

Carl Valentin GmbH
 Neckarstraße 78-86
 D-78056 Villingen-Schwenningen

☎ 0 77 20 / 97 12-0
 Fax: 0 77 20 / 99 49 87-5



Projekt: **NiceLabel Automation – Eingabe am Drucker**

Thema: **Automation Printer Control**

Autor: Klaus Weber

Datum: 25. Nov. 2021

Inhalt

Eingaben an einem Drucker	3
Prinzipieller Ablauf	3
Installation	4
Desktop-Icons	5
Konfiguration	6
Statusanzeige	7
Datenaustausch	7
Eingabe	7
Zusätzliche Eingaben	8
Zusätzliche Eingabe als Info-Text	9
Ende- oder Fehlermeldung	9
Ausgabe	9
Neustart der Eingabeaufforderung	9
Aktivitätsanzeige	10
Hilfe	11
Startparameter	11
Hinweise	12
Systemvoraussetzungen	12
Reaktionszeiten	12
Automatischer Start als Dienst	12
NiceLabel Automation – PowerForms Suite – LMS	13
Konfiguration - Beispiel	13
Zusammenfassung	22
Browser-Statusanzeige	25
Labelstar Office	27

Cloud-Version	28
---------------------	----

AUTOMATION PRINTER CONTROL

Eingaben an einem Drucker

Automation Printer Control ist ein Tool, das eine spezielle Kommunikation zwischen NiceLabel Automation und aktuellen Valentin-Druckern (SH3) ermöglicht.

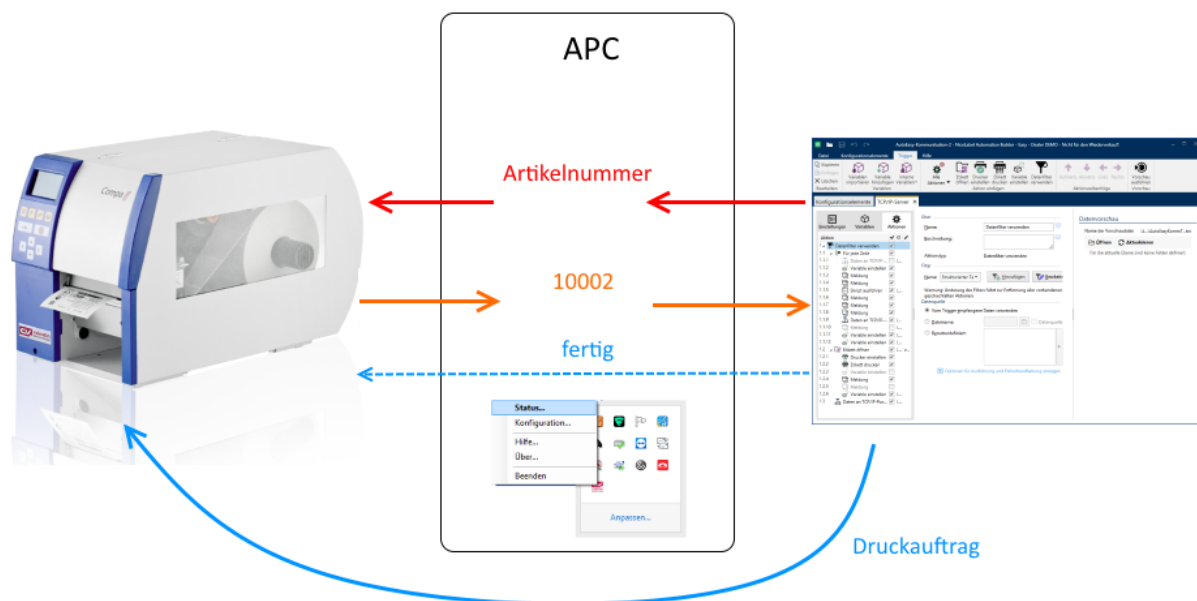
Hierbei werden frei definierbare Eingaben am Drucker abgefragt und die entsprechenden Werte an Automation übertragen.

Prinzipieller Ablauf

APC wird als eigenständiger Dienst auf dem PC (Server) gestartet und ermöglicht die Anbindung 'beliebig' vieler Drucker an Automation. Jede Verbindung zu einem Drucker läuft in einem unabhängigen Thread, so dass eine parallele Verarbeitung gewährleistet ist.

Sobald der APC-Dienst gestartet wird, sucht er nach einem definierten Automation-Port. Automation sendet daraufhin einen Text als Willkommensmeldung beispielsweise *Artikelnummer*.

APC setzt diesen Text in ein spezielles Protokoll um und sendet es an den angeschlossenen Drucker, an dem diese Meldung angezeigt wird. Die eingegebene Nummer wird an APC zurückgesendet, dort entsprechend extrahiert und an Automation weitergeleitet. In Automation wird der Ausdruck generiert und im Normalfall an denselben Drucker gesendet.



HINWEIS!

Es können keine Etiketten verwendet werden, auf denen eine Benutzer-Eingabe definiert wurde.



HINWEIS!

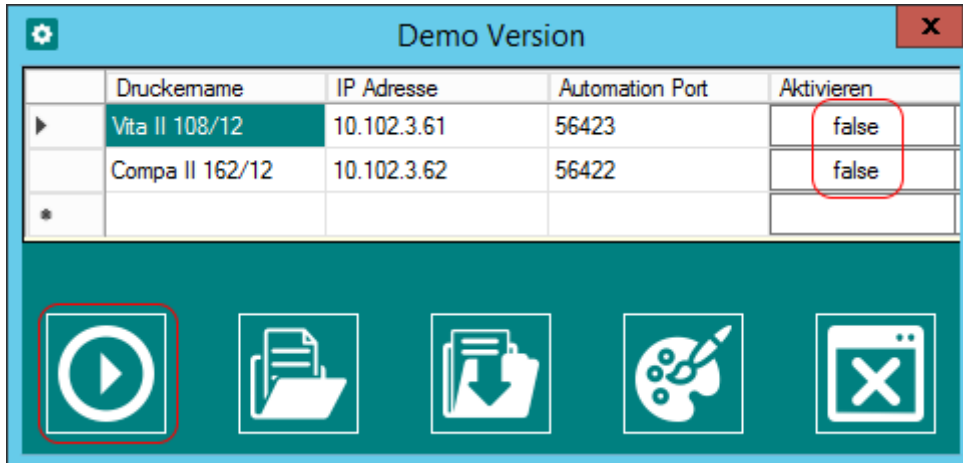
Auch wenn von der Verwendung von NiceLabel Automation ausgegangen wird, wird momentan davon abgeraten, NiceLabel Druckertreiber zu verwenden.

Installation

Das Programm besteht aus zwei Teilen:

- dem Dienst (Service)
- die Statusanzeige inkl. Konfiguration

Nachdem das Programm installiert wurde, wird der Konfigurations-Dialog automatisch geöffnet.



In der Voreinstellung sind zwar zwei Drucker als Beispiel vorhanden, jedoch ist keiner davon freigeschaltet. Somit wird verhindert, dass der Dienst versehentlich ein falsches Gerät beim Start ansteuert.

Hier sind die Drucker einzurichten, freizuschalten (Feld *Aktivieren* auf true) und zu aktivieren.

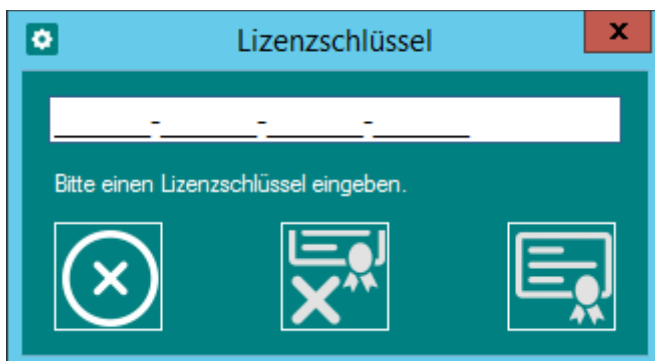
Weil es zu diesem Zeitpunkt noch keine gültige Lizenz gibt, wird diese als nächstes abgefragt.

Ohne Lizenzschlüssel ist nur ein Drucker verwendbar.



Der Dialog kann auch zunächst verlassen werden, um sich erst einmal mit den Beispieldateien vertraut zu machen und anschließend die Konfiguration vorzunehmen. Die Beispiele sind zu finden unter:

Dokumente\Carl Valentin GmbH\Automation Printer Control



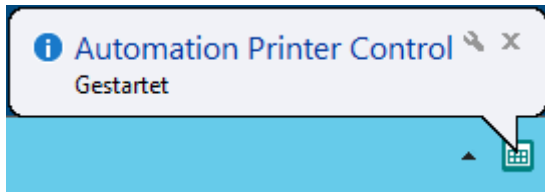
Nach Eingabe des Schlüssels wird der Dienst automatisch neu gestartet, damit die neue Konfiguration und der Lizenzschlüssel aktiviert werden können.

Desktop-Icons



Das Programm für die Statusanzeige wird über das auf dem Desktop neu erzeugte Icon „APC Status“ gestartet.

Hierdurch wird in der Taskleiste ebenfalls ein APC-Icon sichtbar. Per Rechtsklick auf dieses Icon kann die Statusanzeige aktiviert werden.



Bei vielen aktiven Druckern kommen entsprechend viele Benachrichtigungen. Aus diesem Grund können die Task-Benachrichtigungen über das Kontextmenü abgeschaltet werden.



HINWEIS

Für den Betrieb von APC ist nur der Dienst zwingend notwendig. Das Statusprogramm muss nicht unbedingt gestartet werden.

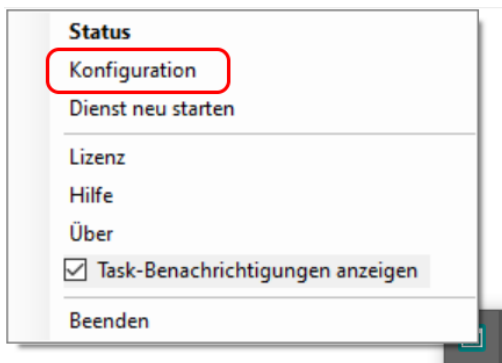


Zur Info:

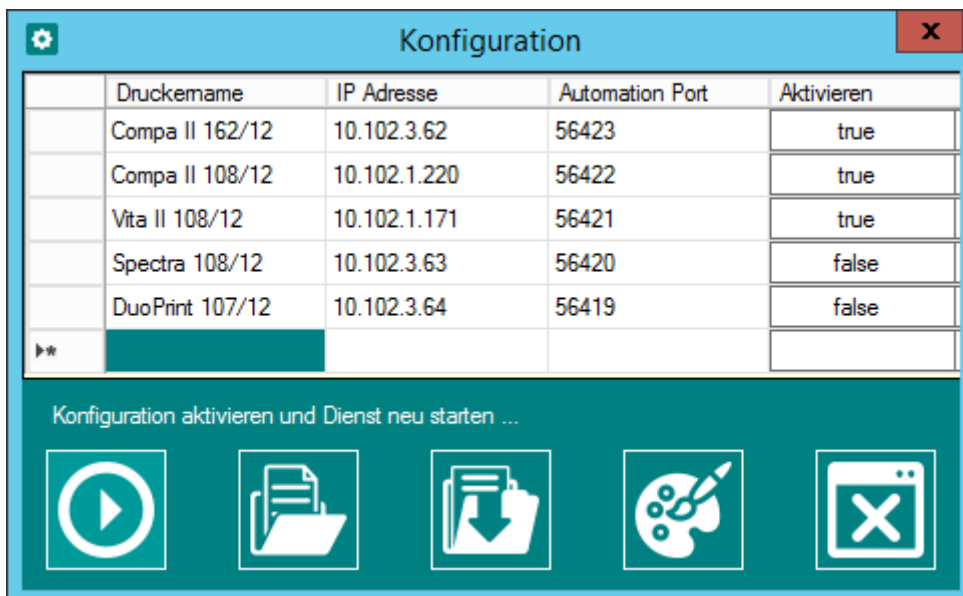
Alternativ zur Statusanzeige steht in der Server-Version noch eine Browseransicht zur Verfügung.

Konfiguration

Die Konfiguration kann auch manuell gestartet werden. Per Rechtsklick auf das APC-Icon in der Taskleiste ist der Menüpunkt *Konfiguration* auszuwählen.



Neue Drucker können hier hinzugefügt werden. Alternativ dazu kann eine bereits bestehende Konfiguration geladen werden. Die neu bearbeitete Konfiguration kann ebenfalls gesichert werden.



Nachdem alle gewünschten Drucker eingetragen wurden, kann diese Konfiguration über den ersten Schalter (Start) aktiviert werden.



HINWEIS!

Die Aktivierung der Konfiguration ist für den Dienst zwingend notwendig. Der Dienst wird in diesem Fall automatisch neu gestartet.

Alternativ zum Konfigurator kann die Konfiguration (xml-Datei) auch mit einem Editor geändert und anschließend der Dienst im Kontextmenü oder im Dienste-Verzeichnis manuell neu gestartet werden.

Die Konfigurationsdatei apc.xml ist zu finden unter:

C:\ProgramData\Carl Valentin GmbH\Automation Printer Control

Statusanzeige

In der Statusanzeige werden alle definierten Drucker angezeigt.

Der Status der einzelnen Drucker wird hierbei in unterschiedlichen Farben dargestellt, so dass sofort ersichtlich ist, welche Drucker gerade verwendet werden und ob Fehler aufgetreten sind.

Statusübersicht der verwendeten Drucker					
	Druckername	IP Adresse	Status	Anzahl Ausdrucke	Letzter Ausdruck
▶	Compa II 162/12	10.102.3.62	#Idle	2	06.07.2018 14:54:13
	Compa II 108/12	10.102.1.220	Artikelnummer	9	06.07.2018 14:53:50
	Vita II 108/12	10.102.1.171	ERROR:Datensatz~60002 falsch	11	06.07.2018 14:53:38
	Spectra 108/12	10.102.3.63	Disabled	0	01.01.0001 00:00:00
	DuoPrint 107/12	10.102.3.64	Disabled	0	01.01.0001 00:00:00



Über den Schalter mit dem Zahnradsymbol kann die Anzahl der Ausdrucke auf 0 gesetzt werden.

Mit dem LOG-Symbol kann eine einfache Protokollierung aktiviert werden. Logdateien werden in dem Verzeichnis C:\logfiles abgespeichert.



HINWEIS!

Die Statusmeldung: *Druckauftrag Nr. xxx ausgeführt* zeigt 1 weniger an, als die Anzahl der Ausdrucke. Der Grund hierfür ist die Zählweise ab 0.

Datenaustausch

Das steuernde Programm (z.B. NiceLabel Automation) muss in der Lage sein, auf einem definierten Port an Localhost als Server zu arbeiten. Es muss dafür sorgen, dass Eingangsdaten an APC gesendet und die passenden Werte verarbeitet werden können. Eine Abschlussmeldung wird erwartet und an den Drucker als Rückmeldung gesendet.

Der Datenaustausch erfolgt nach den folgenden Regeln:

Eingabe

Für die Eingabe am Drucker können bis zu 9 Abfragen gleichzeitig definiert werden. Die einzelnen Abfragen müssen durch ein Semikolon voneinander getrennt werden.

Beispiel: Artikelnummer;Lot;Preis

Zu beachten ist, dass die jeweiligen Texte nicht mehr als 16 Zeichen lang sein dürfen.

Alle Texte stehen am Drucker (SH3) direkt zur Verfügung und können ohne Verzögerung nacheinander eingegeben werden.

Vorgabewerte

Um falsche Eingaben zu reduzieren, können Vorgabewerte definiert werden. Diese werden nach dem ~ Symbol angegeben.

Beispiel: Artikelnummer~12345

Der Vorgabewert bestimmt auch gleichzeitig die maximale Anzahl von Zeichen. Gibt man keine neue Nummer ein, wird 12345 zurückgegeben.

Definierte Eingaben

Die Drucker sind in der Lage, falsche Eingaben direkt zu vermeiden. Mit Hilfe folgender Erweiterungen können die Eingaben sicherer gemacht werden:

#NUM	nur Ziffern 0...9
#LNUM	Ziffern oder leer
#SNUM	leer, Ziffern mit Vorzeichen
#ALPHA	Buchstaben
#ALNUM	Buchstaben oder Ziffern
#AUTO	automatische Zuordnung von Buchstaben oder Ziffern
#TRIM	Inhalt bis zum ersten Leerzeichen

Beispiel: Artikelnummer~12345#NUM

Wird z.B. bei der Verwendung von #NUM ein Buchstabe eingegeben, ertönt ein Beep und das Zeichen wird ignoriert.

Eingabemasken

Manchmal beinhalten Seriennummern Sonderzeichen, wie z.B. Striche. Um diese nicht extra eingeben zu müssen, können sie in den Vorgabewert mit integriert werden.

Beispiel: Seriennnummer~12_345-AB

Folgende Sonderzeichen können zur Maskierung verwendet werden:

"," ". " ":" "/" " " " _ " "*" " _ "

Zusätzliche Eingaben

Nach einer ersten Eingabe ist es möglich, weitere Eingaben abzufragen. Diese könnten beispielsweise aus einer Datenbank kommen, die in Abhängigkeit einer Artikelnummer ggf. notwendig sind.

Eine weitere Abfrage beginnt mit dem Befehl ADDREQ: gefolgt von einem Klammerausdruck, der ebenfalls aus mehreren Einzelabfragen bestehen kann.

Beispiel: ADDREQ:{Extra~59,95}

Falls für eine abschließende Verarbeitung der Daten auch die zuvor eingegebenen Daten benötigt werden, sind diese entsprechend anzuhängen.

Beispiel: ADDREQ:{Extra~59,95};Artikelnummer;Lot;Preis

Wobei hier für Artikelnummer, Lot und Preis die jeweils zuvor eingegebenen Werte zu verwenden sind.

Beispiel: ADDREQ:{d;e};Wert_a;Wert_b;Wert_c

Die gesamte Sammlung der Eingaben wird zum Schluss von APC an Automation gesendet.

Ergebnis: Wert_a;Wert_b;Wert_c;Wert_d;Wert_e

Zusätzliche Eingabe als Info-Text

Weitere Eingabetexte können auch lediglich als Information für den Bediener angezeigt werden.

Beispiel: Transferband~xyz einlegen

Als Bestätigung dieser Anzeige liefert APC an Automation die zweite Zeile zurück:

xyz einlegen

Bei der Konfiguration muss dies berücksichtigt werden.

Ende- oder Fehlermeldung

Damit APC dem Drucker (Bediener) eine Rückmeldung anzeigen kann, ist eine abschließende Meldung notwendig.

Fehlermeldung: ERROR: Text Zeile 1~Zeile 2

Abschluss: SUCCESS: Text Zeile 1~Zeile 2

Bei der Abschlussmeldung ist nur der Doppelpunkt : wichtig, egal was davor steht.

Ausgabe

APC sammelt die eingegeben Werte und packt sie hintereinander, getrennt mit einem Semikolon und sendet diese Daten an Automation.

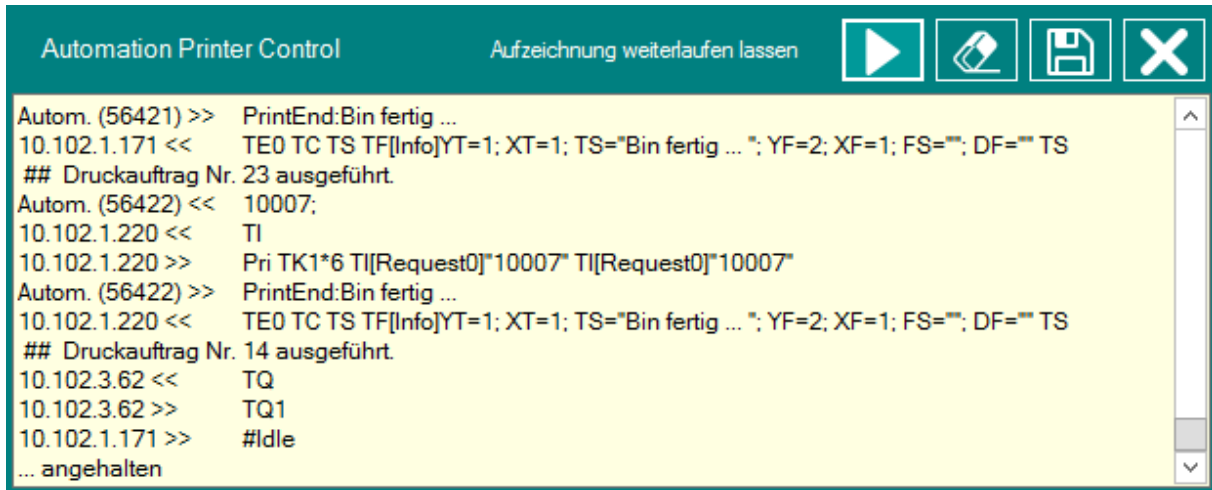
Mehr Rückmeldungen sind für Automation nicht vorgesehen.

Neustart der Eingabeaufforderung

Normalerweise wird nach jedem Druckauftrag eine neue Eingabeaufforderung automatisch gestartet. Sollte diese nicht erscheinen, kann sie über die USB-Tastatur mit F12 angefordert werden. Notfalls muss der Drucker erneut eingeschaltet werden.

Aktivitätsanzeige

Wenn man auf das APC-Icon in der Taskleiste mit der linken Maustaste klickt, wird oberhalb der Taskleiste ein Fenster geöffnet. Hierin wird ein Großteil der Kommunikation zwischen Automation und den Druckern angezeigt. Dieser Dialog dient lediglich der Anzeige, dass das Programm arbeitet und zum Debugging.



Das Fenster kann entweder über den Schließen-Schalter oder über einen nochmaligen Links-Klick auf das APC-Icon wieder geschlossen werden.

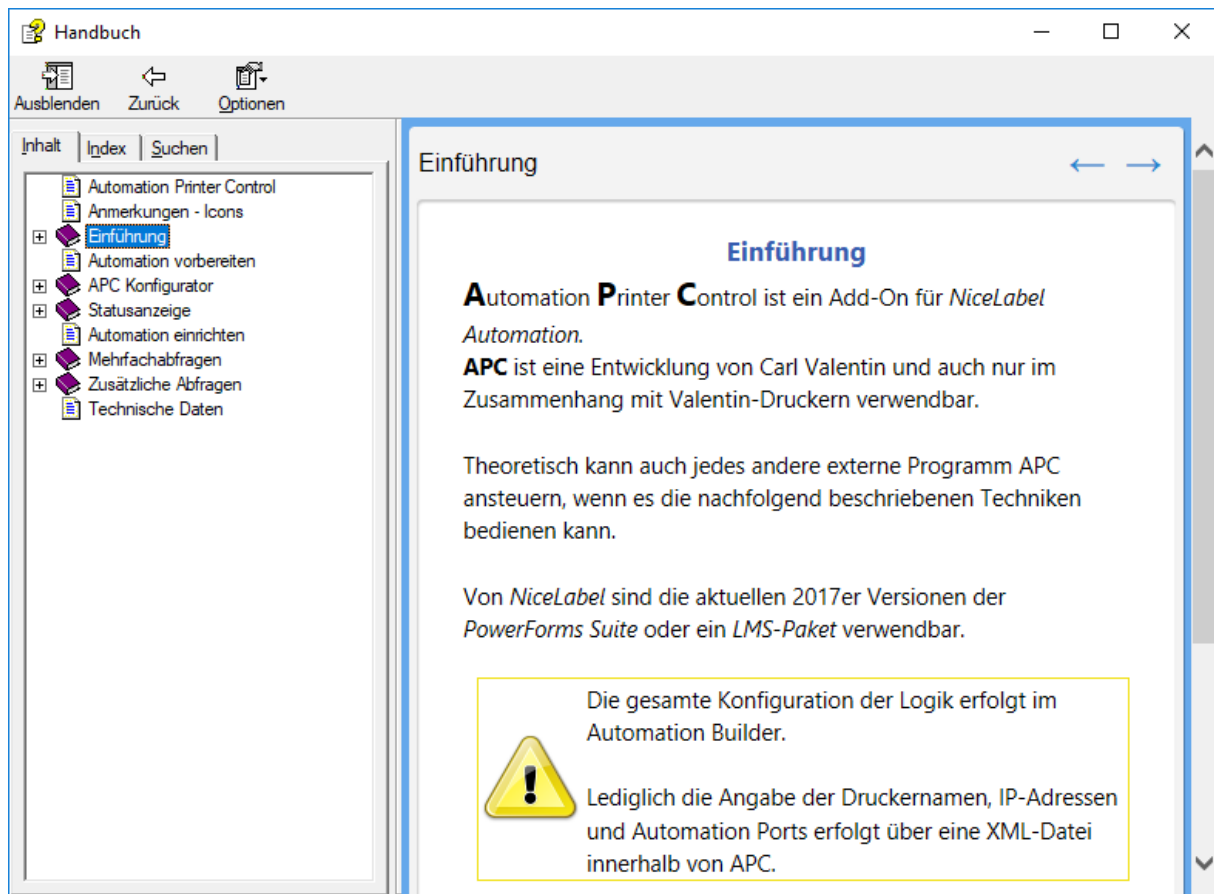


HINWEIS!

Bei dauerhaft geöffnetem Fenster steigt der Speicherverbrauch entsprechend des Inhaltes des Fensters.

Hilfe

Über das APC-Icon in der Taskleiste kann im Kontextmenü mit dem Menüeintrag *Hilfe* die Hilfe-Datei geöffnet werden. In der Hilfe sind weitere Informationen zum Programm vorhanden.



Startparameter

Die einzelnen Dialoge lassen sich mit Hilfe von Parametern auch direkt starten.

Beispiel: APC_Status.exe -konfig

<i>Parameter</i>	<i>Dialog</i>
-konfig	Konfiguration
-status	Statusanzeige
-komm	Übertragene Befehle



HINWEIS!

Nur die dick dargestellten Bereiche werden vom Programm ausgewertet.

Hinweise

- Die Zuordnung zwischen den angelegten Druckern und den in Automation definierten Ports muss zusammenpassen. Wird eine Portnummer geändert, muss dies auch in der APC-Konfiguration erfolgen. Der Dienst muss entsprechend neu gestartet werden.
- Die Browser-Version der Statusanzeige kann nur im Zusammenhang mit einem eingerichteten IIS erfolgen.
- Die entsprechenden Ports für die Verbindungen zu APC müssen in der Firewall ggf. freigeschaltet werden. APC selbst verwendet die lokalen Ports **8733** und **8740**.
- Alle vier Minuten wird der Kommunikationskanal zwischen Dienst und Statusanzeige für jeweils eine Sekunde angehalten und neu gestartet, damit belegte Ressourcen freigegeben werden können. In dieser kurzen Pause kann es zu einer leichten Verzögerung in der Anzeige kommen.

Systemvoraussetzungen

- Windows 7 SP1 bzw. Windows Server 2008 R2 oder neuer
- .NET Framework 4.5
- Bei der Serverversion: Aktiver IIS (Microsoft Internet Information Services)

Reaktionszeiten

Die gestarteten Threads arbeiten parallel und sind somit abhängig vom System. D.h. das Programm kann nicht gewährleisten, dass Reaktionen auf Eingaben in Echtzeit ablaufen.

Verbindungen zu neu eingeschalteten Druckern erfolgen mit max. drei Sekunden Timeout und einer Sekunde Reaktionszeit für andere Prozesse.

Die Statusanzeige wird im Abstand von 500 Millisekunden aktualisiert.

Automatischer Start als Dienst

Der Dienst wird normalerweise auch ohne Anmeldung am System gestartet (getestet auf einem Server 2012 R2).

Es kann sinnvoll sein, für den APC-Dienst eine Abhängigkeit einzurichten, so dass z.B. zuerst die Dienste von NiceLabel-Automation gestartet werden und erst danach der APC-Dienst.

Ohne diese Abhängigkeit kann es sein, dass der APC-Dienst zuerst gestartet wird. Ein bereits angeschlossener Drucker zeigt daraufhin in seiner Anzeige an:

```
Verbinde Server  
Bitte warten...
```

Diese Anzeige erscheint so lange, bis der Automation-Dienst ebenfalls läuft. Erst danach erscheint die normale Eingabeaufforderung, z.B. Artikelnummer. Dieser Vorgang kann einige Minuten dauern.

NiceLabel Automation – PowerForms Suite – LMS



WICHTIG!

Wurde eines der beiden NiceLabel Software-Pakete als Update neu installiert, kann es sein, dass sich der *Builder* nicht starten lässt.

Abhilfe: folgenden Pfad öffnen

%AppData%\..\Local\NiceLabel\NiceLabel xxx

Vorhandene .config-Dateien löschen. Dieses Problem kann immer wieder auftreten.



WICHTIG!

In Automation sind leider verschiedene Bugs vorhanden. Beispiel: Kann in einem Skript (SQL-Server) der Server nicht erreicht werden, wird die Port-Nummer für die Rückmeldungen auf ungültig gesetzt. Weiterhin sind verschiedene Fehlermeldungen irreführend und lenken vom eigentlichen Problem ab.

Abhilfe: Keine

APC ist als AddOn für NiceLabel Automation gedacht. Außer der Konfiguration ist keine extra Einrichtung oder zusätzliche Trigger notwendig.

APC erweitert somit einen TCP/IP-Trigger zu einem Drucker-Trigger.

Sobald die entsprechenden NiceLabel- und APC-Dienste laufen, arbeiten die Programme automatisch zusammen.

Konfiguration - Beispiel

Es ist empfehlenswert, für jeden Drucker einen eigenen Trigger in Automation einzurichten.

Im Verzeichnis *Dokumente\Carl Valentin GmbH\Automation Printer Control* ist eine Beispielkonfiguration vorhanden, die nachfolgend ausführlich beschrieben wird.



Die jeweils aktuell mitgelieferten Beispiele können leicht von den Bildern/Beschreibung abweichen.

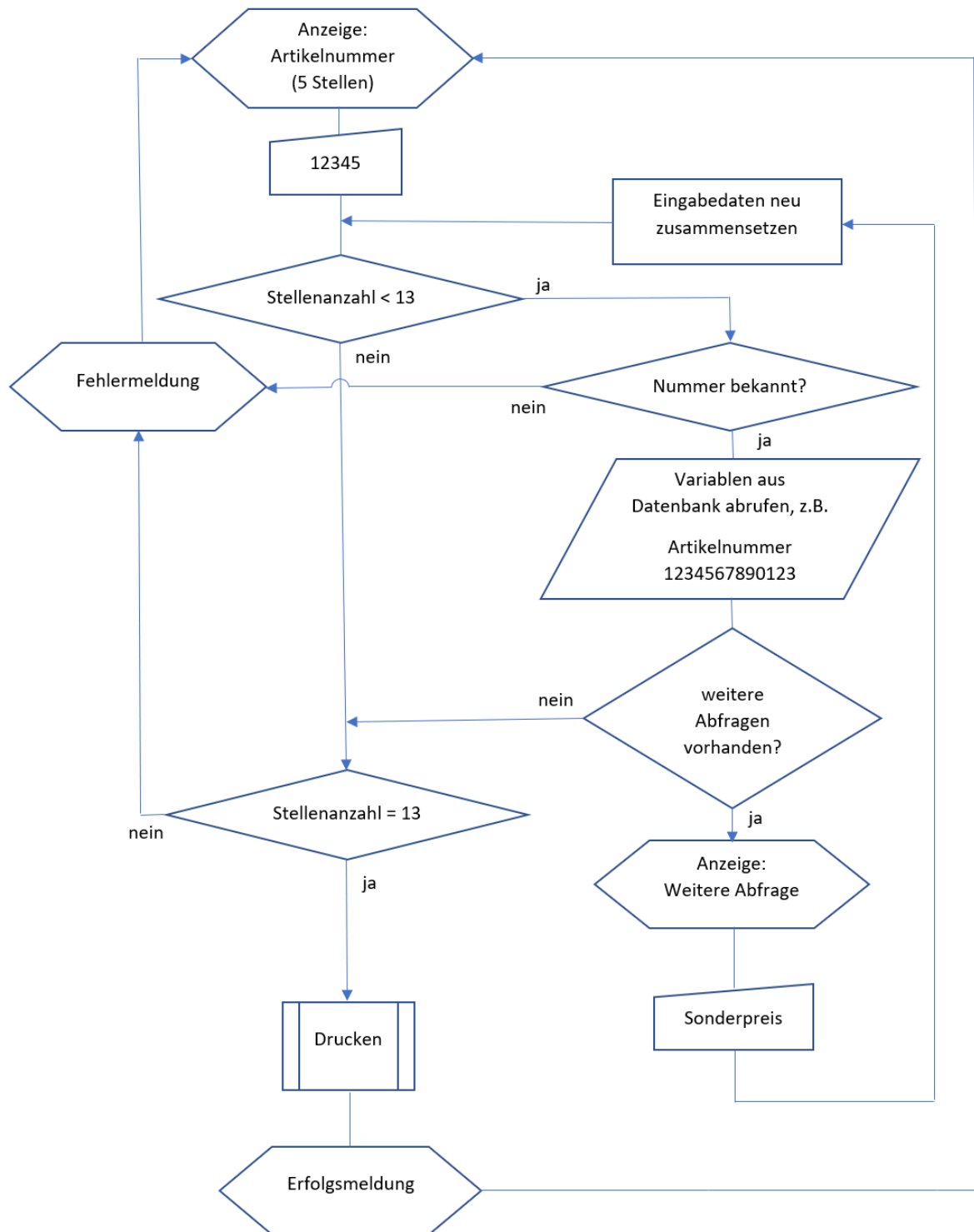
Grundüberlegung

Eine einfache Konfiguration besteht aus einem Port, einem Eingangsfiler, das die eingehenden Daten aufschlüsselt, einer Logik zum Befüllen eines Etikettes und zum Schluss dessen Ausdruck.

So eine einfach strukturierte Darstellung kann als Produktionslinie bezeichnet werden.

Sind mehrere Produktionslinien vorhanden, werden diese einfach kopiert und angepasst. Jede Produktionslinie kann separat betrieben und bei Bedarf (z.B. für Wartungsarbeiten) angehalten werden, ohne die anderen Linien zu beeinflussen.

Flussdiagramm: Prinzipieller Aufbau



Das Beispiel zeigt eine Möglichkeit mit weiteren Abfragen, deren Texte in einer Datenbank definiert sind.

Ohne weitere Abfragen kann der Verzweigungsblock entfallen. Lediglich die Variablensaufbereitung muss vorhanden sein.

Im Block Drucken wird das Etikett geöffnet, mit Daten befüllt, der Drucker ausgewählt und gedruckt.

Diese Konfiguration in NiceLabel Automation

Für die Kommunikation zwischen Automation und APC wird eine lokale Socket-Verbindung verwendet. D.h. in Automation ist ein TCP/IP-Server Trigger einzurichten.

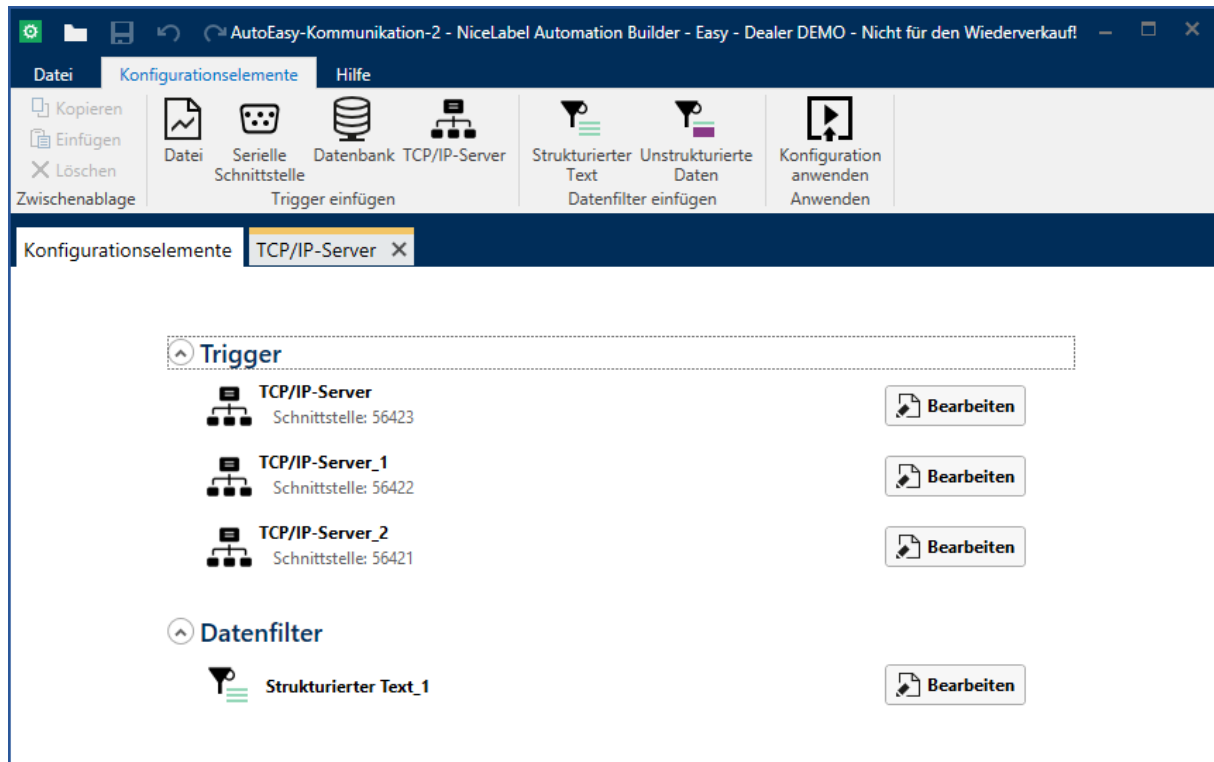


HINWEIS!

Die Beispielkonfiguration funktioniert auch in der Easy-Version von Automation. Der Zugriff auf die verwendete SQL-Datenbank wird daher mit einem Skript erledigt.

Die Verarbeitung der Werte aus der Datenbank wird zum Teil nur zur Veranschaulichung der Funktionsweise verwendet und gehört normalerweise auf das Etikett. Für die Verwendung einer Datenbank-Tabelle direkt auf dem Etikett, ist ein Filter in der Tabellenkonfiguration der Datenbank zu definieren.

Das folgende Beispiel zeigt eine Konfiguration mit drei Linien.



Bei der Entwicklung einer Konfiguration wird die erste Linie bis zur vollen Funktionalität definiert.

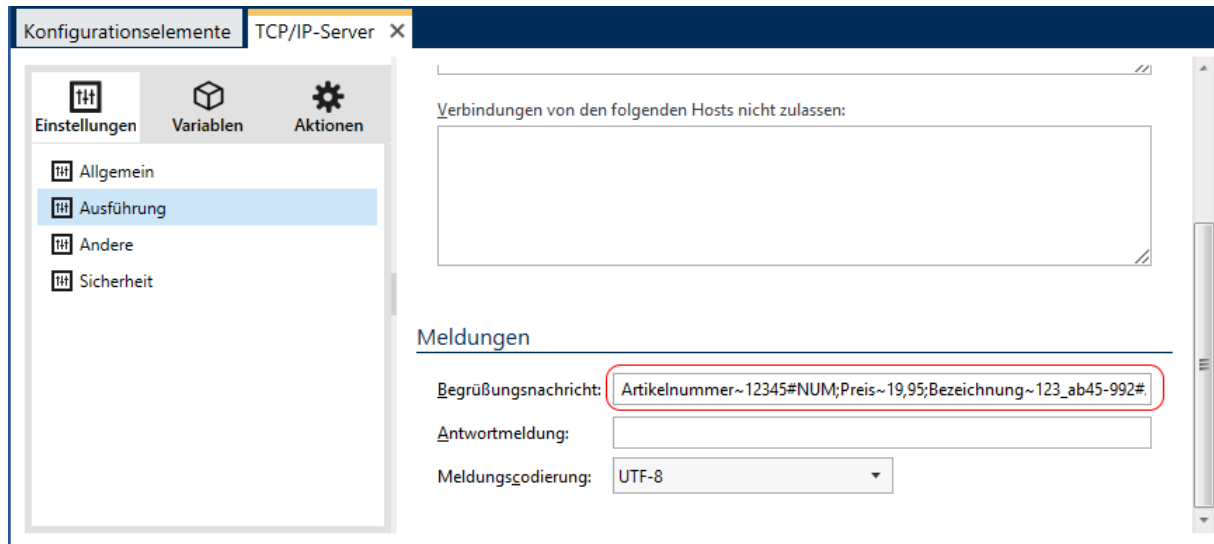
Alle weiteren Linien können über das Kontextmenü (Rechtsklick auf den Trigger) kopiert und eingefügt werden. Bei diesen Kopien müssen lediglich der Port und der Drucker angepasst werden, falls der Rest identisch sein soll.

Über den Schalter *Bearbeiten* gelangt man in den Konfigurationseditor des Automation Builders. Hier kann man die Einstellungen, Variablen und Aktionen definieren.

Begrüßungsmeldung

Die *Begrüßungsnachricht* ist die erste Meldung, die Automation an APC sendet, sobald sich beide Dienste per Socket miteinander verbinden.

Aus dieser *Begrüßungsnachricht* erstellt APC die Abfrage, die an den Drucker gesendet wird.



Artikelnummer~12345#NUM;Preis~19,95;Bezeichnung~123_ab45-992#AUTO

In diesem Beispiel werden am Drucker nacheinander die *Artikelnummer*, der *Preis* und eine *Bezeichnung* abgefragt und dabei jeweils ein Vorgabewert angezeigt.

Ohne eine Änderung am Drucker vorzunehmen, würde er folgende Daten an APC zurücksenden:

12345 19,95 123_ab45-992

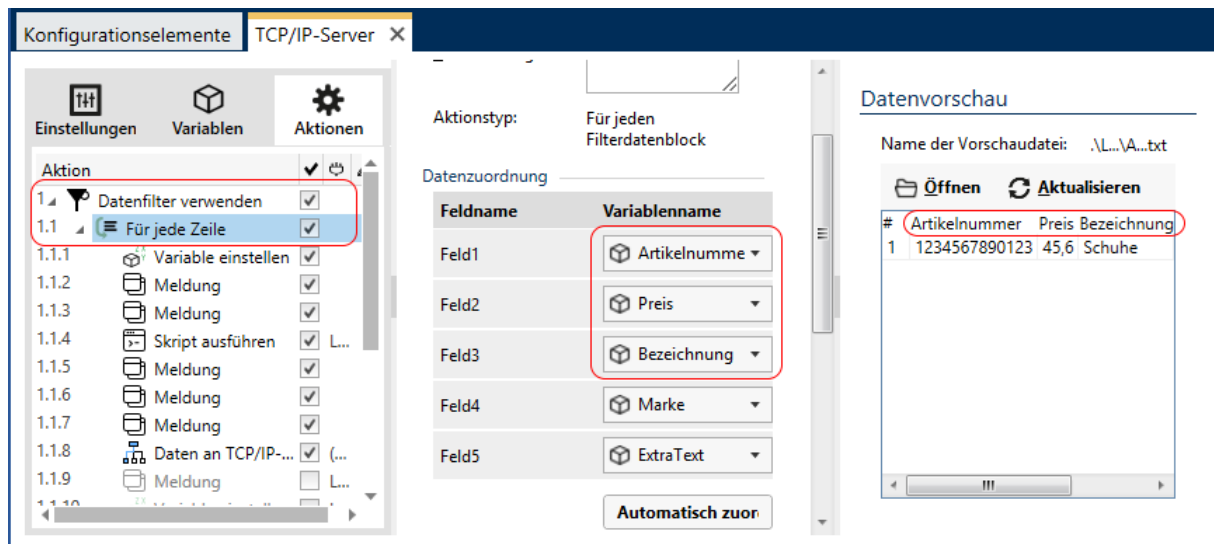
In Automation muss ein Eingangsfilter definiert werden, das diese Werte den entsprechenden Variablen zuordnen kann.

Eingangsfilter

Ein Eingangsfilter besteht aus zwei Aktionen: *Datenfilter verwenden* und *Für jede Zeile*.

Der *Automation Builder* liefert eine einfache Möglichkeit, ein Datenfilter zu importieren. Hierfür wird eine einfache csv-Textdatei erstellt, die in der ersten Zeile die Spaltennamen und in der zweiten Zeile Beispielwerte enthält.

Diese Datei ist in der Datenvorschau zu öffnen. In der Datenzuordnung können passende Variablen automatisch erzeugt werden.



Für jede Zeile bedeutet, dass in der Textdatei auch mehrere Zeilen mit Daten stehen könnten (Beispiel Datei-Trigger).

Es ist wichtig, dass genau in der Reihenfolge, in der die Daten empfangen werden, das Datenfilter zu definieren ist.

Somit erhält man folgende Zurodnung:

Artikelnummer = 12345 Preis = 19,95 Bezeichnung = 123_ab45-992



HINWEIS!

Genau genommen sind hier die Angaben für den Preis und die Bezeichnung überflüssig, weil sie später aus der Datenbank genommen werden. Es soll lediglich die Funktionsweise gezeigt werden.

Bei dem Datenfilter ist darauf zu achten, dass als Trennzeichen ein Semikolon eingetragen ist.

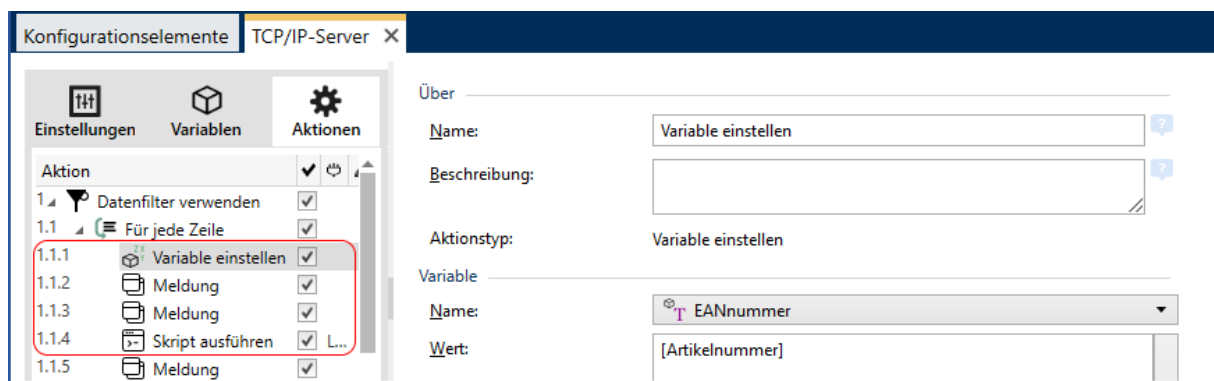
Weitere Verarbeitung

Die Logik der weiteren Verarbeitung kann beliebig sein.

1. In diesem Beispiel wird davon ausgegangen, dass auf dem Etikett eine 13-stellige EAN-Nummer angezeigt wird. D.h. aus der einfach einzugebenden 5-stelligen Artikelnummer soll eine 13-stellige EAN-Nummer werden.
2. Es soll eine Möglichkeit vorhanden sein, ggf. weitere Abfragen (z.B. Sonderpreis) zu generieren oder auch nur Hinweistexte anzuzeigen. Diese Abfragen werden aus einer Datenbank entnommen.

In der Aktion 1.1.1 wird der zuvor definierten Variable *EANnummer* der Wert von *Artikelnummer* zugewiesen. Dies ist eine wichtige Bedingung für die weiteren Aktionen.

EANnummer = 12345



Die Meldungen 1.1.2 usw. dienen lediglich als Debug-Info und können über das rechte Häkchen deaktiviert werden.

VB Skript

Einen wichtigen Beitrag liefert das Skript 1.1.4. In den Pro- oder Enterprise-Versionen können hierfür natürlich direkte SQL-Aktionen genommen werden, die schneller in der Ausführung und leichter zu definieren sind.

Auszug aus dem Skript:

```
sql = "SELECT * FROM Schuhe WHERE Artikelnummer = '" +
      CStr([Artikelnummer].value) + "'"

...
szResult = CStr(rst.Fields("EAN").Value)
EANnummer.Value = szResult
...
szResult = CStr(rst.Fields("Marke").Value)
Marke.Value = szResult
...
If (Len(rst.Fields("ExtraText").Value) > 0) Then
    szResult = CStr(rst.Fields("ExtraText").Value)
    ZusatzAbfrage.Value = szResult
Else
    ZusatzAbfrage.Value = ""
End If
```

Aus der Datenbanktabelle Schuhe wird der Datensatz mit der entsprechenden Artikelnummer genommen. Passend dazu werden die Variablen *EANnummer*, *Marke* und *ZusatzAbfrage* befüllt.

Bedingung: Das Skript wird nur ausgeführt, wenn: `Len([Artikelnummer]) < 13`

D.h. es wird nur im ersten Durchlauf ausgeführt und nicht bei Zusatzabfragen.

Die weiteren Meldungen dienen wieder nur zur Debug-Info.

Weitere Abfrage am Drucker anzeigen

In der Aktion 1.1.8 wird eine weitere Abfrage an APC gesendet, wenn folgendes erfüllt ist.

Bedingung: `(Len([ZusatzAbfrage]) > 0 Or Len([DritteAbfrage]) > 0) And
Len([Artikelnummer]) < 13`

D.h. nur wenn es in der Datenbank Einträge für weitere Abfragen gibt, wird diese Aktion aktiv.

Zusätzliche Abfragen werden mit `ADDREQ:{ [...] ; [...] }` gekennzeichnet. Anschließend werden die bisherigen Eingaben aus der Willkommensabfrage angehängt.

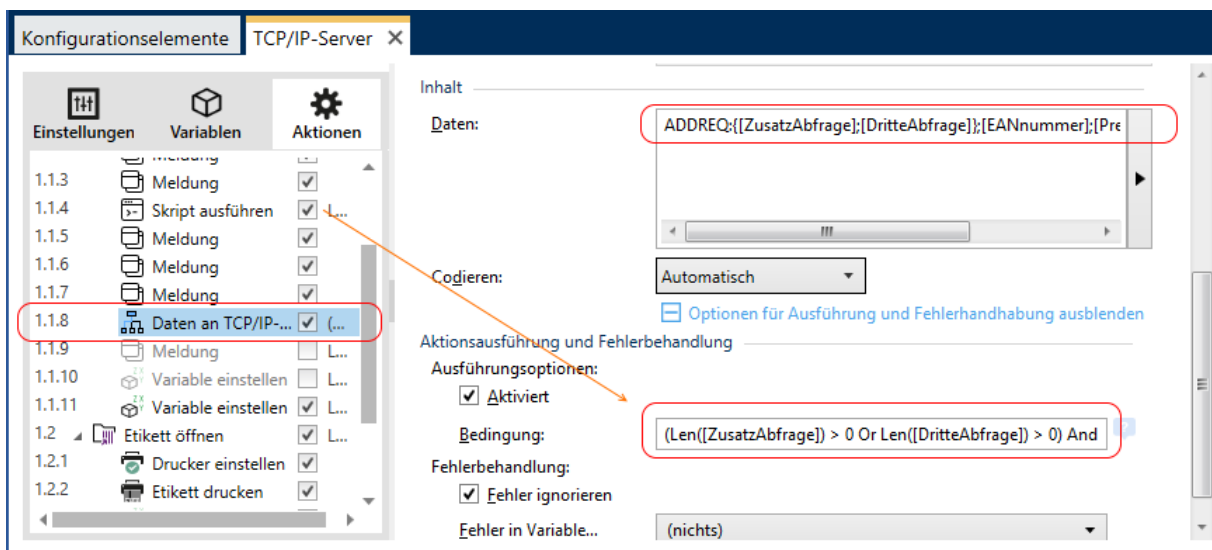
Daten: `ADDREQ:{ [ZusatzAbfrage]; [DritteAbfrage] }; [EANnummer]; [Preis];
[Bezeichnung]; [Marke];`

Damit diese Aktion beim nächsten Durchgang nicht mehr durchlaufen wird, wird anstelle der *Artikelnummer* die *EANnummer* weitergegeben. Die Variable *EANnummer* wurde im Skript mit der richtigen Nummer befüllt.

`Artikelnummer = 10005`

`EANnummer = 4052936045262`

Im folgenden Dialog sind diese Zusammenhänge noch einmal sichtbar:



Aus dem Skript kommen die möglicherweise vorhandenen zusätzlichen Abfragen. Diese werden als Datenpaket verwendet und an den TCP/IP Absender, also APC, gesendet.

Funktionsweise: Zusätzliche Abfrage

Angenommen, in der Datenbank steht „Extra Text“ für eine zusätzliche Abfrage. Als Artikelnummer wird 10005 eingegeben. Somit wird die Variable wie folgt gesetzt:

`ZusatzAbfrage = Extra Text`

In der Druckeranzeige erscheint der Text *Extra Text* in der ersten Zeile.

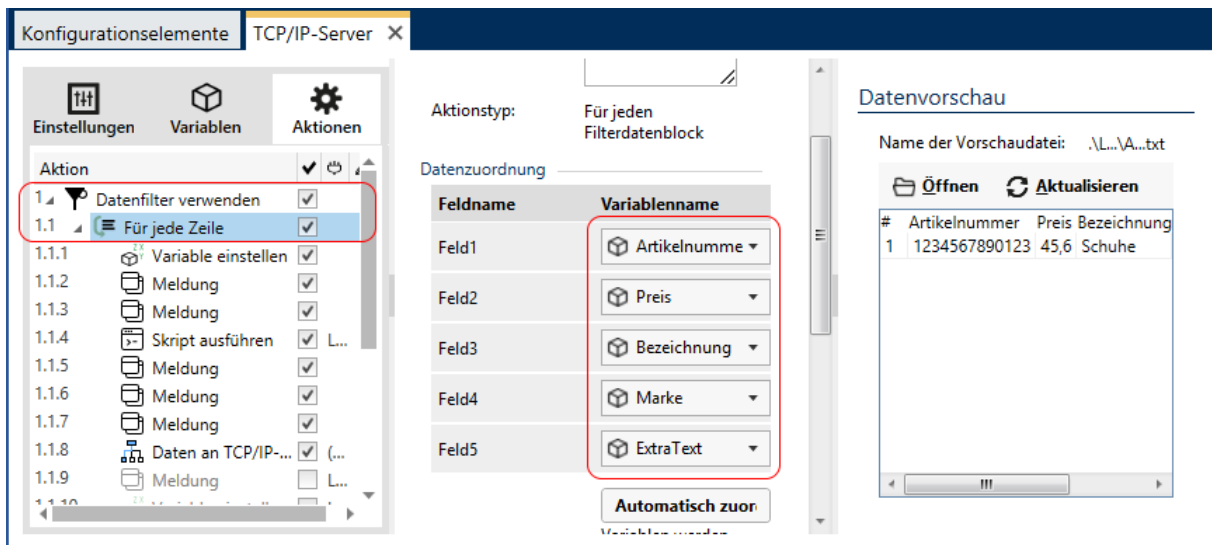
An APC wird übergeben:

`ADDREQ:{Extra Text; []};4052936045262;119,95;Wanderstiefel;Jack Wolfskin;`

Am Drucker wird z.B. der Text 'Sonderangebot' eingegeben.

APC nimmt die neue Eingabe und hängt sie hinten an die übrigen Werte an. Diese neuen Daten werden an Automation gesendet:

4052936045262;119,95;Wanderstiefel;Jack Wolfskin;Sonderangebot;



Bei diesem Durchlauf werden nun alle Variablen aus dem Eingangsfilter mit Daten befüllt.

Artikelnummer = 4052936045262 Preis = 119,95

Bezeichnung = Wanderstiefel Marke = Jack Wolfskin

ExtraText = Sonderangebot

In der Aktion 1.1.1 wird die Variable diesmal mit der richtigen EAN-Nummer befüllt:

EANnummer = 4052936045262

d.h. sie ist nun 13 Stellen lang, womit weitere Aktionen, z.B. das Skript, beeinflusst werden.

Fehlermeldung?

In der Aktion 1.1.11 wird eine Fehlermeldung generiert, falls der Datensatz falsch war.

ERROR:Datensatz~[Artikelnummer] falsch

Bedingung: `Len([Marke]) = 0 And Len([Artikelnummer]) < 13`

D.h. wenn es im Datenbank-Feld Marke keinen Eintrag gegeben hat, muss es eine falsche Artikelnummer gewesen sein.

Druckvorgang starten

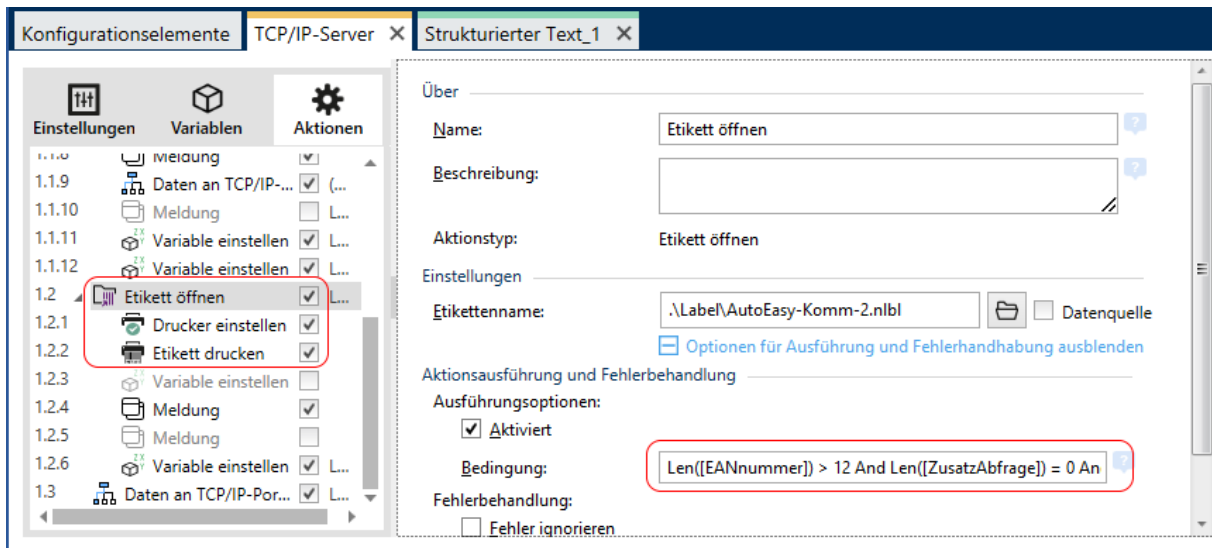
In der Aktion 1.2 wird das Etikett nur unter folgender Bedingung geöffnet:

Bedingung: `Len([EANnummer]) > 12 And Len([ZusatzAbfrage]) = 0 And Len([DritteAbfrage]) = 0`

D.h. nur wenn es eine vollständige EAN-Nummer und keine weiteren Abfragen gibt, wird das Etikett zur weiteren Verarbeitung geöffnet.

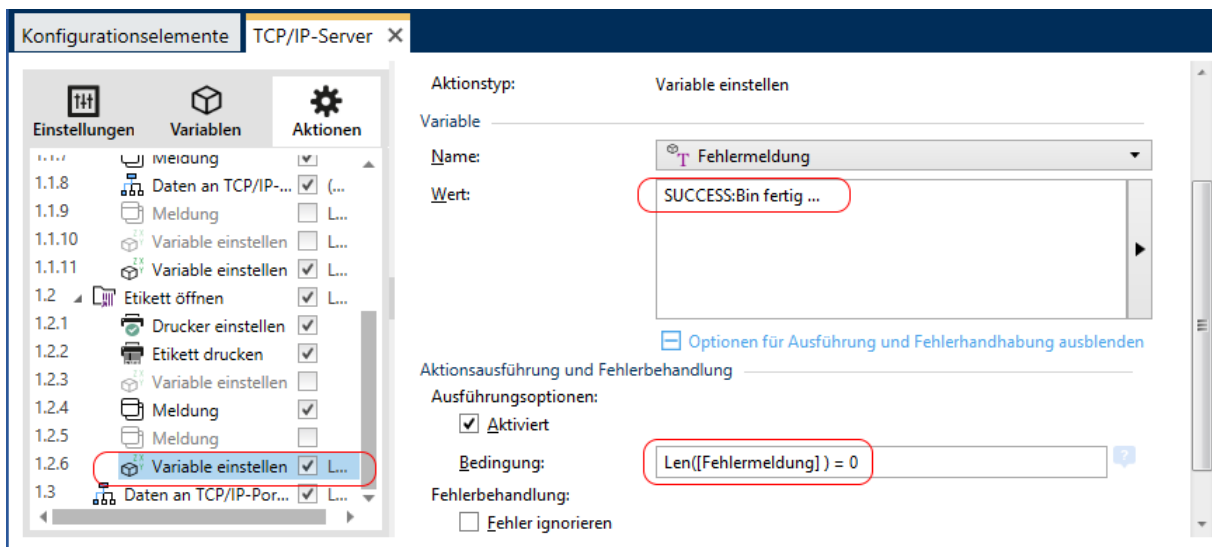
Da die Variable *ZusatzAbfrage* nun leer ist und die Variable *EANnummer* mehr als 12 Zeichen lang ist, wird der Druckvorgang gestartet.

Zuerst wird in der Aktion 1.2 das zu druckende Etikett geöffnet. Danach in 1.2.1 der Drucker ausgewählt und anschließend in 1.2.2 der Druckvorgang gestartet.



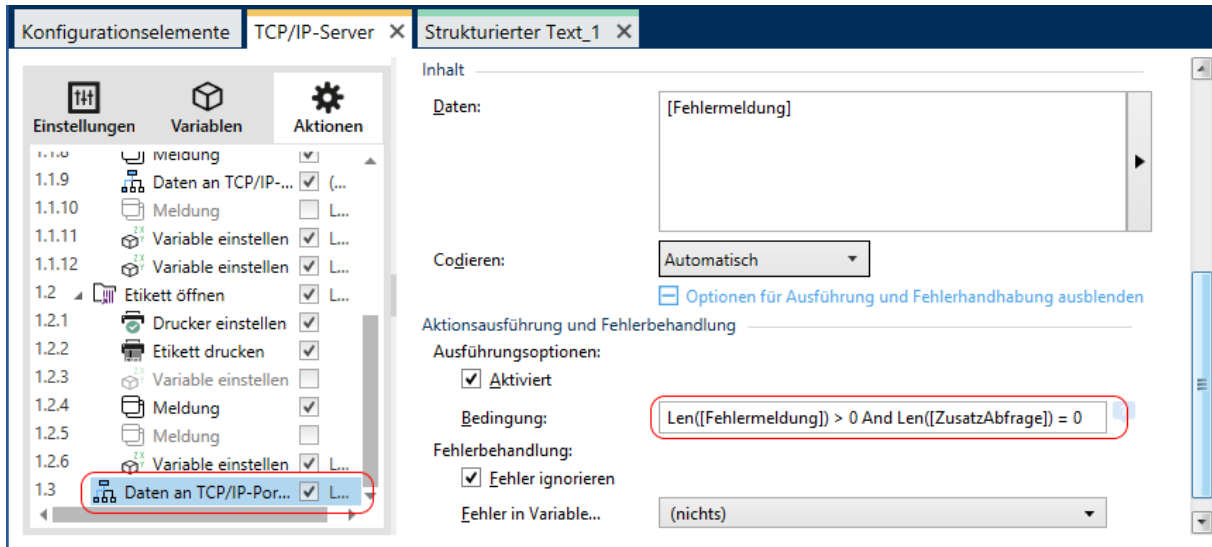
Sollten hierbei Fehler auftreten, werden sie in der Variablen *Fehlermeldung* abgespeichert, ansonsten ist sie leer.

In der Aktion 1.2.6 wird eine Erfolgsmeldung generiert, falls kein Fehler aufgetreten ist.



Abschlussmeldung

Zum Schluss wird noch eine Meldung an APC gesendet, dass der Ausdruck erfolgreich war oder ggf. ein Fehler aufgetreten ist.



Zusammenfassung

D.h. durch ein geschicktes Zusammenspiel von Variablen und deren Inhalten werden verschiedene Aktionen ausgelöst oder gesperrt.

Verschiedene Aktionen lassen sich deaktivieren, so dass sie nur als Hilfestellung (Debug) dienen.

Gerade im Zusammenhang mit einer weiteren Abfrage spielen die Bedingungen eine entscheidende Rolle. Ohne die Verwendung zusätzlicher Variablen und deren abhängige Befüllung mit Daten wird solch ein Ablauf kaum zu realisieren sein.

Ein erster Ansatz ist z.B. die verwendete fünfstellige Artikelnummer, die erst vor dem gewünschten Ausdruck oder für eine zusätzliche Abfrage auf dreizehn Stellen geändert wird. So wird im gezeigten Beispiel das Skript nur im ersten Durchlauf ausgeführt. Das Skript sperrt sich somit selbständig für einen möglichen zweiten Durchlauf.

Ablauf testen

Im *Automation Manager* kann in der Log-Aufzeichnung jede einzelne Aktion auf Fehler überprüft werden.



Fehlersuche – Bugs

Im Automation *Manager* wird farbig angezeigt, wenn ein Fehler aufgetreten ist, weil z.B. die Etikettendatei nicht gefunden wurde. Leider stoppt der Manager die Ausführung, so dass der gesamte Ablauf durcheinandergerät.

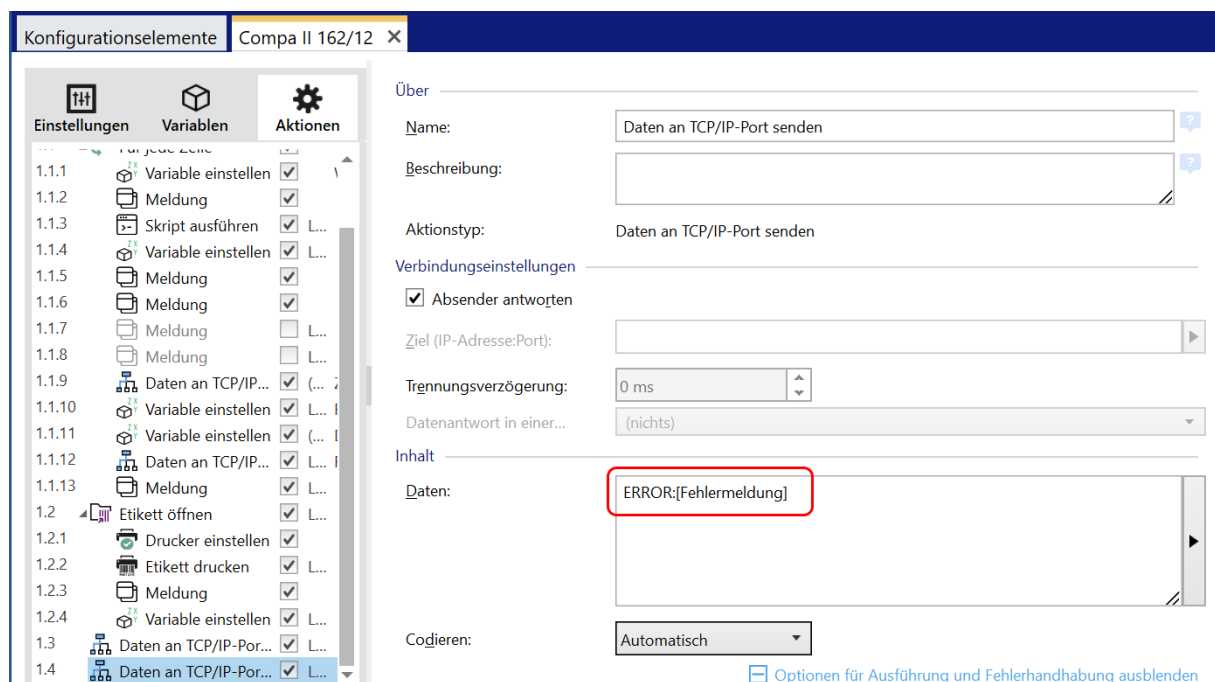
Im Manager kann unter *Log* überprüft werden, bei welcher Aktion ein Fehler aufgetreten ist.

Wird im *Builder* unter Fehlerbehandlung das Häkchen bei *Fehler ignorieren* gesetzt, läuft die Steuerung dagegen weiter. Die erzeugte Fehlermeldung sollte anschließend noch mit „ERROR:“ erweitert werden zu: ERROR:[Fehlermeldung]

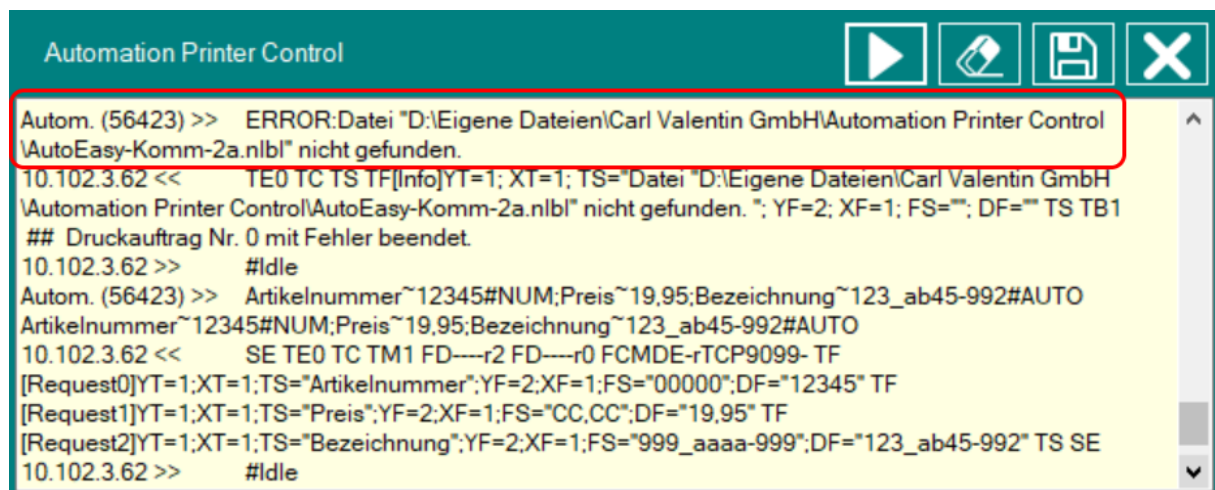
Beispiel:

Die gewünschte Erfolgsmeldung aus der Aktion 1.2.4 lautet „SUCCESS:Bin fertig...“. Diese Meldung wird in der Aktion 1.3 an APC gesendet.

Liegt ein Automation-Fehler vor z.B. „Datei xyz nicht gefunden“, wird in der Aktion 1.4 vor die Fehlermeldung ERROR: gestellt, so dass APC auch die von Automation generierte Fehlermeldung erkennt.



Ist z.B. die angegebene Etikettendatei falsch, würde dies im APC-Aktivitätsfenster angezeigt werden:



WICHTIG: Wird eine Konfiguration neu geladen und gestartet, muss auch der APC-Dienst neu gestartet werden. Geöffnete APC-Dialoge müssen danach ebenfalls neu geöffnet werden.

Zu beachten ist weiterhin, dass nicht jeder Fehler von Automation sauber abgefangen wird. Tritt z.B. im Skript ein Fehler auf, funktioniert die Aktion 1.4 nicht mehr, weil angeblich die Socket-Verbindung unterbrochen wurde. Da es keinen direkten logischen Zusammenhang zwischen den beiden Fehlern gibt, ist dies wahrscheinlich ein Bug in Automation. D.h., wenn APC nicht sauber funktioniert, sollten zuerst die log-Einträge im *Manager* überprüft werden.

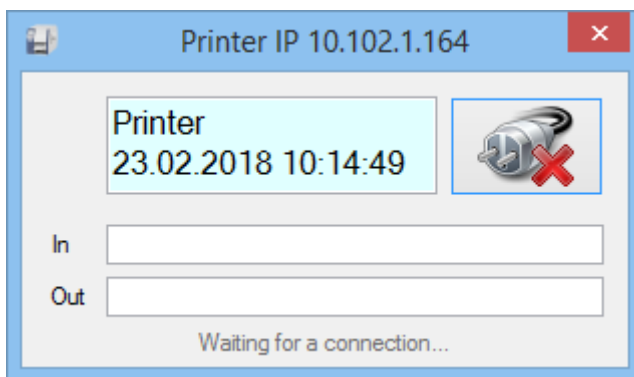
Falls auf dem PC der Pfad **C:\logfiles** existiert, wird dort eine log-Datei **APC-Logfile.txt** abgelegt, die max. 100 kB groß werden kann. Aus dem vorher gezeigten Statusfenster kann eine vollständige Protokollierung abgespeichert werden.

Testumgebung

Im Installationspaket ist eine Automation-Konfiguration sowie eine Access-Datenbanktabelle enthalten, die z.B. in eine SQL-Datenbank importiert werden kann.

In dieser Datenbank werden Artikelnummern von 10001 – 10012 verwendet.

Um mehrere Drucker gleichzeitig testen zu können, kann das ebenfalls mitgelieferte Programm DruckerDummy.exe auf einem weiteren PC verwendet werden. Dieses Tool sendet per Zufallsprinzip korrekte und falsche Artikelnummern an APC. Mehrfachabfragen werden allerdings nicht unterstützt.



Das Programm zeigt im oberen Rahmen (Caption) die IP-Adresse des virtuellen Druckers an. Diese Adresse ist im APC-Konfigurator anzugeben. Die Kommunikation, also die Eingabe der Artikelnummer und Rückmeldungen finden zwischen APC und dem virtuellen Drucker statt.

Da davon auszugehen ist, dass der virtuelle Drucker auch für den eigentlichen Druckvorgang verwendet wird, muss auf dem Automation-PC ein entsprechender Druckertreiber installiert sein. Dessen Port ist auf nul: zu setzen.

Browser-Statusanzeige

Wenn der Status des Servers von einem anderen PC (oder Smartphone) aus angezeigt werden soll, kann die Browseransicht gestartet werden.

Anstelle von nicelabel-srv (bzw. localhost) ist hierbei die Adresse des eigenen Servers anzugeben.

<https://nicelabel-srv:8740/Default.aspx>



HINWEIS!

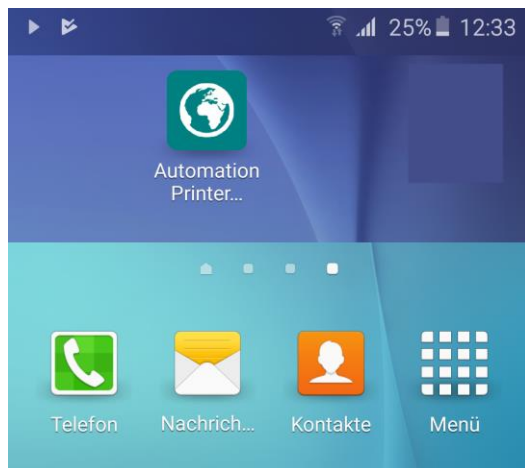
Auf dem Server muss der IIS aktiviert sein.

Druckername	IP-Adresse	Status	# Ausdrücke	Letzter Ausdruck
Compa II 162/12	10.102.3.62	Seriennummer	0	13.07.2018 12:21:19
Compa II 108/12	10.102.1.220	Artikelnummer, Extra Text	38	13.07.2018 12:42:28
Vita II 108/12	10.102.1.171	Offline	0	13.07.2018 12:21:19
Spectra 108/12	10.102.3.63	Disabled	0	01.01.0001 00:00:00
DuoPrint 107/12	10.102.3.64	Disabled	0	01.01.0001 00:00:00

Copyright © 2018 - Carl Valentin GmbH

Auf einem Smartphone ist in einem Browser die Adresse des Servers einzugeben.

Bei einem Android Chrome Browser kann über die drei Punkte neben der Adresszeile ein Kontextmenü geöffnet und der Punkt *Zum Startbildschirm zufügen* gewählt werden. Damit wird ein Icon erstellt, das einen schnellen Zugriff erlaubt:



10.102.1.186:8740/Default.aspx

Automation Printer Control

Druckername	IP-Adresse	Status	# Ausdrücke	Letzter Ausdruck
Compa II 162/12	10.102.3.62	Seriennummer	0	13.07.2018 12:21:19
Compa II 108/12	10.102.1.220	ERROR:Datensatz~60002 falsch	31	13.07.2018 12:39:19
Vita II 108/12	10.102.1.171	Offline	0	13.07.2018 12:21:19
Spectra 108/12	10.102.3.63	Disabled	0	01.01.0001 00:00:00
DuoPrint 107/12	10.102.3.64	Disabled	0	01.01.0001 00:00:00

Labelstar Office

Anstelle der Aktionen 1.2 – 1.2.2 könnte auch ein Skript verwendet werden, mit dem man Labelstar Office Etiketten drucken kann.

```

Set objApp = CreateObject("LSOffice.Application")

' Initialisierung
objApp.Initialize()

' Fehlerbehandlung
If (objApp.HasError) Then
    WScript.Echo objApp.LastError.Message
    WScript.Quit
End If

' Etikettenname und Druckernamen aus Variablen entnehmen und Etikett öffnen
Set objLabel = objApp.OpenLabel(Etikettenname)
' Drucker festlegen
objLabel.ActivePrinter = Druckernamen

' Einem Etikettenobjekt einen Wert zuweisen
Set objField = objLabel.GetFieldByName("ARTNR")
objField.SetContent Artikelnummer

' Alternativ dazu eine Artikelnummer aus einer auf dem Etikett aktiven
Datenbank als Datensatz verwenden
objLabel.SelectRecord (Artikelnummer)

' Etikett n mal ausdrucken
objLabel.Print 1

'Aufräumen
Set objField = Nothing
Set objLabel = Nothing
Set objApp = Nothing

```

Verwendbare Befehle können folgender Beschreibung entnommen werden:

C:\Program Files (x86)\Carl Valentin GmbH\Labelstar Office\Automation\Referenzhandbuch.pdf

z.B. Label-Object – SelectRecord-Methode

Cloud-Version

In den Beispielen ist auch eine Cloud-Konfiguration vorhanden.

In dieser Version wird davon ausgegangen, dass alle notwendigen Dateien in der NiceLabel Cloud im Dokumentenspeicher abgespeichert sind.

Die Dreifachabfrage am Anfang wurde auf die Abfrage der Artikelnummer reduziert. Dafür wurde das Etikett mit der Cloud-Datenbank verbunden und Datenbankfelder direkt verwendet. D.h. alle Inhalte, die zur Laufzeit nicht extra eingegeben werden müssen, kommen aus der Datenbank. Im Prinzip also die Kurzversion der in den vorhergehenden Abschnitten beschriebenen Konfiguration.



WICHTIG!

In der Etikettendatei und im Skript der Automation-Konfiguration sind jeweils die aktuellen Verbindungsdaten der Cloud noch anzupassen.

Im Automation-Manager ist die Cloud-Konfiguration hinzuzufügen und zu starten. APC und Automation arbeiten danach mit den Daten aus der Cloud.