

the professional
labeling software

www.nicelabel.com
info@nicelabel.com



Programming Your System to Include a Labeling Solution

A Programming Integration White Paper

Version 20080418-01

© 2008 Euro Plus d.o.o. & Niceware® International, LLC
All rights reserved.

www.nicelabel.com

1	Executive Summary.....	3
2	API Integration Method (Programming Method)	4
	2.1 Objects, Methods, Properties	4
	2.2 NiceLabel Print Engine(s)	4
3	NiceLabel's Application Programming Interface (API).....	6
	3.1 Using NiceLabel API.....	8
	3.2 NiceLabel Wrapper	8
	3.3 Examples of API Integration	9
	3.3.1 ActiveX Automation	9
	3.3.2 .Net Integration	10
	3.4 Example of Web API Integration	10
	3.5 Example of Pocket API Integration	11
	3.6 More API Examples with the NiceLabel Integration Pack	12
4	NiceLabel Integration Solutions.....	13
	4.1 NiceLabel Developer Series.....	13
	4.1.1 NiceLabel SDK	13
	4.1.2 NiceLabel PocketSDK	14
	4.1.3 NiceLabel WebSDK.....	14
	4.2 NiceLabel Pro	15
5	Conclusion	16
	Appendix.....	17
	Additional Resources	17
	Glossary	17
	Technical Contacts	18

1 Executive Summary

As the global business competition tightens, businesses increasingly refocus their attention to the customer and adopt the “market of one” strategy. This approach has been driving a surge in IT system customization, to the point where many companies run their business on completely customized systems. When faced with the daunting task of adding a complete new functionality to such a system, integration is not only the best, but often the only choice available.

This white paper focuses on the API programming method of printing functionality integration, a method that, while it may not be the easiest of all possibilities to implement, provides the most freedom in its versatility. This method is intended for:

- Extensively customized infrastructures that cannot support a new layer of software, and demand full integration
- IT systems, which demand absolute and unlimited customization of label printing processes
- IT systems, which demand a completely seamless integration of a new functionality into existing applications

This method of integration is the easiest to adapt to and learn by the end-users of the technologies, as the printing solutions are not intricately connected to the existing infrastructure, but rather build into it, effectively becoming a part of it. Indeed, such an approach to integration requires some modification of the current infrastructure and previous programming experience.

The programming integration is based on the premise that for a printing system to be truly and absolutely integrated in an existing IT environment, it must become part of the environment. When this is done optimally, the resulting solutions can avoid changing the processes and operation of existing systems but they expand them to some degree.

Programming integration, compared to other integration methods, is more difficult to implement. It demands advanced programming skills and the ability to insert the NiceLabel printing building-block into the existing software, to maintain it, and to develop it with time as the IT system changes, to adapt to fulfill its needs.

2 API Integration Method (Programming Method)

Windows operating systems provide different methods of communication between applications. NiceLabel supports Windows connectivity features such as ActiveX Automation, Dynamic Data Exchange (DDE) and other mechanisms that allow other applications to take full control of NiceLabel's print engine.

ActiveX is the Microsoft implementation of the COM (Component Object Model), a standard defining the support for reusable, language-independent and multi-platform components. With COM, it is possible for one component to communicate with another and take advantage of its functions. NiceLabel shares its label processing and printing power with an external application through its print engine.

Label design is done in the software edition NiceLabel Pro. Your application embeds the features of NiceLabel print engine within itself so the print engine literally becomes a building-block of your application. Your application issues commands to the print engine and takes total control over it. The communication is bidirectional and you not only have insight into the print engine, but also the print job status and the printer status.

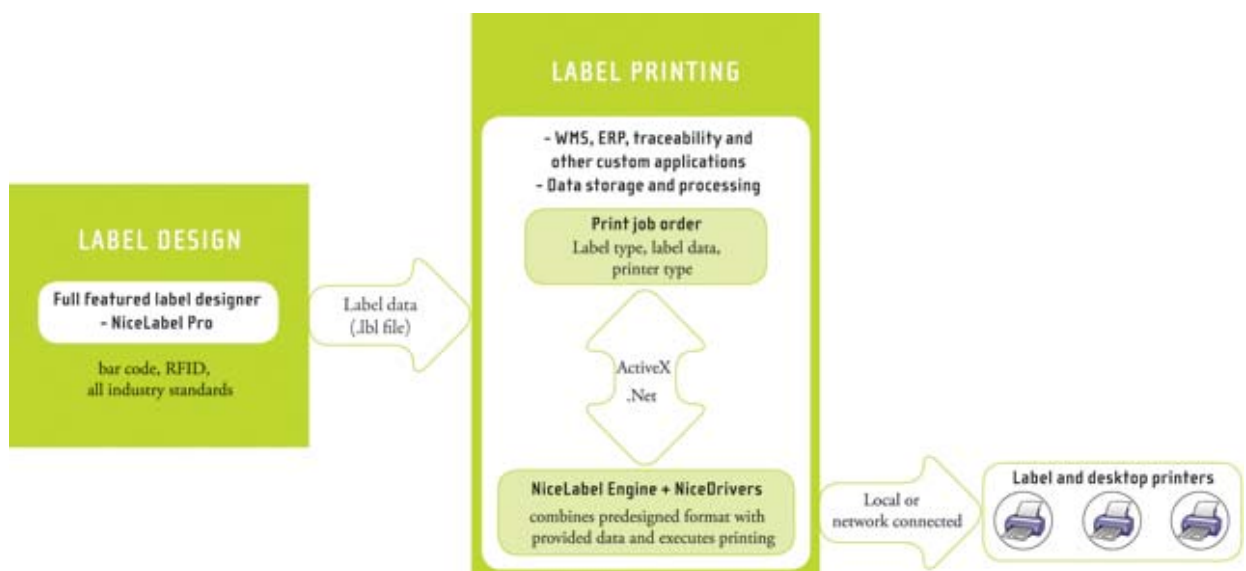


Figure 1: The label design-printing process with NiceLabel print engine

2.1 Objects, Methods, Properties

The 'objects' inside API (Application Programming Interface) that you can change are typically the visible objects on the label, but they can also include NiceLabel database connections, NiceLabel functions, NiceLabel variables etc.

The 'method' is actually the function or procedure of the object. This is the action you can perform on the object, and in some cases, receive a result (feedback).

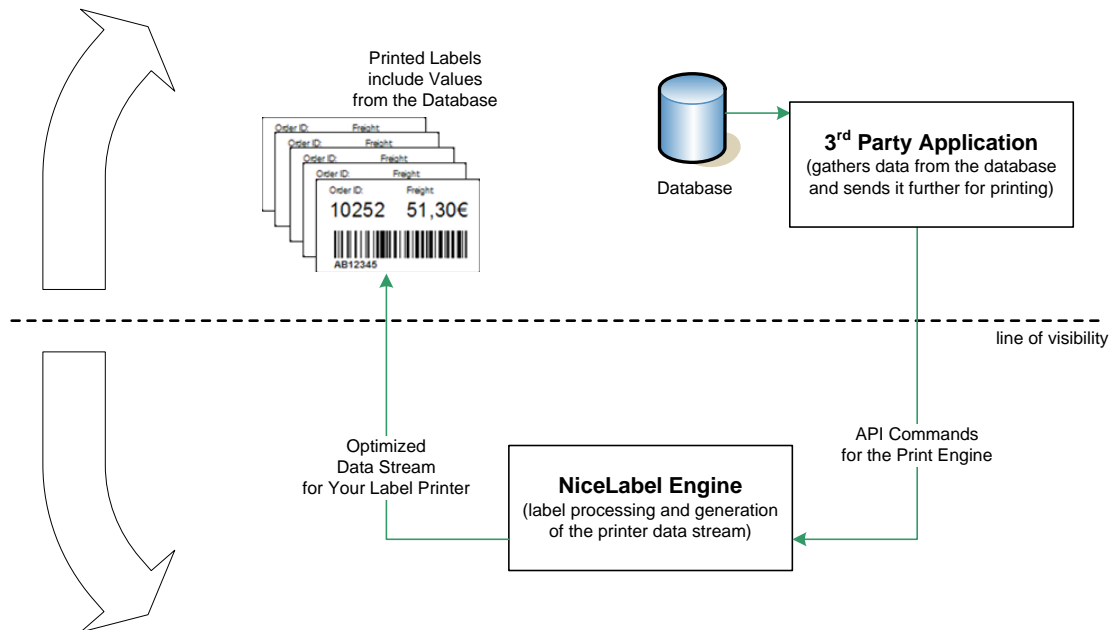
The 'property' is the attribute of the object.

2.2 NiceLabel Print Engine(s)

The print engine is the component that is responsible for the label printing process. It is essentially a small ActiveX building-block that shares the NiceLabel optimal label printing know-how with any application that uses it. The print engine exposes its functions (properties,

methods) with its API. Your application can use the functions available in the print engine, as if those functions were originally a part of the application.

Processes **VISIBLE** to the end user



Processes **NOT VISIBLE** to the end user

Figure 2: NiceLabel print engine runs hidden from the user

Your application controls the label printing process (acquires the label data, sends the data to the print engine, executes label printing), leaving the end user unhindered by the NiceLabel print engine, which is running beneath the application.

Before you can control the NiceLabel print engine, your application must start it by creating an instance of the print engine in the system memory. You can control several print engines at a time, speeding up printing in high-volume label-printing situations. The client application then communicates with the print engine, obtains the list of objects, sees their functions, and changes the values. The available objects and functions are defined by the exposed API calls.

NiceLabel print engines support communication with desktop, mobile, and Web applications.

With NiceLabel API solutions, you can stop worrying about printing the labels, and instead focus on your application, while NiceLabel takes care of optimized printing.

3 NiceLabel's Application Programming Interface (API)

An application programming interface (API) is a source code interface that an operating system, library or service provides to support requests made by computer programs. The NiceLabel API defines all objects, methods and properties which the client application can use, as if they were included in its own native code.

In most cases, the client application commands the print engine to open the label, set values of variable fields, change printers, and print out labels. The API, on the other hand, allows you to modify the label design of your label to some extent through API calls. You cannot create a new label or put new objects on the existing label with the API, but you can modify some of the formatting and placement of the existing objects.

Some methods and properties of the NiceLabel API are accessible in read-only mode and only allow you to verify the current definition/value/state, while other methods act as properties, which you can modify from within your application.

NiceLabel introduces two API modes, level 1 and level 2.

- Level 1 (basic method) allows you to open the label, set the variable values and print it out. This method is useful, if you do not need a label preview and never need to change the label design.
- Level 2 (advanced method) allows you to open the interface to the label and see the entire label structure exposed. You can have greater influence on the label design and object's properties (formatting, values, placement), and you can see the label preview.

For details about available API calls in either of the methods, see the NiceLabel Programming Guide.

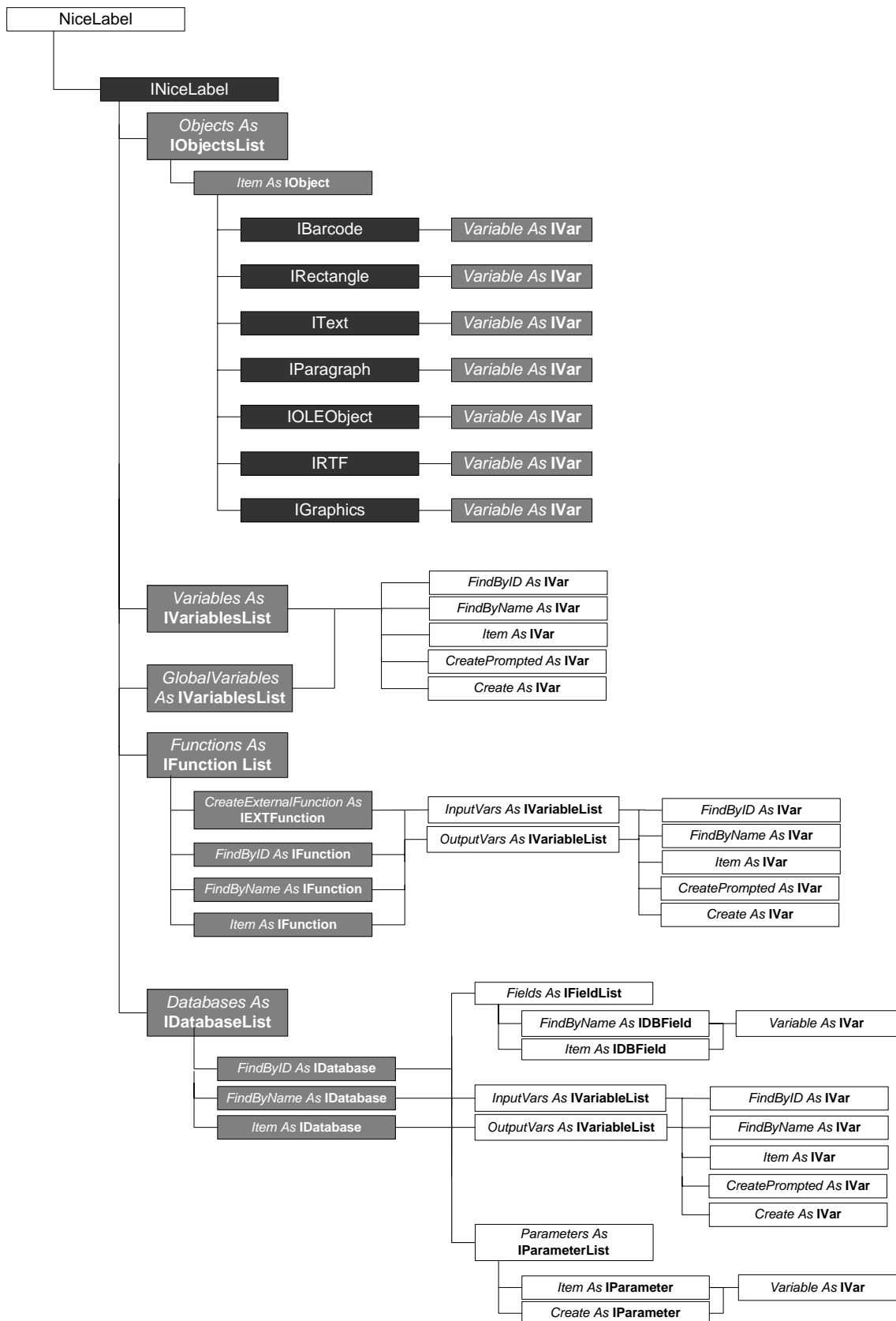


Figure 3: NiceLabel's level 2 (advanced method) API exposed

3.1 Using NiceLabel API

Typically, you would use NiceLabel API in the following cases:

- You are a programmer and have access to the application source code.
- You develop applications for Windows desktop, Windows mobile, or Web environment.
- You develop the software that lacks optimal label-printing capabilities, but still need to print the labels to different label printers.
- You do not want to invest money/time/resources into learning specific printer programming languages in order to be able to print out labels. You do not want to learn how to print different bar codes, or how to comply with the different data-encoding standards.
- All you need is the software you will use for the label design (NiceLabel Pro), and the component that you will control to print out the same labels (NiceLabel print engine) with data coming from your application.

The usual steps of implementation of the NiceLabel print engine are:

1. **Define a label:** Before you can print the label using the print engine, you must create it in NiceLabel Pro and test its printing.
2. **Create a reference (type library):** The type library is the file that contains the descriptions of NiceLabel API. All available objects, methods and properties are described in there. You must import the type library into your developer platform (Microsoft Visual Studio, Borland Delphi, etc.) The entire NiceLabel then becomes accessible.
3. **Create an instance of the print engine:** You must define the link to the NiceLabel API so you can use an instance of NiceLabel print engine and all its available functions from your application.
4. **Program the code:** Use the objects, methods, and properties in the source code of your application. This is where you use commands to open the existing label, set variable values to some values, select label printer and print the label.
5. **Check for problems:** When you issue a certain command, you can always verify the command status and print engine status. If the error appears for whichever reason, you will receive the error ID from which it is easy to determine what went wrong. Tracking error IDs also allows you to display error messages to the user.

One of the more advanced features in the NiceLabel API is the ability to preview the label template in a window within your application. It is a very powerful option as it allows the user to confirm the actual label design before he decides to print the label.

As an example of the NiceLabel API power you can take a look at NiceForm, the front-end application generator available in NiceLabel Suite edition. NiceForm is the front-end application; it collects the data from the user, and prints the selected label to selected printer. NiceForm uses the NiceLabel API in the same way as you can from your application.

3.2 NiceLabel Wrapper

NiceLabel natively supports the ActiveX integration method. If you develop ActiveX applications, you can communicate directly with the print engine and command its actions.

However, in the last years the applications based on the Microsoft .NET Framework are becoming more popular. .NET applications cannot communicate directly with the print engine. As a solution NiceLabel print engine ships with the additional component (wrapper) that acts as

a mediator between your .NET application and the print engine. The wrapper has its own ActiveX interface to which you connect from your .NET or ActiveX applications.

The wrapper has a more »modern« ActiveX interface that also offers support for the early binding communication between your application and the print engine.

The benefits of the early binding communication method:

- The code executes faster, your application will perform better.
- Support for Intelli-Sense, you can use code-completion to write the code faster.
- Support for debugging with syntax control.

3.3 Examples of API Integration

3.3.1 ActiveX Automation

The integration process is often referred to as COM, ActiveX, or OLE Automation. The ActiveX Automation interface can be understood as a simple call of a function within your own program. The only difference is that the function is located in another program or .DLL file and not within your own program code.

In your program code, you must first open an instance of the NiceLabel print engine, then open the label, set the label variables, and print out the label. This is the typical workflow; however, the exposed API allows you to perform additional actions with the print engine. For more information, see the NiceLabel Programming Guide.

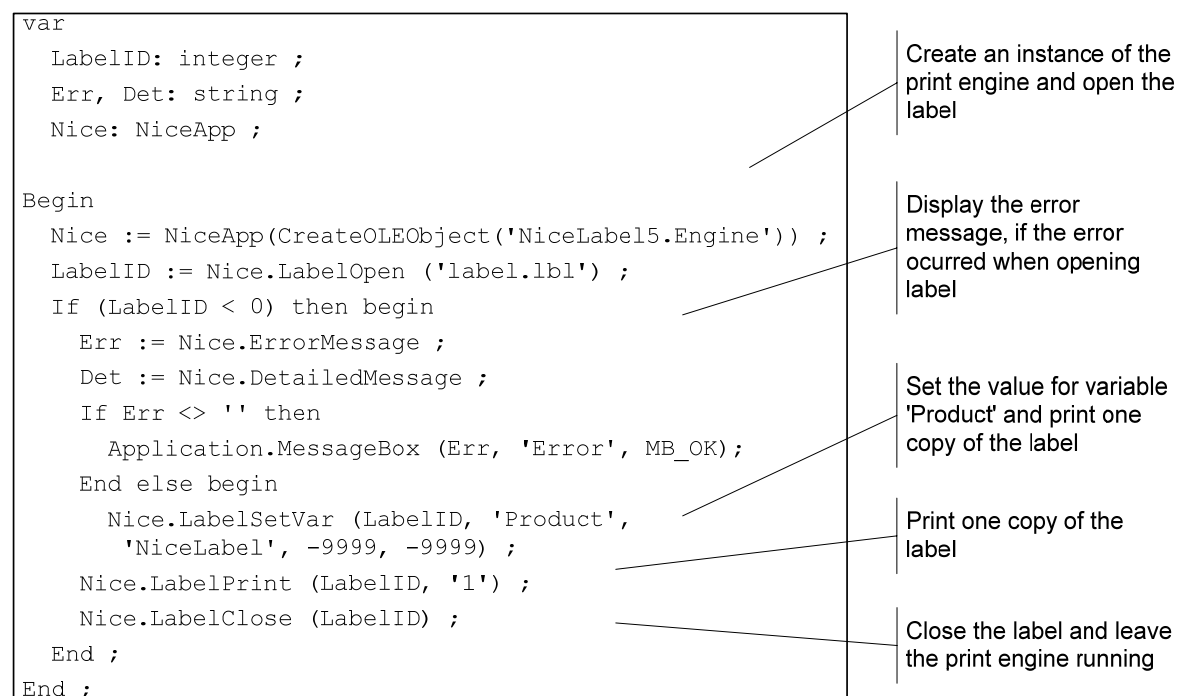


Figure 4: Sample ActiveX Integration from Borland Delphi

3.3.2 .Net Integration

The NiceLabel ActiveX interface can also be used for integration to .NET applications. The add-on wrapper handles proper data interchange between the NiceLabel ActiveX and .NET interfaces.

The sample code shows the following: open a label, retrieve a label variable by name and set its value, use a specific printer and print five (5) labels.

```
'Create an instance of NiceLabel print engine and the Label object
Dim nlApp As New NiceEngine5WR.NiceApp
Dim nlLabel As New NiceEngine5WR.NiceLabel =
nlApp.LabelOpenEx("C:\Sample.lbl")

'Get collection of variables from label
Dim nlVarList As NiceEngine5WR.VariableList = nlLabel.Variables

'Set values for specific variables
Dim tmpVar As NiceEngine5WR.WRVar
tmpVar = nlVarList.FindByName("ProductID")
tmpVar.SetValue("EX100-2")

'Set printer and print 5 labels
nlLabel.PrinterName = "ZEBRA LP 2844-Z"
nlLabel.Print("5")
```

Figure 5: Microsoft Visual Basic.NET Sample

3.4 Example of Web API Integration

With NiceLabel print engine you can print labels using Web browser as your main tool, without any NiceLabel software or even NiceDrivers (printer drivers) installed on the client computers. If you are a developer of ASP.NET applications running on a Windows operating system under Internet Information Services (IIS), you can control the print engine through the provided Web service.

The Web service is capable of the following features:

- Get a list of available printers. The web service will check the system and return a list of all printers installed on the server (where NiceLabel WebSDK is running)
- Get a list of available labels. The web service will return a list of labels (LBL files), which are available on the server and can be used for web printing.
- Get a list of label variables. When the user selects a label to be used, the will return a dataset of variables and their properties, which are available for that label.
- Create a label preview. When the Web application has gathered the variable data from the user, it can make requests to generate the label preview.
- Create a print stream. When the Web application has gathered the variable data from the user, it can make request to generate the print stream. In this step, NiceEngine and a Windows printer driver will be used to generate the print stream.
- Send a print stream to the client: When available, the print stream is sent to the client computer where the ActiveX/Java component inside the Web browser intercepts it, and sends it directly to the selected port, to which the printer is connected.

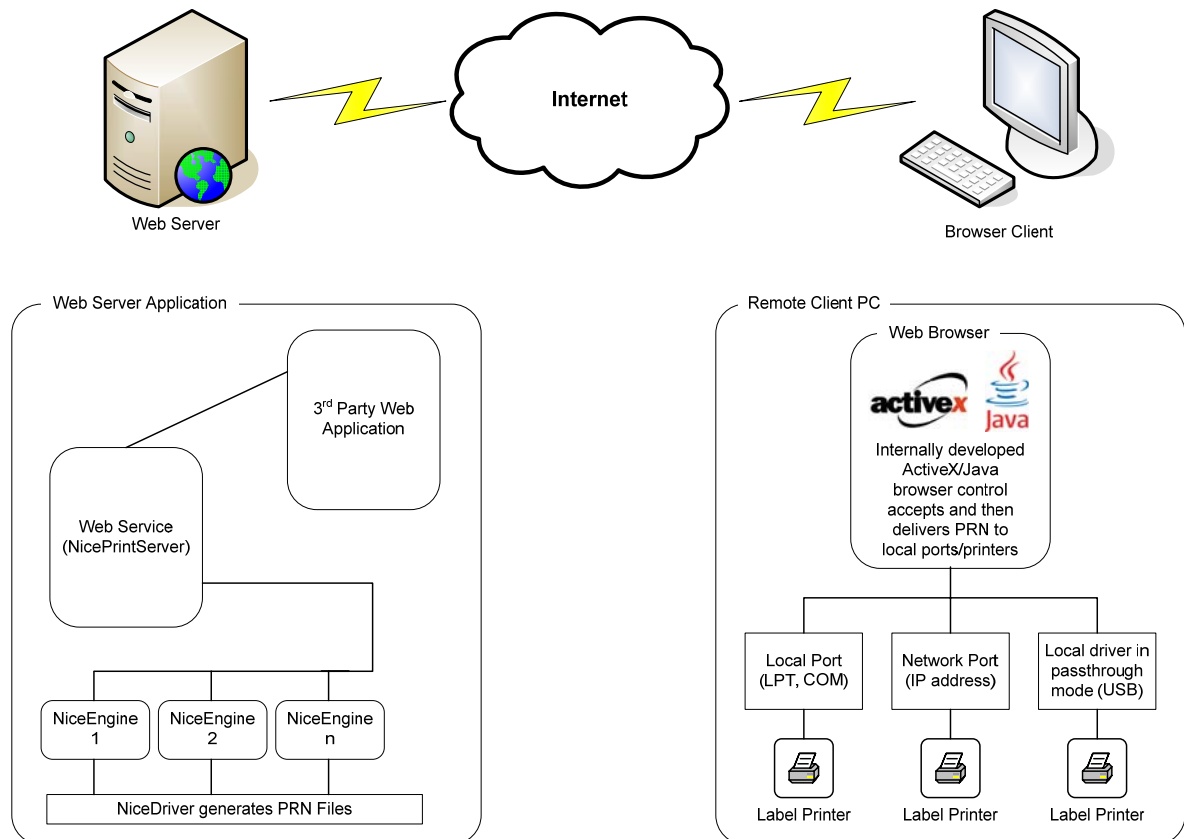


Figure 6: Architectural overview of NiceLabel Web printing

NiceLabel WebSDK Programming Guide provides a sample Visual Studio 2005 Web application that demonstrates how to perform various common tasks. This sample includes solution, project files and complete source code. See the NiceLabel WebSDK Programming Guide at <http://www.nicelabel.com>

3.5 Example of Pocket API Integration

Pocket API provides the methods to print the labels from applications on your mobile Windows devices. Your application creates the instance of Pocket NiceLabel print engine and executes the label printing actions in a similar way than Windows API or Web API print engines.

```
'Create and initialize Pocket NiceLabel Engine
pneEngine = PocketNiceEngine.EngineClassFactory.CreateEngine
pneEngine.Init()

'Open the label file
pneLabel = pneEngine.OpenLabel(LabelPath)

'Get variables from label
pneVariables = pneLabel.LabelVariables

'Get first variable from label
pneVariable = pneVariables.GetAt(0)
'Set lblPrompt's text to the variable's Prompt
lblPrompt.Text = pneVariable.Prompt
```

```
'Set variable's value to user textbox value
pneVariable.Value = txtVarValue.Text

'Create new output port
pneOutput = PocketNiceEngine.EngineClassFactory.CreateOutput
pneOutput.Kind = PocketNiceEngine.OutputKindType.TcpIP
pneOutput.SetTcpIp(IPAddress, TCPPort)

'Set label's output
pneLabel.Output = pneOutput

'Send 1 label to printer
pneLabel.Print("1")
```

Figure 7: Microsoft Visual Basic.NET Sample for NiceLabel Pocket print engine

3.6 More API Examples with the NiceLabel Integration Pack

Additional API samples are available at <http://www.nicelabel.com>. The samples are available in the software package NiceLabel Integration Pack. The Integration Pack will help you learn how to embed NiceLabel print engine into your applications with ActiveX and .NET programming interfaces directly from the programming examples. The samples are available for Visual Basic, Delphi, C++, and other development platforms.

The NiceLabel Integration Pack will also provide the following:

- Sample projects using the desktop NiceLabel print engine from different development environments
- Sample projects using the mobile NiceLabel print engine (Pocket NiceLabel)
- Wrappers for embedding the NiceLabel print engine into .NET applications
- NiceLabel and Pocket NiceLabel ActiveX Programming Guides

4 NiceLabel Integration Solutions

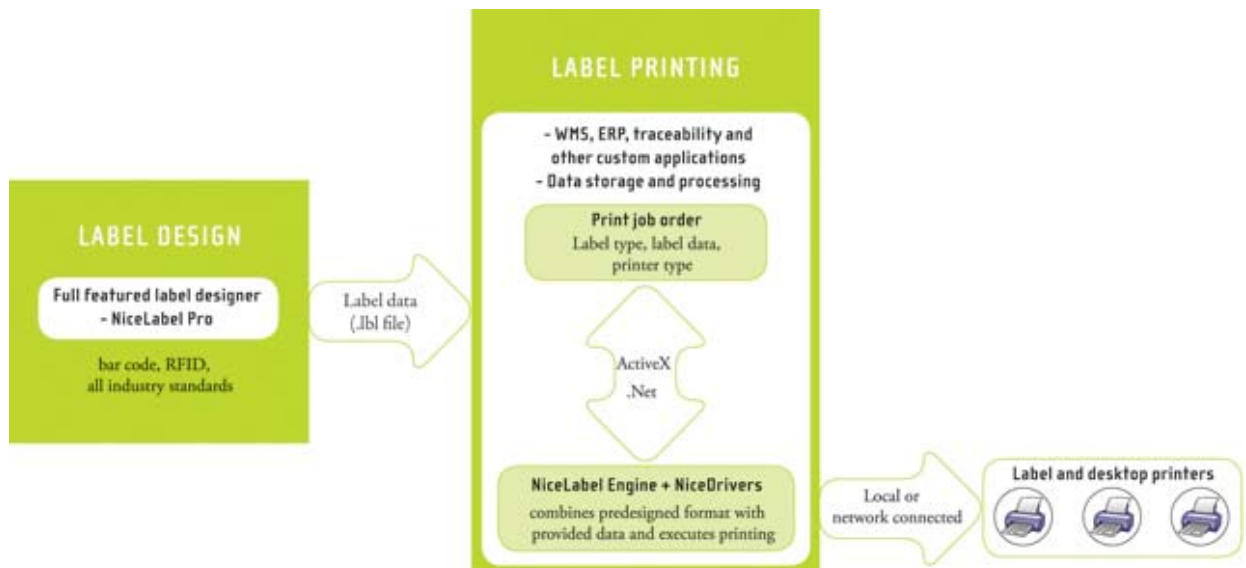
4.1 NiceLabel Developer Series

NiceLabel Developer Series is a line of products, which provides tools to integrate bar code and RFID label design and printing with thermal printer support into 3rd party software applications. It addresses development needs for label printing from desktop, server, mobile or Web-based software applications.

Integration of label printing into your software application offers your software additional functionality, and the end user does not even see the NiceLabel interface, because the software application of your choice uses NiceLabel label printing power in the background.

NiceLabel print engine is available in three different editions, based on the user requirements. Basically, you can print labels from the desktop applications, from mobile applications and from Web applications.

4.1.1 NiceLabel SDK

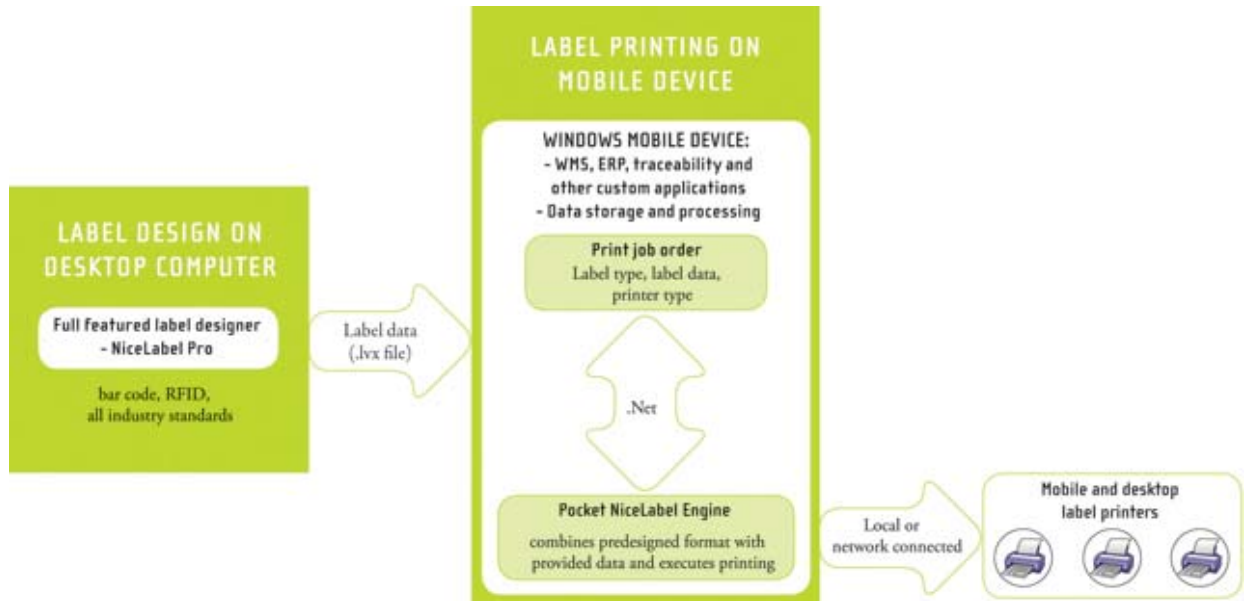


NiceLabel includes full label design functionality of NiceLabel Pro and deployment rights for the easy-to-distribute NiceLabel print engine that is embedded in your desktop application.

The NiceLabel SDK offers the following benefits:

- Powerful label printing engine capable printing labels of any complexity (including bar codes and RFID)
- Optimal printing to any label printer
- User friendly stand-alone label designer available

4.1.2 NiceLabel PocketSDK

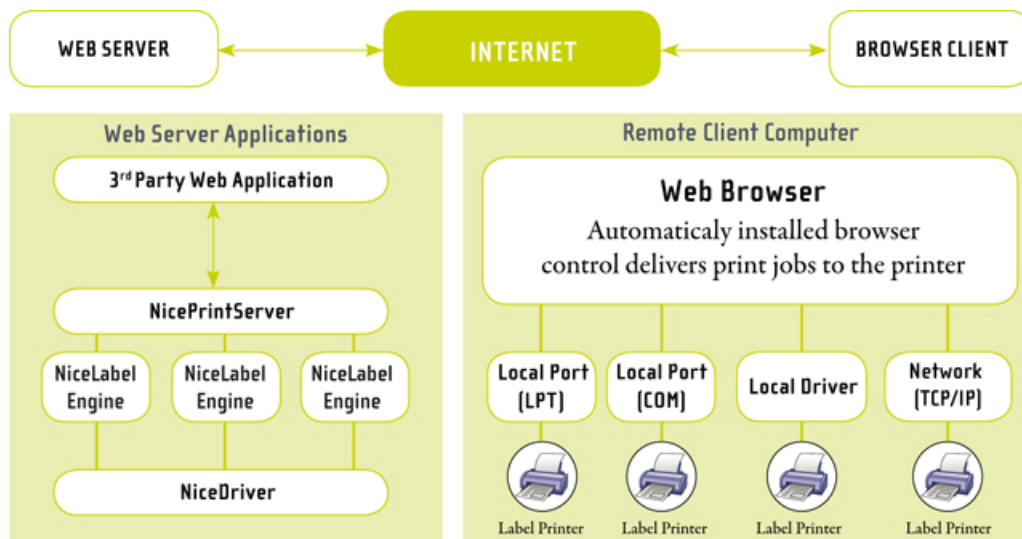


NiceLabel PocketSDK includes all major^{*} label design features of NiceLabel Pro and deployment rights for the easy-to-distribute Pocket NiceLabel Engine that is embedded in the software publisher's application for Windows Mobile devices.

The NiceLabel PocketSDK offers the following benefits:

- A powerful printing engine for thermal label printers (adding a missing printing support to Windows mobile platform)
- Print labels of almost any complexity (including bar codes and RFID)
- Easy integration and deployment without much of a learning curve

4.1.3 NiceLabel WebSDK



NiceLabel WebSDK enables developers to programmatically access the NiceLabel SDK on a Web server and send print jobs to printers connected to a remote desktop (Web printing).

^{*} The print engine available with NiceLabel PocketSDK is a bit under-powered NiceLabel print engine. It does not support all the design features available with the desktop print engine. For example: limited set of functions is supported, VB scripting is not available, variable label objects must be supported internally by the printer, etc.

NiceLabel WebSDK eliminates the need for NiceLabel or a printer driver to be installed on the end user's client computer.

The integration of NiceLabel printing services into an existing Web application is done with the help of NiceLabel Services, which enable the developer to create a comprehensive Web printing solution, including retrieving sets of labels, lists of printers, setting the variable data for the labels, and executing print actions.

Key benefits of NiceLabel WebSDK:

- Powerful label printing engine integrated through a web service interface
- Printing from the web browser without any manual local software installation
- Optimal printing to any local or server connected printer.

4.2 NiceLabel Pro

NiceLabel Pro is labeling software for professional label design and printing. NiceLabel Pro is an easy-to-use, wizard-driven tool for any labeling requirement and includes complete database support, full integration capabilities, RFID smart labeling, compliance labeling, Unicode and multi-language labeling and more.

NiceLabel Pro in the same way as NiceLabel print engine contains the exposed API, so you can also use it in place of the print engine. Compared to the print engine, NiceLabel Pro will occupy more system memory, it needs to load the code for menus, toolbars, etc, even if you will never use these feature while NiceLabel Pro is used for label printing thru API calls.

The wrapper is also available for NiceLabel Pro and it can be used in a .NET environment.

5 Conclusion

The NiceLabel software family offers integration solutions for bar code and RFID smart label printing in any kind of desktop, mobile and network environment. System Development Kits (SDK) solve the problem of label printing on any platform.

API integration is the most complex but also the most powerful label printing integration method. Using programming you can enjoy flexibility and deployment options that are only limited by your imagination and programming skills.

The 3rd party applications need label printing feature, and when quick implementation is required to answer to business needs, the developer won't add the missing functionality on his own, but will look for pre-made building blocks of the necessary code (components) and include them into his application. The NiceLabel print engine knows all about optimal bar code and RFID label printing to all available label printers and makes a perfect choice as a solution provider. The print engine listens for the commands coming from the host application and executes them in order, acting as an independent solution, as well as an integral part of the broader system.

If you want to learn more about the different label printing integration methods or the NiceLabel product range in general, see the literature referenced in the Appendix of this white paper or visit the NiceLabel Web site at www.nicelabel.com.

Appendix

Additional Resources

Additional documentation is available, detailing the programming methods of labeling integration and label design procedure. As any detailed documentation of these processes largely depends on the individual solution, the examples and workflows in this level of documentation is based on NiceLabel products and the methodology of their approach to professional label printing and integration. The documents are available at <http://www.nicelabel.com/Learning-center>.

Label Integration White Papers	<ul style="list-style-type: none"> ▪ White Paper: The Concerns and Benefits of Label Printing Integration ▪ White Paper: Middleware Integration of Label Printing Into Existing IT Systems ▪ White Paper: Designing and Exporting Labels to Your Existing IT Infrastructure
Programming Guides	<ul style="list-style-type: none"> ▪ NiceLabel Programming Guide ▪ Pocket NiceLabel Programming Guide ▪ NiceLabel WebSDK Programming Guide
NiceLabel Developer Series User Guides	<ul style="list-style-type: none"> ▪ NiceLabel SDK User Guide ▪ NiceLabel PocketSDK User Guide ▪ NiceLabel WebSDK User Guide
Pocket NiceLabel Mobile Devices:	<ul style="list-style-type: none"> ▪ White Paper: Mobile Wireless Printing using Pocket NiceLabel ▪ Pocket NiceLabel Quick Start Guide
NiceLabel Pro:	<ul style="list-style-type: none"> ▪ NiceLabel Quick Start Guide ▪ NiceLabel Pro User Guide

Glossary

ActiveX	<p>A type of COM component that is also known as an "ActiveX control". In addition to COM objects, an ActiveX control usually also implements some of the standard interfaces for embedding, user interface, methods, properties, events, and persistence.</p> <p>Because ActiveX components can support the OLE embedding interfaces, they can be included in web pages. ActiveX components can be used from languages such as Visual Basic, Visual C++, Java, VBScript.</p>
Early Binding	<p>A method to implement OLE Automation to communicate with another application. Using the early binding method you set the connections between two applications early in the process, i.e. during design time. In late binding, the application looks for the connection during run time. In general, early binding is the preferred method and executes the code faster.</p>
Software Development Kit	<p>A set of development tools that allows a software</p>

(SDK)

engineer to create applications for a certain software package, software framework, hardware platform, computer system, operating system or similar.

Technical Contacts

Head Office

Euro Plus d.o.o.

Ulica Lojzeta Hrovata 4c
SI-4000 Kranj, Slovenia
+386 4 280 50 00 Tel
+386 4 233 11 48 Fax

www.europlus.si
info@europlus.si

North American Office

Niceware International, LLC

10437 Innovation Drive, Suite 147
Milwaukee, WI 53226
+1 414 476 6423 Tel
+1 414 476 7955 Fax

www.nicewareintl.com
sales@nicewareintl.com
support@nicewareintl.com

French Office

Cobarsoft SARL
Le rempart
32320 Montesquiou
France
+33 (0) 562 709 201 Tel
+33 (0) 562 708 004 Fax

support@nicelabel.fr
www.nicelabel.fr