

Python-Skript: Analyse einer Umsatzsteuer-ID für ACL und IDEA

1 Vorbemerkung

Python wird mit seinen vielfältigen Bibliotheken als Werkzeug innovativer und zukunftsgerichteter Datenanalysen innerhalb des Prüfungsumfeldes zunehmend wichtiger. Wir möchten diese Entwicklung konstruktiv begleiten und erstellen neue Funktionserweiterungen für ACL und IDEA zunehmend auf der Grundlage von Python. Diese können als "benutzerspezifische Funktionen" anschließend in gleicher Form innerhalb beider Programme eingesetzt werden.

In einer weiteren Entwicklungsrichtung integrieren wir ACL- und IDEA-Funktionalität in eigenständige Python-Programme. Dieses ermöglicht weitergehende Automatisierungen unter Einbeziehung innovativer Python-Bibliotheken. Kolleginnen und Kollegen, die hieran Interesse haben, können sich über unsere zuständigen Ansprechpartner informieren. Einige Hinweise finden sich in unserer Veröffentlichung "Continuous Auditing mit IDEA-Skript", NWB-Verlag 2019.



2 Ziel und Inhalt des aktuellen Skriptes

Das beigefügte Python-Skript "USTID_Nr_Pruefung_ROP.py" ergänzt den Standard-Funktionsumfang von ACL und IDEA. Es prüft die in einem Tabellenfeld aufgeführten Umsatzsteuer-Identifikationsnummer-Angaben programmtechnisch im Hinblick auf formale Richtigkeit oder online ihre Gültigkeit auf Basis der Angaben des Bundeszentralamts für Steuern (BZSt) oder der Europäischen Union (VIES). Zusätzlich prüft es nicht nur ihre Zusammenhänge mit den in Tabellenfeldern angegebenen Firmennamen und Adressen, sondern liefert es auch, falls vorhanden, Informationen über die Firmennamen, Adressen und den Gültigkeitszeitraum von als ungültig beurteilten Umsatzsteuer-ID's. Das Skript wird als "Funktion" bereitgestellt. In dieser Form kann es innerhalb der Gleichungseditoren von ACL und IDEA gleichermaßen verwendet werden. Nähere Hinweise finden sich in der nachfolgenden Beschreibung.

3 Programmbestandteile

Zu dieser Python-Funktion gehören die nachstehenden Programmbestandteile:

Bezeichnung	Objekt	Funktion	Kommentar
USTID_Nr_Pruefung_ROP.py	Python-Skript	Programmcode	Einzelne Programm-Module mit dem Hauptmodul "Ust-IdNr. Analyse"

4 Voraussetzungen für den Einsatz des Skriptes

Die Verwendung des aufgeführten Skriptes erfordert einen installierten Python-Interpreter. Für IDEA werden die entsprechenden Bibliotheken ab IDEA Version 10.4 mitgeliefert und bei der Programminstallation berücksichtigt. ACL stützt sich auf hingegen auf eine separat installierte Python-Umgebung. Über diese Anforderung hinaus müssen die von uns bereitgestellten Skripte in bestimmte Datenverzeichnisse kopiert werden. In ACL sind diese Datenverzeichnisse frei wählbar. Sie müssen jedoch in einer Umgebungsvariablen adressiert werden. In Bibliotheksstruktur von IDEA muss die Ablage jeweils innerhalb des Projektordners und dort in dem Verzeichnis "Benutzerdefinierte Funktionen.ILB" (Customs Functions) erfolgen. Die folgenden Installationsangaben geben weitere Hinweise.

5 Installation des Python-Interpreters

Diese ist insbesondere für den Python-Skripteinsatz notwendig. Angesichts der generell wachsenden Bedeutung von Python ist die Installation *der jeweils aktuellen Version* auch sinnvoll. Den letzten Programmstand erhält man über die Internet-Python-Seite:

<https://www.python.org/>

Hier finden sich ggf. auch ergänzende Installationshinweise.

6 Installations-Schritte zu ACL und IDEA

6.1 ACL-Installation (lokale ACL-Versionen)

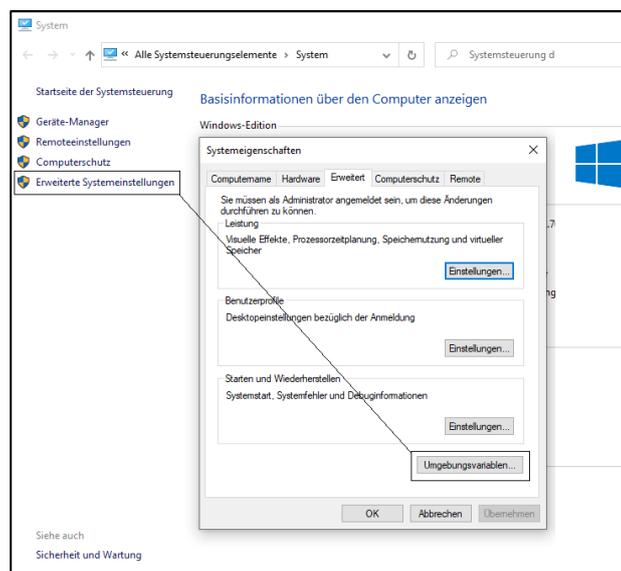
Vor Verwendung der bereitgestellten Skripte müssen auf den Rechner mit der lokalen ACL-Installation neue Verzeichnisse für die zu verwendenden Skripte angelegt und Umgebungsvariable für das Betriebssystem aufgenommen werden. Nachfolgend beschreiben wir das Vorgehen für Windows 10:

- Skriptverzeichnisse einrichten

Es können eines oder mehrere Verzeichnisse verwendet werden. Sie sollten nicht besonders geschützt sein. Die Anlage erfolgt üblicherweise mit dem Windows-Explorer und der Menü-Anweisung "Neue Ordner".

- Umgebungsvariable anlegen (einmalig)

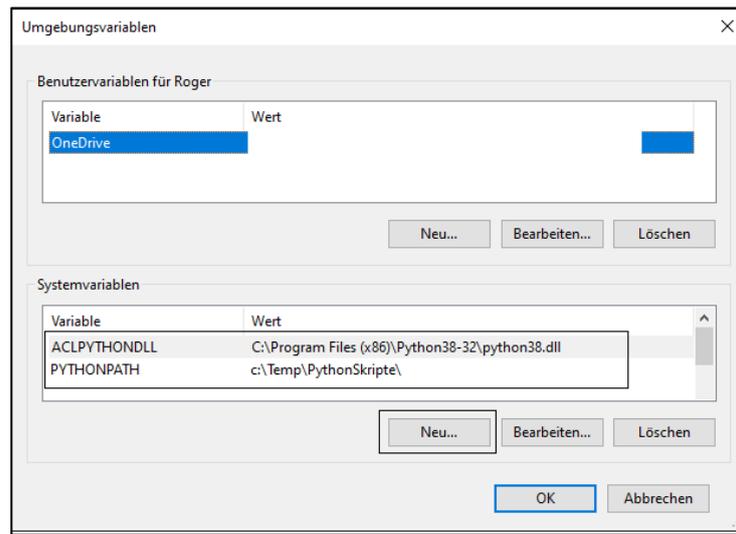
Die Tastenkombination "Windows + Pause" mit der Option "Erweiterte Systemeinstellungen" führt zu dem Schalter "Umgebungsvariablen":



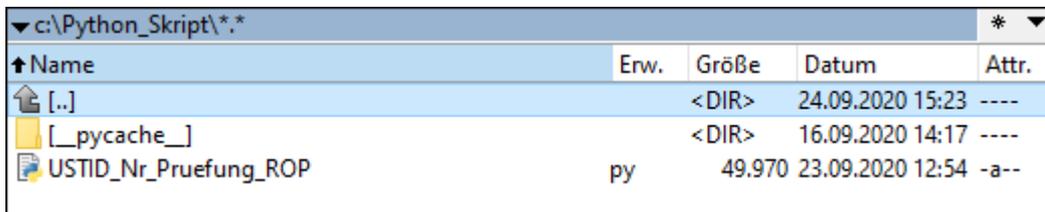
Hier können die erforderlichen Variablen (Systemvariable) ergänzt werden:

Variable	Angabe	Beispiel
PYTHONPATH	Der vollständige Pfad der Ordner, die Sie für Ihre Python-Skripts erstellt haben. Trennen Sie mehrere Ordnerpfade durch ein Semikolon.	C:\temp\PythonSkripte; C:\dev;C:\tmp
ACLPYTHONDLL	Der vollständige Pfad und Dateiname der Python-DLL-Datei im Python-Installationsordner, den Sie mit Analytics oder dem Robots-Agenten verwenden möchten.	c:\Program Files (x86)\Python38-32\python33.dll

Die aufgeführten Angaben sind lediglich Beispiele. Die individuellen Einträge müssen die tatsächliche Installations- und Arbeitsumgebung berücksichtigen.



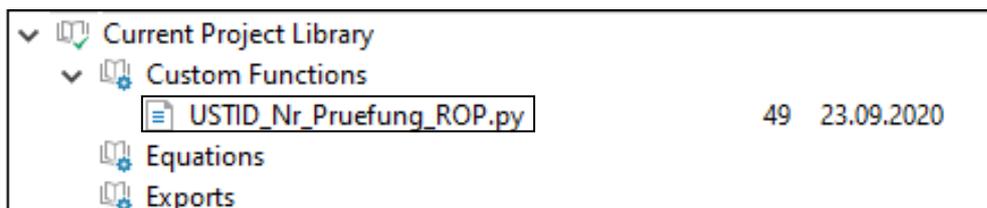
- Skript in angegebenes Verzeichnis kopieren (für jedes neu erhaltene Skript)



Damit wären die Präliminarien für lokale geführte ACL-Prüfsoftware abgeschlossen. Für die ACL-Serverlösung kann es eine abweichende Handhabung geben, die aus den ACL-Handbüchern hervorgeht.

6.2 IDEA-Installation (lokale IDEA-Versionen)

Zur Verwendung unserer Python-Skripte in IDEA müssen diese bei externen Projekten in das jeweilige Projektverzeichnis und dort in den Ordner "Benutzerdefinierte_Funktionen.ILB" (Custom Functions.ILB) gespeichert werden.



Sie gesellen sich dort neben benutzerdefinierte Funktionen, die mit IDEA-VBA-Skript geschrieben wurden und können auch ähnlich eingesetzt werden.

7 Einsatz des Skriptes in ACL-Prüfsoftware

ACL unterstützt die Weitergabe von Argumenten (i.d.R. Tabellenfelder oder steuernde Parameter) an externe Python-Funktionen und den Empfang von Rückgabewerten zur Verwendung innerhalb von Rechenfeldern einer ACL-Tabelle entsprechend den möglichen Rückgabeformaten mit folgenden Funktionen:

- PYDATE() Rückgabe Datum
- PYDATETIME() Rückgabe Zeit
- PYLOGICAL() Rückgabe "Falsch / Richtig"
- PYNUMERIC() Rückgabe Zahl / Wert
- PYSTRING() Rückgabe Zeichen / Text

Die Funktion selbst stellt den Rückgabewert in einem der aufgeführten Formate innerhalb der Rechengleichung zur Verfügung. Hierfür benötigt sie zumindest drei Parameter:

Parameter 1: Skriptbezeichnung und Funktionsbezeichnung

Die Bezeichnung des Skriptes und der hierin kodierten Funktion könnten unterschiedlich lauten, weshalb hier beide Angaben erforderlich sind. Es empfiehlt sich allerdings bei der Skripterstellung auf gleichlautende Bezeichnungen für die Skriptdatei und die in ihr enthaltene Funktion zu achten. Beide Angaben sind gemeinsam in Anführungszeichen einzuschließen und durch ein *Semikolon* zu trennen (*Achtung! Innerhalb des ACL-Handbuches wird fälschlicherweise mehrfach auf die Trennung durch ein Komma hingewiesen. Dieses führt zu einer fehlerhaften Ausgabe*). Beispiel: "USTID_Nr_Pruefung_ROP; USTID_Nr_Pruefung_ROP "

Parameter 2: Länge des Rückgabewertes in Anzahl Zeichen

Parameter 3: Feldbezeichnung, Wert oder steuernder Parameter

Für dies Skript enthält der dritte Parameter die Bezeichnung der auszuführenden Prüfung. Der Anwender kann eine der folgenden Bezeichnungen abhängig von seinem Wunsch auswählen:

- "Formal": Formale Prüfung – ausschließlich durch das Programm
- "Vies": Online-Prüfung - EU-Server (VIES)
- "Vat": Online-Prüfung Bundeszentralamtes für Steuern (BZSt)
- "Ausland": Formale Prüfung für deutsche Ust-IdNr. und Online-Prüfung ausländischer Ust-IdNr durch das Bundeszentralamt für Steuern (BZSt)
- "Tief ": Formale Prüfung für alle Ust-IdNr sowie Online-Prüfung ausländischer Ust-IdNr durch das Bundeszentralamt für Steuern (BZSt) und deutscher Ust-IdNr über die EU-Plattform (VIES)

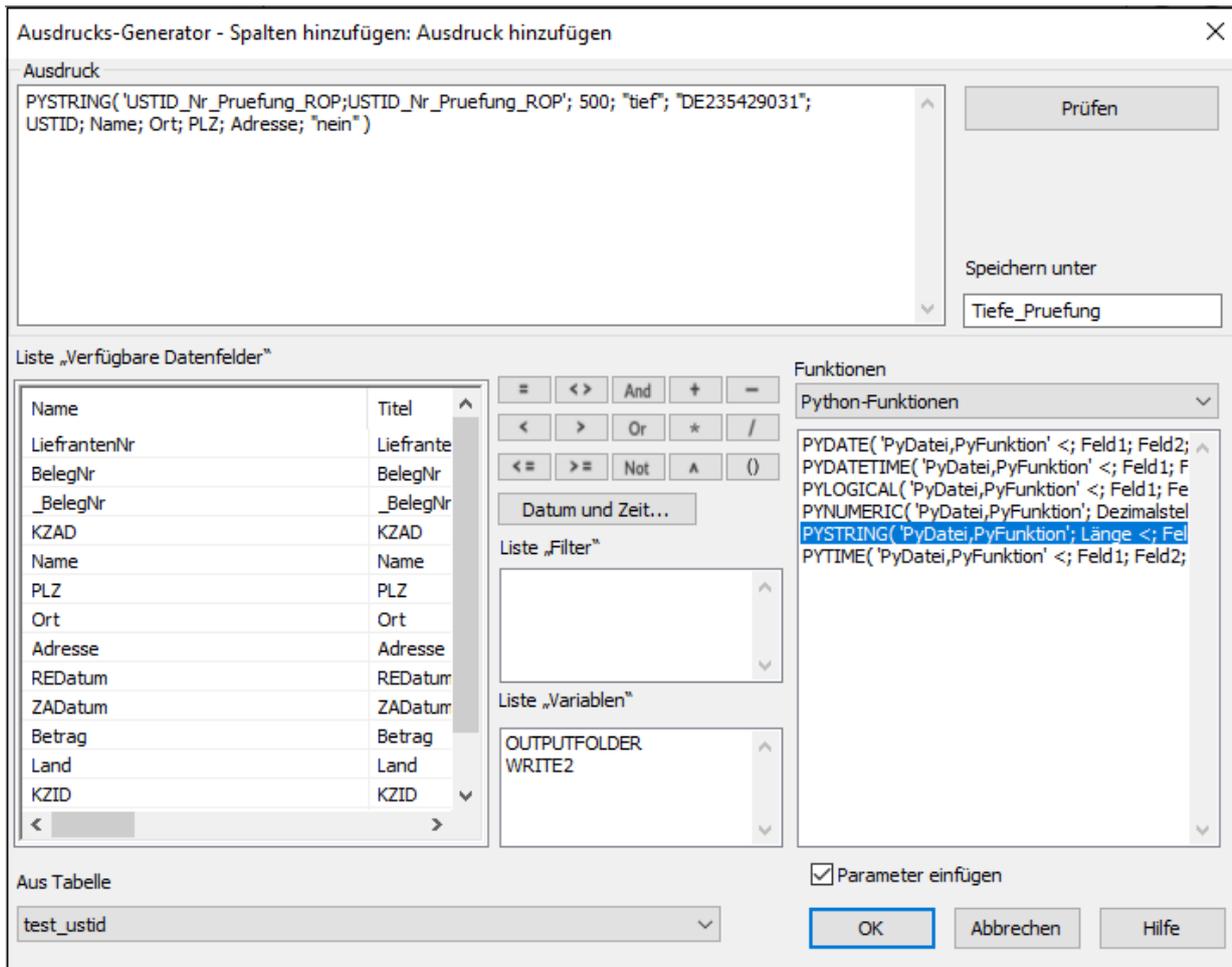
Parameter 4: Die eigene betriebliche deutsche Umsatzsteuer-ID

Für dies Skript enthält der vierte Parameter die deutsche Umsatzsteuer-Identifikationsnummer des prüfenden Betriebs (z.B. "DEXXXXXXXXX")!

Parameter x: Weitere Feldbezeichnungen, Werte oder steuernder Parameter

Üblicherweise enthält der vierte Parameter innerhalb dieses Skripts die Feldbezeichnung aus einer ACL-Tabelle (z.B. "USTID"). Weitere Parameter müssen wie folgt angegeben werden.

Beispiel: PYSTRING("USTID_Nr_Pruefung_ROP; USTID_Nr_Pruefung_ROP "; 500; "Tief"; "DEXXXXXXXXX"; USTID; Name; Ort; PLZ; STRASSE; "nein/ja")



Damit die Inhalte der innerhalb der Funktion verwendeten Tabellenfelder richtig übergeben werden, empfiehlt es sich, die verwendete Feldbezeichnung in eine Funktion Alltrim() einzukleiden. Innerhalb einer Python-Funktion können weitere externe Python-Funktionen aufgegriffen werden, sofern sie alle innerhalb eines einheitlichen Verzeichnisses gespeichert sind.

Beachten Sie bitte, dass die ersten 5 Parameter bis einschließlich der Feldbezeichnung mit der zu prüfenden Umsatzsteuer-ID *zwingend* einzugeben sind. Soweit für die weiteren Felder "Name", "Postleitzahl", "Ort", "Straße" keine Angaben in der ACL-Tabelle vorliegen, können sie durch doppelte Anführungszeichen "" teilweise oder vollständig ersetzt werden:

Beispiel: `PYSTRING("USTID_Nr_Pruefung_ROP; USTID_Nr_Pruefung_ROP "; 500; "Tief"; "DEXXXXXXXXXX"; USTID; ""; ""; ""; ""; "nein")`

Da ACL die Python-Funktion in einem virtuellen Rechenfeld als "Gleichung" berücksichtigt, würden laufend neue Online-Abfragen für das Ergebnisfeld vorgenommen, sobald eine Tabellenzeile auf dem Bildschirm erscheint. Da dieses nicht praktikabel ist, empfiehlt es sich, **keinen neuen Ausdruck in das Tabellenlayout aufzunehmen, sondern ein kleines ACL-Skript für die Ausgabe zu verwenden**, welches bei geöffneter ACL-Tabelle gestartet wird:

```
GROUP
com Variable Prüfungsergebnis mit Python-Funktion einrichten
Pruefungsergebnis = PYSTRING("USTID_Nr_Pruefung_ROP; USTID_Nr_Pruefung_ROP"; 500; "Tief"; "DE235429031"; USTID; Name; Ort; PLZ; Adresse; "nein")
com Alle Felder der geöffneten Tabelle und die Variable in eine neue Tabelle ausgeben
Extract all Pruefungsergebnis to Ergebnisausgabe
End
```

Hiernach wird eine neue "Ergebnistabelle" erzeugt, welche die Variable (hier "Prüfungsergebnis") als Feld mit festem Inhalt (keine Gleichung) beinhaltet.

Das Ergebnis sieht hiernach wie folgt aus:

USTID	Prüfungsergebnis
1 DE235429031	Gueltige deutsche Ust-IdNr. . Firma: ---\n. Quelle: VIES.
2 PL 873-28-98-418	Die angefragte Ust-IdNr. ist gueltig, Firmenname unpruefbar. Firmenort unpruefbar. PLZ unpruefbar. Firmenstrasse unpruefbar. - - . Quelle: Bundeszentralamt fuer Steuern.
3 DE814521375	Gueltige deutsche Ust-IdNr. . Firma: ---\n. Quelle: VIES.
4 DE 25 84 64 250	Gueltige deutsche Ust-IdNr. . Firma: ---\n. Quelle: VIES.
5 FR22424761419	Die angefragte Ust-IdNr. ist gueltig, Firmenname unpruefbar. Firmenort unpruefbar. PLZ unpruefbar. Firmenstrasse unpruefbar. - - . Quelle: Bundeszentralamt fuer Steuern.
6 GB 107 3280 00	Die angefragte Ust-IdNr. ist gueltig, Firmenname unpruefbar. Firmenort unpruefbar. PLZ unpruefbar. Firmenstrasse unpruefbar. - - . Quelle: Bundeszentralamt fuer Steuern.
7 AT U63224727,	Die angefragte Ust-IdNr. ist gueltig, Firmenname unpruefbar. Firmenort unpruefbar. PLZ unpruefbar. Firmenstrasse unpruefbar. - - . Quelle: Bundeszentralamt fuer Steuern.
8 GB 524 3711 68 123	Die angefragte Ust-IdNr. ist ungueltig. Sie ist nicht in der Unternehmerdatei des betreffenden EU-Mitgliedstaates registriert.Hinweis:Ihr Geschäftspartner kann seine gueltige
9 DK13585628	Die angefragte Ust-IdNr. ist gueltig, Firmenname unpruefbar. Firmenort unpruefbar. PLZ unpruefbar. Firmenstrasse unpruefbar. - - . Quelle: Bundeszentralamt fuer Steuern.
10 GB 766 8008 04	Die angefragte Ust-IdNr. ist gueltig, Firmenname unpruefbar. Firmenort unpruefbar. PLZ unpruefbar. Firmenstrasse unpruefbar. - - . Quelle: Bundeszentralamt fuer Steuern.
11 GB613451470	Die angefragte Ust-IdNr. ist gueltig, Firmenname unpruefbar. Firmenort unpruefbar. PLZ unpruefbar. Firmenstrasse unpruefbar. - - . Quelle: Bundeszentralamt fuer Steuern.
12 FR22424761419	Die angefragte Ust-IdNr. ist gueltig, Firmenname unpruefbar. Firmenort unpruefbar. PLZ unpruefbar. Firmenstrasse unpruefbar. - - . Quelle: Bundeszentralamt fuer Steuern.
13 IE 2251597K	Die angefragte Ust-IdNr. ist ungueltig. Sie war im Zeitraum von ... bis ... gueltig (siehe Feld 'Gueltig_ab' und 'Gueltig_bis'). Firmenname unpruefbar. Firmenort unpruefbar. PLZ
14 EL 094327684	Die angefragte Ust-IdNr. ist gueltig, Firmenname unpruefbar. Firmenort unpruefbar. PLZ unpruefbar. Firmenstrasse unpruefbar. - - . Quelle: Bundeszentralamt fuer Steuern.
15 IE 6693587J	Die angefragte Ust-IdNr. ist ungueltig. Sie war im Zeitraum von ... bis ... gueltig (siehe Feld 'Gueltig_ab' und 'Gueltig_bis'). Firmenname unpruefbar. Firmenort unpruefbar. PLZ
16 EL 094327684	Die angefragte Ust-IdNr. ist gueltig, Firmenname unpruefbar. Firmenort unpruefbar. PLZ unpruefbar. Firmenstrasse unpruefbar. - - . Quelle: Bundeszentralamt fuer Steuern.
17 IE 8/Y/93637V	Die angefragte Ust-IdNr. ist gueltig, Firmenname unpruefbar. Firmenort unpruefbar. PLZ unpruefbar. Firmenstrasse unpruefbar. - - . Quelle: Bundeszentralamt fuer Steuern.
18 EL94327684	Formale ungueltige europaeische Ust-IdNr. aus Griechenland
19 IE 9950 9588	Die angefragte Ust-IdNr. ist gueltig, Firmenname unpruefbar. Firmenort unpruefbar. PLZ unpruefbar. Firmenstrasse unpruefbar. - - . Quelle: Bundeszentralamt fuer Steuern.

8 Einsatz des Skriptes in IDEA-Prüfsoftware

IDEA ermöglicht den Einsatz von Python-Skripten *lediglich bei lokalen Installationen und nicht für die IDEA-Server* Komponente. Innerhalb des Gleichungseditors wird dabei lediglich eine Funktion (im Bereich Zeichen-Funktionen) mit der Bezeichnung "Python()" unterstützt.

Die Funktion selbst stellt den Rückgabewert ausschließlich in einem Zeichenformat innerhalb der Rechengleichung zur Verfügung. Hierfür benötigt sie mindestens zwei Parameter:

Parameter 1: Skriptbezeichnung (und Funktionsbezeichnung)

Die Bezeichnung des Skriptes und der hierin kodierten Funktion *müssen identisch sein!* Die gemeinsame Skript- und Funktionsbezeichnung ist in Anführungszeichen einzuschließen.

Parameter 2: Feldbezeichnung, Wert oder steuernder Parameter

Für dies Skript enthält der zweite Parameter die Bezeichnung der auszuführenden Prüfung. Der Anwender kann eine der folgenden Bezeichnungen abhängig von seinem Wunsch auswählen:

- "Formal": formale Prüfung
- "Vies": Prüfung auf Basis des Servers der europäischen Kommission
- "Vat": Prüfung auf Basis des Servers des Bundeszentralamtes für Steuern
- "Ausland": nur formale Prüfung für deutsche Ust-IdNr. und Prüfung von ausländischen Ust-IdNr. auf Basis des Servers von Bundeszentralamt für Steuern
- "Tief ": tiefe Prüfung auf Basis unterschiedlichen Server.

Parameter 3: Feldbezeichnung, Wert oder steuernder Parameter

Für dies Skript enthält der dritte Parameter die Umsatzsteuer-Identifikationsnummer der Benutzer (z.B. "DEXXXXXXXXXX").

Parameter 4: Die eigene betriebliche deutsche Umsatzsteuer-ID

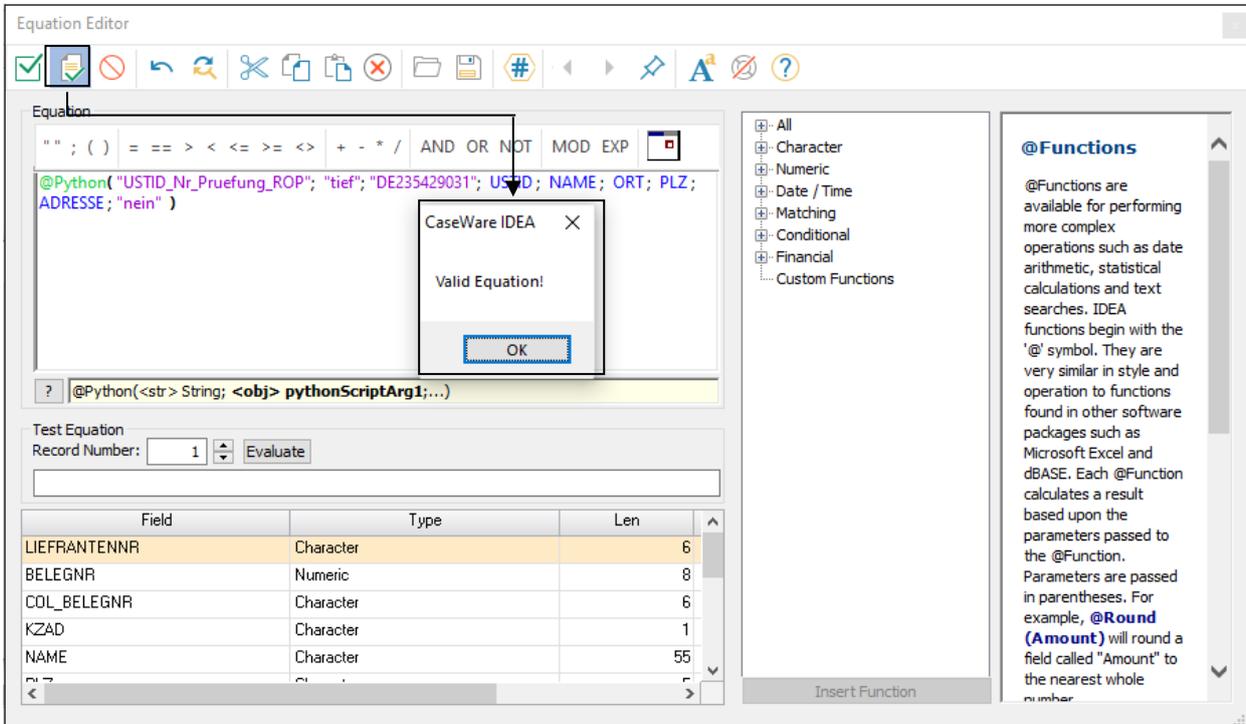
Für dies Skript enthält der vierte Parameter die deutsche Umsatzsteuer-Identifikationsnummer des prüfenden Betriebs (z.B. "DEXXXXXXXXXX")!

Parameter x: Feldbezeichnungen, Werte oder steuernde Parameter

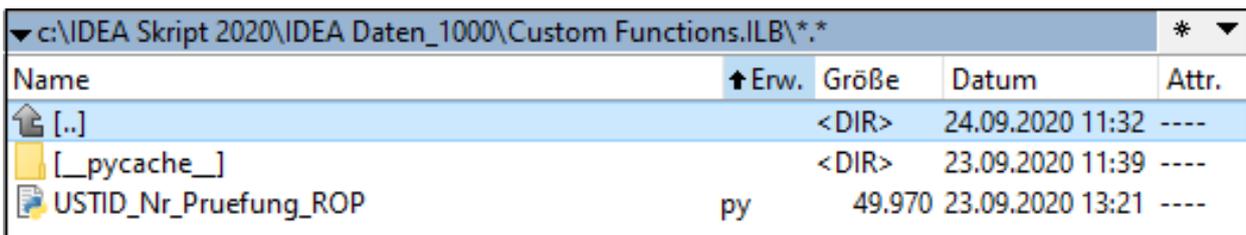
Es *müssen* so viele Parameter in dem Python-Aufruf von IDEA angegeben werden, wie in der Python-Funktion kodiert wurden! Ggf. sind auch hier doppelte Anführungszeichen als Ersatz für fehlende Felder (NAME, ORT etc.) zu verwenden.

Beispiel: @PYTHON("USTID_Nr_Pruefung_ROP "; "Tief"; "DEXXXXXXXXXX"; "USTID"; NAME; Ort; PLZ; STRASSE; "nein")

In IDEA besteht ebenfalls das Problem, dass virtuelle Rechenfelder lediglich die Gleichung (Python-Funktion) und nicht deren Ergebnis ablegen. Auch in diesem Fall würde jeweils erneut eine Server-Abfrage vorgenommen, sobald ein entsprechendes Feld auf dem Bildschirm erscheint. Um dieses zu verhindern, ist es erforderlich, das **Ergebnisfeld mit dem Feldtyp "Zeichen" (nicht virtuelles Zeichenfeld!)** anzulegen und die Python-Funktion in der gewünschten Weise aufzurufen.



Achtung: Vor einer ersten Verwendung in IDEA muss jede neue Python-Funktion innerhalb des Gleichungseditors mit der Verifizierungsfunktion **"Überprüfen" am oberen Bildschirmrand** validiert werden (siehe oben). Erst hiernach wird sie für die IDEA-Nutzung durch Transfer in den Python-Cache (Ordner "Benutzerdefinierte Funktionen") freigegeben:



Das Ergebnis:

USTID	TIEF_PRUEFUNG
1 DE235429031	Gültige deutsche USt-IdNr. . Firma: ---\n. Quelle: VIES.
2 PL 873-28-98-418	Die angefragte USt-IdNr. ist gültig. Firmenname unprüfbar. Firmenort stimmt nicht ueberein. PLZ unprüfbar. Firmenstrasse unprüfbar. - - . Quelle: Bundeszentralamt fuer Steuern.
3 AT U63224727.	Die angefragte USt-IdNr. ist gültig. Firmenname unprüfbar. Firmenort stimmt nicht ueberein. PLZ unprüfbar. Firmenstrasse unprüfbar. - - . Quelle: Bundeszentralamt fuer Steuern.
4 FR22424761419	Die angefragte USt-IdNr. ist gültig. Firmenname unprüfbar. Firmenort stimmt ueberein. PLZ unprüfbar. Firmenstrasse unprüfbar. - - . Quelle: Bundeszentralamt fuer Steuern.
5 ES 828318236	Die angefragte USt-IdNr. ist gültig. Firmenname stimmt nicht ueberein. Firmenort unprüfbar. PLZ unprüfbar. Firmenstrasse nicht angefragt. - - . Quelle: Bundeszentralamt fuer Steuern.
6 ES 863618474	Die angefragte USt-IdNr. ist gültig. Firmenname stimmt nicht ueberein. Firmenort unprüfbar. PLZ unprüfbar. Firmenstrasse nicht angefragt. - - . Quelle: Bundeszentralamt fuer Steuern.
7	

9 Abschließende Hinweise

Die aufgeführten Beispiele verweisen auf eine differenzierte Handhabung innerhalb von ACL- und IDEA-Prüfsoftware. Nach der erstmaligen Einrichtung sollten Python-Skripte aber in beiden Umgebungen problemfrei laufen. Wir werden daher zukünftig verstärkt Python-Lösungen anbieten, die sich für alle Prüfprogramme eignen.

10 Weiter Hinweise

Alle Skripte (Funktionserweiterungen) stellen wir im Rahmen eines *kollegialen Gedankenaustausches kostenfrei* ausschließlich zur Anwendung für Prüfungszwecke zur Verfügung. Diese Bereitstellung erfolgt ohne jede Gewährleistung für eine bestimmte Funktion oder Fehlerfreiheit. Jede weitere Verwendung, z.B. für Seminar- oder Schulungszwecke ist ausdrücklich untersagt! Sollten Anwender einen solchen Einsatz feststellen, danken wir für einen kurzen Hinweis.

Die aufgeführten Funktionserweiterungen werden laufend ergänzt. Angaben zu deren Inhalt und Stand finden sie auf unserer Internetseite:

www.odenthal-auditsoftware.de/

Die Skripte können mit Hilfe eines dort angeordneten Formulars angefordert werden. Anregungen und Fehlerhinweise nehmen wir gerne entgegen.

Wir wünschen viel Erfolg bei der Anwendung.

Roger Odenthal Nicodem Choula Teghuo