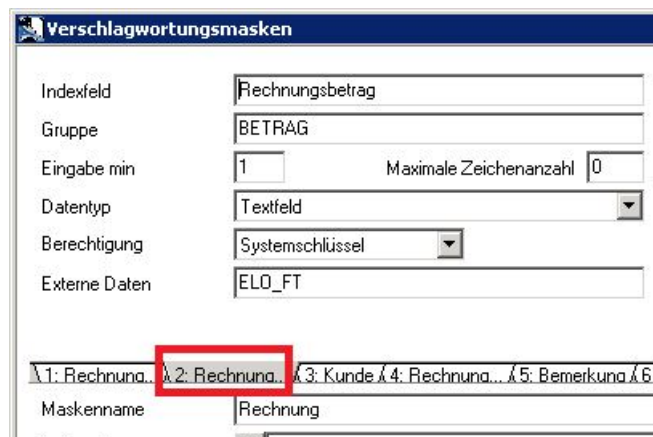
Um den Rechnungsbetrag an die Kurzbezeichnung anzuhängen, muss dieser als erstes ausgelesen werden, wir speichern diesen also in einer Variable zwischen:

```
dim rechnungsBetrag
```

```
rechnungsBetrag = ELO.getObjAttrib(1)
```

Die Funktion GetObjAttrib(int AttribNo) gewährt Zugriff auf die Indexfelder. Die Zahl in der Klammer gibt das jeweilige Indexfeld an. Die Indexfelder sind von 0-49 durchnummeriert.

Die jeweilige Nummer des Indexfeldes kann im Verschlagwortungsmasken-Editor nachgesehen werden:



Da die Nummerierung für die Skript-Funktion bei 0 anfängt, müssen wir von der angezeigten Zahl den Wert 1 abziehen. Es ergibt sich also oben stehendes `getObjAttrib(1)`.

Nach dieser Zeile Code steht in der Variablen `rechnungsBetrag` der Inhalt des Indexfeldes, später bei der Verarbeitung des Musterdokumentes wird dort also „623,54 €“ stehen.

Nun könnten wir den zwischengespeicherten Wert an die Kurzbezeichnung anhängen:

```
ELO.objShort = ELO.objShort & rechnungsBetrag
```

Damit würde der Rechnungs-Betrag allerdings direkt ohne Leerzeichen an die Kurzbezeichnung angefügt werden, daher fügen wir noch ein Leerzeichen dazwischen ein.

```
ELO.objShort = ELO.objShort & " " & rechnungsBetrag
```

Im Anschluss müssen wir noch dafür sorgen, dass die per Skript vorgenommene Änderung auch „gespeichert“ wird. Dazu können wir in der Postbox (das Connector-Dokument befindet sich zum Skript-Lauf immer in der Postbox - unabhängig davon, welche Ablage-Option verwendet wird) die folgende Funktion nutzen:

```
ELO.addPostboxFile("")
```

Wenn Sie jetzt alles speichern, können Sie den ersten Durchlauf mit dem Skript bereits durchführen. Ggf. ist es notwendig die Postbox nach der Connector-Verarbeitung noch einmal zu aktualisieren, damit die Änderung der Kurzbezeichnung sichtbar wird.

Das Skript sieht bis zu diesem Zeitpunkt folgendermaßen aus:

```
Set ELO = CreateObject("ELO.office")
dim rechnungsBetrag
rechnungsbetrag = ELO.getObjAttrib(1)
ELO.objShort = ELO.objShort & " " & rechnungsBetrag
ELO.addPostboxFile("")
```

Schritt 3

Wer schon Skripting-Erfahrung hat, kann das Skript aus Schritt 2 nun erweitern - da die Thematik dadurch aber stark an Komplexität zunimmt, schreiben wir für unsere Übung in jedem Schritt ein neues Skript.

In diesem Schritt wollen wir ein Zeichen austauschen. In unserer Rechnung ist der Betrag mit einem Komma als Dezimal-Trennzeichen angegeben. Da wir im Beispielfall jedoch überwiegend im englischsprachigen Raum agieren benötigen wir den Betrag stattdessen mit einem Punkt als Dezimal-Trennzeichen. Diese Änderung lässt sich nicht durch eine Textüberegel realisieren, so dass wir (wer hätte das von einem Skripting-Tutorial erwartet ☺) erneut zu einem Skript greifen.

Zu Beginn gehen wir genauso vor, wie wir es bereits in Schritt 2 getan haben - wir benötigen zuerst ein ELO-Objekt:

```
Set ELO = CreateObject("ELO.office")
```

Da wir erneut mit dem Rechnungsbetrag arbeiten wollen, können wir die nächsten zwei Zeilen ebenfalls übernehmen:

```
dim rechnungsBetrag
rechnungsbetrag = ELO.getObjAttrib(1)
```

Unser Rechnungsbetrag (623,54 €) steht uns nun wieder in der Variablen `rechnungsbetrag` zur Verfügung. Um ein Zeichen in dieser Zeichenfolge (eng. String) auszutauschen können wir String-Funktionen verwenden (Tipp: wenn Sie mehr über die möglichen String-Funktionen wissen möchten, suchen Sie doch im Internet einmal nach „VB Script String Functions“).

Wir benötigen an dieser Stelle die Funktion „`Replace`“ (dt. Ersetzen).

Diese Funktion benötigt (neben einigen optionalen Argumenten, die wir an dieser Stelle nicht benötigen) folgende Parameter:

`Replace (Zeichenkette, ZuErsetzendesZeichen, ErsatzZeichen)`

Als **Zeichenkette** übergeben wir unseren Rechnungsbetrag, es handelt sich bei diesem Parameter um die Zeichenkette in der ein Zeichen ausgetauscht werden soll.

Das **ZuErsetzendeZeichen** ist natürlich unser Komma, also das Zeichen das ersetzt werden soll; wir übergeben also ein in Anführungszeichen eingebettetes Komma `" , "`

Für das **ErsatzZeichen** bleibt also folgerichtig nur noch der Punkt übrig, der statt des Kommazeichens Verwendung finden soll. Darum übergeben wir einen in Anführungszeichen eingebetteten Punkt `" . "`

Da in unserem Rechnungsbetrag nur ein einziges Komma vorkommt, können wir den Austausch also sehr einfach mit der folgenden Skript-Zeile vornehmen:

```
rechnungsbetrag = Replace(rechnungsbetrag, " , " , ".")
```

Im Anschluss müssen wir den veränderten Rechnungsbetrag in das entsprechende Indexfeld zurückschreiben. Dafür verwenden wir statt des `getObjAttrib(int)`, welches zum Auslesen eines Indexfeldes verwendet wurde, nun das `setObjAttrib(int)`:

```
Call ELO.setObjAttrib(1, rechnungsbetrag)
```

und speichern die Änderung, siehe auch Schritt 2.

```
ELO.addPostboxFile("")
```

Unser fertiges „Schritt 3“-Skript sieht also folgendermaßen aus:

```
Set ELO = CreateObject("ELO.office")
dim rechnungsbetrag
rechnungsbetrag = ELO.getObjAttrib(1)
rechnungsbetrag = Replace(rechnungsbetrag, " , " , ".")
Call ELO.setObjAttrib(1, rechnungsbetrag)
ELO.addPostboxFile("")
```

Schritt 4

Skripte sind auch sehr hilfreich dabei, uns vor Falscheingaben zu schützen. An dieser Stelle entwickeln wir deshalb ein Skript, um den Eintrag im Feld **Rechnungsbetrag** auf Plausibilität zu prüfen.

Für ein solches Skript müssen wir uns als erstes überlegen, wie denn prinzipiell ein Automatismus aussehen könnte, der diese Prüfung durchführt. Es wäre nun denkbar, ein wirklich umfangreiches Skript zu schreiben, das eine Prüfung auf verschiedenste mögliche Fehler durchführt; für unser Beispiel soll es aber ausreichen, wenn geprüft wird, ob das letzte Zeichen der durch den Connector ausgelesenen Zeichenkette tatsächlich ein €-Zeichen ist.

Das Skript soll also im Hintergrund bleiben, solange das Indexfeld **Rechnungsbetrag** mit einem €-Zeichen abschließt.

Wir beginnen, wie immer, mit unserem ELO-Objekt. Da wir erneut mit dem Rechnungsbetrag arbeiten wollen, können wir auch die anderen beiden bereits bekannten Zeilen weiterverwenden:

```
Set ELO = CreateObject("ELO.office")
dim rechnungsBetrag
rechnungsbetrag = ELO.getObjAttrib(1)
```

Um nun das letzte Zeichen des Rechnungsbetrages mit dem €-Zeichen vergleichen zu können müssen wir dieses Zeichen zuerst isolieren. Dazu deklarieren wir in der zweiten Zeile unseres Skriptes noch eine weitere Variable:

```
Set ELO = CreateObject("ELO.office")
dim rechnungsBetrag, vergleichsZeichen
rechnungsbetrag = ELO.getObjAttrib(1)
```

Um aus der Zeichenfolge (eng. String) **rechnungsbetrag** das letzte Zeichen (bzw. das erste Zeichen von rechts) herauszubekommen, greifen wir erneut auf eine String-Funktion zurück. Diesmal:

```
Right ( Zeichenkette, Laenge )
```

Die Funktion Right liefert uns eine bestimmte Anzahl von Zeichen zurück und beginnt dabei auf der rechten Seite der Zeichenkette, also beim letzten Zeichen. Wir verwenden die Parameter also wie folgt:

Zeichenkette ist wie gehabt unsere Rechnungsnummer. Es handelt sich um die Zeichenkette bei der Zeichen von rechts aus zurückgegeben werden sollen.

Da wir nur das letzte Zeichen haben wollen (welches im Optimal-Fall unser €-Zeichen ist) verwenden wir eine **Laenge** von **1**, also nur das erste Zeichen von rechts.

Wir verwenden die Funktion also jetzt um in unsere Variable **vergleichsZeichen** das letzte Zeichen aus **rechnungsbetrag** einzulesen:

```
vergleichsZeichen = Right ( rechnungsbetrag, 1 )
```

Jetzt müssen wir prüfen, ob das **vergleichsZeichen** mit € übereinstimmt. Im Folgenden werden wir, um einen Fehler zu simulieren, statt des €-Zeichens ein \$-Zeichen verwenden, so dass wir sehen können, dass das Skript auch eingreift, wenn das notwendig ist. Solange kein Fehler vorliegt, bleibt das Skript nämlich brav im Hintergrund und wir bemerken nichts von seiner Existenz.

```
if vergleichsZeichen <> "$" Then
```

```
    MsgBox "Es wurde ein falscher Wert in Rechnungsbetrag eingetragen."
```

Die erste der beiden Zeilen liest sich „übersetzt“ folgendermaßen:

Falls (eng. if) der in **vergleichsZeichen** gespeicherte Wert ungleich (<>) mit dem Dollar-Zeichen (\$) ist, dann (eng. then) führe die folgende Zeile aus.

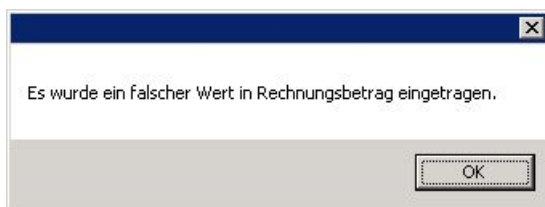
Die zweite der Zeilen bewirkt nichts anderes als die Einblendung eines Nachrichten-Fensters (eng.: Message Box) erscheint und den in Anführungszeichen hinterlegten Text (den Sie beliebig variieren können) anzeigt.

Wir schließen dann mit

End if

Was nichts anderes bedeutet, als dass die if-Verzweigung in dieser Zeile endet.

Führen wir die Connector-Verarbeitung mit dem Skript aus, so erhalten wir also folgende Nachricht:



Und unser fertiges „Schritt 4“-Skript sieht jetzt folgendermaßen aus:

```
Set ELO = CreateObject("ELO.office")
dim rechnungsBetrag, vergleichsZeichen
rechnungsBetrag = ELO.getObjAttrib(1)
vergleichsZeichen = Right ( rechnungsBetrag, 1 )
if vergleichsZeichen <> "$" Then
    MsgBox "Es wurde ein falscher Wert in Rechnungsbetrag eingetragen."
End if
```

Führen Sie jetzt ruhig einen Testdurchlauf durch, um die Funktionalität der Nachrichten-Box zu prüfen. Damit das Skript so arbeitet, wie wir es ursprünglich geplant haben, dürfen Sie aber nicht vergessen, das Dollar-Zeichen wieder durch das Euro-Zeichen zu ersetzen, so daß die entsprechende Skript-Zeile nun lautet:

```
if vergleichsZeichen <> "€" Then
```

Abschluss

Es gibt sehr viele weitere Möglichkeiten, um per Skript zusätzliche Anforderungen zu erfüllen oder Fehlerbehandlungen einzubauen. Prinzipiell sind Ihrer Phantasie dabei kaum Grenzen gesetzt. Sie können sowohl generelles VB-Skripting zur Programmierung Ihrer Skripte verwenden als auch die über die ELOoffice-OLE-Schnittstelle bereitgestellten Funktionen nutzen. Letztere können Sie der OLE-Schnittstellenbeschreibung entnehmen, die Ihnen der ELOoffice Support (support@elo.com) auf Anfrage gerne zur Verfügung stellt.

Wir hoffen, dass die oben aufgeführten Skripte Ihnen einen kleinen Einblick in die Welt des ELOoffice-Skriptings gewähren - und Sie mit Hilfe dieser kleinen Skripte als „Schema-Anleitung“ vielleicht auch die eine oder andere spezielle Anforderung, die Ihnen begegnet, lösen können.

Wenn Sie jetzt Spaß am Skripting gefunden haben sollten, können Sie sich bereits auf den nächsten Teil unserer ELOoffice-Connector-Skripting-Reihe freuen. Dieser wird sich dann an Fortgeschrittene wenden und Ihnen die Kombination der Anwendungsfälle sowie eine Möglichkeit zur schöneren und aussagekräftigeren Aufbereitung der Skript-Rückmeldung aufzeigen.