

The logo for NiceLabel, featuring the word "NiceLabel" in white text on a dark blue rounded rectangular background. A small green circle with a white arrow pointing up and to the right is positioned at the top right corner of the dark blue rectangle.

NiceLabel

NiceLabel Automation

Version 1.5

Release Notes

Rev-1602

Table of Contents

What is New in NiceLabel Automation?	2
New Label Template Features.....	2
Batch Printing Support.....	2
Support for Codabar Barcode Symbology (1.5.1)	3
Support for Master/Detail Table Connections in Labels.....	3
Support for “Collect Records” Database Connection Option	4
Stand-alone SSCC Barcode Support	4
Updated Application Identifiers for GS1 Support (1.5.1).....	5
Support for HIBC Specification version 2.5 (1.5.1).....	5
Updated Printing Subsystem.....	5
Automatic Font Replacement (1.5.1).....	5
Updates to Non-restrictive Data Processing (1.5.1)	6
Optimized Printing of Fixed Data on the Labels	6
Improved Quality of the Printing of PDF Documents	7
Retrieving Printer Status for non-NiceLabel Printer Drivers.....	7
Support for Unlimited Printing Quantity	8
Other Updates.....	9
Overriding the Database Connection Strings.....	9
Simplified Connections to Microsoft Excel and Microsoft Access Data Sources	10
User Guides and Help Are Available in French and German Languages.....	10
Retry Option Added to the “Execute SQL Statement” Action (1.5.1).....	11
Support for Optional Elements in the XML Schema (1.5.1).....	11
Windows 10 Support	11
Periodical Refresh of Printer Driver List.....	11
Support for UTF-16 Big Endian Encoding.....	12
Label Variable with Multiline Formatting	12
Code128 Coding Mode Algorithm Optimization.....	13
Automatically Detect and Fix Corrupted Log Database.....	14
Updated Toolbar in Script Editors.....	14
Code 39 Barcode Length is no Longer Limited.....	15

What is New in NiceLabel Automation?

New Label Template Features

Batch Printing Support

The batch printing functionality that can be defined in NiceLabel Pro label designer is now also available when printing labels from NiceLabel Automation.

The batch printing mode allows you to do the following:

- Defining more labels inside the same label. You can have Header, Main and Tail labels. Double-sided printing is also available, if you have such a printer. The Header/Tail labels print just once per batch, the Main labels print in the required quantity.
- Defining the quantity of labels that identify a batch. Usually, the whole print job consists of one batch.
- Defining actions after the Header/Main/Tail label prints. The available list of actions is determined by the printer driver you are using. For thermal label printers, the actions usually are “cut label” or “pause printer until the label is removed”, for office printers the action is usually “eject page”.

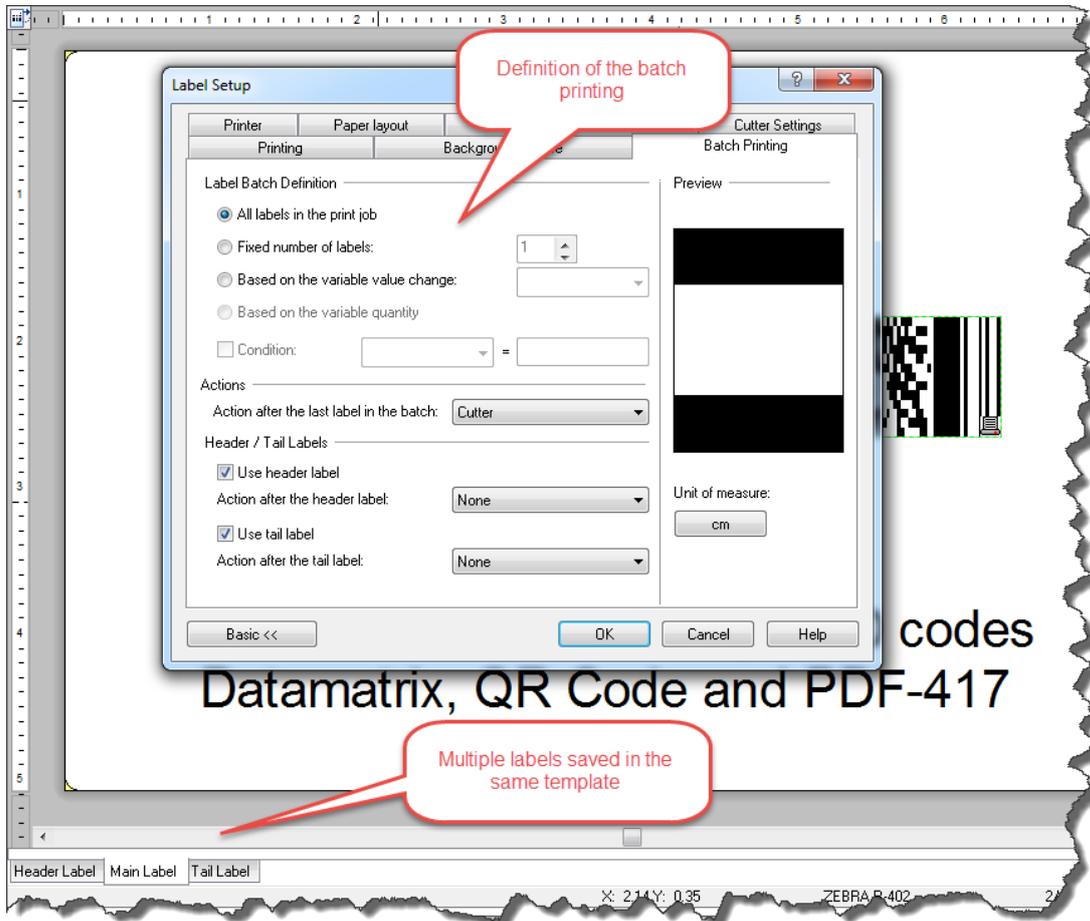


Figure 1: Batch printing supported in Automation products

Support for Codabar Barcode Symbology (1.5.1)

Based on the requirement expressed by NiceLabel users, support for Codabar barcode symbology has been added to NiceLabel Automation.

Support for Master/Detail Table Connections in Labels

A Master/Detail connection in the label represents a case when you have made two (or) more database connections on the label and have created connections between the tables. In this case, each record in the master connection has one or more records associated in the detail connection. The combined data from the master and detail table must print on the same label.

When two or more tables are dependent, the master table will switch to the next record only after all of the records in the detail table have been used.

Support for “Collect Records” Database Connection Option

When the option **Collect Records** is enabled in the label template, the defined number of consecutive records is “collected together” and will print on the same label, not on different labels.

For example, if you collect 10 records in the same dataset, a single label will be printed, not 10 labels.

The records are merged together using the defined delimiter.

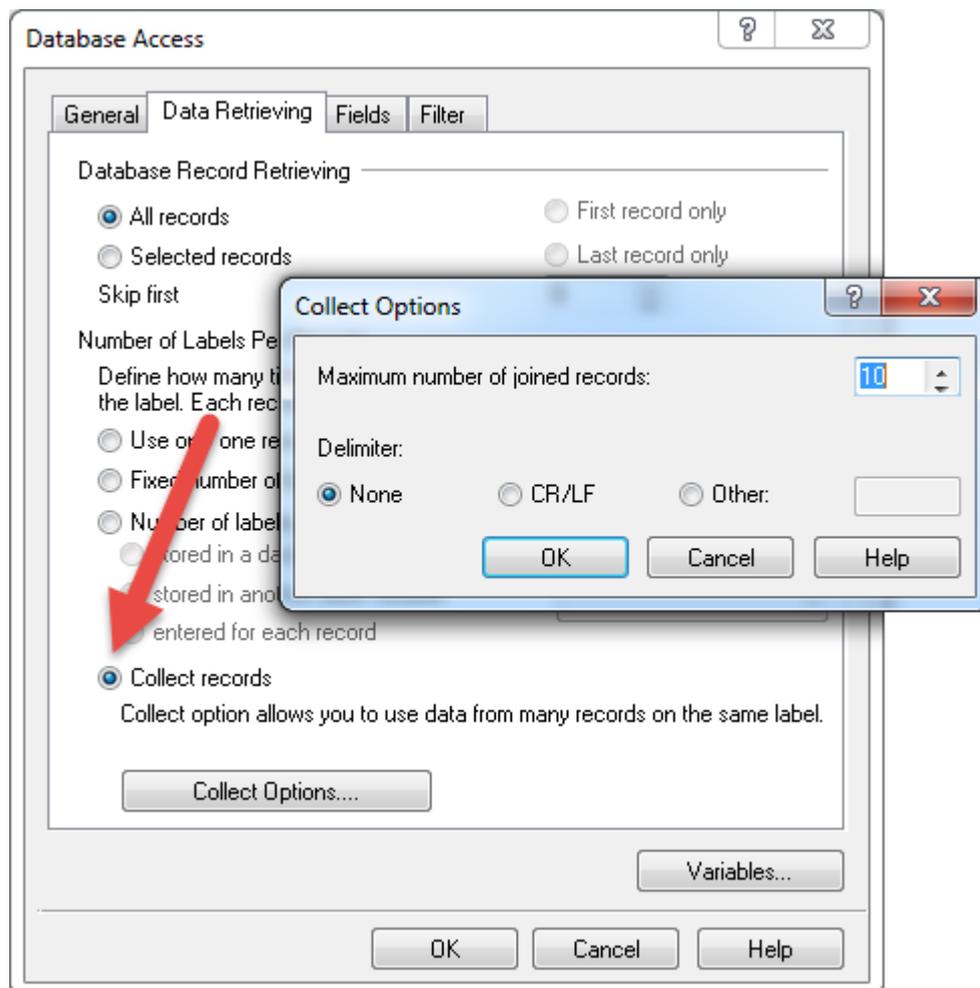


Figure 2: Option "Collect records" is supported in NiceLabel Automation

Stand-alone SSCC Barcode Support

SSCC stands for Serial Shipment Container Code and is part of the GS1 Logistic Specification. The SSCC is in fact a GS1 Application Identifier (00) and has been therefore supported since the first version of Automation. The AI(00) usage is frequent enough to be made available by the NiceLabel software as a new type of barcode – the SSCC barcode.

When selecting this barcode, the NiceLabel will guide you throughout the definition of the SSCC components (Extension Digit, GS1 Company Prefix, Serial Reference).

Updated Application Identifiers for GS1 Support (1.5.1)

NiceLabel software has been updated accordingly with the changes that were given by the GS1 General Specifications Version 15, January 2015 (issue 2) and Version 16, January 2016.

The following Application Identifiers have been removed from GS1 specification: 8100, 8101, 8102. They are still part of the software for backward compatibility.

The following Application Identifiers have been added in January 2016 update: AI394n (Percentage discount of a coupon), AI8012 (Software version), AI8111 (Loyalty points of a coupon).

The following Application Identifiers have been added in January 2015 update: 7005 (Catch Area), 7006 (First Freeze Date), 7007 (Harvest Date), 7008 (Species for Fishery Purposes), 7009 (Fishing Gear Type), 7010 (Production Method).

<p>NOTE: NiceLabel Pro release 6.5.1 is required for designing labels with the latest set of Application Identifiers.</p>

Support for HIBC Specification version 2.5 (1.5.1)

Health Industry Business Communications Council (HIBCC) is an industry supported and internationally accredited nonprofit standards development organization with a global reach. The council develops standards that meet the unique requirements of the world's healthcare providers, such as HIBC.

HIBCC released version 2.5 of the HIBC specification, which is now supported with NiceLabel Automation.

Updated Printing Subsystem

Automatic Font Replacement (1.5.1)

You might decide to design your label templates to print the text objects using fonts formatted as built-in printer (internal) fonts. However, when printing such label using a different kind or printer, the selected fonts might not be available on the new printer. The new printer probably supports an entirely different set of internal fonts.

Similar problem might occur if the TrueType font that is used on the label is not installed on the target PC, from which NiceLabel Automation will be printing the labels.

NiceLabel Automation can be configured to automatically replace the fonts used on the label with compatible fonts. You can configure the font mapping based the font names. If the original font is not found, NiceLabel Automation will tries to use the first available replacement font as defined in the mapping table. If no suitable replacement font is found, Arial TrueType font is used.

Updates to Non-restrictive Data Processing (1.5.1)

The previous version of NiceLabel Automation included an option named “Non-restrictive data processing”. This option was included in the trigger properties. Its role was to control the parameters during the printing process:

1. **The excessive value provided for the variable was ignored.**
For example, a variable is configured to accept a maximum number of 10 characters. If you assign 15 characters to the variable, the last 5 of them were cut, and only the first 10 were used.
2. **Setting values to non-existing label variables was ignored.**
For example, you were trying to assign the value “100” to a variable named ProductID, which was not defined in the label. The “Non-restrictive data processing” option ignored such issues.

The latest NiceLabel Automation release enables you to set these two options with individual checkboxes. You can fine-tune the parts of the “data processing” that you want to enable or disable.

With release 1.5.1, these two options are available in all editions of NiceLabel Automation software (in previous releases, they were available in Automation Pro and Automation Enterprise only).

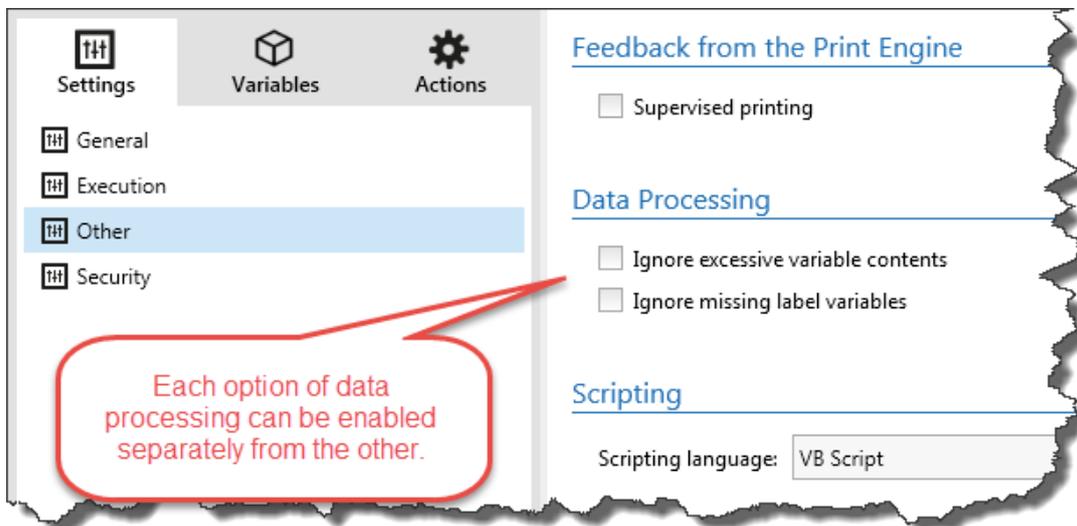


Figure 3: Ignoring errors in data processing

Optimized Printing of Fixed Data on the Labels

When the label quantity is more than 1, the print job will contain more labels, as many as are specified in the Print Label action. New NiceLabel Automation optimizes the label objects that are encoded in the print job.

The variable objects that do not change during the label printing will be sent to the printer just once with the first label. Previously, the variable object that was assigned with some value was sent to the printer for each label; it did not change during the print.

Improved Quality of the Printing of PDF Documents

When you place PDF documents as image objects on the label, they will be converted into images when the label gets printed.

Previously, PDF image objects were converted to an image using 300 DPI resolution. This could cause poor quality printouts when using higher-resolution printers, as the image inside a PDF had to be stretched to the new resolution, or the problems would be down-scaling graphics to 200 DPI printers.

In version 1.5, the rendering quality of PDF graphics matches the resolution of the target printer. Such a PDF will print optimally to the printer for which it was created.

Retrieving Printer Status for non-NiceLabel Printer Drivers

NiceLabel Automation can query all printer drivers, including non-NiceLabel printer drivers, for the live printer status. Based on the printer status response, you can decide to run different actions in the trigger. Moreover, when NiceLabel Automation is activated with a license from the Control Center, the printer statuses of all local printers are reported to the Control Center, which can be configured to send out automated alerts in case of printer problems.

Typical problems reported from the printer might be “Out of Paper”, “Toner Low”, “Out of Toner”, etc. These messages are dependent on the printer model/driver that you are using. Generally speaking, the statuses that you can see in the Windows Spooler are also reported to NiceLabel Automation.

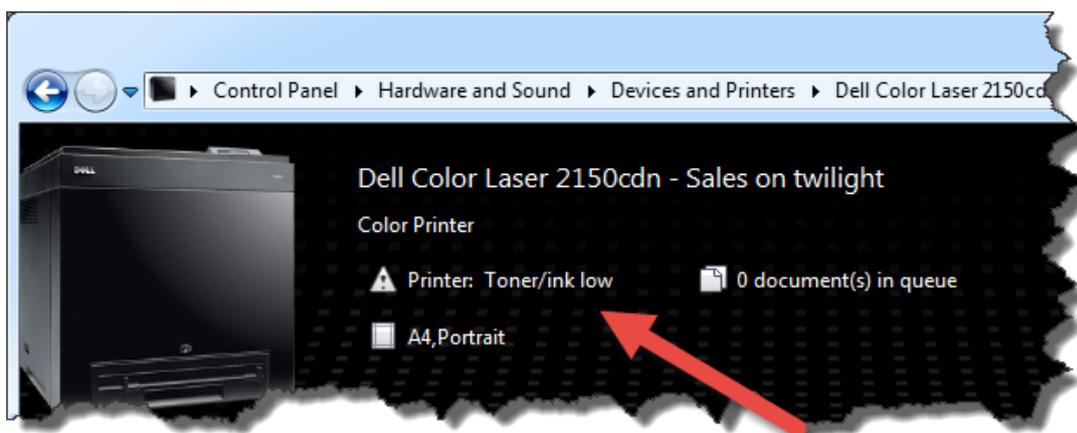


Figure 4: Printer status as shown in the printer driver

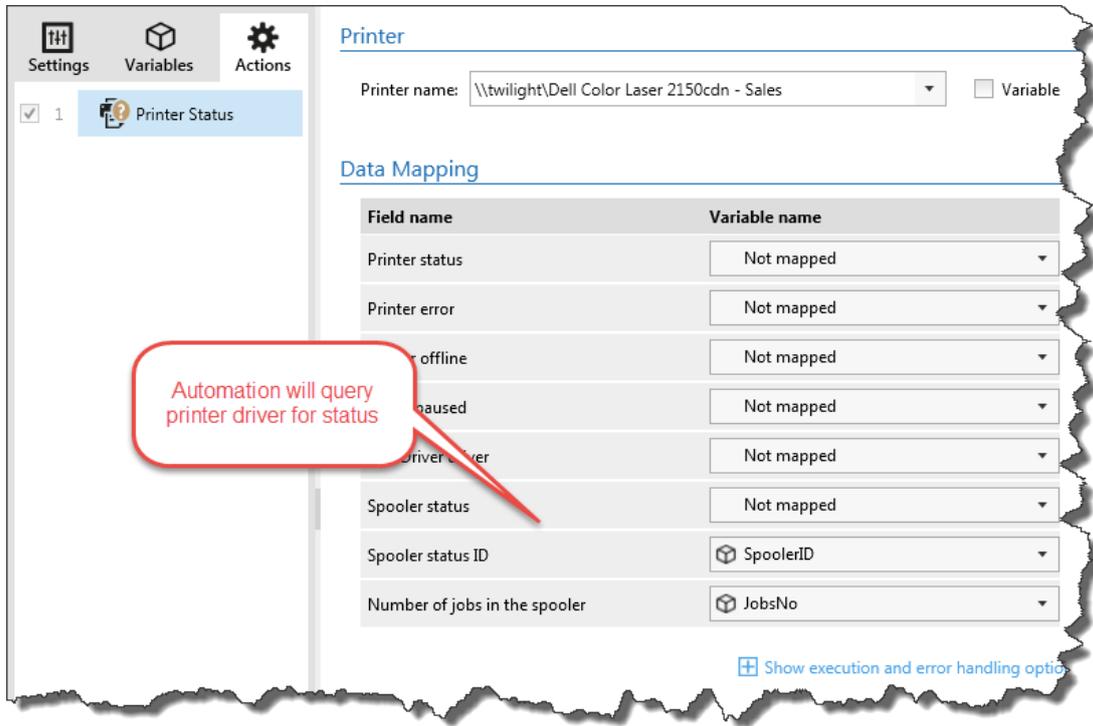


Figure 5: Action Printer Status will ask the driver for printer status

```
"SpoolerID";"SpoolerStatus"
"131072";"Toner/Ink Low"
```

Figure 6: Printer status saved in two defined variables

Support for Unlimited Printing Quantity

The option "All (unlimited quantity)" supports all printing scenarios in Automation 1.5.

Typically, you would use this option in two scenarios.

Scenario 1: You command the printer to continuously print the same label until it is switched off, or when it receives a command to clear its memory buffer. In this scenario, you must use the NiceLabel printer driver to print your labels. When you print the fixed label, just one print job is sent to the printer, with the quantity set to "unlimited". Label printers have a parameter to the print command to indicate "unlimited" printing.

When the label is not fixed but includes objects that change during the printing, such as counters, then the printed quantity will be set to the maximum quantity that is supported by the printer. For example, some printers have an internal quantity limit set to 32,000 labels.

NOTE: You must use NiceLabel printer drivers.

Scenario 2: The trigger doesn't provide any data, but rather only acts as a signal showing that the "event has happened/now let's print labels". The logic to acquire the necessary data is on the label. A connection to a database is configured on the label.

At each trigger, the label must connect to the database and print all records from the database. In this case, the option "unlimited" is understood as "print all records from the database".

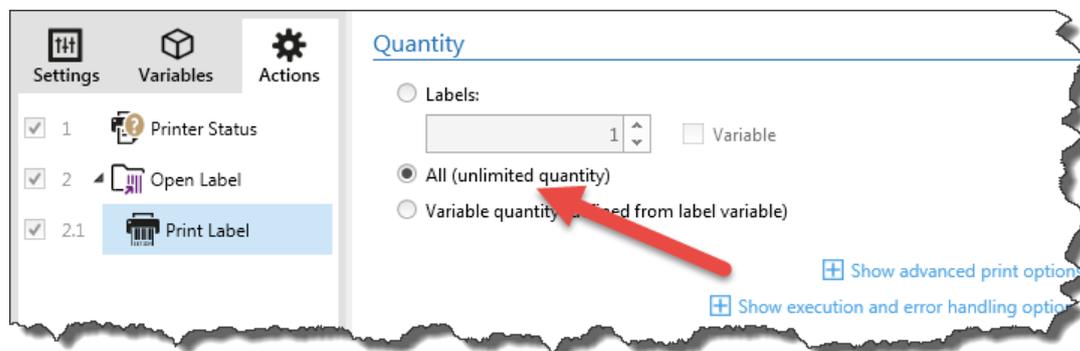


Figure 7: Supported print scenarios for "unlimited quantity"

Other Updates

Overriding the Database Connection Strings

When you configure a database trigger or define the action Execute SQL Statement, you have to provide a "connection string". That is a string that specifies information about a data source and the means of connecting to it. It includes attributes such as the name of the server and the database, as well as security information such as username and password.

When you configure and run the triggers in the same environment, once defined connection string can be used for testing in the development as well as in the shop floor.

However, there are occasions, where the connection string must change before the trigger can be used in the production, such as:

- You configure the trigger in your environment and will later transfer it into the client's environment. You use your own database during configuration and testing, but will have to use the client's database.
- You use the multi-tier landscape with development/quality assurance/production environments. The database used during the development and testing is not the same as the production database.

NiceLabel Automation 1.5 addresses such deployments and allows you to override the database connections strings used in the configurations without modifying the trigger configuration itself. You can define an XML configuration file that contains a table of source and replacement database servers and databases.

When the Automation Service detects the config file, all matching server names and databases in the connection strings will be replaced with the new names.

Simplified Connections to Microsoft Excel and Microsoft Access Data Sources

The modern-day machines running Windows operating system usually have the 64-bit variant of Windows installed. In such an environment, Automation Service will run as a 64-bit application, also requiring 64-bit drivers to access database sources.

The Automation Builder that you use for developing and testing your configuration always runs as a 32-bit application. As such, it will use 32-bit database drivers. While label preview works on-screen in Automation Builder, you might see error messages in Automation Manager reported by the Service.

When you access databases in your configurations, Automation Service from version 1.5 will first try to connect to the database using 64-bit drivers. If no such drivers are available on the machine, Automation Service will fall back to using 32-bit drivers.

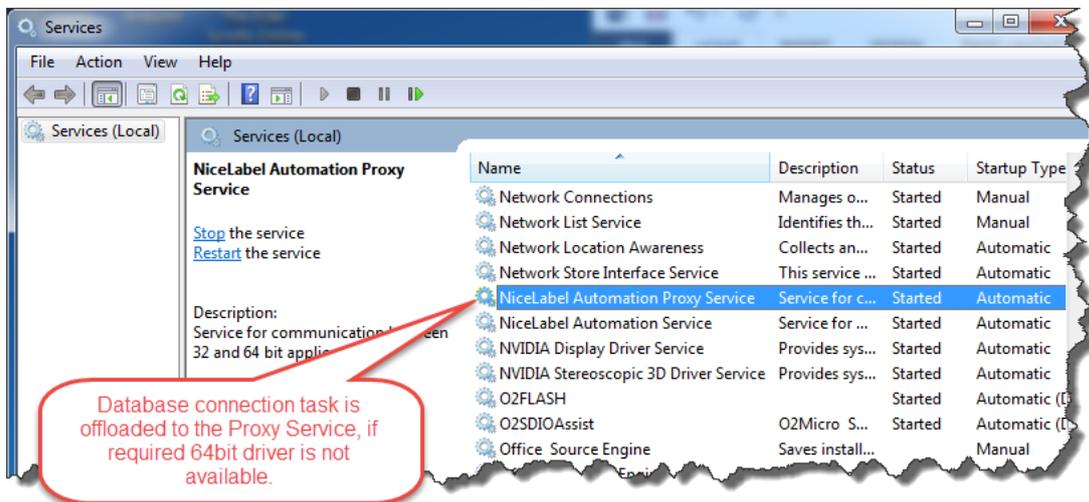


Figure 8: NiceLabel Automation Proxy Service will connect to the database, if 64bit driver is not available

User Guides and Help Are Available in French and German Languages

NiceLabel Automation user guide and online help system have been localized into French and German languages.

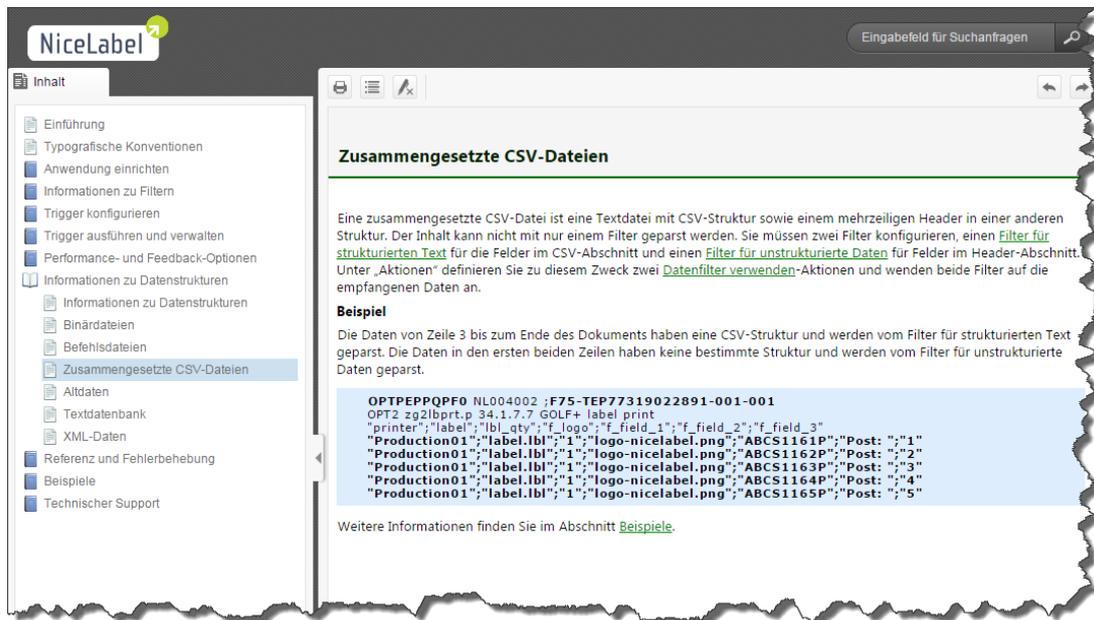


Figure 9: Online help system in German language

Retry Option Added to the “Execute SQL Statement” Action (1.5.1)

You can configure the Execute SQL Statement action to retry establishing a connection with the database server continuously in case the first attempt was not successful. If the action continues to fail after the defined number of attempts, an error is raised.

Support for Optional Elements in the XML Schema (1.5.1)

XML Data Filter supports optional elements. Optional element can be XML element, attribute or data block. When element is configured as optional, XML data parsing does not fail with an error, when the element is not available in the provided data. Any variables connected to optional elements are set to empty values when such element is missing.

This conforms to the XSD indicator `minOccurs`, which specifies the minimum number of times an element can occur. If value of `minOccurs` is 0, the element is optional, it might be present in the data or not.

Windows 10 Support

NiceLabel Automation has been tested to work on Windows 10 operating system, ensuring the compatibility with the new operating system when it will ship.

Periodical Refresh of Printer Driver List

NiceLabel Automation will refresh the list of available printer drivers at startup and each time the Windows system notifies it about the addition of a new printer. However, if the Windows firewall is not configured correctly, the information about newly added network

printers is not sent to the Automation and it might not see the new printer until the Service is restarted.

To better fit into environments where the firewalls cannot be modified to allow certain inbound protocols, Automation Service will periodically refresh the list of available printer drivers. If new printer drivers are found, they can be used for printing without a Service restart.

The default refresh time interval has been set to 10 minutes and can be modified by a setting in a configuration file.

Support for UTF-16 Big Endian Encoding

When you work with Unicode data and select the UTF-16 encoding mode in Automation what you are actually selecting is the UTF-16 **Little Endian** mode. The Unicode data that Automation receives might be encoded in the other variant of the UTF-16 data encoding, which is UTF-16 **Big Endian**.

NiceLabel Automation supports UTF-16 Big Endian text encoding in triggers and actions.

The term **endian** refers to the convention used to encode the data bytes. The difference is in how the data is internally stored, and in which order the data is stored. The Intel x86 processor uses the Little Endian architecture, while other system that NiceLabel Automation integrates with might use the Big Endian mode.

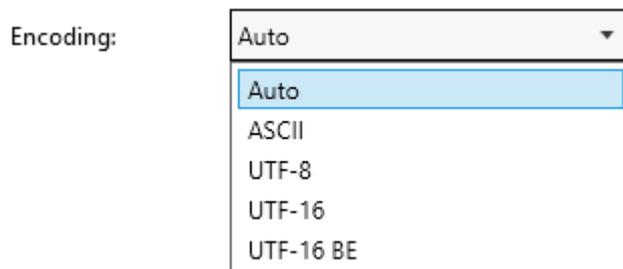


Figure 10: UTF-16 Little Endian and Big Endian encodings are both supported

This update allows easier communication with wider range of system/technologies, such as AS/400 or SAP systems, without the need for the internal conversion into Little Endian mode, which was necessary with previous version.

Label Variable with Multiline Formatting

NiceLabel Pro label designer supports multiline property in the settings of variables. The option allows you to break a long string of data in multiple lines of data. When you feed this data into a Text object, it will become multiline text object.

This option will be useful to existing users with such label templates who are migrating from older NiceLabel integration platform with NiceWatch to Automation. No modification of existing labels will be necessary to print them from Automation.

However, when you are creating new labels and need to use multiline data, the recommended way is by using the Text Box or RTF Box objects. These two have native support for multiline contents with automatic word-wrapping.

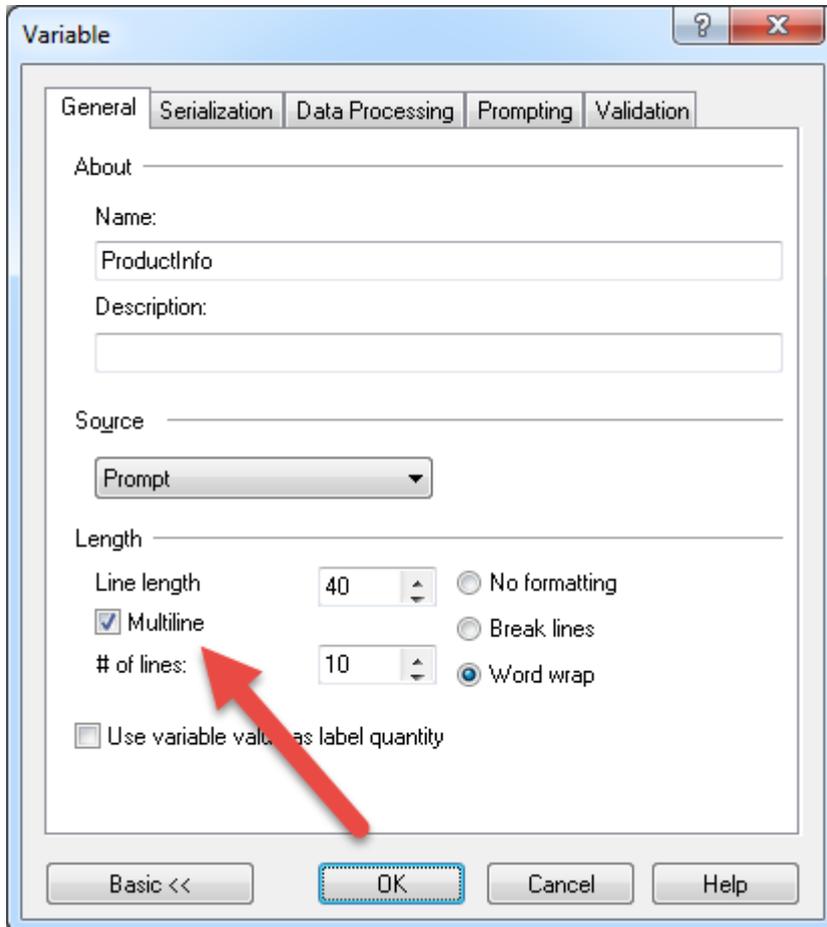


Figure 11: Multiline variable is supported

Code128 Coding Mode Algorithm Optimization

The barcode Code128 includes the data/command elements that can encode the same data using different methods, which results in shorter or longer size of the barcode symbol. Obviously, Automation strives to generate the shortest possible barcode symbol.

The new version of Automation implements an optimized Code128 encoding algorithm resulting in the shortest barcodes. The barcode will occupy less space on the label.

For example, you can encode the data "A12345" in many variants. Here are two:

- <ModeB>A<ModeC>1234<ModeB>5
- <ModeB>A1<ModeC>2345

Obviously, the latter variant provides a better result as less overhead data is encoded in the barcode. The new algorithm included with Automation 1.5 will impact your Code128 barcodes in the similar way.

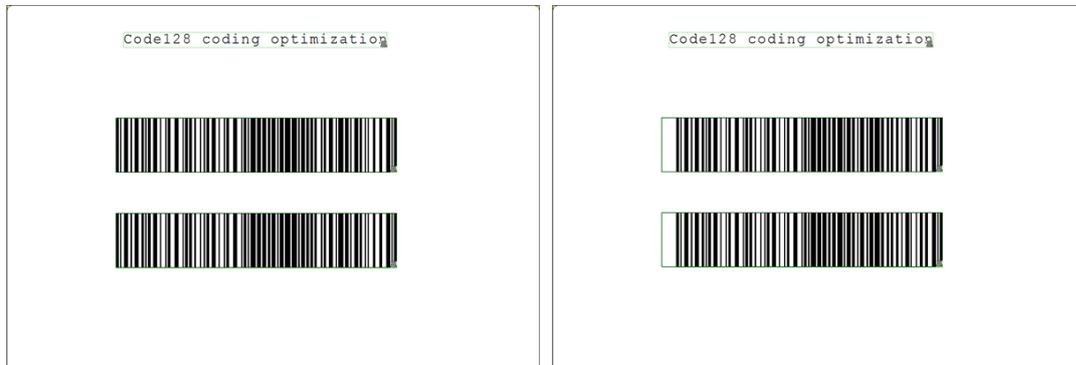


Figure 12: The before-after screenshot of the Code 128. The right barcode is optimized and with a smaller footprint.

Automatically Detect and Fix Corrupted Log Database

Automation stores the history log for each trigger event in the local SQLite database.

In some cases, the database files can become corrupted, usually caused by abrupt termination of machine functionality, such as power outage without using UPS system or some other hard disk related problems.

When the Automation Service starts the log database is checked for consistency. When the structural/data problems are detected in the log database, Automation will replace the database files with the initial (empty) version of the files.

NOTE: Automation local log files are not meant to be used as long-term print history logging. Automation retains the last 7 days-worth of data in the local SQLite database. If you need longer history storage and want to consolidate printing events from multiple NiceLabel machines, consider using NiceLabel Control Center product.

Updated Toolbar in Script Editors

The order of the toolbar commands used in the script editor control has been changed. First in the order are commands that allow you to verify or test-run the entered code, followed by variable-related commands, followed by commands to export/import the entered script to file on the disk.

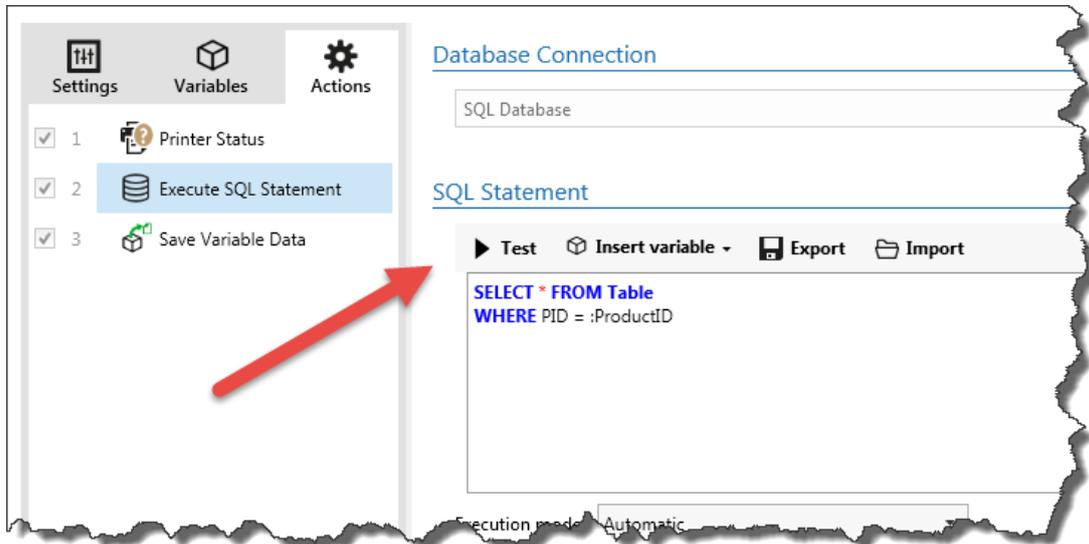


Figure 13: More intuitive order of commands in the toolbar

These changes will be visible in the following actions: “Execute script”, “Send custom commands”, and “Execute SQL statement”.

Code 39 Barcode Length is no Longer Limited

By definition, you can enter any number of characters for Code 39. In practice, the barcode Code 39 quickly becomes very long.

NiceLabel Automation no longer limits the Code 39 barcode contents to 50 characters.