

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 Designers 6.3

Release Notes

Rev-1409

Table of Contents

What is New in NiceLabel Designers	2
Label Design	2
DotCode Barcode Symbology	2
Overriding the Default Values of Variables.....	2
Updated Support for Transparent Graphics	2
Support for Food Allergens Built-in through Custom Functions.....	3
Printing JPEG Images to Postscript Printers in CMYK Color Space	4
Relative Positioning of Elements on Both Sides of the Label	4
Importing PDF Files as Images	5
Support for the Full Justification of Text in an RTF Object	6
Support for HIBC Specification version 2.4.....	6
Updated List of GS1 Application Identifiers.....	6
Support for the Custom RTF (Rich Text Format) Control Codes	7
Fitting Barcode to Label Size	7
New Antenna Settings for RFID Support.....	8
Default Printer Selected in NiceLabel Pro applies to NicePrint	8
Compressing the Embedded Images is a Default Option	8
Adapting Aztec Barcode Defaults for Industrial Ink-Jet Printers	9
Form Application Design	9
Setting Object Properties.....	9
Visual Indicator for Actions Defined for “On Variable Change” Event	10
Hiding the Title Bar	11
Using Font Names in Combo Box and List Box Objects	11
Setting the Initial State of an Object.....	12
Option to Allow/Forbid Duplicates Even When Values are not Sorted	13
Description Field for a Variable.....	14

What is New in NiceLabel Designers

Label Design

DotCode Barcode Symbology

DotCode is fairly new 2D barcode symbology, targeting industrial applications. This symbology was designed for the unique identification of objects in a relatively small area, or for direct marking by low precision marking technologies, ink-jet and lasers used in direct part marking (DPM) applications.

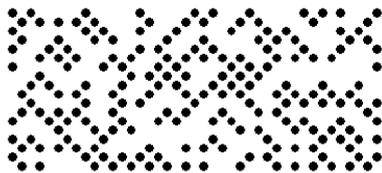


Figure 1: DotCode symbol encoding 'http://www.nicelabel.com'

It is variable size, dot-matrix error-correction symbology. The overall symbol shape is adjustable; rectangular to nearly square shaped symbols are possible.

It is ready for GS1 applications and suitable for encoding application identifiers (AIs). NiceLabel exposes it also as a 'GS1 DotCode' barcode.

Overriding the Default Values of Variables

When you define the default values for variables in the label, they are saved inside the label file and used every time that you do not specify the custom values. To print labels with custom values, your printing application simply sends the values to the label variables just before the printout.

Version 6.3 introduces another possibility to override the default values. You can create a text file with custom values. The file must have the same name as the label file, but with the extension `.defaults`. The file must contain field-value pairs, such as:

```
VariableName1 = Value1  
VariableName2 = Value2
```

Figure 2: Field-value pairs from `.defaults` file override the variable values

Updated Support for Transparent Graphics

NiceLabel can already print the images in transparent mode, when you use NiceLabel printer drivers. Transparent images are without any background, but use a see-through background. You can stack the transparent images on top of each other on the label and the top image will not cut the rectangular shape out of the images lower in the stack.



Figure 3: Support for transparent images (left) compared to the previous version (right)

The new NiceLabel adds support for printing transparent images to non-NiceLabel thermal printer drivers and also to Postscript and full-color printers.

Only some image file formats support transparency. The popular formats are PNG, PSD, and WMF.

Support for Food Allergens Built-in through Custom Functions

The European Union’s mandatory food labeling legislation, formalized in Regulation (EU) No. 1169/2011, not only specifies what information must be shown, but also how that information is presented, including font, color, and contrast. The new regulation deals in detail only with font size, but leaves the exact type of allergen highlighting to the user. The allergens in the ingredient list must be printed in a typeset or font that distinguishes them from the rest of the list (such as bold, italic, underlined, inverse, different color).

NiceLabel’s approach to food allergens printing was introduced several months ago with the VBScript add-on, which was somehow difficult to operate by non-technical users.

NiceLabel Pro 6.3 includes new built-in VBScript custom functions. The food allergen formatting logic is available with a call of the single custom function. The code itself is hidden “under the hood”. A single line of code is needed.

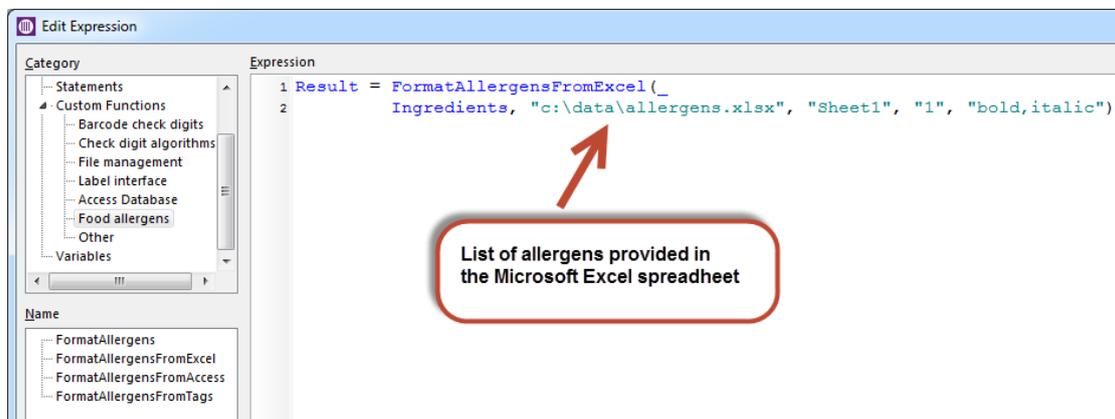


Figure 4: Executing selected food allergen function

The user can execute the function and provide all of the variable information in its parameters, such as the location of the list of allergens and the ingredients. Various custom functions are provided for the different types of databases used, such as Microsoft Excel and Microsoft Access.

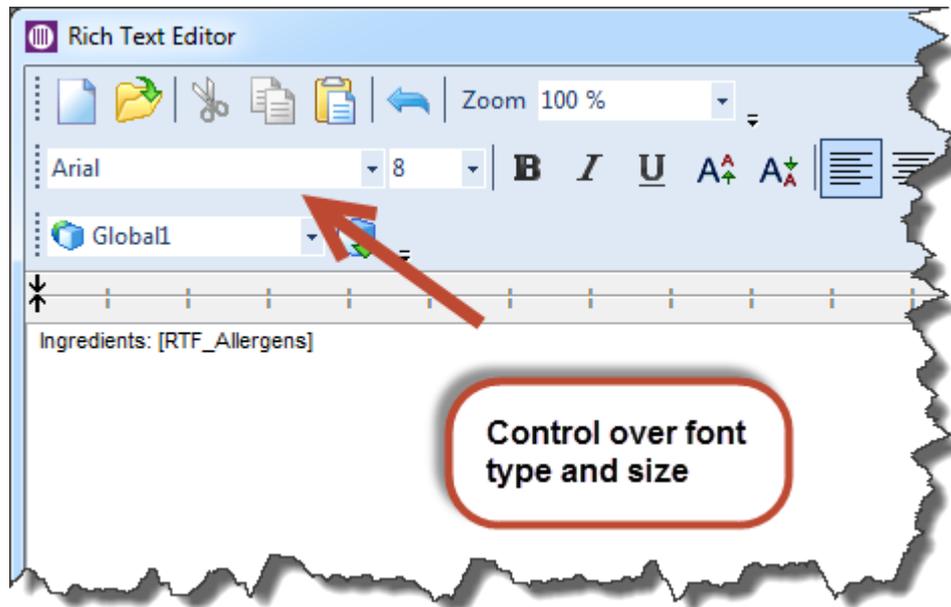


Figure 5: You control the font properties in the Rich Text Editor. You can merge ingredients with other data.

Printing JPEG Images to Postscript Printers in CMYK Color Space

If you own a Postscript printer and use JPEG images with CMYK color space on the label, NiceLabel will send the image to the printer as-is. This will allow for accurate color printouts as no internal conversion from CMYK to RGB and back to CMYK color space will take place internally in NiceLabel and in the printer driver.

When using RGB images or printing CMYK images to non-Postscript printers, the image will still be internally converted to RGB color space.

Relative Positioning of Elements on Both Sides of the Label

NiceLabel already supports the relative positioning of label objects. A relative-positioned object is such that it changes its position on the label on-the-fly when its master object changes size or position.

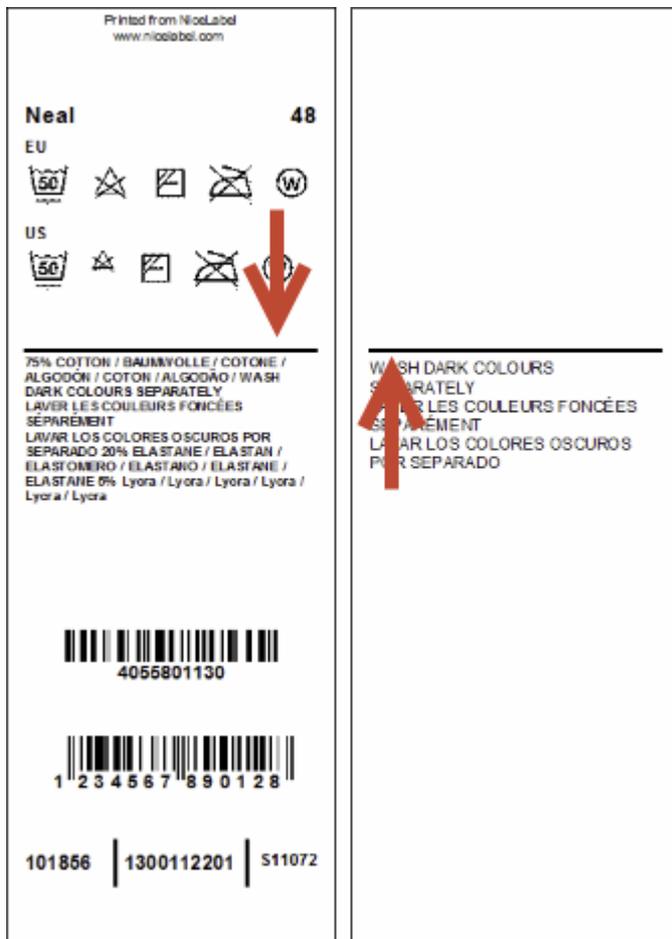


Figure 6: The line on the back side is automatically aligned with the object on the first side

The relative positioning is used to reduce the number of label variations. The same label design (single .LBL file) can be used for printing various products. The objects will change their positions to keep the distances between them and the label size will be resized to accommodate to the amount of data.

With NiceLabel 6.3 you can create a relative link between the objects on both sides of the label. Usually, you will want to link the central object on the back side with the central object on the front page in order to keep the central label seam in the same location on both sides.

The relative positioning and variable label sizes are mandatory features for any successful printing implementation in the textile/garment industries. They will help you keep the number of various label templates in the manageable range.

Importing PDF Files as Images

When the on-demand printed label is used on the final product, the companies might design the graphical label layout using some other software, such as Adobe Illustrator, Adobe Photoshop, or CorelDraw Graphics Suite. The design can then be saved as a PDF file.

In NiceLabel 6.3, you can load the graphics templates from PDF files into the Picture object. When the PDF content loads, all of the vector objects are converted to bitmaps at a quality level matching the target printer resolution.

NiceLabel can then add some last-minute information, such as date/time stamps, LOT numbers, and unique counters and combine it with the graphics template. The final printout is done using NiceLabel technology on a full-color printer.

Support for the Full Justification of Text in an RTF Object

The RTF label object now supports full justification, the same as the TextBox object. When enabling this option, the text within the object will be aligned to the left and right side of the object.

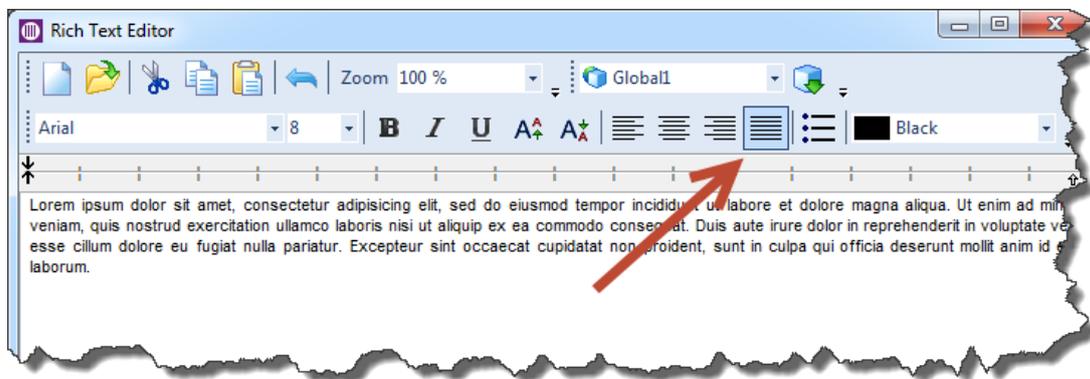


Figure 7: Full justification support in the RTF editor

Support for HIBC Specification version 2.4

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

Earlier this year, HIBCC released a HIBC specification version 2.4, which is now supported in NiceLabel Designers.

Updated List of GS1 Application Identifiers

The GS1 support has been updated with the changes from the GS1 General Specifications Version 14, January 2014.

New GS1 Application Identifiers are supported: AI(16) Sell by date, AI(713) National Healthcare Reimbursement Number (NHRN) - Brazil DRN, AI(8010) Component/Part Identifier (CPID) and AI(8011) Component/Part Identifier Serial Number (CPID SERIAL). AI(8007) International Bank Account Number (IBAN) has updated length of data.

Support for the Custom RTF (Rich Text Format) Control Codes

When you use insert variables into the Rich Text object, their content is used inside the object as-is. The RTF control codes will be interpreted as text and will not be parsed as the control codes. You can enable the parsing of the RTF control codes, so they are executed at the print time. You would use this option if the text formatting was already applied to the content you intend to use in the RTF object, so that you do not have to do it again.

For example, the variable can contain the following value:

```
This will print in \b bold\b0 formatting.
```

And the result will be formatted text at printout:

```
This will print in bold formatting.
```

To enable the parsing of the RTF control codes provided in the value of the variable, the variable name must begin with the prefix "RTF_". This will instruct NiceLabel to interpret RTF codes as such, not as plain text.

This allows for easy merging of formatted data provided by several variables within the same Rich Text Box object.

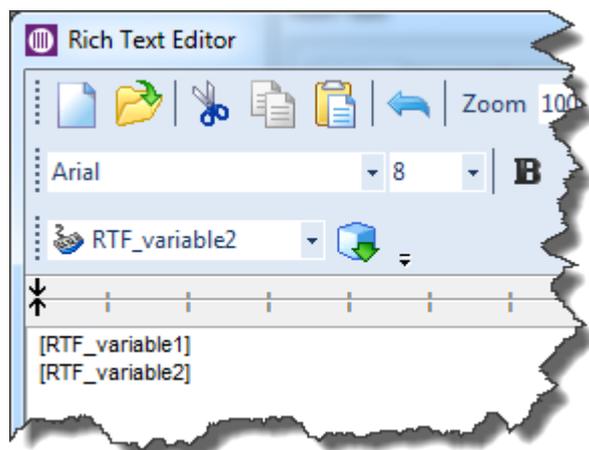


Figure 8: Merging variables within the Rich Text Box itself

Fitting Barcode to Label Size

When a new barcode is added on the label, its size will not exceed the width of the label.

Depending on the size of the label and selected barcode type, the barcode symbol might be drawn outside of the label boundaries. To simplify the label design, each new barcode object placed in the label will be resized to fit in the drawing area.

New Antenna Settings for RFID Support

Passive RFID tags utilize an induced antenna coil voltage for operation. The induced voltage provides a power source for a RFID tag and allows the tag to send and receive the data.

The radio-frequency signal can be effectively radiated when the antenna in the RFID-enabled printer is at the correct distance from the RFID tag embedded in the label.

RFID antenna offset can now also be a negative number for fine-tuning the antenna location.

The user interface introduces a new setting for applying the antenna power, the amount of radio-frequency signal emitted from the printer.

Default Printer Selected in NiceLabel Pro applies to NicePrint

In the label designer (the application NiceLabel Pro) you can already select a 'default printer', which can be used as a preferred printer. NiceLabel default printer should be a printer driver that you frequently use for label printing.

The same printer is now recognized by NicePrint. This is useful in those cases you open a label file that was created for a specific printer driver not available in the current system. In this case, NicePrint will automatically select the default printer defined in the NiceLabel system.

Compressing the Embedded Images is a Default Option

In NiceLabel designer, you can choose to embed the image inside the label (.LBL file). This makes the solution more portable (the images are traveling with the label itself), but also increases the size of the label file. The file size increment also depends on the type of images that are saved inside the label file.

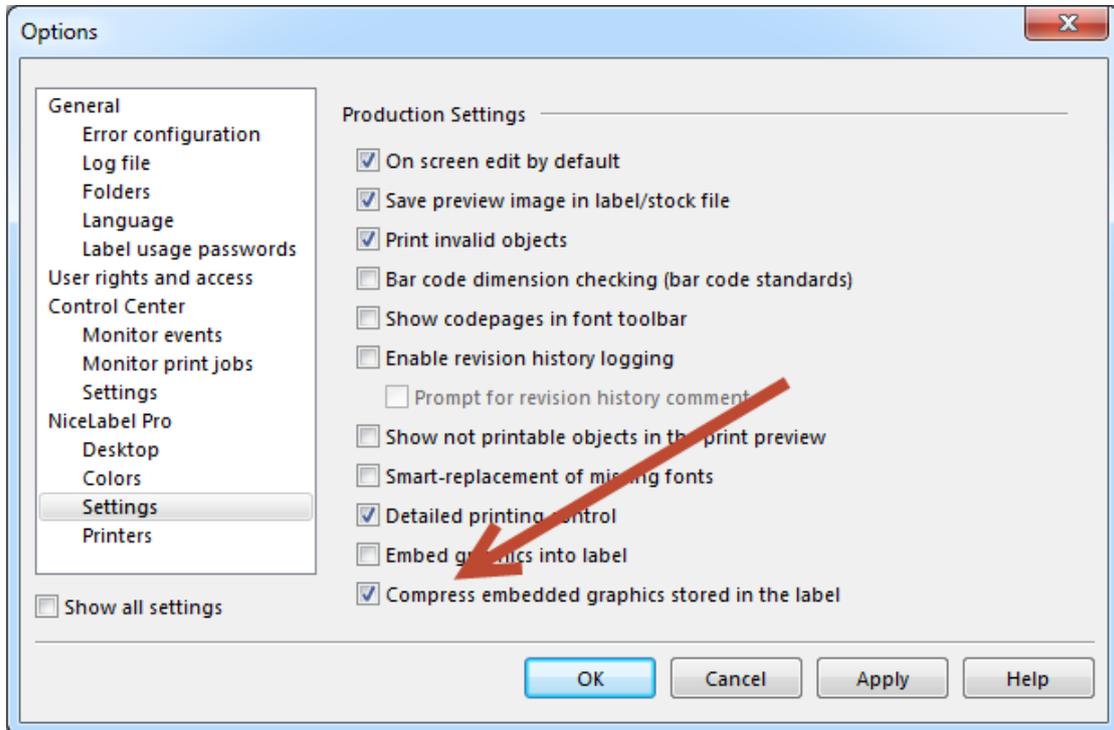


Figure 9: Enabling the compression of the embedded images

NiceLabel already supports the compression encoding mechanism, which greatly reduces the size of the label file with the embedded images.

This option is now enabled by default, ensuring the minimum size of label files.

Adapting Aztec Barcode Defaults for Industrial Ink-Jet Printers

Some low-resolution printers (industrial ink-jet) can print small 2D barcodes with just one ink-drop per barcode element. The minimal expansion for Aztec barcode was 3, which is OK for high resolution printers, but it prevented printing the barcode on some low-resolution printers with small heads.

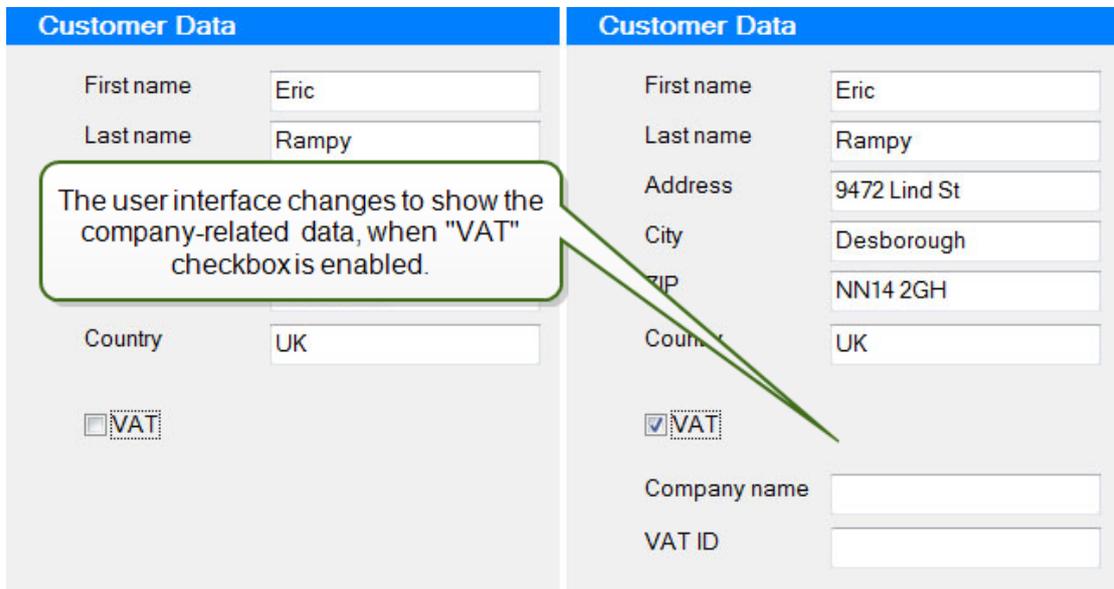
The minimal expansion that is allowed when generating a graphic image of the Aztec barcode was changed to 1.

Form Application Design

Setting Object Properties

The new action Set Object Properties allows you to modify the user interface and functionality of the form application while it is running.

You can create an application that will use a simplified user interface, but it can switch to a more advanced user interface, displaying more options.



Another example: the print button is disabled and the user cannot click it as long as there are problems with the entered data. When the problem has been resolved, the print button becomes available.

This is a powerful new feature that you can use to visually modify your applications while they are already running.

You can change the following object properties on-the-fly:

- Object position
- Object size
- Object enable/disable state
- Object visibility
- Object color
- Object font properties (name, color, size, style)

Visual Indicator for Actions Defined for “On Variable Change” Event

When creating the application in NiceForm, the actions are usually executed by clicking the command buttons. However, more complex applications require actions to run on more occurrences for more specific events, such as “when the application starts”, “when the application stops”, “when the value of some monitored variable changes”, and similar.

The more variables you have defined for your application, the more difficult it is to find the ones that run actions, when their values change.

To help you identify variables with “on variable change” event, NiceForm will display a visual indicator next to them.

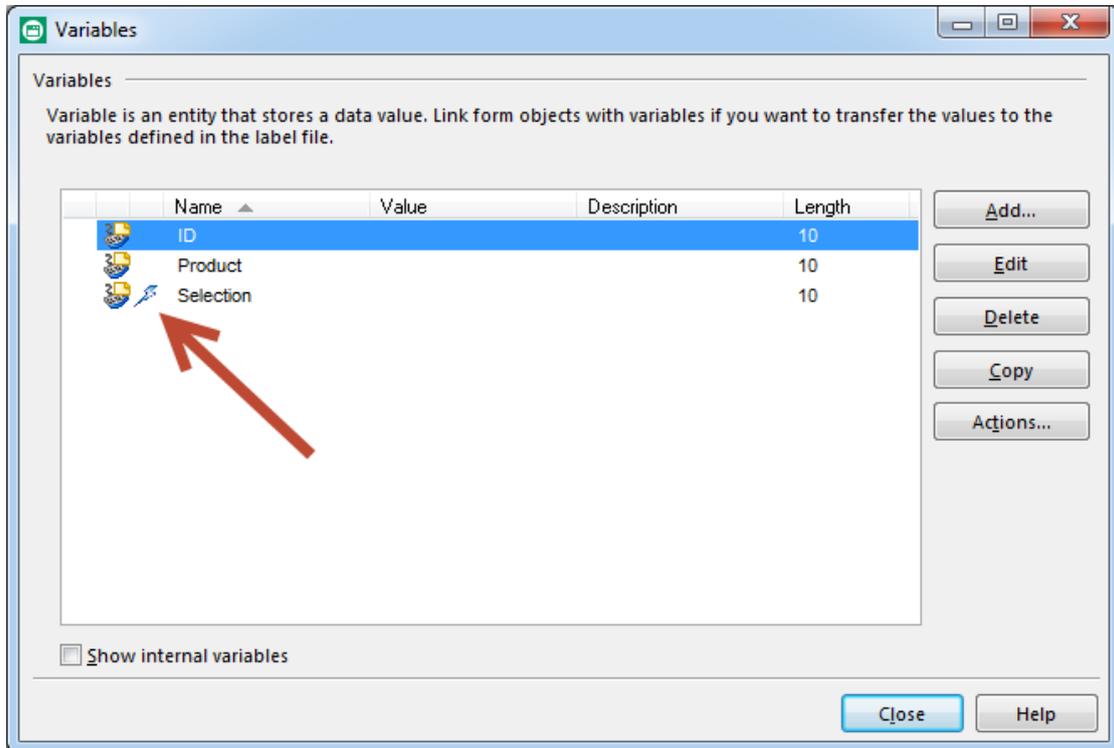


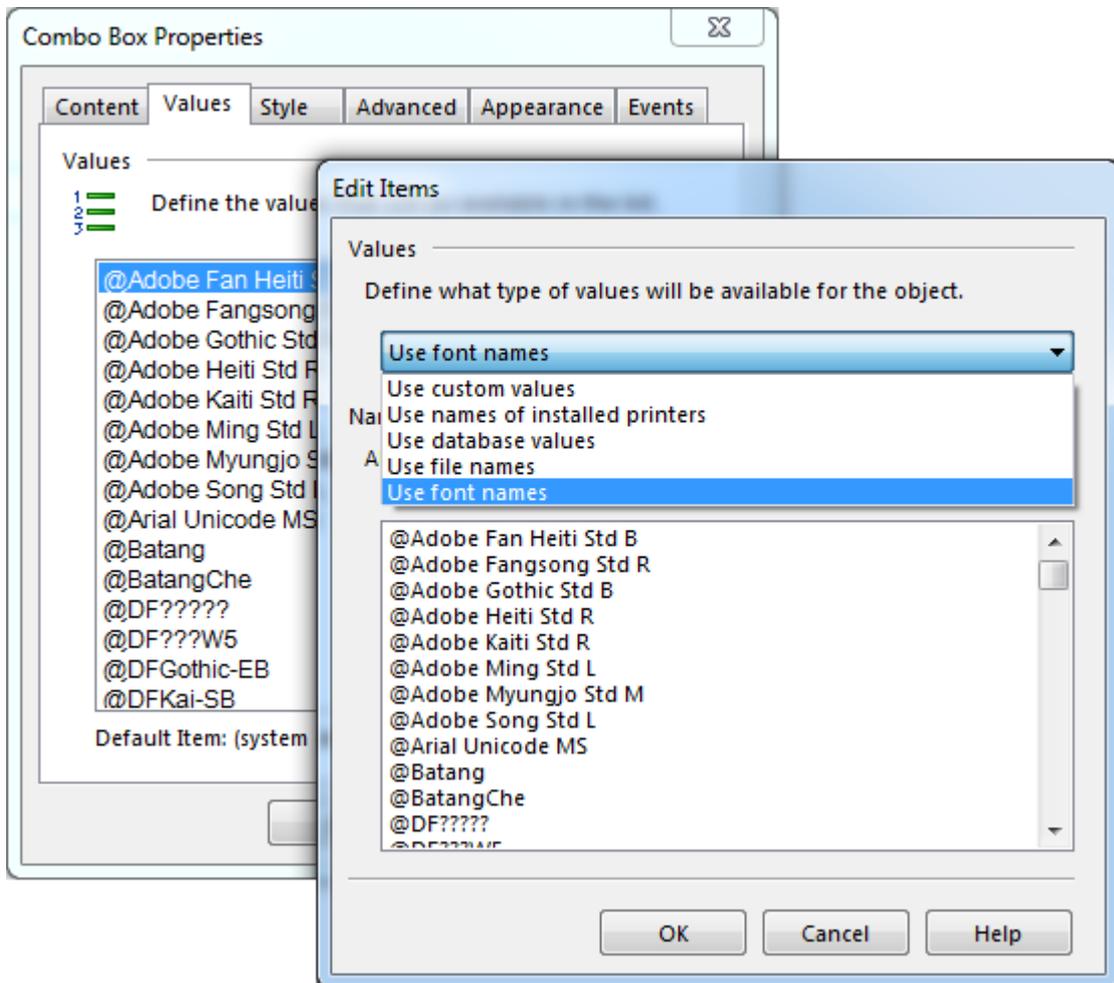
Figure 10: The variables that have defined actions for an *OnChange* event are clearly identified

Hiding the Title Bar

The applications created with the NiceForm application generator can run without displaying the title bar. This feature allows for creating applications that work in a kiosk mode where a title bar is not needed.

Using Font Names in Combo Box and List Box Objects

Combo box and list box objects have a new source of data that they can display. They can collect a list of fonts installed on the local computer and make them available for the selection.

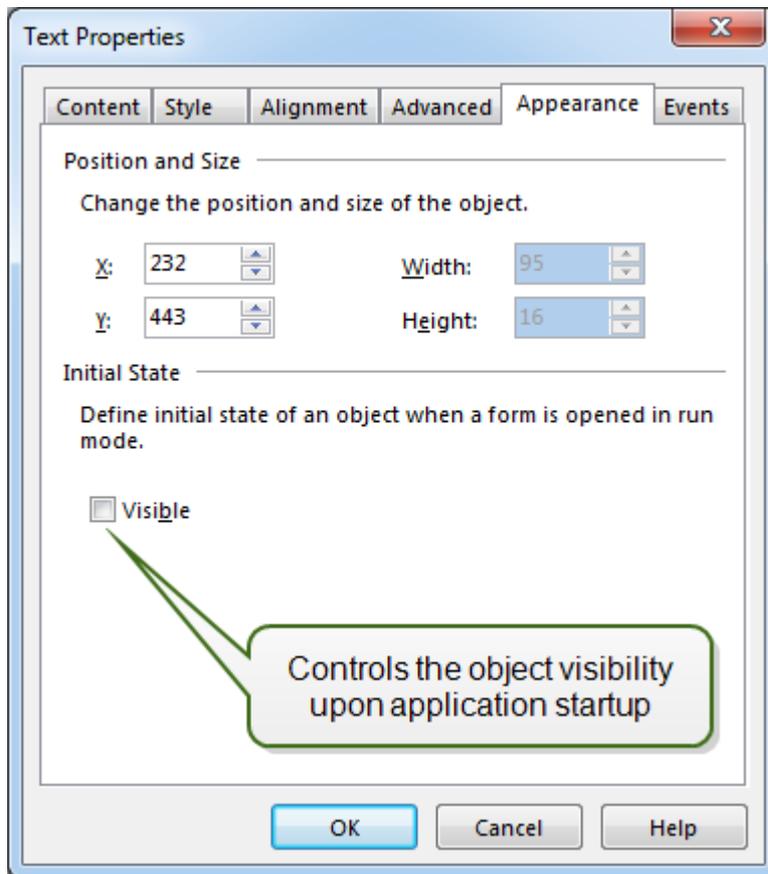


This feature is tightly connected to the new action **Set Object Properties**.

The user can select the font in the combo box and apply it to the objects while the application is running. You can change the font that is used for the button, or in the edit field, or in the text object.

Setting the Initial State of an Object

The form objects now support the visibility property. The objects can be visible or hidden, and you control it through the action **Set Object Properties**.



You will always see your objects in the design mode. However, when switching to runtime, the visibility depends on the logic within the application. You can also control the initial state of each object - whether the object should be visible when a form is started in the run mode or if it should be hidden.

Option to Allow/Forbid Duplicates Even When Values are not Sorted

The Combo box, list box, radio group and button group objects now have support for configuring whether or not duplicates are allowed even when items in these objects are not sorted. Previously, the items needed to be sorted, and the user was not allowed to modify the 'Allow duplicates' property.

The option to remove duplicates is useful when the form objects display values from external third party sources, which you have no control over, such as a customer's databases.

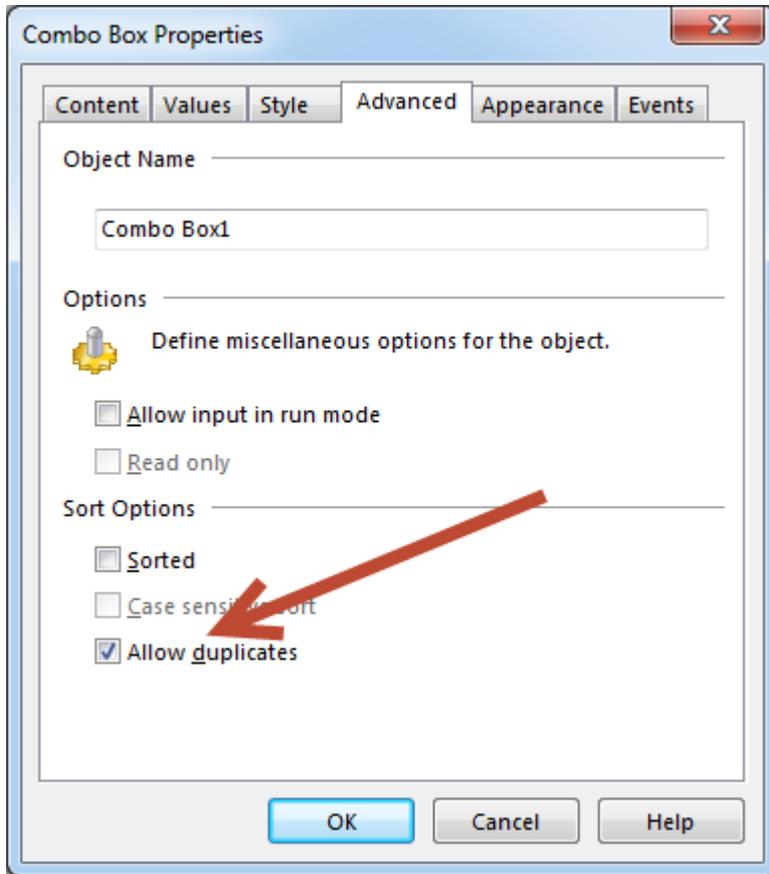


Figure 11: You can remove duplicated values in the list

Description Field for a Variable

The variables in NiceForm now have a description field where a user can enter more information about the variable, such as why the variable is needed, why the specified default value is used, where in the actions it is used and similar descriptions.

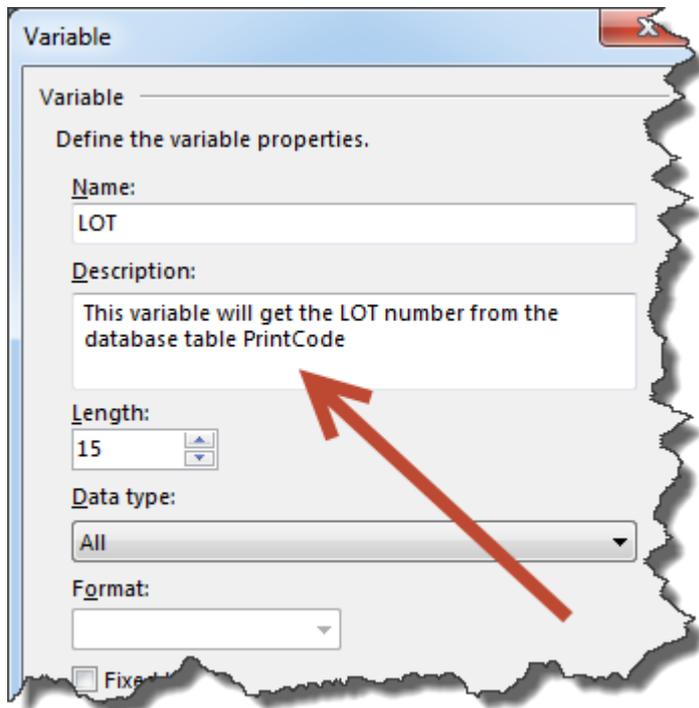


Figure 12: Describing the purpose of the variables