valentin
DRUCKSYSTEME

NICELABEL AUTOMATION

Automation Printer Control

Quality . Tradition . Innovation

Particulars on delivery, appearance, capacity, dimensions and weight reflect our knowledge gained at the time of printing.

Subject to modifications.

Constant development of the devices may be responsible for discrepancies arising between the documentation and the device.
The current version is available under www.carl-valentin.de

Trade marks

All the specified brands and trademarks represent registered brands or trademarks of their owners. They may not be specifically identified. The lack of identification permits to a conclusion as to that it does not involve a registered brand or trademark.

valentin
DRUCKSYSTEME

Carl Valentin GmbH

Postfach 3744
78026 Villingen-Schwenningen
Neckarstraße 78 – 86 u. 94
78056 Villingen-Schwenningen

Phone +49 7720 9712-0
Fax +49 7720 9712-9901

E-Mail info@carl-valentin.de
Internet www.carl-valentin.de

Table of Contents

1 Entries at a printer

Automation **P**rinter **C**ontrol is a tool which permits a special communication between NiceLabel Automation and current Valentin printers (SH 3).

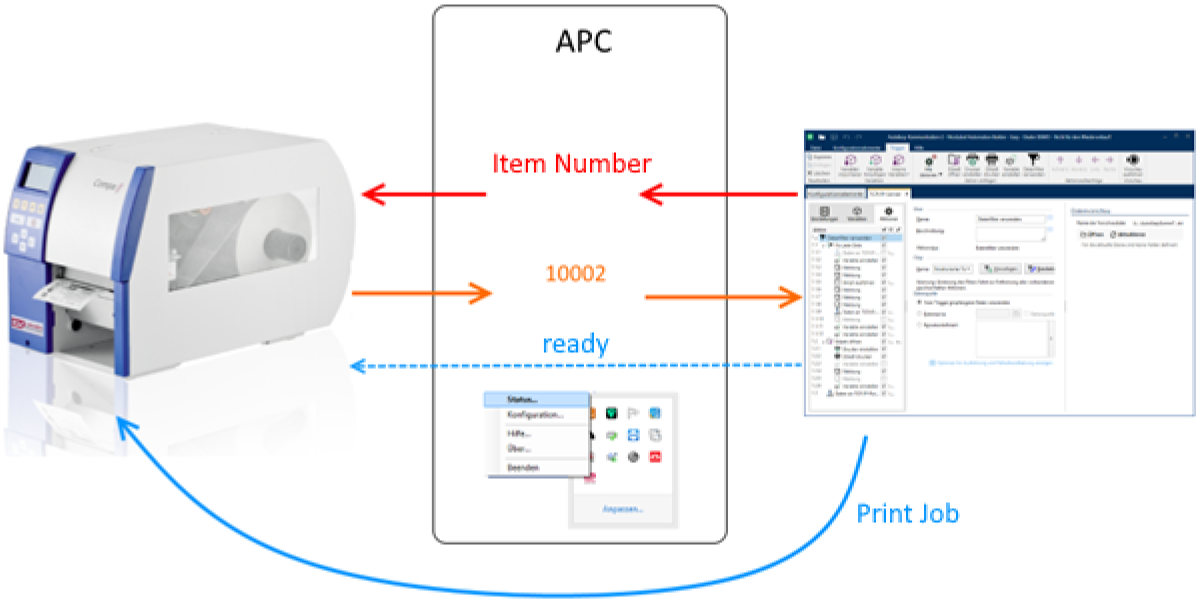
Hereby free-definable entries at the printer were demanded and the appropriate values were transmitted to Automation.

1.1 Principle process

APC is started as independent service on a PC (server) and permits the connection of any number of printers to Automation. Each connection to a printer runs in an independent thread in order to ensure a parallel processing.

As soon as the APC service is started, it looks for a defined Automation port. Thereupon Automation sends a text as a welcome message, e.g. *Item number*.

APC converts this text into a special protocol and sends it to the connected printer at which the message is shown. The entered number is returned to APC, extracted accordingly and passed to Automation. The printout is generated in Automation and under normal conditions send to the same printer.



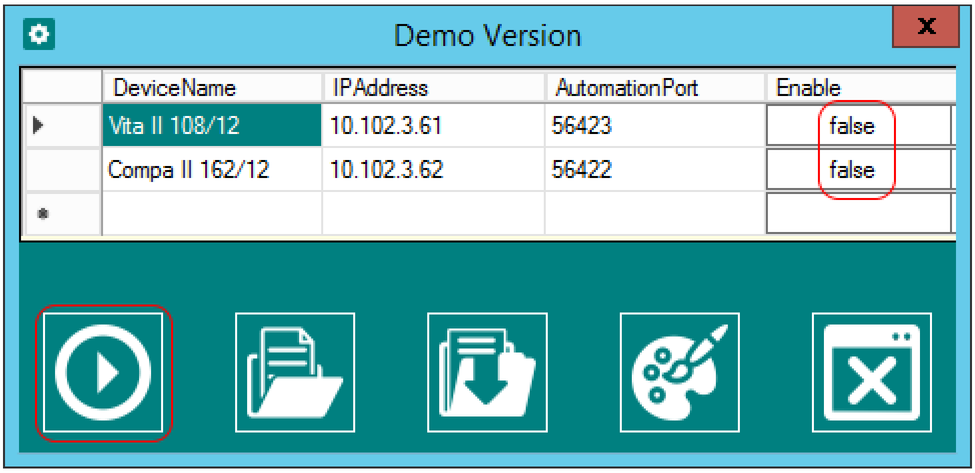
2 Installation

The program consists of two parts:

- service
- status display including configuration

APC-ServerIIS.exe

After the program was installed, the Configuration dialog is opened automatically.



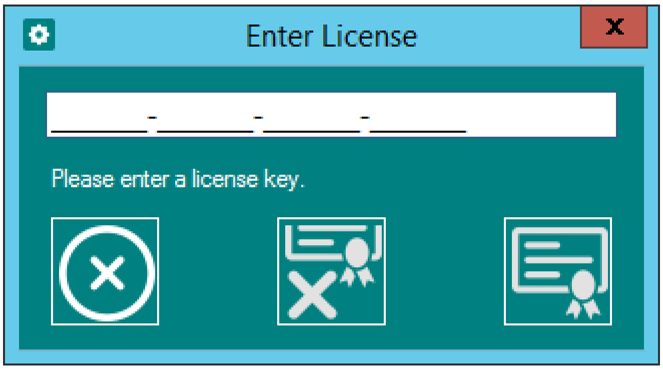
Demo Version

	DeviceName	IPAddress	AutomationPort	Enable
▶	Vita II 108/12	10.102.3.61	56423	false
	Compa II 162/12	10.102.3.62	56422	false
*				

In the pre-setting two printers are available as an example, however none is released. This prevents that the service controls inadvertently wrong equipment when starting.

Setup, release (enable to true) and activate the printers.

Because there is no valid license at this time, this is queried next.



Enter License

_______-_______-_______-_______

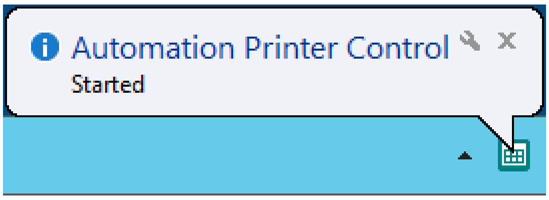
Please enter a license key.

Afterwards the service is started anew automatically, in order to activate the new configuration and the license key.

2.1 Desktop icons

The program for the status display is started by the APC icon newly created on the desktop.

Hereby an APC icon is visible in the task bar. Right-click on this icon to activate the status display.



Automation Printer Control

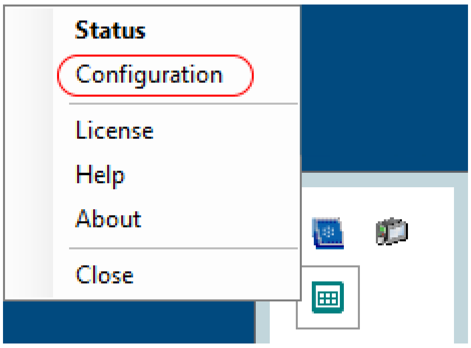
Started

For information:

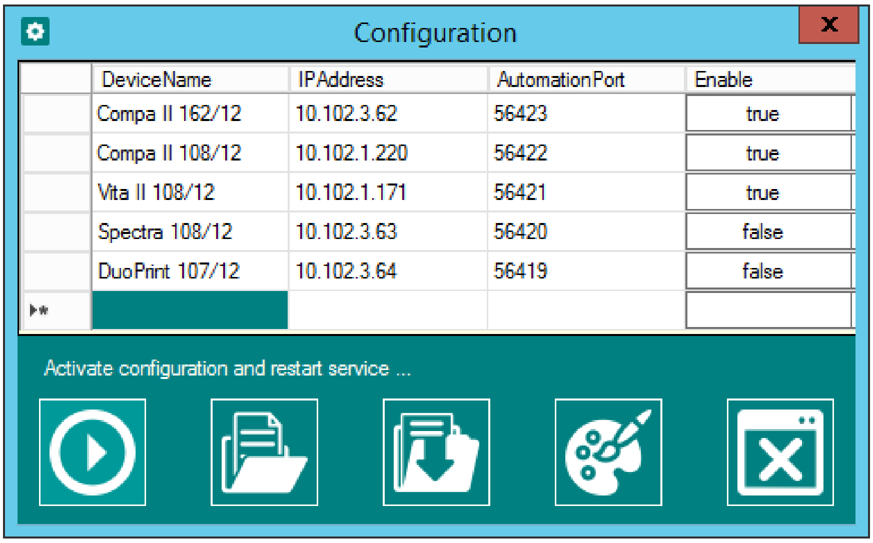
Alternatively to the status display, in the server version a browser view is additionally available.

2.2 Configuration

The configuration can be started manually. Right-click the Valentin icon in the task bar then select the menu item *Configuration*.



New printers can be added manually here. Moreover an already existing configuration can be loaded or the new edited configuration can be saved.



	DeviceName	IPAddress	AutomationPort	Enable
	Compa II 162/12	10.102.3.62	56423	true
	Compa II 108/12	10.102.1.220	56422	true
	Vita II 108/12	10.102.1.171	56421	true
	Spectra 108/12	10.102.3.63	56420	false
	DuoPrint 107/12	10.102.3.64	56419	false
▸*				

Activate configuration and restart service ...

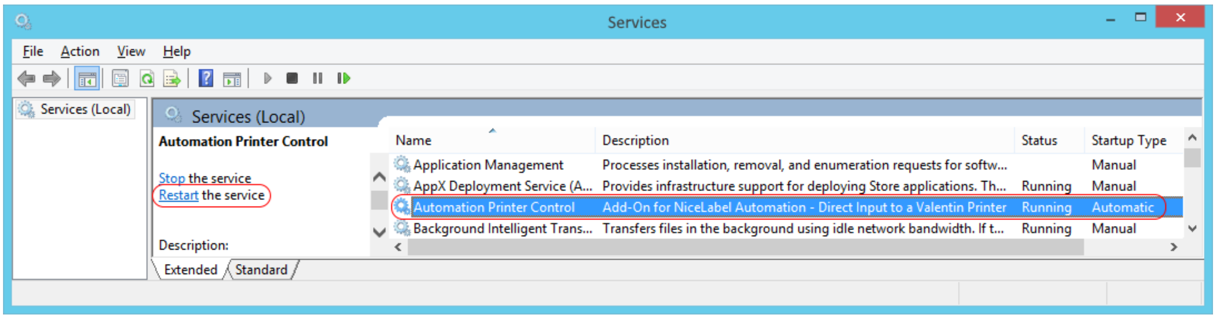
After all desired printers were entered; this configuration can be activated by the first switch (Start).

NOTICE!
The activation is mandatory for the service.

The configuration file *apc.xml* is located under the following path:

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

Alternatively the xml file can be modified in an editor and then the service can be manually started anew in the services directory.



Automation Printer Control

Stop the service
Restart the service

Description:

Name	Description	Status	Startup Type
Application Management	Processes installation, removal, and enumeration requests for softw...		Manual
AppX Deployment Service (A...	Provides infrastructure support for deploying Store applications. Th...	Running	Manual
Automation Printer Control	Add-On for NiceLabel Automation - Direct Input to a Valentin Printer	Running	Automatic
Background Intelligent Trans...	Transfers files in the background using idle network bandwidth. If t...	Running	Manual

3 Data Exchange

The controlling program (e.g. NiceLabel Automation) must be able to work on a defined port at localhost as server. It must ensure that input data can be send to APC and the appropriate values can be processed. A final notice is expected and send to the printer as response.

The data exchange takes place according to the following rules:

3.1 Input

For the input at the printer, 9 queries can be defined at the same time. The individual queries must be separated by a semicolon.

Example: Item number;Lot;Price

Pay attention that the respective texts may not be longer than 16 characters.

All texts are available directly at the printer (SH3) and can be entered without delay successively.

Pre-set values

In order to reduce wrong entries, pre-set values can be defined. These are indicated after the symbol ~.

Example: Item number~12345

The pre-set value determines at the same time the maximum number of characters.

Pre-set values

In order to reduce wrong entries, pre-set values can be defined. These are indicated after the symbol ~.

Example: Item number~12345

The pre-set value determines at the same time the maximum number of characters.

Defined entries

The printers are able to avoid wrong entries directly. With help of the following extensions, the entries can be made safer:

#NUM	only numbers 0...9
#LNUM	numbers or blank
#SNUM	blank, signed numbers
#ALPHA	letters
#ALNUM	letters or numbers
#AUTO	automatic allocation of letters or numbers
#TRIM	contents up to the first blank

Example: Item number~12345#NUM

E.g. when using #NUM and a letter is entered, a beep is audible and the digit is ignored.

Input masks

Sometimes serial numbers contain special characters, e.g. dash. In order to not have to enter them additionally, they can be integrated in the pre-set value.

Example: Serial number~12_345-AB

The following special characters can be used for masking:

```
","    "."    ":"    "/"    " "    "_"    "*"    "-"
```

3.2 Additional Entries

After the first entry it is possible to enquire further entries. They can come e.g. from a database which are probably necessary in dependence of an article number.

A further query begins with the command ADDREQ: followed by a bracket term which can contain of several single queries.

Example: ADDREQ:{Extra~59,95}

If for a concluding processing of the data also the before entered data is needed, these are to be attached accordingly.

Example: ADDREQ:{Extra~59,95};Item number;Lot;Price

For article number, lot and price, the before entered values are to be used.

Example: ADDREQ:{d;e};Value_a;Value_b;Value_c

At the end, the entire collection of inputs is send from APC to Automation.

Result: Value_a;Value_b;Value_c;Value_d;Value_e

NOTICE!

In the current version, an additional query can only be processed once.

3.3 Final Message or Error Message

In order that APC can show a response to the printer (user), a final message is necessary.

Error message: ERROR: Text Line 1~Line 2

Conclusion: SUCCESS: Text Line 1~Line 2

For the final message only the colon is important, regardless what is indicated in front of it.

3.4 Output

APC collects the entered values and packs them one behind the other, separated with a semicolon and sends this data to Automation.

More responses for the application are not intended.

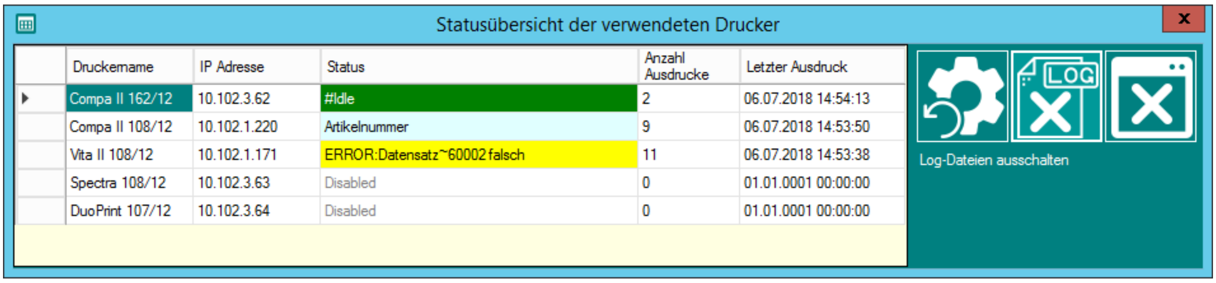
3.5 Restart after the Command Prompt

Normally after each print order, a new command prompt is started automatically. If the command prompt should not appear, use F12 on the USB keyboard to request it. If necessary, the printer must be switched on again.

3.6 Status Display

In the status display all defined printers are shown.

The status of the individual printers is represented in different colors, so that it is immediately visible which printer is in use or if errors occurred.



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

With the switch with the number symbol the number of printouts can be set to 0.

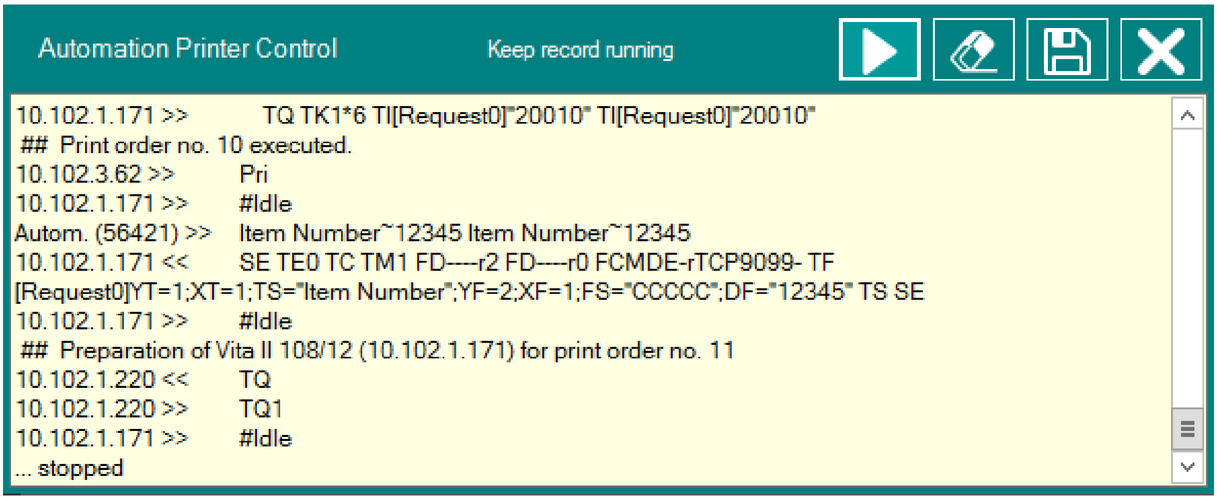
With the time symbol a simple logging can be activated. The log files are saved in the directory *C:\logfiles*.

NOTICE!

The status display: *Print order no. xxx executed* shows 1 less than the number of printouts. The reason for this is the counting method starting from 0.

3.7 Further status window

Click with the left mouse button on the Valentin icon in the task bar to open a window above the task bar. Herein a majority of communication between Automation and printers is shown. This dialog serves only for displaying, that the program works and to debugging.



```
10.102.1.171 >>        TQ TK1*6 TI[Request0]"20010" TI[Request0]"20010"
 ##  Print order no. 10 executed.
10.102.3.62 >>        Pri
10.102.1.171 >>       #Idle
Autom. (56421) >>     Item Number~12345 Item Number~12345
10.102.1.171 <<       SE TE0 TC TM1 FD----r2 FD----r0 FCMDE-rTCP9099- TF
[Request0]YT=1;XT=1;TS="Item Number";YF=2;XF=1;FS="CCCCC";DF="12345" TS SE
10.102.1.171 >>       #Idle
 ##  Preparation of Vita II 108/12 (10.102.1.171) for print order no. 11
10.102.1.220 <<       TQ
10.102.1.220 >>       TQ1
10.102.1.171 >>       #Idle
... stopped
```

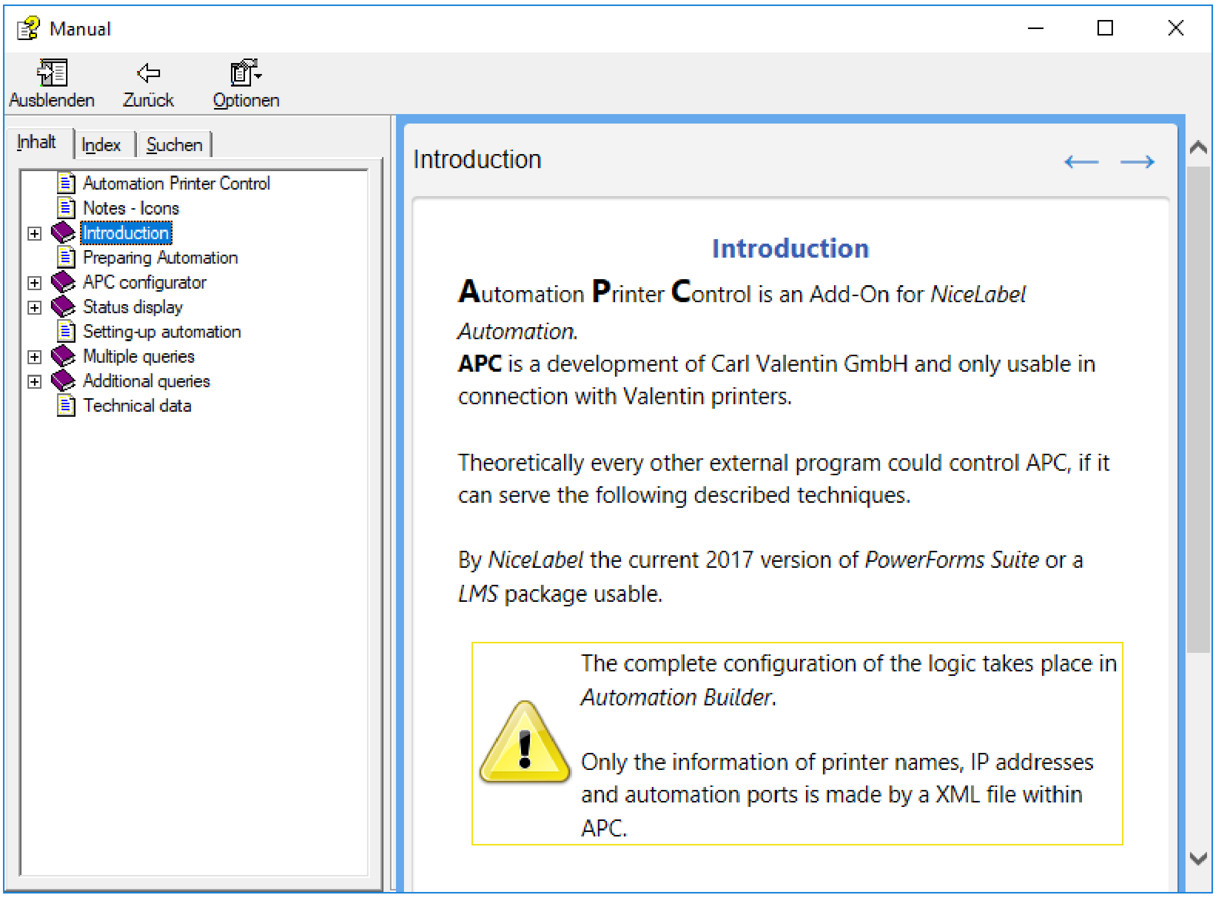
In order to close the window, click the closing switch or left click again the Valentin icon.

NOTICE!

With permanently opened window the memory consumption rises according to the contents of the window.

4 Help

In order to open the help file, click the APC icon in the task bar and select from the context menu *Help*.



Introduction

Automation **P**rinter **C**ontrol is an Add-On for *NiceLabel Automation*.
APC is a development of Carl Valentin GmbH and only usable in connection with Valentin printers.

Theoretically every other external program could control APC, if it can serve the following described techniques.

By *NiceLabel* the current 2017 version of *PowerForms Suite* or a *LMS* package usable.

The complete configuration of the logic takes place in *Automation Builder.*

Only the information of printer names, IP addresses and automation ports is made by a XML file within APC.

5 Start Parameters

The individual dialogs can be started directly with parameters.
Example: APC_Status.exe -config

Parameter	*Dialog*
-**config**	Configuration
-**stat**us	Status display
-**comm**	Transferred commands

NOTICE!
Only the thick represented ranges are evaluated by the program.

6 Notes

- The allocation between the created printers and the ports defined in Automation must fit. If a port number is changed, this must be effected in the APC configuration. The service must be started anew.
- The browser version of the status display can take place only in connection with a created IIS.
- The appropriate ports for the connection to APC must be released (if necessary) in the firewall. APC uses the local ports 8733 and 8740.
- Every four minutes the communication channel between service and status display is stopped for one second each and then restarted, in order to release occupied resources. In this short break, there might be a delay in the display.

7 System Requirements

- Windows 7 SP2 and/or Windows Server 2008 R2 or newer
- .NET Framework 4.5
- For server version: Active IIS (Microsoft Internet Information Services)

8 Reaction Times

The started threads work parallel and are dependent from the system. I.e. the program cannot ensure that reactions on inputs run in real time.

Connections to new created printers take place with max. three seconds timeout and one seconds reaction time for other processes.

The status display is updated at a distance of 500 milliseconds.

9 Automatic Start as Service

Normally the service is started without registration at the service (tested on a server 2012 R2).

It can be meaningful to install dependence for the APC service so that e.g. first the NiceLabel Automation services are started and then the APC service.

Without this dependence, the APC service could be started at first. An already connected printer shows in its display:

```
Connect server
Please wait…
```

This display appears as long as the automation service runs equally. Only after that the normal input prompt appears, e.g. article number. This procedure can take some minutes.

10 NiceLabel Automation – PowerForms Suite – LMS

APC is intended as AddOn for NiceLabel Automation. Except the configuration no extra set-up or additional trigger is necessary.

APC extends a TCP/IP trigger to a printer trigger.

As long as the corresponding NiceLabel services and APC services run, the programs work together automatically.

10.1 Configuration

It is surely possible to query by the input of a printer ID, several printers by an unique Automation port. On the basis of this ID, the configuration must ensure that the printout is printed on a specific printer.

That much interleaved configurations are a challenge but also error-prone.

An example configuration is in the following directory: *Documents\Carl Valentin GmbH\Automation Printer Control*, which is described in detail in the following.

Basic consideration

A single configuration consists of a port, an input filter which estimates the incoming data, logic for filling the label and its printout.

Such a simple structured representation can be called production line.

If there are several production lines, these are simply copied and adapted. Each production line can be operated separately and stopped if necessary (e.g. maintenance), without affecting the other lines.

A simple configuration

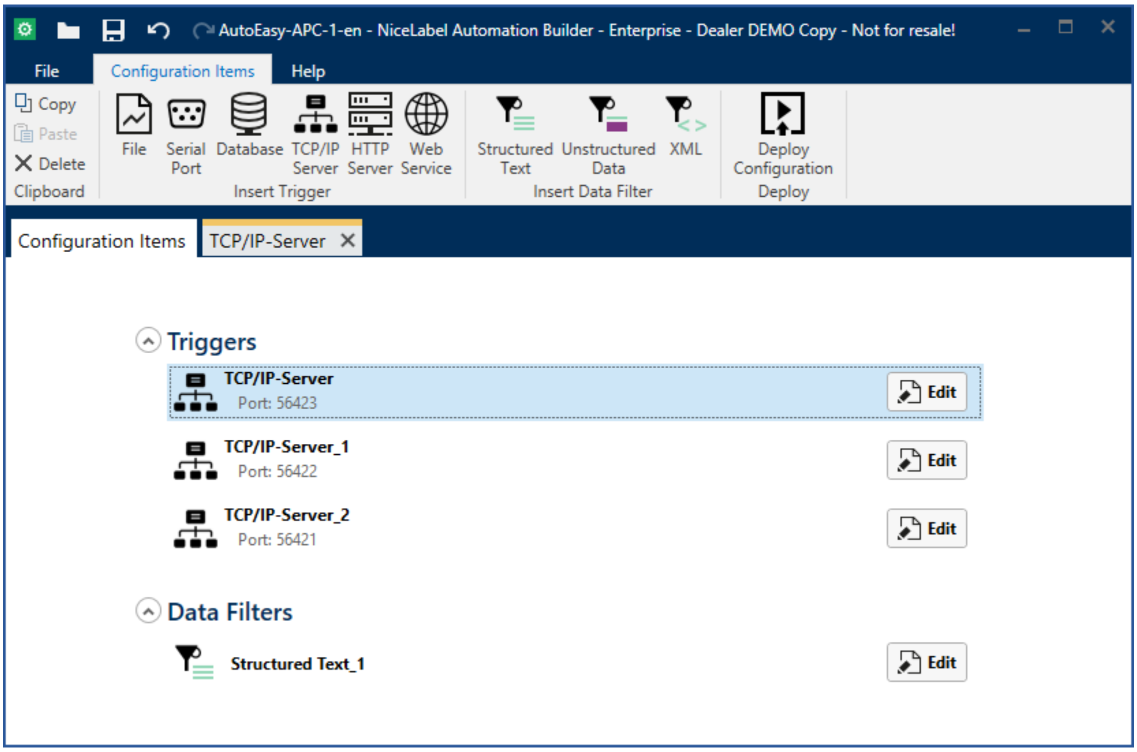
For the communication between Automation and APC, a local socket connection is used. I.e. in Automation a TCP/IP server trigger is to be set.

NOTICE!

The example configuration works also in the Easy version of Automation. The access to the used SQL database is handled with a script.

The processing of values from the database is used partially as an illustration of the function method and normally belongs on the label.

The following example shows a configuration with three lines.



Triggers

- TCP/IP-Server, Port: 56423
- TCP/IP-Server_1, Port: 56422
- TCP/IP-Server_2, Port: 56421

Data Filters

- Structured Text_1

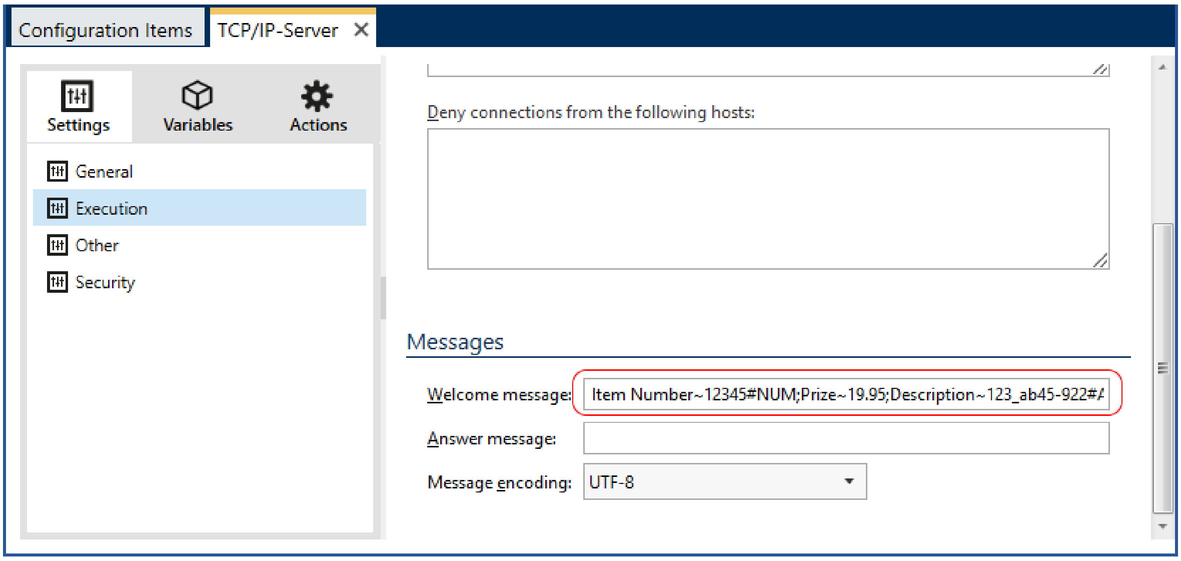
With each line only the port and the printer must be adapted, if the rest should be identical.

With the switch *Edit* it is possible to arrive to the configuration editor of Automation Builder. Here settings, variables and actions can be defined.

Welcome message

The *Welcome message* is the first message that Automation sends to APC as soon as both programs are connected per socket.

From the *Welcome message*, APC creates the query which is send to the printer.



Deny connections from the following hosts:

Messages

Welcome message: Item Number~12345#NUM;Prize~19.95;Description~123_ab45-922#

Answer message:

Message encoding: UTF-8

```
Item Number~12345#NUM;Price~19,95;Description~123_ab45-992#AUTO
```

In this example, the *article number*, *price* and *description* are queried at the printer one after the other, and a pre-set value is shown in each case.

Without making a change at the printer, the printer would return the following data to APC:

12345 19,95 123_ab45-992

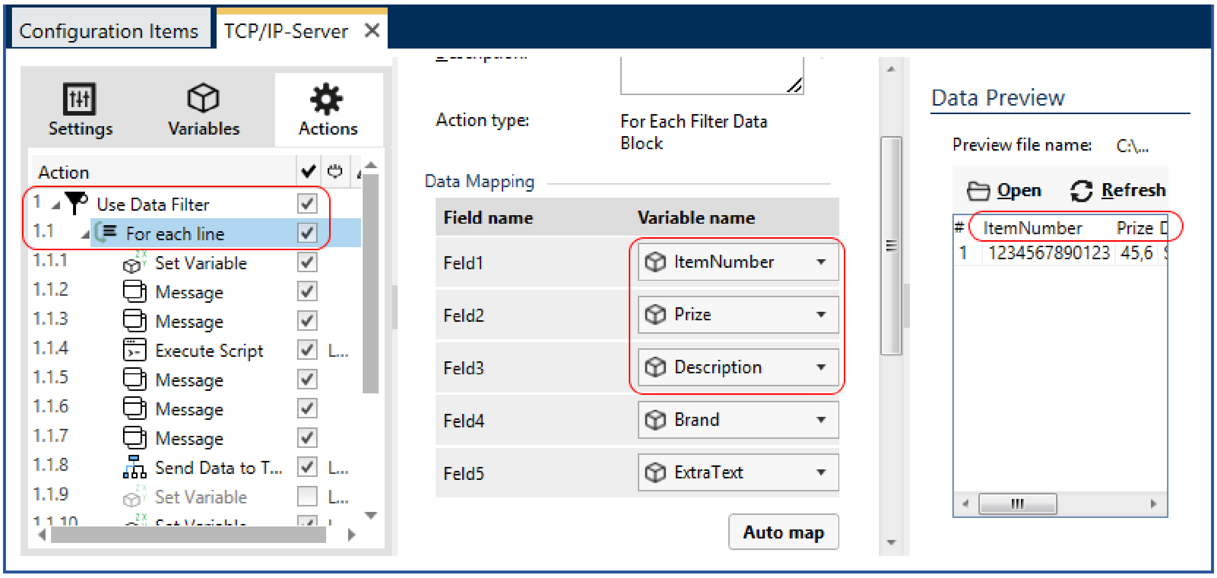
In Automation an input filter must be defined which can allocate these values to the corresponding variables.

Input filter

An input filter consists of two actions: *Use Data Filter* and *For each line*.

The *Automation Builder* supplies a simple possibility to import a data filter. For this, a simple csv text file is created which contains in the first line the column name and in the second line example values.

This file is to be opened in the data preview. In the data allocation, suitable variables can be created automatically.



Action	✓	
1 Use Data Filter	☑	
1.1 For each line	☑	
1.1.1 Set Variable	☑	
1.1.2 Message	☑	
1.1.3 Message	☑	
1.1.4 Execute Script	☑	L...
1.1.5 Message	☑	
1.1.6 Message	☑	
1.1.7 Message	☑	
1.1.8 Send Data to T...	☑	L...
1.1.9 Set Variable	□	L...

Action type: For Each Filter Data Block

Data Mapping

Field name	Variable name
Feld1	ItemNumber
Feld2	Prize
Feld3	Description
Feld4	Brand
Feld5	ExtraText

Preview file name: C:\...

#	ItemNumber	Prize
1	1234567890123	45,6

This means for each line: in the text file can stand multiple lines with data.

I.e. a data filter is to be defined exactly in the order in which the data will be received.

This is the received allocation:

ItemNummer = 12345 Prize = 19,95 Description = 123_ab45-992

NOTICE!

Strictly speaking, the indications for the price and the description are unnecessary because they are taken later from the database. The function method is to be shown only.

With the data filter, make sure that a semicolon is entered as separator.

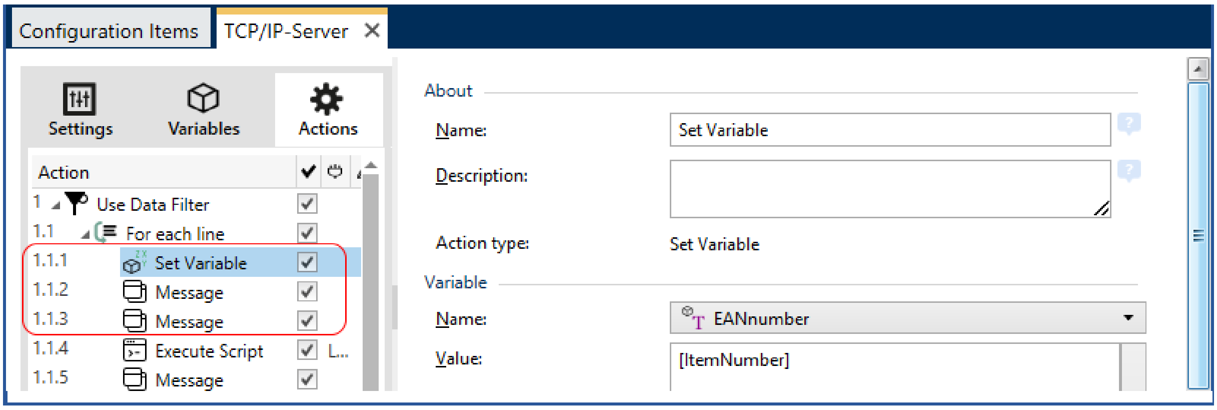
Further processing

The logic of the further processing can be arbitrary.

1. In this example is assumed that on the label a 13 digits EAN number is indicated. I.e. the easy to enter 5 digits article number should change to a 13 digits EAN number.
2. It should be a possibility to generate further queries if necessary or to indicate only reference text. These queries are taken from the database.

In the action 1.1.1 the value of ItemNumber is assigned to the before defined variable EANnumber. This is an important condition for the further actions.

EANnumber = 12345



Action	✓	
1 Use Data Filter	☑	
1.1 For each line	☑	
1.1.1 Set Variable	☑	
1.1.2 Message	☑	
1.1.3 Message	☑	
1.1.4 Execute Script	☑	L...
1.1.5 Message	☑	

About

Name: Set Variable

Description:

Action type: Set Variable

Variable

Name: EANnumber

Value: [ItemNumber]

The messages 1.1.2 and 1.1.3 serve only as debug information and can be deactivated by the right checkmark.

VB script

The script 1.1.4. supplies an important contribution. In the professional or enterprise version direct actions can be taken which are faster in the execution.

Extract from the script:

```
sql = "SELECT * FROM Schuhe WHERE Artikelnummer = '" +
       CStr([ItemNumber].value) + "'"
...
szResult = CStr(rst.Fields("EAN").Value)
EANnumber.Value = szResult
...
szResult = CStr(rst.Fields("Marke").Value)
Brand.Value = szResult
...
If (Len(rst.Fields("ExtraText").Value) > 0) Then
    szResult = CStr(rst.Fields("ExtraText").Value)
       AddQuestion.Value = szResult
    Else
       AddQuestion.Value = ""
End If
```

From the database table Schuhe, the data record with the corresponding article number is used. In line with this the variables *EANnumber*, *Brand* and *AddQuestion* are filled.

Condition: The script is executed only, if: Len([ItemNumber]) < 13

The messages 1.1.5 – 1.1.7 serve again only for debug information.

Show further query at the printer

In the action 1.1.8 a further query is send to APC if the following condition is fulfilled:

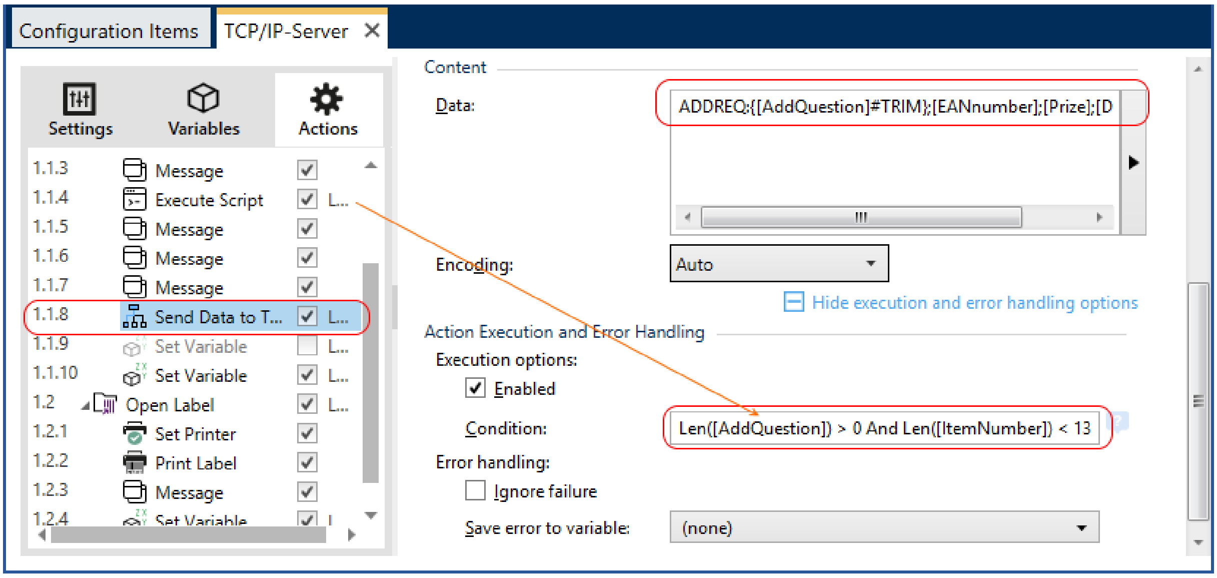
```
Len([AddQuestion]) > 0 And Len([ItemNumber]) < 13
```

I.e. only if there are entries for further queries in the database, this action becomes active.

Additional queries are marked with `ADDREQ:{[AddQuestion];[Add2ndQuestion];[…]};` Afterwards, the past queries (variables) are attached.

```
ADDREQ:{[AddQuestion]};[EANnumber];[Prize];[Description];[Brand];
```

In the following dialog, the connections are again visible:



1.1.3	Message	☑	
1.1.4	Execute Script	☑	L...
1.1.5	Message	☑	
1.1.6	Message	☑	
1.1.7	Message	☑	
1.1.8	Send Data to T...	☑	L...
1.1.9	Set Variable	□	L...
1.1.10	Set Variable	☑	L...
1.2	Open Label	☑	L...
1.2.1	Set Printer	☑	
1.2.2	Print Label	☑	
1.2.3	Message	☑	
1.2.4	Set Variable	☑	L

Content

Data: ADDREQ:{[AddQuestion]#TRIM};[EANnumber];[Prize];[D

Encoding: Auto

Hide execution and error handling options

Action Execution and Error Handling

Execution options:

☑ Enabled

Condition: Len([AddQuestion]) > 0 And Len([ItemNumber]) < 13

Error handling:

□ Ignore failure

Save error to variable: (none)

From the script come the possibly existing additional queries. These are used as data packet and send to the TCP/IP sender, i.e. APC.

Operating principle: Additional query

Assumed in the database stood a value for an additional query. The number 10005 was set as article number. The variable is set as follows:

```
AddQuestion = Extra Text
```

In the printer display appears the text *Extra Text* in the first line.

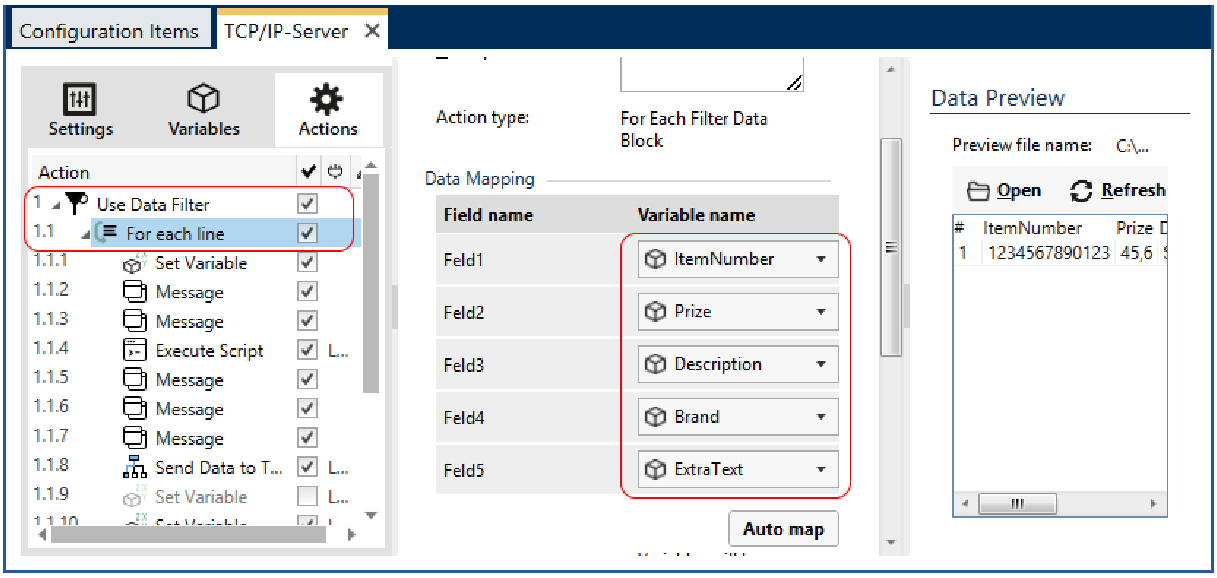
passed to APC

`ADDREQ:{Extra Text;[]}``;4052936045262;119,95;Hiking boots;John Westfield;`

Now at the printer the text e.g. 'Sale' is entered.

APC takes the new entry and append it at the end to the remaining values. This new data is send to Automation:

4052936045262;119,95;Hiking boots;John Westfield;Sale;



Configuration Items | TCP/IP-Server

Action	✓		
1 Use Data Filter	☑		
1.1 For each line	☑		
1.1.1 Set Variable	☑		
1.1.2 Message	☑		
1.1.3 Message	☑		
1.1.4 Execute Script	☑	L...	
1.1.5 Message	☑		
1.1.6 Message	☑		
1.1.7 Message	☑		
1.1.8 Send Data to T...	☑	L...	
1.1.9 Set Variable	☐	L...	

Action type: For Each Filter Data Block

Data Mapping

Field name	Variable name
Feld1	ItemNumber
Feld2	Prize
Feld3	Description
Feld4	Brand
Feld5	ExtraText

Data Preview

Preview file name: C:\...

#	ItemNumber	Prize
1	1234567890123	45,6

With this cycle, all variables from the input filter are filled with data.

```
ItemNumber = 4052936045262     Prize = 119,95
Description = Hiking boots     Brand = John Westfield
ExtraText = Sale
```

In the action 1.1.1, at this time the variable is filled with the new number:

```
EANnumber = 4052936045262
```

I.e. this is now 13 digits long. The actions 1.1.4, 1.1.8 and 1.1.11 cannot be executed no more.

I.e. it continues automatically with the action 1.2.

Error message?

In the action 1.1.11 an error message is generated, if the data record was wrong.

```
ERROR:Data Set~[ItemNumber] wrong
```

Condition: `Len([Brand]) = 0 And Len([ItemNumber]) < 13`

I.e. if the field Brand does not contain an entry, it must have been a wrong article number.

Start print procedure

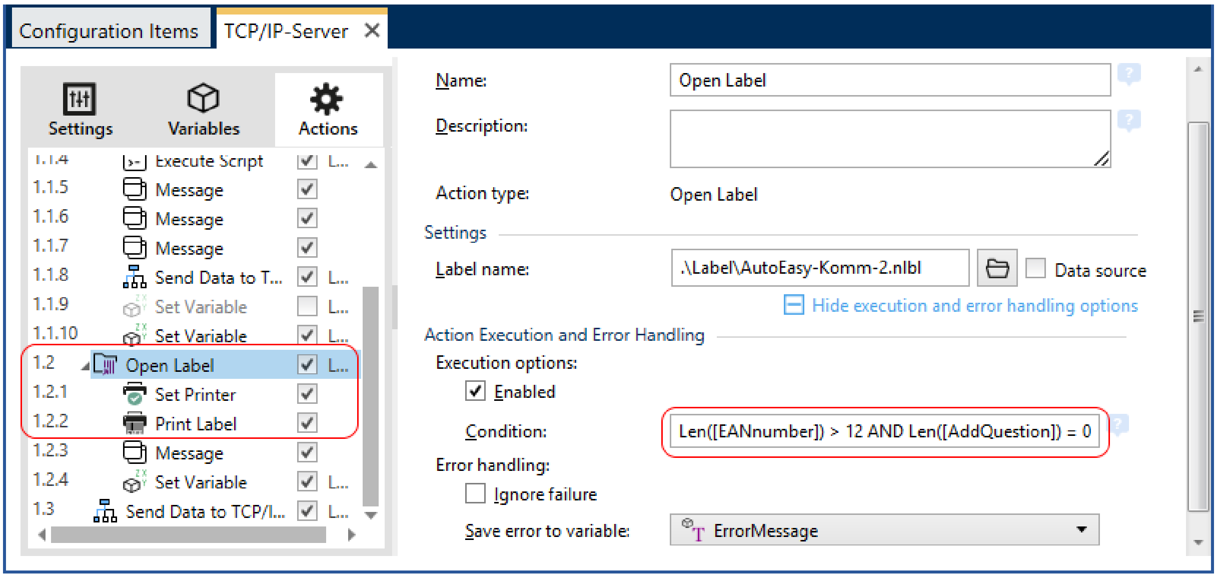
In the action 1.2, the label is opened under the following condition only:

Condition: `Len([EANnumber]) > 12 And Len([AddQuestion]) = 0`

I.e. only if there is a complete EAN number and no more queries, then the label is opened for further processing.

As the variable *AddQuestion* is empty now and the variable *EANnumber* is more than 12 digits long, the printing is started.

At first, in action 1.2 the label which is to print is opened. Afterwards, in 1.2.1 the printer is selected and then in 1.2.2 the printing is started.


Configuration Items | TCP/IP-Server ✕

Settings | Variables | Actions

1.1.4 Execute Script ☑ L...
1.1.5 Message ☑
1.1.6 Message ☑
1.1.7 Message ☑
1.1.8 Send Data to T... ☑ L...
1.1.9 Set Variable □ L...
1.1.10 Set Variable ☑ L...
1.2 Open Label ☑ L...
1.2.1 Set Printer ☑
1.2.2 Print Label ☑
1.2.3 Message ☑
1.2.4 Set Variable ☑ L...
1.3 Send Data to TCP/I... ☑ L...

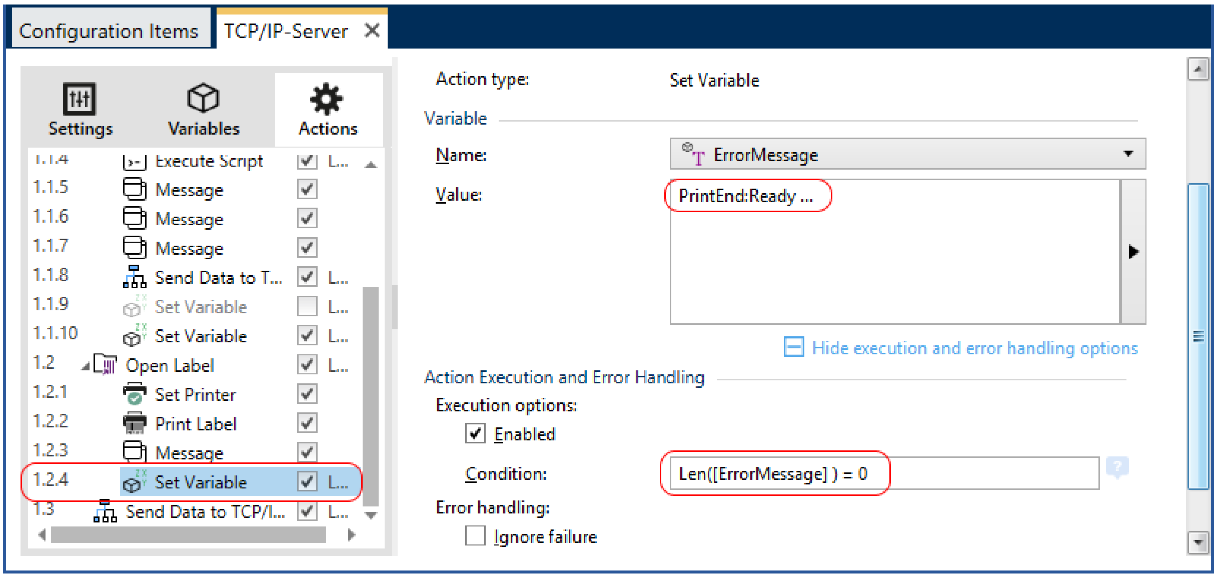
Name: Open Label
Description:
Action type: Open Label

Settings
Label name: .\Label\AutoEasy-Komm-2.nlbl □ Data source
Hide execution and error handling options

Action Execution and Error Handling
Execution options:
☑ Enabled
Condition: Len([EANnumber]) > 12 AND Len([AddQuestion]) = 0
Error handling:
□ Ignore failure
Save error to variable: ErrorMessage

If errors occur they are saved in the variable *ErrorMessage*, otherwise the variable is empty.

In the action 1.2.4 a success message is generated, if no error occurred.


Configuration Items | TCP/IP-Server ✕

Settings | Variables | Actions

1.1.4 Execute Script ☑ L...
1.1.5 Message ☑
1.1.6 Message ☑
1.1.7 Message ☑
1.1.8 Send Data to T... ☑ L...
1.1.9 Set Variable □ L...
1.1.10 Set Variable ☑ L...
1.2 Open Label ☑ L...
1.2.1 Set Printer ☑
1.2.2 Print Label ☑
1.2.3 Message ☑
1.2.4 Set Variable ☑ L...
1.3 Send Data to TCP/I... ☑ L...

Action type: Set Variable

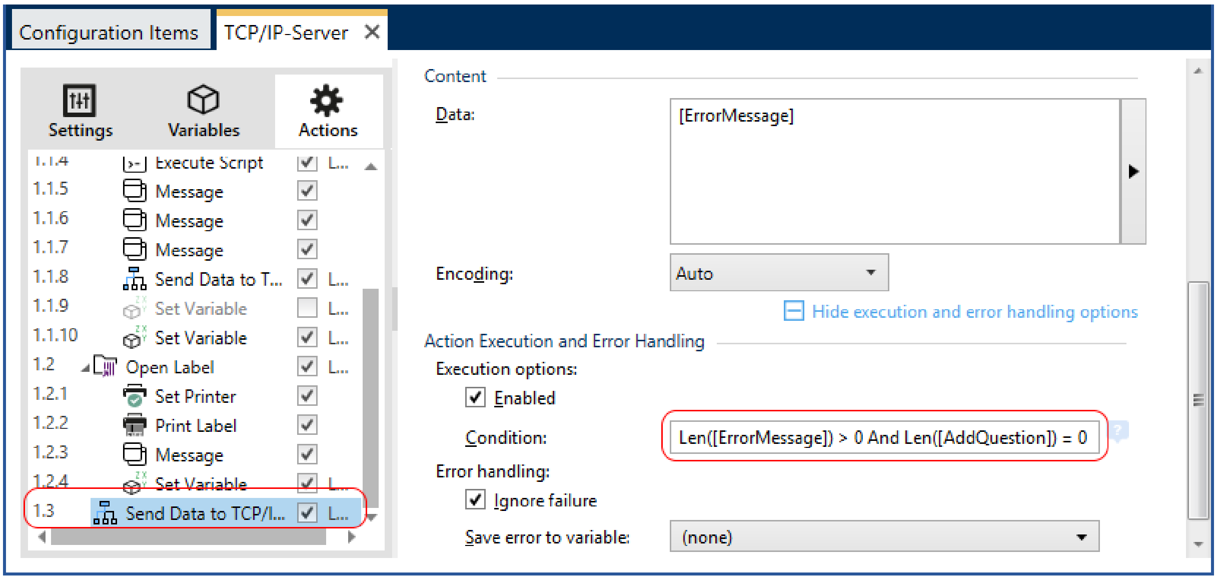
Variable
Name: ErrorMessage
Value: PrintEnd:Ready ...

Hide execution and error handling options

Action Execution and Error Handling
Execution options:
☑ Enabled
Condition: Len([ErrorMessage]) = 0
Error handling:
□ Ignore failure

Final message

At the end a message is send to APC that the print was successful or that an error arose.



1.1.4	Execute Script	☑	L...
1.1.5	Message	☑	
1.1.6	Message	☑	
1.1.7	Message	☑	
1.1.8	Send Data to T...	☑	L...
1.1.9	Set Variable	□	L...
1.1.10	Set Variable	☑	L...
1.2	Open Label	☑	L...
1.2.1	Set Printer	☑	
1.2.2	Print Label	☑	
1.2.3	Message	☑	
1.2.4	Set Variable	☑	L...
1.3	Send Data to TCP/I...	☑	L...

Content

Data: [ErrorMessage]

Encoding: Auto

Hide execution and error handling options

Action Execution and Error Handling

Execution options:

☑ Enabled

Condition: Len([ErrorMessage]) > 0 And Len([AddQuestion]) = 0

Error handling:

☑ Ignore failure

Save error to variable: (none)

I.e. with the skilful interaction of variables and their contents, different actions are released or closed.

Different actions can be activated, so that they serve only as assistance (debug).

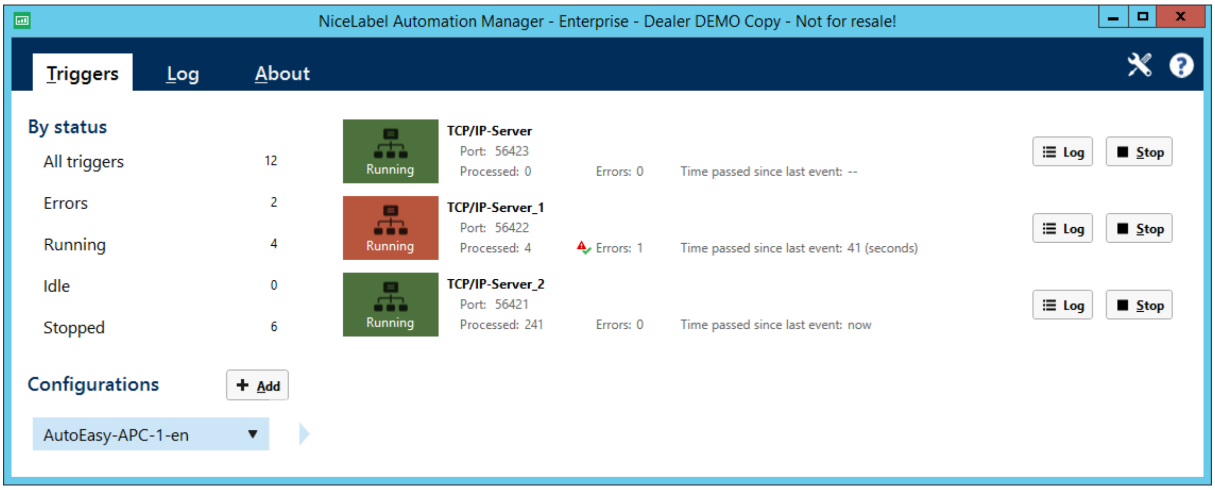
10.2 Summary

In connection with a further query the conditions play a decisive role. Without the use of additional variables and their dependent filling with data, such a process is hardly to realize.

The first approach is e.g. the used 5-digit article number which is only changed before the desired printout or for an additional query for 13-digits. In the shown example, the script is only executed in the first run. The script will be locked independently for a second run.

Process testing

In the log recording of the *Automation Manager* each action can be examined for errors.


NiceLabel Automation Manager - Enterprise - Dealer DEMO Copy - Not for resale!
Triggers
Log
About
Add
Log
Stop
Running

By status

All triggers	12
Errors	2
Running	4
Idle	0
Stopped	6

Configurations

AutoEasy-APC-1-en

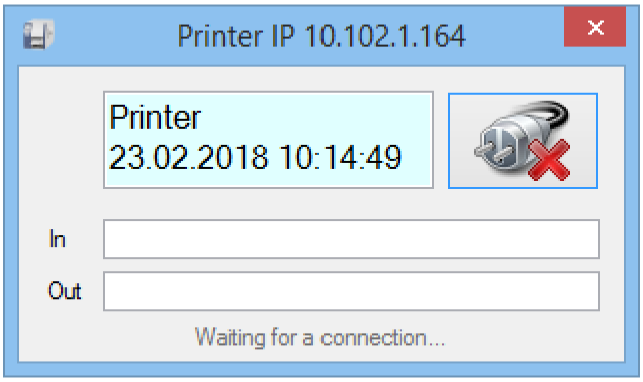
Running	TCP/IP-Server Port: 56423 Processed: 0	Errors: 0	Time passed since last event: --	Log Stop
Running	TCP/IP-Server_1 Port: 56422 Processed: 4	Errors: 1	Time passed since last event: 41 (seconds)	Log Stop
Running	TCP/IP-Server_2 Port: 56421 Processed: 241	Errors: 0	Time passed since last event: now	Log Stop

11 Test Environment

The installation package contains an Automation configuration and an Access database table which can be imported e.g. into a SQL database.

In this database, article numbers from 10001 to 10012 are used.

In order to test multiple printers at the same time, the provided program PrinterDummy.exe can be used on a further PC. This tool sends correct and wrong article numbers per random principle to APC. However, multiple queries are not supported.



Printer IP 10.102.1.164

Printer
23.02.2018 10:14:49

In

Out

Waiting for a connection...

12 Browser Status Display

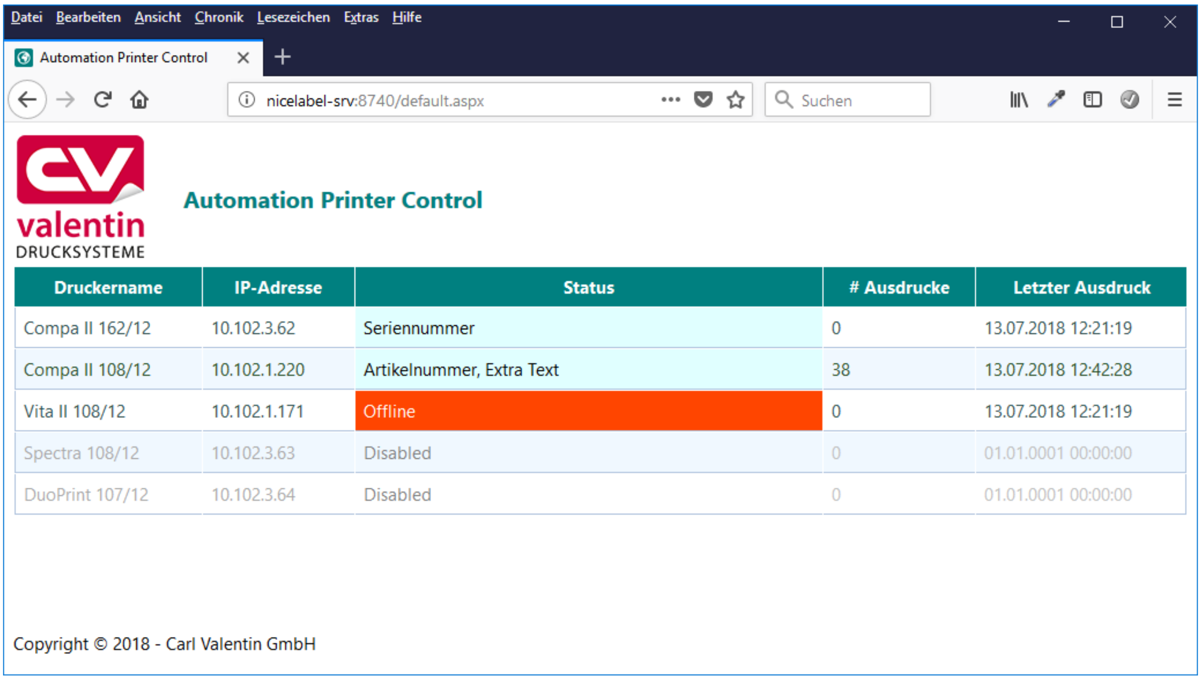
If the status of the server should be shown from another PC (or smartphone), the browser view can be started.

Instead of nicelabel-srv (or localhost), the address of the own server is to be entered.

http://nicelabel-srv:8740/Default.aspx

NOTICE!

IIS must be activated on the server.



Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
Automation Printer Control
nicelabel-srv:8740/default.aspx
Suchen
valentin
DRUCKSYSTEME

Automation Printer Control

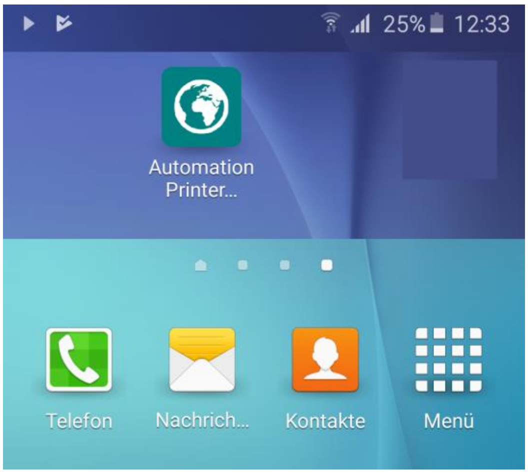
Druckername	IP-Adresse	Status	# Ausdrucke	Letzter Ausdruck
Compa II 162/12	10.102.3.62	Seriennummer	0	13.07.2018 12:21:19
Compa II 108/12	10.102.1.220	Artikelnummer, Extra Text	38	13.07.2018 12:42:28
Vita II 108/12	10.102.1.171	Offline	0	13.07.2018 12:21:19
Spectra 108/12	10.102.3.63	Disabled	0	01.01.0001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

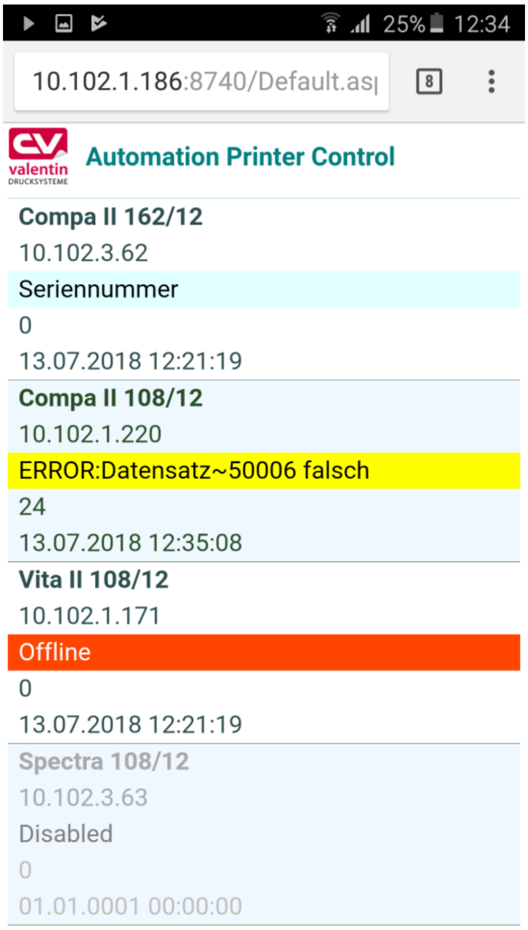
Copyright © 2018 - Carl Valentin GmbH

On a smartphone, the server address is to be entered in a browser.

On a smartphone, the server address is to be entered in a browser.

With an Android Chrome browser, by means of the three points next to the address line, open the context menu and select the *Add to start screen*. Herewith an icon is created which permits fast access:





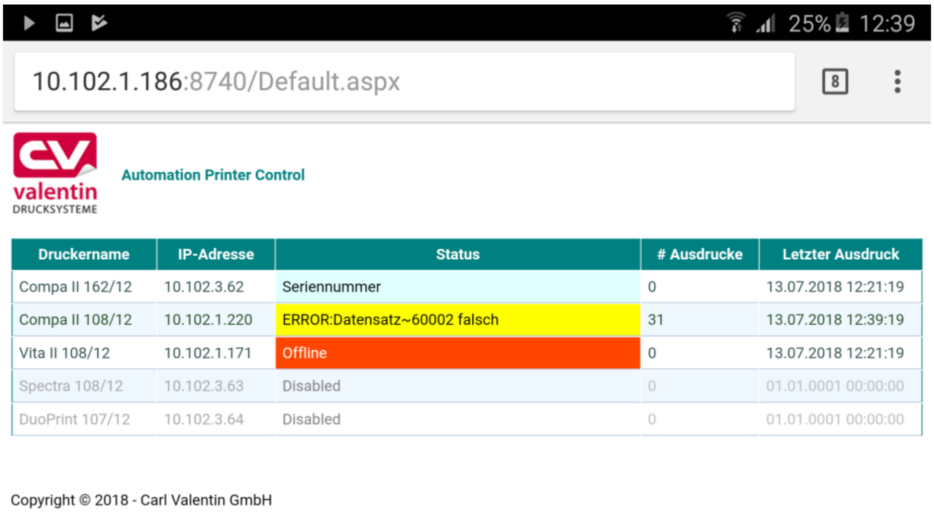
Automation Printer Control

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~50006 falsch
24
13.07.2018 12:35:08

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



Automation Printer Control

Druckername	IP-Adresse	Status	# Ausdrucke	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

Copyright © 2018 - Carl Valentin GmbH

Carl Valentin GmbH
Neckarstraße 78 – 86 u. 94 . 78056 Villingen-Schwenningen
Phone +49 7720 9712-0 . Fax +49 7720 9712-9901
info@carl-valentin.de . www.carl-valentin.de