

Python-Skript: IBAN-Analyse für ACL und IDEA

1 Ziel und Inhalt des Skriptes

Das beigefügte Python-Skript "IBAN_Analyse_ROP.py" ergänzt den Standard-Funktionsumfang von ACL und IDEA. Es prüft die in einem Tabellenfeld aufgeführten IBAN-Angaben im Hinblick auf formale Richtigkeit. 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.

2 Programmbestandteile

Zu dieser Python-Funktion gehören die nachstehenden Programmbestandteile:

Bezeichnung	Objekt	Funktion	Kommentar
IBAN_Analyse_ROP.py	Python-Skript	Programmcode	Einzelne Programm-Module mit Hauptmodul "IBAN Analyse"

3 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.

4 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.

5 Installations-Schritte zu ACL und IDEA

5.1 ACL-Installation (lokale ACL-Versionen)

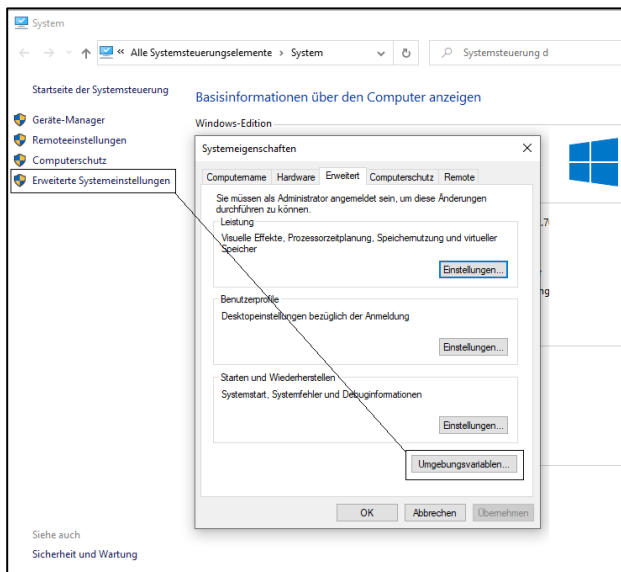
Vor Verwendung der bereitgestellten Skripte müssen auf den Rechner mit der lokalen ACL-Installation neue Verzeichnisse für die zu verwendenden Skripte angelegt und Umgebungsvariable für das Betriebssystem aufgenommen werden. Nachfolgend beschreiben wir das Vorgehen für Windows 10:

- 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)

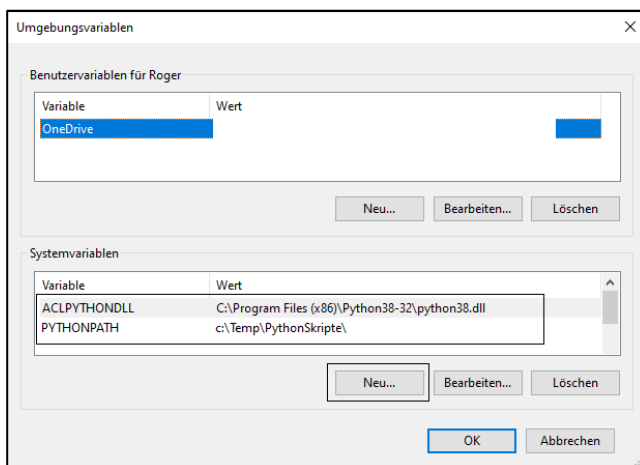
Die Tastenkombination "Windows + Pause" mit der Option "Erweiterte Systemeinstellungen" führt zu dem Schalter "Umgebungsvariablen":



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\python38.dll

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



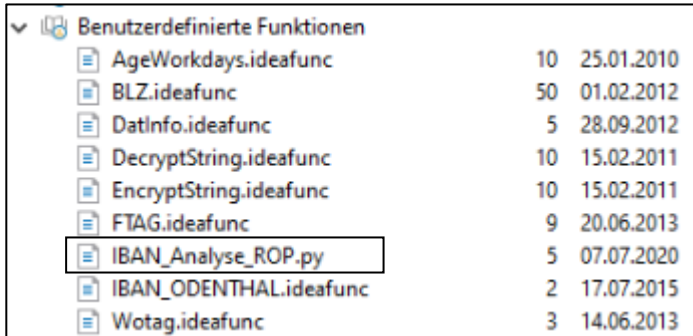
- Skript in angegebenes Verzeichnis kopieren (für jedes neu erhaltene Skript)

Name	Erw.	Größe	Datum	Attr.
...		<DIR>	07.07.2020 12:51	----
[_pycache_]		<DIR>	07.07.2020 12:51	----
hello	py	85	30.06.2020 16:07	-a--
IBAN_Analyse_ROP	py	4.242	07.07.2020 10:25	-a--

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.

5.2 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.



Benutzerdefinierte Funktionen		
AgeWorkdays.ideafunc	10	25.01.2010
BLZ.ideafunc	50	01.02.2012
DatInfo.ideafunc	5	28.09.2012
DecryptString.ideafunc	10	15.02.2011
EncryptString.ideafunc	10	15.02.2011
FTAG.ideafunc	9	20.06.2013
IBAN_Analyse_ROP.py	5	07.07.2020
IBAN_ODENTHAL.ideafunc	2	17.07.2015
Wotag.ideafunc	3	14.06.2013

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

6 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 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-Handbuchs wird fälschlicherweise mehrfach auf die Trennung durch ein Komma hingewiesen. Hiernach kommt es zu einer fehlerhaften Ausgabe*). Beispiel: "IBAN_Analyse_ROP; IBAN_Analyse_ROP"

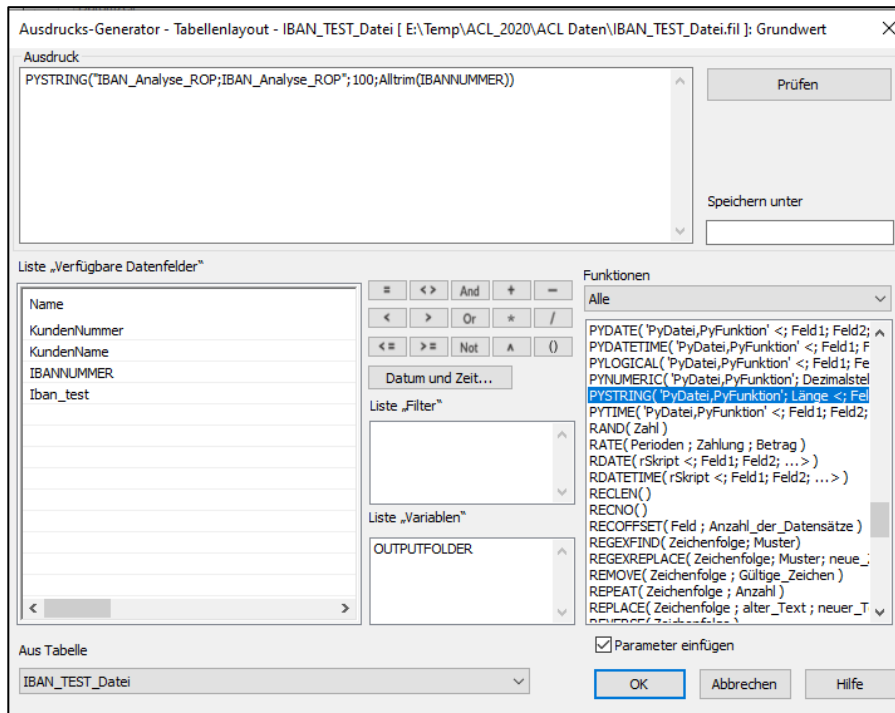
Parameter 2: Länge des Rückgabewertes in Anzahl Zeichen

Parameter 3: Feldbezeichnung, Wert oder steuernder Parameter

Üblicherweise enthält der dritte Parameter die Feldbezeichnung aus einer ACL-Tabelle (z.B. "IBAN_Angabe").

Parameter x: Weitere Feldbezeichnungen, Werte oder steuernder Parameter

Beispiel: PYSTRING("IBAN_ANALYSE_ROP; IBAN_ANALYSE_ROP"; 100; "IBANFELD")



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.

Das Ergebnis:

	KundenNummer	KundenName	IBANNUMMER	Iban_test
1	10012	Koby Huber	DE8350890000131579209	Gueltige IBAN aus Deutschland
2	10013	Bryant Singleton	CH0200700110000387896	Gueltige IBAN aus Schweiz
3	10016	Sullivan Rivas	DE93508526510075078568	Gueltige IBAN aus Deutschland
4	10025	Josie Humphrey	DE91506521240017112582	Gueltige IBAN aus Deutschland
5	10052	Kole Long	DE91505922000007211490	Gueltige IBAN aus Deutschland
6	10061	Rylan Gilbert	DE9350890000019011208	Gueltige IBAN aus Deutschland
7	10063	Alec Baldwin III	LI2608802001003488101	Gueltige IBAN aus Liechtenstein
8	10088	Liana Nielsen	DE02120300001017387992	Gueltige IBAN aus Deutschland
9	10098	Anastasia Clarke	DE83508900000108208902	Gueltige IBAN aus Deutschland
10	10126	Paula Goff	DE83508900000131579403	Gueltige IBAN aus Deutschland
11	10137	Gonzalo Montgomery	CH0200767000C51001987	Gueltige IBAN aus Schweiz
12	10154	Gerardo Gould	DE93508526510065002305	Gueltige IBAN aus Deutschland
13	10223	P Townsend	PL01987456231564987458216598	Ungueltige IBAN aus Polen
14	10243	Mateo Buckner	IV9142957569845621459	Ungueltige IBAN aus Lettland
15	10246	Sabastian Bach Duke	LI3659875421364852720	Ungueltige IBAN aus Liechtenstein
16	10250	Eleanor Rigby	CH0200781125534343504	Gueltige IBAN aus Schweiz
17	10250	Leslie Stafford	DE91506521240112009816	Gueltige IBAN aus Deutschland
18	10291	Paxton Stanton	DE91506521240112009816	Gueltige IBAN aus Deutschland
19	10307	Quintin Craft		Ungueltige IBAN aus unbekanntes Land

6 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.

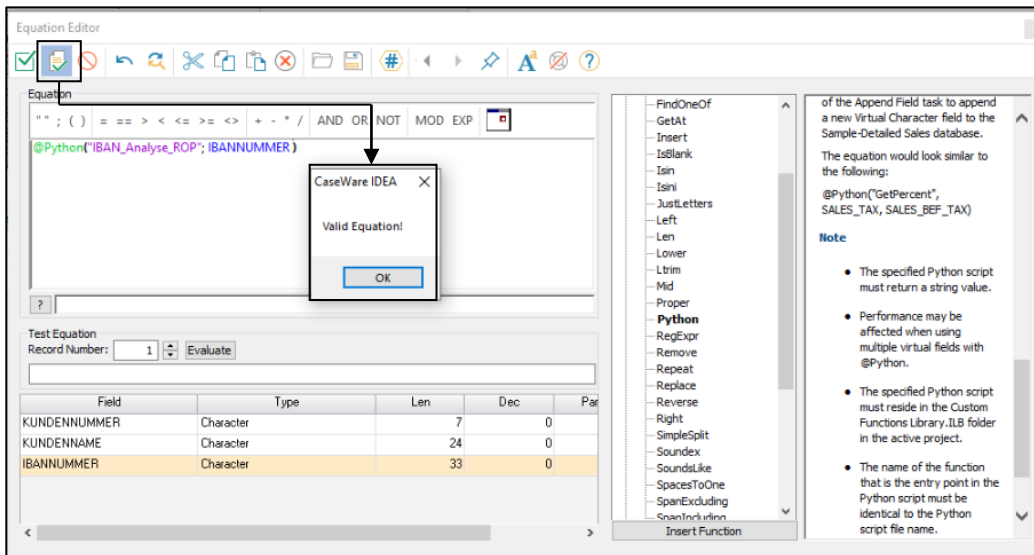
Parameter 2: Feldbezeichnung, Wert oder steuernder Parameter

Üblicherweise enthält der zweite Parameter die Feldbezeichnung aus einer IDEA-Tabelle (z.B. "IBAN_Angabe").

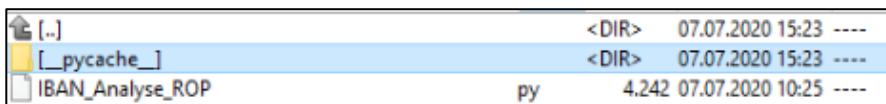
Parameter x: Feldbezeichnung, Wert oder steuernder Parameter

Es *müssen* so viele Parameter in dem Python-Aufruf von IDEA angegeben werden, wie in der Python-Funktion kodiert wurden!

Beispiel: @PYTHON("IBAN_ANALYSE_ROP"; IBANFELD)



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



Das Ergebnis:

IBAN_TEST	KUNDENNUMMER	KUNDENNAME	IBANNUMMER	IBAN_TEST
1	10012	Koby Huber	DE83508900000131579209	Gueltige IBAN aus Deutschland
2	10013	Bryant Singleton	CH0200700110000387896	Gueltige IBAN aus Schweiz
3	10016	Sullivan Rivas	DE93508526510075078568	Gueltige IBAN aus Deutschland
4	10025	Josie Humphrey	DE91506521240017112582	Gueltige IBAN aus Deutschland
5	10052	Kole Long	DE91505922000007211490	Gueltige IBAN aus Deutschland
6	10061	Rylan Gilbert	DE93508900000019011208	Gueltige IBAN aus Deutschland
7	10063	Alec Baldwin III	LI2608802001003488101	Gueltige IBAN aus Liechtenstein
8	10088	Liana Nielsen	DE02120300001017387992	Gueltige IBAN aus Deutschland
9	10098	Anastasia Clarke	DE83508900000108208902	Gueltige IBAN aus Deutschland
10	10126	Paula Goff	DE83508900000131579403	Gueltige IBAN aus Deutschland
11	10137	Gonzalo Montgomery	CH0200767000C51001987	Gueltige IBAN aus Schweiz
12	10154	Gerardo Gould	DE93508526510065002305	Gueltige IBAN aus Deutschland
13	10223	P Townsend	PL01987456231564987458216598	Ungueltige IBAN aus Polen
14	10243	Mateo Buckner	IV9142957569845621459	Ungueltige IBAN aus Lettland
15	10246	Sabastian Bach Duke	LI3659875421364852720	Ungueltige IBAN aus Liechtenstein
16	10250	Eleanor Rigby	CH0200781125534343504	Gueltige IBAN aus Schweiz
17	10250	Leslie Stafford	DE91506521240112009816	Gueltige IBAN aus Deutschland
18	10291	Paxton Stanton	DE91506521240112009816	Gueltige IBAN aus Deutschland
19	10307	Quintin Craft		Ungueltige IBAN aus unbekanntes Land

6 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.

8 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