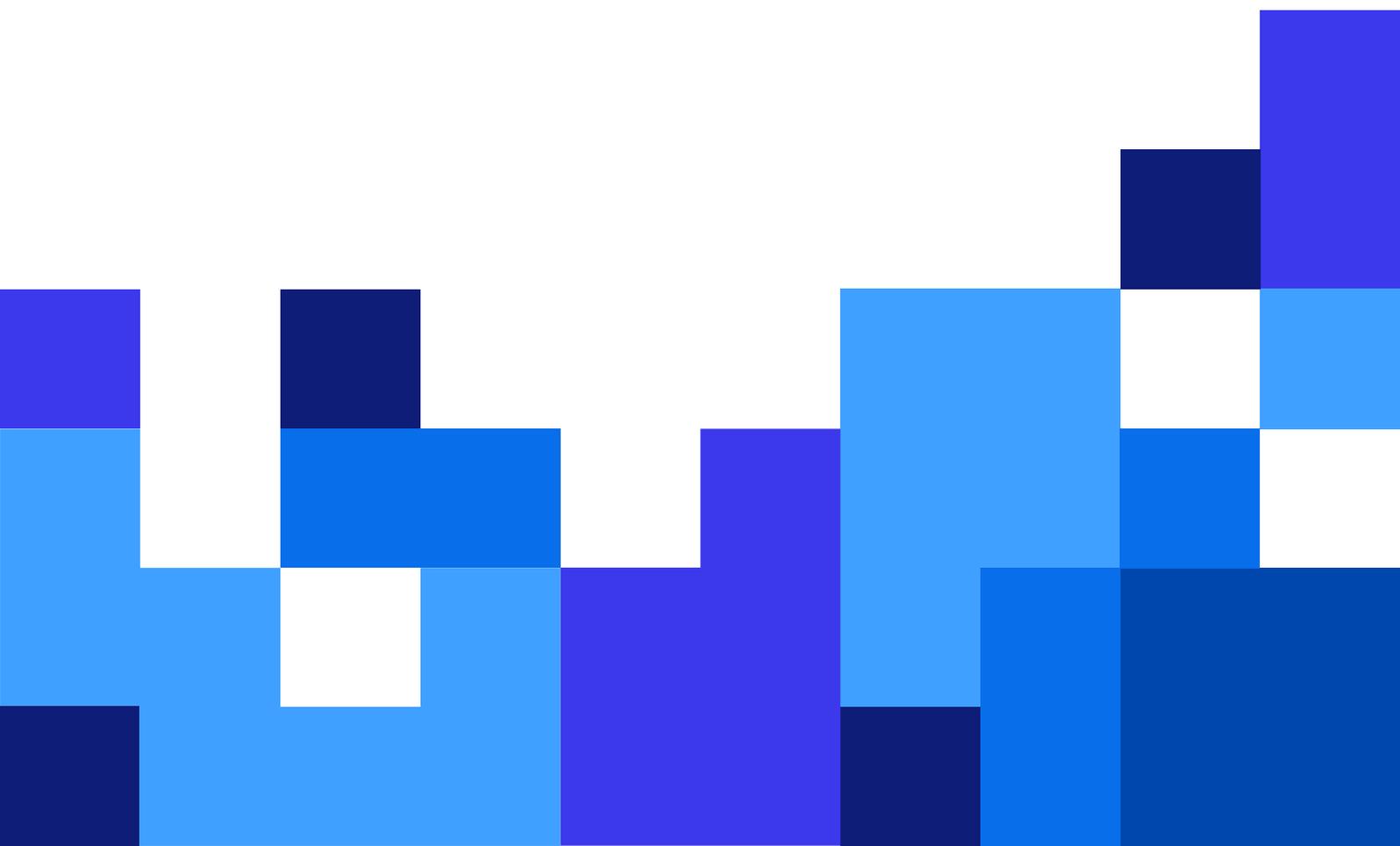


# .NET API User Guide

Rev-2022-4



# Table of Contents

<b>1. Introduction</b> .....	<b>3</b>
1.1. Who Should Use This Guide? .....	3
1.2. Typographical Conventions .....	3
1.3. Sections .....	3
1.4. What is NiceLabel .NET API .....	4
1.4.1. Introduction to NiceLabel .NET API .....	4
1.4.2. NiceLabel .NET API Components .....	4
<b>2. Deploying and Licensing</b> .....	<b>6</b>
2.1. System and Software Requirements .....	6
2.2. Deploying NiceLabel .NET API .....	6
2.3. Licensing .....	7
<b>3. Using NiceLabel .NET API</b> .....	<b>8</b>
3.1. Getting Started .....	8
3.2. Referencing NiceLabel .NET API .....	8
3.3. Label Design .....	8
3.4. Typical Workflow Tasks .....	9
3.4.1. Print Engine Management .....	9
3.4.2. Label Template Management .....	9
3.4.3. Printer Management .....	10
3.4.4. Graphical Preview and Digital Archiving .....	10
3.4.5. Label Printing .....	10
3.4.6. Print Job Monitoring .....	11
3.4.7. Exception Handling .....	11
3.4.8. Print Logging .....	12
<b>4. Control Center Integration</b> .....	<b>13</b>
4.1. Document Management .....	13
4.2. Logging .....	13

# 1. Introduction

## 1.1. Who Should Use This Guide?

This guide is intended for software developers who want to programmatically integrate printing of NiceLabel label templates into their own application. While such approach allows seamless integration into third party applications, it requires understanding of .NET Framework technologies and object oriented programming techniques. You will also need experience with C# programming language and Visual Studio to understand and use the sample applications. For easier non-programmable printing integration, consider using NiceLabel Automation integration module instead.

## 1.2. Typographical Conventions

Class names are displayed as `Consola.Font.Red`.

File names and properties are written as `filename.ext`.



### NOTE

Points of special consideration appear with green backgrounds.

## 1.3. Sections

This guide is organized into the following sections:

- **What is NiceLabel .NET API?** Presents an overview of the NiceLabel .NET API.
- **Deploying and licensing.** Describes the installation and licensing process to ensure all resources are available for integration.
- **Using NiceLabel .NET API.** Best practices and suggestions for working with the NiceLabel Print Engine.
- **Control Center Integration.** Instructions on how to integrate with Control Center management module for document management and print history logging.
- **Technical Support.** Where to go for additional assistance.

## 1.4. What is NiceLabel .NET API

### 1.4.1. Introduction to NiceLabel .NET API

NiceLabel .NET API is .NET Framework based library that enables software developers to seamlessly print NiceLabel label templates and interact with NiceLabel management module.

NiceLabel Print Engine is a rendering engine that is packaged as part of the NiceLabel .NET API. It can be the basis for a variety of products for previewing and printing NiceLabel label templates at different stages of the professional print workflow. NiceLabel Print Engine is designed to deliver reliable high-speed printing of labels and documents.

NiceLabel .NET API is built on Microsoft .NET Framework and is the preferred choice for developers of .NET applications. It enables users to quickly add full-featured professional label printing capability to their own applications.

NiceLabel Print Engine was designed from ground up with modern multi-core processors in mind. It comes with multi-threading support out of the box to easily deliver the best possible performance on any hardware.

### 1.4.2. NiceLabel .NET API Components

NiceLabel .NET API Libraries are installed along with the NiceLabel product that you install on each printing client.

You have to install the NiceLabel .NET API Libraries on each printing client. For more information see [Deploying and Licensing](#).

The NiceLabel .NET API includes:

- **NiceLabel .NET API Libraries.** A set of managed .NET assemblies that your application will refer to. These assemblies include NiceLabel Print Engine that will be used to interface with label templates and manage print jobs.
- **NiceLabel .NET API Installation and Deployment Guide.** This document.
- **NiceLabel .NET API Programming Reference.** Contains information about the entire NiceLabel .SDK namespace. See the file `SDK Reference.chm`.
- **NiceLabel Printer Drivers.** NiceLabel has developed printer drivers for more than 4000 different printer models. We recommend using our native printer drivers so you can benefit from the optimized print streams. For a complete list of downloadable drivers, please visit <https://www.nicelabel.com/downloads/printer-drivers>.
- **NiceLabel .NET API samples.** A set of sample applications that will help you get started and demonstrate capabilities of the API. Each Visual Studio project demonstrates how to utilize a

specific feature of the NiceLabel .NET API. Please refer to the `ReadMe.txt` file included with each project as well as the inline code comments for specific details.

## 2. Deploying and Licensing

### 2.1. System and Software Requirements

In order to successfully implement NiceLabel .NET API, your infrastructure needs to fulfill the system requirements. To install and run this product, you need at least a minimum supported configuration.

Please consult general [NiceLabel System Requirements](#). To develop your application you will need a development environment that can be used with .NET Framework 4.5 such as Microsoft Visual Studio 2012 or newer.

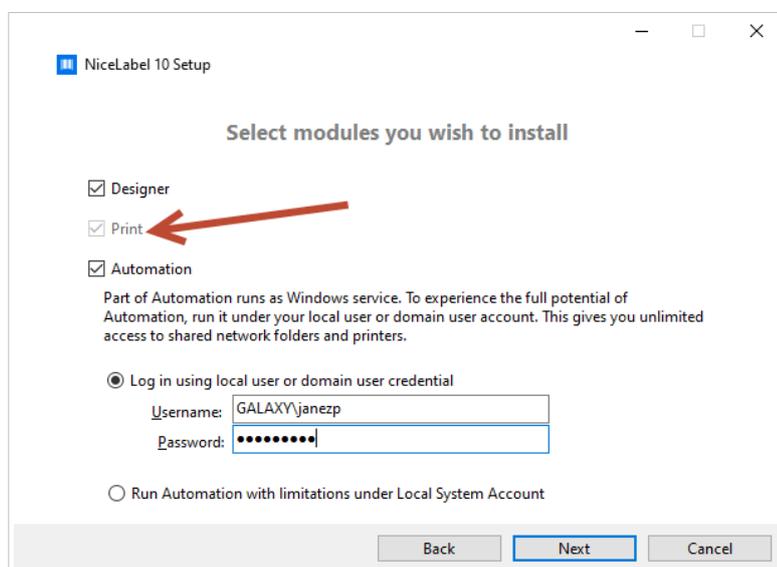
### 2.2. Deploying NiceLabel .NET API

The NiceLabel .NET API installation package installs the following:

- **SDKFiles folder.** This folder contains the following.
- **Subfolders with DEMO/Sample applications.** These folders include C# sample applications (in source code). They reference the files from `SDKFiles` folder. Project files can be opened, compiled and executed. Each folder includes also a short description about the project.
- **SDK Reference.chm.** The NiceLabel .NET API help file.

When you download NiceLabel .NET API ZIP file from the web, make sure to first unblock it. If you do not unblock it, Windows will treat the content of ZIP file as untrusted and you might see the load errors such as “Cannot load assembly. Error details: System.IO.FileLoadException”, when you load NiceLabel .NET API files in your applications. Right-click the ZIP file and click Unblock button.

Use the universal NiceLabel installer to install NiceLabel .NET API on the client computer. The API libraries are installed as part of Print components.



Additionally, NiceLabel LMS users can also connect to NiceLabel Control Center to use files in the Document Storage and to collect the history of printing events. For more information, see the topic [Control Center Integration](#).

If you are using NiceLabel Cloud, you can start using your NiceLabel .NET API after you install and sign in your NiceLabel Print client to the NiceLabel Cloud:

1. Open NiceLabel Print.
2. Go to File > About.
3. Under Account, click Sign in.
  - The Connect to NiceLabel Cloud window appears.
4. Type in the name of the NiceLabel Cloud server you are connecting to and click Connect.
5. The Sign in to your NiceLabel Cloud window appears. Sign in using your Microsoft or Google Account.

If you are using NiceLabel .NET API in combination with NiceLabel Print that you activate in your Control Center, you can also:

- Pre-configure activation settings using the `product.config` configuration file. This allows you to perform standardized repeated installations in your labeling environment. See all available preconfiguration options in [the Knowledge Base article](#).
- Pre-configure the activation settings using command line interface. This way, you can make your repeated installations faster. You can find the license key definition command in [the Knowledge Base article](#).

## 2.3. Licensing

Before you can use the NiceLabel .NET API on a workstation the license has to be activated.

To activate the license, run NiceLabel Designer or Automation after installation and follow the on-screen guidance for activation.

NiceLabel .NET API is available with PowerForms Suite, LMS Pro and LMS Enterprise editions of NiceLabel 10. Control Center integration is available only in LMS editions. Printers used with NiceLabel .NET API count in the same license quote as from other NiceLabel modules. See the document [NiceLabel 10 Licensing](#) for more information on licensing.

For software vendors that develop web or cloud based products, higher volume off-the-shelf products or niche applications that use predefined label templates, NiceLabel offers a set of developer-only products which might be better suited for such processes. Contact our sales department to learn more.

## 3. Using NiceLabel .NET API

### 3.1. Getting Started

The easiest way to get started is to consult the samples NiceLabel has prepared to demonstrate various capabilities of NiceLabel .NET API.

Samples are located in the following folder:

```
c:\Users\Public\Documents\NiceLabel 10\SDK\Samples
```

The folder where you have extracted the NiceLabel .NET API ZIP file.

The sample applications have been written in C# with Visual Studio 2013. Each sample focuses on a typical task and is accompanied with a `ReadMe.txt` file and code comments which provide important notes for integrating with NiceLabel .NET API components.



#### NOTE

We recommend to start with **SimpleTestApplication** sample.

### 3.2. Referencing NiceLabel .NET API

To include the NiceLabel Print Engine in an application, you will need to create a reference to the `SDK.NET.Interface.dll` assembly.

This file can be found in the `SDKFiles` folder, where you have extracted the NiceLabel .NET API ZIP file.

It is recommended to keep the files of NiceLabel .NET API in a separate folder to your application's executable. When the reference to `SDK.NET.Interface.dll` file is made, this file may be copied to your application's bin folder.

This file can be found in the installation folder of NiceLabel which is typically:

```
c:\Program Files\NiceLabel\NiceLabel 10\bin.net
```

### 3.3. Label Design

NiceLabel .NET API components have no label design capabilities and no user interface. It is strictly a print engine which is embedded into other applications.

Label templates, which the NiceLabel Print Engine will use for printing, must be created in the application NiceLabel 10. This is the label design tool that you use to create graphical layout of label templates, define object properties, and variable fields.

You will use NiceLabel Designer to create your label templates, and you will use NiceLabel .NET API to merge user data, and to manage label printing.

## 3.4. Typical Workflow Tasks

### 3.4.1. Print Engine Management

All NiceLabel .NET API features are exposed through the `NiceLabel.SDK.IPrintEngine` interface.

When the application refers the `SDK.NET.Interface.dll` assembly, it can access `PrintEngineFactory` static class. Before using any other properties, the application needs to set the `SDKFilePath` property. This is necessary for the library to access the NiceLabel .NET API files, which can exist on different locations.

Once the location is known, the application can retrieve the `IPrintEngine` interface by accessing the `PrintEngineFactory.PrintEngine` property.

After retrieving the `PrintEngine`, a call to `Initialize()` method of the returned `IPrintEngine` instance needs to be done.

```
IPrintEngine printEngine = PrintEngineFactory.PrintEngine;  
printEngine.Initialize();
```

Before terminating the application, `Shutdown()` method of the `IPrintEngine` instance should be called to release all resources used by the library.

### 3.4.2. Label Template Management

Access to label template files will be crucial for all workflows. The `OpenLabel()` method allows your application to manage which template(s) your users will have access to. Templates can be opened from files on your client's file system or from `System.IO.Stream` objects for users that want to deploy them embedded in their application or maintained in a database.

Additionally, NiceLabel LMS users can use Document Storage Server for document management of label files.



#### NOTE

See the sample **SimpleTestApplication**.

### 3.4.3. Printer Management

NiceLabel Print Engine provides a list of locally available printer drivers to your application and is provided as a convenience to reduce coding on your side.

Printer selection follows this hierarchy:

1. The `PrinterName` property is set to a valid printer name.
2. The printer name stored on the template is found.
3. The system's default printer.



#### NOTE

See the sample `PrinterSelectionSample`.

### 3.4.4. Graphical Preview and Digital Archiving

Label previews are a vital part of any print workflow system. Print production personnel as well as consumers rely on online- and hard copy previews as assurances of final job quality.

NiceLabel .NET API-based applications will use the same print engine for both previews and final output, thus ensuring that what you see is indeed what you get.

With the `GetLabelPreview()` method, your application can retrieve a graphical preview of the next label that will be printed. This method is very convenient for displaying a true preview so that the user can see what will be actually printed.

The `PrintToGraphics()` method will generate a series of images of all the labels that would be printed. This can be useful for achieving purposes or working with proof-of-concepts.



#### NOTE

See the samples `GetLabelPreviewSample` and `PrintToGraphicsSample`.

### 3.4.5. Label Printing

NiceLabel technology allows you to print to any printer with a Windows driver. In addition, NiceLabel output provides printer commands for more than 4000 label printers allowing you to use native printer features, such as barcodes, font and counters, thus improving printing performance.

NiceLabel .NET API offers three different printing modes:

- **Synchronous Printing.** This mode maintains a connection to the print process. In this mode, the application sends data to the print process and keeps the connection to it established as long as the print process is busy. Calling the `Print()` method returns feedback about the print job.

- **Asynchronous Printing.** This mode processes the print job in its own thread. This boosts performance and increases the number of print jobs that can be processed in a time frame. When using the `PrintAsync()` method, the caller is notified via event handlers when print process status changes.
- **Session Printing.** NiceLabel's unique concept of session printing assures that every print job will be printed exactly as you intended. If you need to make sure that your print jobs are not interrupted when using shared printers or you need to print different labels on a single page, NiceLabel makes it easy with session print.

All labels within a single session print are sent to the printer as a single job, ensuring the correct printing order.

Each session printing includes three stages: `StartSessionPrint()` initializes a new printing session, `SessionPrint()` adds a print request to the current session and can be called multiple times, and `EndSessionPrint()` finalizes the current session.



#### NOTE

See the sample `SimpleAsyncPrintSample`.

### 3.4.6. Print Job Monitoring

Regardless of the label printing mode you chose, NiceLabel .NET API lets you monitor the printing process through simple, event-based print job status updates. This detailed access to your printing process allows you to control and respond correctly to any situation.



#### NOTE

See the sample `SimpleAsyncPrintSample`.

### 3.4.7. Exception Handling

All exceptions from the NiceLabel Print Engine will be of type `SDKException`. This class contains detailed information about errors that might be encountered.



#### NOTE

See the sample `ExceptionHandlingSample`.

### 3.4.8. Print Logging

NiceLabel LMS users can benefit from NiceLabel .NET API ability to log print events from each client into the Control Center's consolidated history database. The collected information can be drilled-in and examined from Control Center's centralized web interface.

See the **NiceLabel .NET API Programming Reference** for details.

## 4. Control Center Integration

Control Center is a web-based management application that is at the heart of the NiceLabel Label Management System. [NiceLabel LMS](#) is a next-generation all-inclusive label management system for businesses that operate a quality management system and for regulated enterprises. It includes everything you need to standardize, centralize and control your labeling. NiceLabel .NET API allows applications developers to seamlessly integrate and leverage Control Center's Document Management System and consolidated Print History.

### 4.1. Document Management

NiceLabel Control Center products include a document management system optimized for labeling that is accessible through Web Distributed Authoring and Versioning (WebDAV) protocol. The files inside this repository can be subject to lifecycle management controls, such as version control and a multi-step approval process.

All your label-related files can be stored within an online centralized Document Storage, governed by lifecycle management and user access control. NiceLabel .NET API will be able to gain access into the Document Storage and retrieve the requested file.

### 4.2. Logging

NiceLabel .NET API can log print events from each client into the Control Center's consolidated history database.