

Label Cloud Data Integration

Guide to linking data to labels with NiceLabel Label Cloud

Rev-2020-02 ©NiceLabel

Contents

INTRODUCTION	4
LINKING TO DATABASES	5
Link to file databases	5
Link to Azure SQL user database provided with Label Cloud	5
Link to external databases	6
DIRECT INTEGRATION	7
Cloud trigger	7
Operation	8
Technical details	9
IoT print using Cloud API	10
Operation	11
Technical details	11
Thin client printing	13
Operation	14
Technical details	14
File drop to SFTP	15
Alternative on-premises integration	15
REFERENCES	17
APPENDIX A: INTEGRATION BUNDLE	1
Configuring Control Center	1
Uploading files	1
Registering Web application	1
Creating Cloud Integration	2
Registering Cloud Integration in Developer Portal	2
Registering cloud printers	3
Configuring mockup website	3
Configuring integration server	4
Configuring Web Client	5
Using integration sample	6
IoT print using Cloud API	6
Cloud trigger	7
Thin client example	7
APPENDIX B: CLOUD TRIGGER API	1
Executing Cloud trigger API	1
Registering as integrator	2
www.nicelabel.com	

API Methods	2
Method PRINTERS	2
Method PRINT	2
Method LABELCATALOG	3
Method PREVIEW	4
Method PRINTJOB	4
Method VARIABLES	5
Method PRINTERSTATUS	6
Data structures	6
Incoming data payloads	6
Feedback messages	8

Introduction

This document provides information about what options you have with NiceLabel Label Cloud to print labels with data originating in your business applications.

Dependent on how the access to your data is granted by your security policy you can:

- **Create a link to your database from NiceLabel software.** In this case, you would connect a database to your labels or forms and typically use NiceLabel user interface to print labels. You would create NiceLabel PowerForms printing applications.
- **Integrate NiceLabel printing into your business system.** In this case, you would initiate label print from your existing applications, but NiceLabel Automation would be the print engine in the background to process the data and image the labels.

Linking to databases

The business applications that you interact with daily must save all updates, changes and new information you input in some organized storage. We refer to such storage “a database”. There are various types of databases, but most commonly used for label data are relational databases. These can be simple file-based storages, such as Microsoft Access or Microsoft Excel files, or can be more robust and high-performance SQL databases.

NiceLabel software can read data from any type of a database as long as you:

1. Have a way to connect to the database (e.g. a driver for your database)
2. Permissions to connect to the database (e.g. username and password)

You can use the data from the database as-is, but you can also generate custom views to either filter the data from one table or merge data from several tables into a resulting dataset. You can create views visually with a graphical designer or using SQL statements. You would use fields from the dataset as a data source for label or form objects, creating dynamic objects. Each dynamic object updates its content per each item in a dataset.

Link to file databases

File databases offer an alternative to professional SQL-type of databases and are much easier to use for a simple projects. You might have Microsoft Access databases or Microsoft Excel spreadsheets to store your product information, or you export the data from a business system into file databases. Using file databases might also be indicator the company is not using the single source of truth for the data.

When using NiceLabel Label Cloud you store the labeling assets in the cloud document management system. Typically, you would store label templates and images, however, you can also store file databases.

The downside to this option is read-only access to the files. You cannot write data to these databases. To update the databases with new data, you have to upload the file again.

As an alternative, you can configure labeling solutions to load data from locally available file databases. You can have your Microsoft Access and Microsoft Excel files stored on a local or network attached storage. In this case, the files are accessible in read-write mode and can be modified from within NiceLabel software.

Link to Azure SQL user database provided with Label Cloud

Each Label Cloud Business (and above) subscription comes with the access to Microsoft Azure SQL database. This is Microsoft SQL database hosted in Microsoft Azure environment. The database size depends on the Label Cloud edition that you use.

The database is not to be mistaken with the product database that Label Cloud uses to store labeling assets and print history for your subscription. This is a separate user database that you have full access to and can manage yourself. You can use this database for your own purposes, including storing your product information.

A lot of times, users export master data from their business systems into this database and make it available to NiceLabel software. Your company's security policies might not grant NiceLabel access directly to the database of your business system, but NiceLabel can access this Azure SQL database.

You can use Microsoft SQL Server Management Studio to manage the database.

Link to external databases

NiceLabel applications available from your Label Cloud account can also access data from any other database source you might have. The system requirement is that you must have the appropriate database driver installed on the workstation running NiceLabel software. If you use NiceLabel Web printing technology, then the database driver can be installed on the server (if you want a server to initiate the connection to the database) or on the workstation (if you wish workstation to initiate a connection).

NiceLabel will work with any Windows database driver (e.g., ODBC and OLEDB drivers).

Direct integration

When you use a cloud-based business system to drive the company's processes, and it comes to a question of label printing, typically, one question is unearthed.

How will you print labels from software that runs in the cloud to your printers that are installed on-premises?

This chapter provides information and samples for the configuration steps to enable label printing from a cloud-based application using NiceLabel software that also runs in the cloud and can communicate with the on-premises printers.

This document is accompanied by sample files that demonstrate the direct integration options:

1. NiceLabel Automation configuration (MISX)
2. Sample label (NLBL)
3. Website created in AngularJS framework that acts as a mockup for your cloud system. The Website demonstrates the three integration options explained below.

Cloud trigger

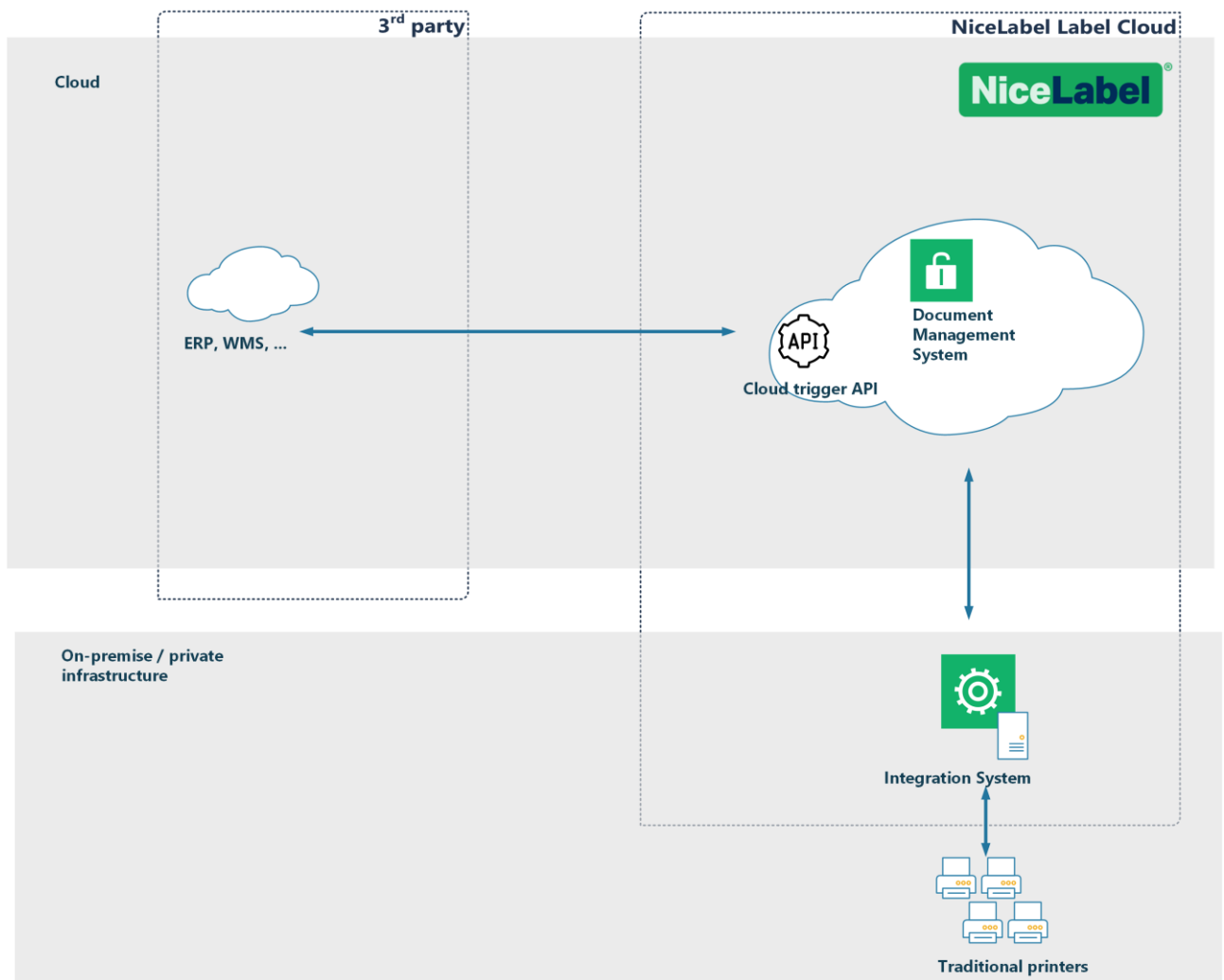
This type of printing is also known as "server-based proxy printing". In this case, your cloud application communicates with the server-based proxy (NiceLabel Automation) that does the processing and prints to local print queues available on the server. You do not have to modify the firewall to allow the data from your cloud application to pass into NiceLabel Automation. You send your data to API in the Label Cloud, which forwards it down to NiceLabel Automation using a secure Azure Service Bus.

BENEFITS

- Support for any printer brand/model.
- Can do "data enrichment" (merging data from various external systems on the same label).
- Can return label preview.

REQUIREMENTS

- On-premise installation of the NiceLabel integration system (application NiceLabel Automation).
- Printer drivers for all printers installed on a server with NiceLabel Automation.
- Network-attached printers. Printing to locally-connected printers, such as USB printers, is available when they are shared in a network.



Operation

Cloud trigger allows you to integrate your Label Cloud with an existing cloud-based business system, and to print labels on your local printers. Modern IoT printers can communicate directly with the cloud, but this is not something legacy printers can do. To print labels to legacy printers, you need an integration system installed on-premises, where the printers are.

Your cloud-based business system cannot communicate with the on-premises software or devices. There is a firewall protecting the local network from the unsolicited requests from the internet.

When the cloud-based business system produces an output, you send it to a NiceLabel cloud-hosted **Cloud trigger API** (proxy), which then relays it for processing to the on-premises NiceLabel Automation. The message is sent using a secure **Azure Service Bus**, Microsoft cloud messaging service. You send the message as an HTTPS request using GET or POST methods.

NOTE: The structure of the payload that you send into the API must match the configuration for NiceLabel Automation. You can configure Automation to adapt to the standard XML/JSON messages that your system can already generate.

As a result, the on-premises integration system merges the label template from NiceLabel Label Cloud DMS (Document Management System) with the data received from the cloud business system producing a printout to local printers.

To summarize, Cloud Trigger enables a secure and transparent way to integrate your local label printing with applications that communicate over the open internet and are not available in the same network with the printers.

Workflow:

1. Send data from the cloud-based system to NiceLabel API in Label Cloud (HTTPS request).
2. The on-premises integration system receives the data and processes the request as configured defined in the NiceLabel Automation configuration file.
3. Labels print on local printers. The status response is sent back to the cloud-based system. A response can also contain label preview or other custom data that you need.

Technical details

The print request is made to the NiceLabel API at:

<https://labelcloudapi.onnicelabel.com/api/CloudTrigger/{triggerId}>

where you provide:

1. `{triggerId}`: "unique identifier" of your Cloud Trigger as defined in the Automation configuration
2. Custom request headers to provide your user credentials.
 - Set `Api-Version` to "v1"
 - Set `Ocp-Api-Subscription-Key` to subscription key.
3. Label data in either query string (GET method) or in the message body (POST method).

The provided label data is then processed in the integration system and used for printing labels. You can choose the format type for the input data (e.g., CSV, XML, JSON, etc.) You must create the matching configuration in the Automation trigger to parse and process the data.

In the example provided with this integration pack, the JSON message is sent to Label Cloud API:

<https://labelcloudapi.onnicelabel.com/api/CloudTrigger/Api-CloudIntegrationDemo-Print>

Sample outbound JSON message:

```
{
  "FilePath": "/folder/label.nlbl",
  "FileVersion": "",
  "Quantity": "1",
  "Printer": "",
  "PreviewFormat": "PNG",
  "Variables": [
    {
      "Product_name": "Syringe",
      "GTIN": "00311234567901",
      "LOT": "ABC123",
      "BestBefore": "10.05.22",
      "SSCC": "0383123456000001",
      "Count": "10"
    }
  ]
}
```

```
}
```

The above message tells NiceLabel Automation to create a PNG preview of label `label1.nlbl` and use the name-value pairs from the `Variables` object.

In this integration example, NiceLabel Automation has been configured to work with the specific JSON structure pictured above. You do not have to use the same Automation configuration for your real-case scenario. You can create your own NiceLabel Automation configuration to adapt to the data structure you already use in your application.

For more information how to configure and run the integration sample, see the APPENDIX A. All details about API definition are available in APPENDIX B.

IoT print using Cloud API

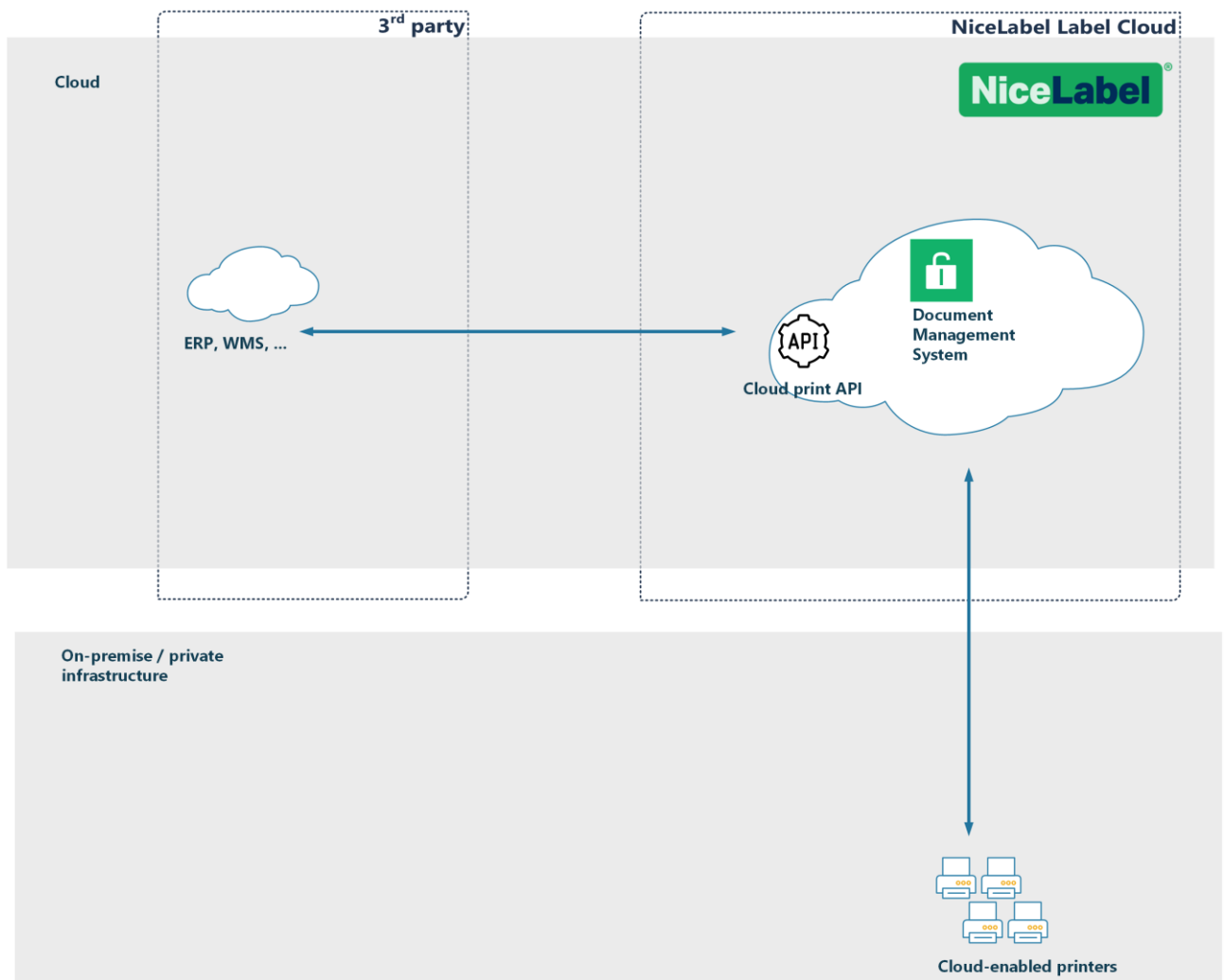
This type of printing is also known as “direct printing”. In this case, your cloud application prints directly to cloud-enabled printers. No local NiceLabel footprint is necessary. There are no locally installed NiceLabel software, such as NiceLabel Automation, or printer drivers.

BENEFITS

- No local NiceLabel footprint.

REQUIREMENTS

- Cloud-enabled printers that can connect to Label Cloud service. Any Zebra Link-OS printer and SATO CL4NX / CL6NX printers will work.
- A reliable internet connection.



Operation

The term “cloud printers” identifies smart label printers that can connect to Label Cloud print service and receive print jobs from it. Cloud printing enables users to print from any application or device to cloud-connected printers regardless of printer location. Cloud printing also eliminates the need for printer drivers. The cloud print service creates print jobs for the target printers and delivers them to the printers via the cloud.

The scheme above explains the architecture of cloud printing. The printer registers and connects to the Label Cloud print service. When the user submits a print request, the cloud service merges the label template from DMS with the received data, creates a matching print job (e.g., ZPL for Zebra printers, SBPL for SATO printers, etc.), and delivers the print job to the requested printer over the internet.

Technical details

When the Cloud print add-on is enabled for your NiceLabel Label Cloud subscription, the Cloud Print API becomes available, and it exposes several HTTP REST methods for print integration. Before you can use any of the methods, you must register your cloud-enabled printers in the Label Cloud so they become visible to the service.

For example, there is a GET method “Printers” to retrieve a list of all registered printers and their live statuses. There is also a POST method “Print” to print the selected label from DMS to the cloud printer with the data from your cloud application.

The API payload can be provided as JSON or as XML data.

The print request is made to the NiceLabel API at:

<https://labelcloudapi.onnicelabel.com/PrintApi/Print/{printerName}>

where you provide:

1. `{printerName}`: the name of your cloud-connected printer as defined when you registered the printer. To get the list of registered printers, you must execute the "GetPrinters" method.
2. Custom request headers to provide your user credentials.
 - Set `Api-Version` to "v1"
 - Set `Ocp-Api-Subscription-Key` to subscription key.
3. Label data in either JSON or XML payload.

The sample JSON payload could look like:

```
{
  "FilePath": "/folder/GS1-128.nlbl",
  "FileVersion": "",
  "Quantity": 1,
  "Variables": [
    {
      "Name": "Product_name",
      "Value": "Syringe"
    },
    {
      "Name": "GTIN",
      "Value": "00311234567901"
    },
    {
      "Name": "LOT",
      "Value": "ABC123"
    },
    {
      "Name": "BestBefore",
      "Value": "10.05.22"
    },
    {
      "Name": "SSCC",
      "Value": "03831234560000001"
    },
    {
      "Name": "Count",
      "Value": "10"
    }
  ],
  "PrinterSettings": ""
}
```

The provided label data is then processed by the API and used for printing labels. The label is always loaded from the DMS.

All details about API definition are available in the [Developer Portal](#).

To register the printer in Label Cloud and enable integration with the Developer Portal, see the [Control Center user guide](#) (Label Cloud edition), chapter **4.5.5 Understanding Cloud Print**.

Thin client printing

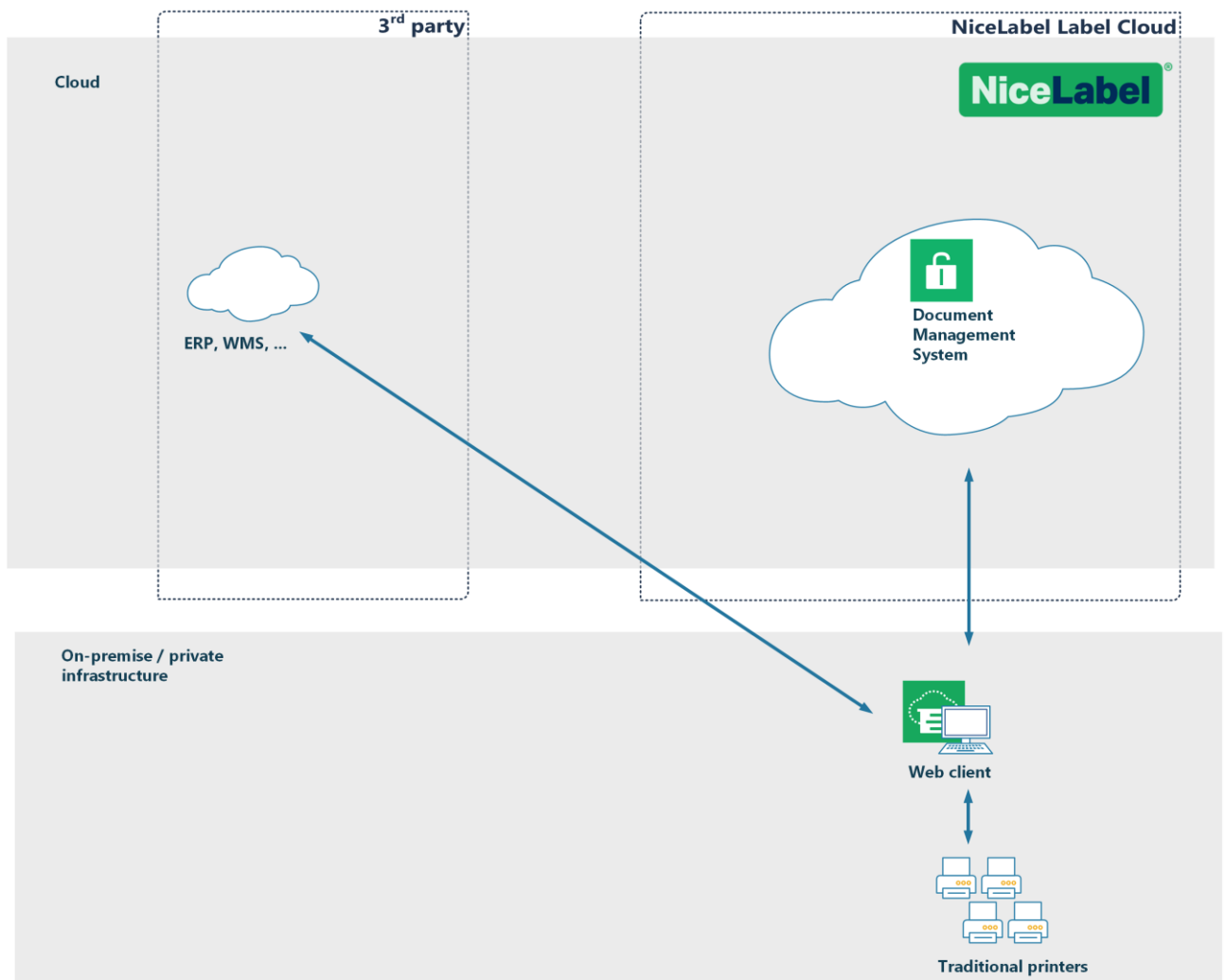
This type of printing is also known as “client-based proxy printing”. In this case, your cloud application communicates with the local proxy (NiceLabel Web Client) that does the processing and prints labels to local print queues available on the workstation.

BENEFITS

- Support for any printer brand/model.
- Can do “data enrichment” (merging data from various external systems on the same label).
- Local processing on the workstation.
- Printing to all local or network printers with the driver installed.
- Can display label preview in the Web client.

REQUIREMENTS

- There is some local NiceLabel footprint – a small thin client is installed on each workstation.



Operation

In this scenario, the cloud-based business system invokes NiceLabel Web Client that runs the Web application designed for label printing. The Web application is invoked from the cloud application via an URI request with label data provided in the query string. The Web application uses any locally installed printer driver to print data on a label.

The user interface of that Web application is created in NiceLabel Designer and runs as a PowerForms application. The user interface of the sample Web application included with this integration pack is minimalistic, only containing a printer selection.

Technical details

NiceLabel Web Client registers a custom URI scheme on the computer. This is a technology behind how the cloud-based application can start the desktop application and send data into it.

The following URI method is called from the cloud-based system:

```
nicelabelwebclient:?server=https://<account>.onnichelabel.com/print&openMode=2&application=<applicationname>&variable=Var1=123&variable=Var2=ABC
```

where you provide:

1. **<account>**: name of your Label Cloud account.

www.nicelabel.com

2. `<applicationname>`: name of your Web application as defined in the Control Center.
3. `openMode=2`: when a new request for Web Client is received, the already running Web application receives new values of variables
4. `variable=Var1=123`: set value of variable `Var1` to 123

For more information for the URI parameters, see the [NiceLabel Web Printing installation guide](#), chapter **8 Adjusting the Software**.

In this example, the Web application allows selecting a default printer where labels will print. You can design your application's user interface any way you want, including displaying the label preview.

For more information how to run the integration sample, see the APPENDIX A.

File drop to SFTP

In this printing type, your cloud application uses a standard SFTP protocol to drop a file over the internet. SSH File Transfer Protocol (SFTP) is designed to provide secure file transfer capability. The connection between the client and the FTP server is encrypted to allow passwords and other sensitive information to be transferred securely over the network.

NiceLabel Automation is configured to monitor folders, where SFTP stores the delivered files. This must be a locally mounted file system, visible to NiceLabel Automation. Once a file arrives, NiceLabel Automation picks it up and processes it.

BENEFITS

- Support for any printer brand/model.
- Can do "data enrichment" (merging data from various external systems on the same label).
- Can return label preview.

REQUIREMENTS

- On-premise installation of the NiceLabel integration system (application NiceLabel Automation).
- Printer drivers for all printers installed on a server with NiceLabel Automation.
- Network-attached printers. Printing to locally-connected printers, such as USB printers, is available when they are shared in a network.
- NiceLabel Automation must be able to access the file system where files from SFTP are stored.

Alternative on-premises integration

When your business system runs locally in the same network as NiceLabel integration system, there are other integration options you can choose to deliver data for label printing into NiceLabel.

NiceLabel Automation can accept data through the following options:

- **File drop.** Your application saves the data into selected local or network folder. You can use the data format supported by your application and then create matching configuration to parse data in

NiceLabel Automation. Typically, this would be XML, JSON or CSV structure, but can be anything else.

- **Serial port.** It is used to receive data from devices connected with a serial port (RS-232), such as barcode scanners or weight scales.
- **Database.** NiceLabel Automation will monitor the specified database for changes. When a new record is detected, the label is automatically printed. NiceLabel can monitor the database of your business system directly if you allow the connection. Many times, the master data is exported from the business system into an intermediate database that NiceLabel can access.
- **TCP/IP socket.** It allows raw data connection from network devices. NiceLabel can operate in the server or client mode.
- **HTTP server.** It is used to receive data from clients sending HTTP messages. This is a native method of communication from web applications. Usually, the format of the message is JSON, but NiceLabel supports any other data structure as well.
- **Web Service.** It is used to receive SOAP messages from clients. The underlying protocol is HTTP, and the message is XML-formatted.

References

1. NiceLabel Label Cloud [getting started](#)
2. NiceLabel Web Printing [installation guide](#)
3. NiceLabel Control Center [user guide](#). Chapter **Understanding Cloud Print**.
4. NiceLabel Automation [user guide](#). Chapter **Cloud Trigger**.
5. NiceLabel [Developer Portal](#), where you register your Cloud Trigger and obtain the subscription key.

Americas
+1 262 784 2456
sales.americas@nicelabel.com

EMEA
+386 4280 5000
sales@nicelabel.com

Germany
+49 6104 68 99 80
sales@nicelabel.de

China
+86 21 6249 0371
sales@nicelabel.cn

www.nicelabel.com



APPENDIX A: Integration bundle

The integration bundle demonstrates all available types of integrated print from the cloud application. Your cloud-based application is simulated with the mockup JavaScript application. The mockup application can send data for printing to:

1. **Cloud Print API** to demonstrate direct printing to cloud-enabled printers.
2. **Cloud Trigger API** to demonstrate server-based printing using a Cloud Trigger.
3. **NiceLabel Web Client** to demonstrate thin client printing.

The sample files are provided in the [LabelCloudDataIntegrationPack.zip](#) file.

The sample is configured for NiceLabel Label Cloud account "demosystem". You must update the sample application to work with your Label Cloud account. Follow the steps in the following chapters.

Configuring Control Center

Uploading files

1. Login to your Label Cloud's Control Center.

<https://<account>.onniceLabel.com/dashboard>

replace <account> with your actual Label Cloud account name.

2. Download and install **NiceLabel Web Client**.

[Download Software](#)

NiceLabel 2019
NiceLabel WebClient

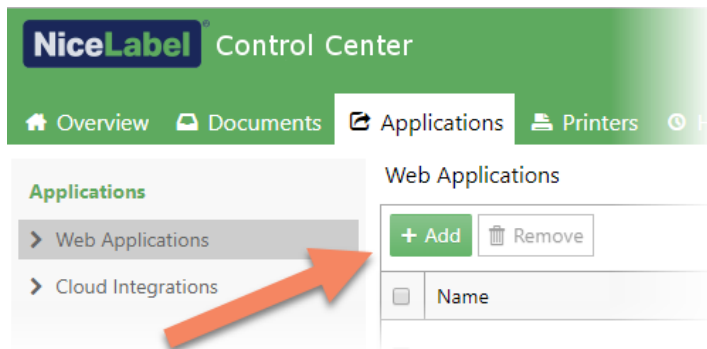
3. In **Documents** tab, create a new folder `/Demo/LabelCloudDataIntegration`.
4. Open provided file [LabelCloudDataIntegrationPack.zip](#). Upload the contents of folder **Document Storage** into the folder **CloudIntegration** inside DMS.

Registering Web application

In Web application tab you define the application that NiceLabel Web Client can run. We use the Web Application for "thin client" printing demo.

Do the following:

1. In **Applications** tab, select **Web Applications** in the left-hand menu and click **+Add**.



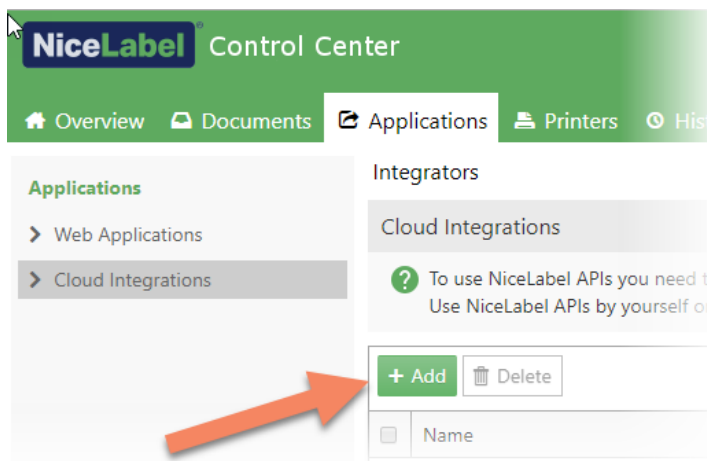
2. In **Name**, enter **CloudIntegration-ThinClient**.
3. In **Solution or Label files**, click Browse and select the solution **/Demo/LabelCloudDataIntegration/ThinClient.nsln**.
4. In **Authorized users and groups**, add at least one user (yourself). The user receives the email invitation and can load the Web Application in NiceLabel Web Client. All users that you authorize will be able to run the Web Application.
5. Click **Save**.

Creating Cloud Integration

The Cloud Integration that you create in Control Center is part of the authentication mechanism that allows you to consume **Cloud Print API** and **Cloud Trigger API**. We use the same Cloud Integration for "IoT print using Cloud API" and "Cloud trigger" printing demo.

Do the following:

1. In **Applications** tab, select **Cloud Integrations** in the left-hand menu and click **+Add**.



2. In **Name**, type in a custom name for the integration. You can type in any name.
3. Click **Save**.

Registering Cloud Integration in Developer Portal

Before using the Cloud Print API or Cloud Trigger API, you must register **Cloud Integration** in the **Developer Portal**. This action links your Label Cloud account with the API. As a result, you get a subscription key that authorizes you to execute the API calls.

Do the following:

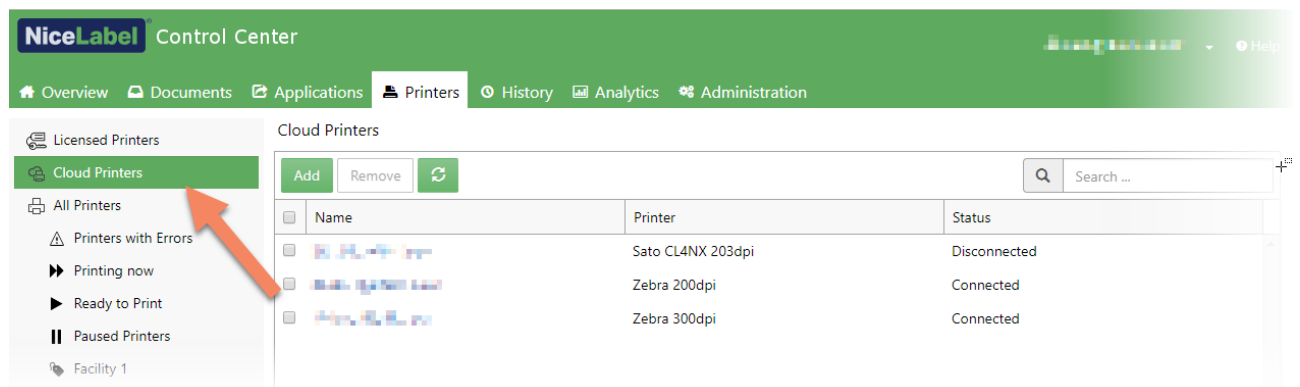
1. Open the Control Center [user guide](#) or online help. Read the chapter **4.5.5.7 Getting Ready To Use Cloud Print API**.
2. Follow the provided instructions to complete the following:
 - a. Set up a new account on the [Developer Portal](#).
 - b. Create a new subscription.
 - c. Register the subscription (from Developer Portal) with the Cloud Integration (from the NiceLabel Control Center).

The subscription key becomes active and you must use it in custom HTTP header **Ocp-Apim-Subscription-Key** with each call to the API.

Registering cloud printers

Before you can execute the Cloud Print API, you must register your cloud-enabled printers with your Label Cloud account.

Follow the instructions in the Control Center [user guide](#) or online help. Read the chapter **Understanding Cloud Print**.



Configuring mockup website

1. Open the provided file [LabelCloudDataIntegrationPack.zip](#).
2. Extract the subfolder **Website** to a temporary location on your disk.
3. Go to folder [Website](#).
4. Open [index.html](#) in text editor.
5. Search for string "nicelabelwebclient". Update the URI reference:

```
nicelabelwebclient:?server=https://demosystem.onnicelabel.com/print
```

replace "demosystem" with the name of your Label Cloud account.

6. Save the file.
7. Open [scripts/config.js](#) in text editor.
8. Enter your subscription key as value for the **subscriptionKey**.

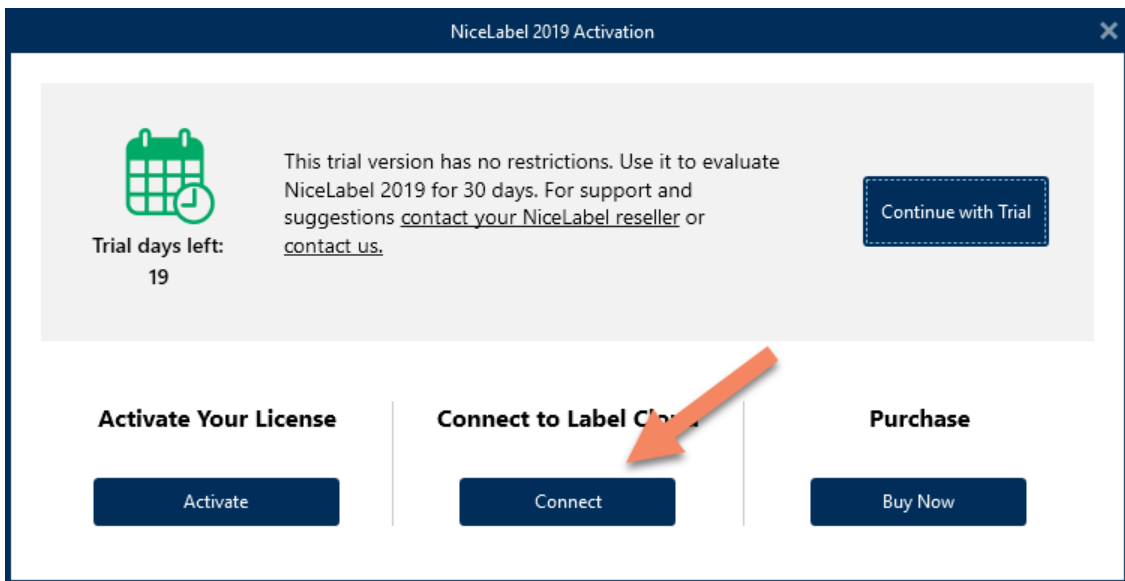
NOTE: You can get the "subscriptionKey" from Products tab in [Developer Portal](#).

Leave values for **label** and **uniqueIdentifier** unchanged. Parameter "label" specifies name of the label file to use and parameter "uniqueIdentifier" specifies the unique identifier for the cloud trigger (as defined in the Automation configuration).

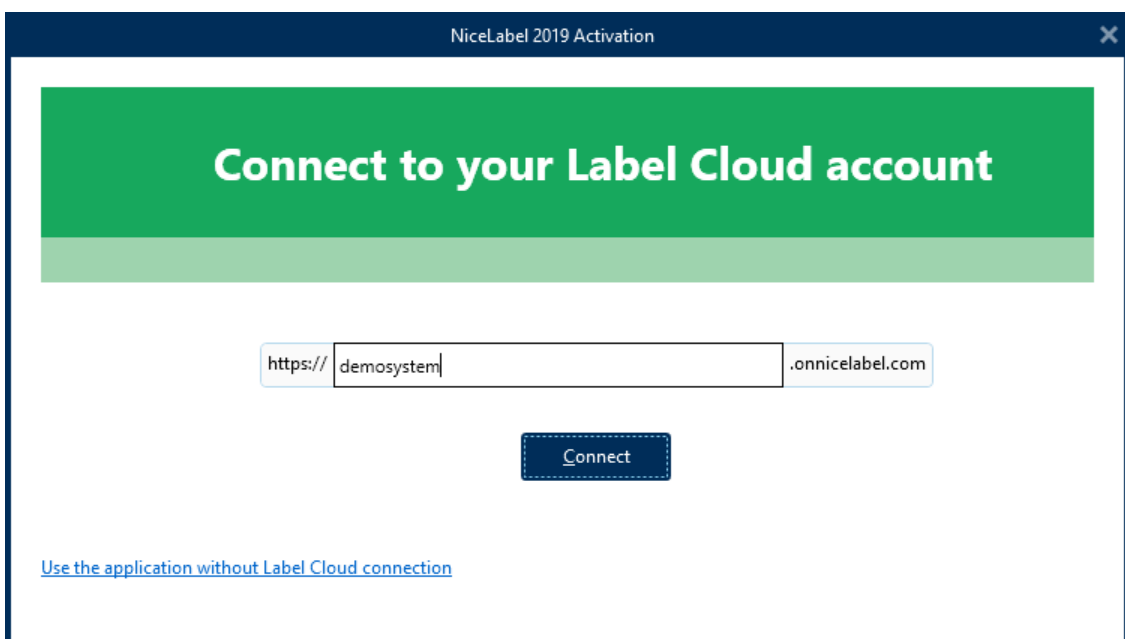
9. Save the file.

Configuring integration server

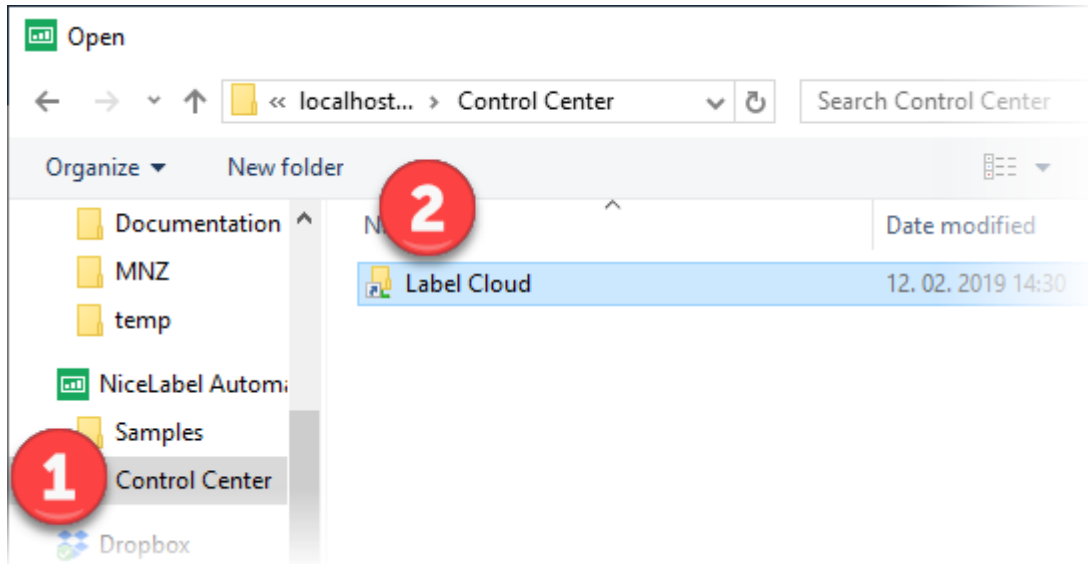
1. Run **NiceLabel Automation Manager**.
2. Click **Connect** button in "Connect to Label Cloud" section to start activation process.



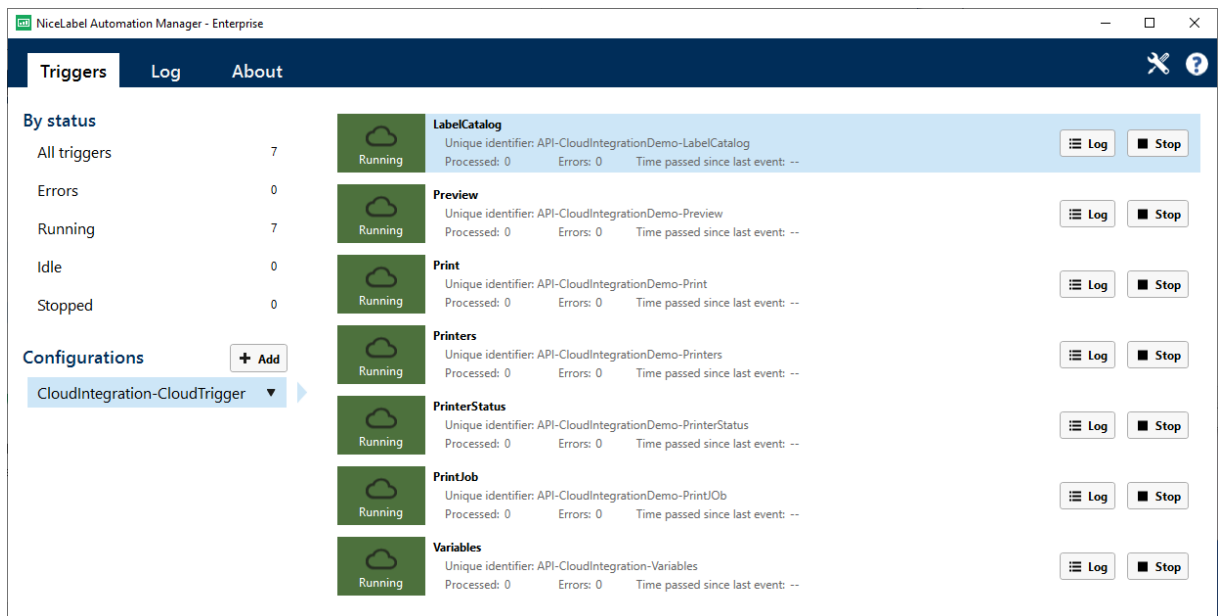
3. Enter the name of your Label Cloud account.
For example, replace "demosystem" in the screenshot below with your actual Label Cloud account name.



4. Click **Connect**.
5. Select either Microsoft or Google authentication, dependent on how you have logged-in into Control Center from the invitation email.
When you are done, NiceLabel is activated on the current workstation.
6. In Automation Manager, click **+Add** and browse for the Automation configuration file (.MISX) from Document Storage. Following the previous steps, you have copied the file to folder [/Demo/LabelCloudDataIntegration](#), so load it from there.



7. Start all triggers in the configuration.



Configuring Web Client

To use the NiceLabel WebClient for “thin client” printing, do the following:

1. Open NiceLabel solution **ThinClient.nsln** in NiceLabel Designer for editing.

2. Select the “question mark” Picture object.
3. Double click it to open its properties.
4. In **Events** tab, click **Actions** for **On Click** event.
5. In **Open Document/Program** action, change the file name option to point to the URL where you have published the mockup website. If you intend to run the mockup website from local disk and not through a web server, you can delete the action **Open Document/Program** altogether.
6. Save the changes.

Using integration sample

To use the integration sample, do the following:

1. Go to the temporary folder in your disk, where you extracted files from [LabelCloudDataIntegrationPack.zip](#) file.
2. Go to [Website](#) folder.
3. Double click **index.html**.
The mockup of a cloud system opens in your browser.

Select printer: These printers are registered in the cloud.

GTIN	Product Name	SSCC	Count	Best before	LOT	Print
00311234567901	Syringe	0383123456000001	10	10.05.22	ABC123	Print
00311234567918	Duo Pen Needle	0383123456000002	50	01.10.23	ABC999	Print
00311234567925	Phaseal Protector	0383123456000003	100	01.02.21	CDE323	Print

Each outbound XML request and inbound response are logged in the browser console (press F12 key, then select Console tab to see the messages).

IoT print using Cloud API

To execute the cloud printing, do the following:

1. For the integration method, select **IoT print using Cloud API**.
2. Select the printer from the list. This is a dynamic list of printers. It displays printers that are registered in your Label Cloud account. If you do not see any printers in the list, make sure to register them first. See the chapter **Registering cloud printers**.
3. Click **Print** button to print label of the selected product.

With each Print request the mockup website constructs an JSON message and sends it as HTTP request to the **Cloud Print API**, which processes the XML data, creates print job, sends it to a cloud-connected printer and provides a feedback.

Cloud trigger

To execute this server-based integration, do the following:

1. For the integration method, select **Cloud trigger**.
2. Select the printer from the list. This is a dynamic list of printers. It displays printers that are available on the computer, where you have installed NiceLabel integration system (NiceLabel Automation).
3. Click **Print** button to print label of the selected product.
4. Click **Preview** button to see the label preview of the selected product.

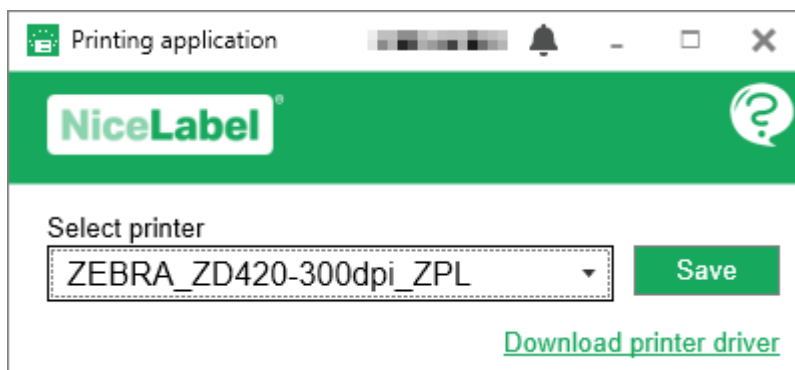
With each Print or Preview request the mockup website constructs an XML message and sends it as HTTP request to the **Cloud Trigger API**, which redirects the message to on-premises NiceLabel Automation. NiceLabel Automation processes the XML data, executes the required actions (such as print or preview) and provides a feedback.

Thin client example

To execute this client-based integration, do the following:

1. For the integration method, select **Thin client**.
2. Click **Print** button to invoke the **NiceLabel Web Client**.
NOTE: You must have NiceLabel Web Client installed on the computer, where you run the mockup web application.
3. Log in to the NiceLabel Web Client with the Microsoft or Google account. Select "Remember me" option to remember your login credentials for subsequent runs.

The Web Client runs and displays the Web application.



4. Select your local printer from the list where you want to print the label.
5. Click **Save** to remember the printer for the future.
6. Keep the Web application running.
7. Whenever you click **Print** button in the mockup application, the product data is transmitted to the Web Client in the query string and label prints to the selected local printed.

The

8. If you do not have the driver for your label printer installed, click **Download printer driver** and install the driver.

APPENDIX B: Cloud trigger API

This appendix describes the necessary steps to enable Cloud trigger printing. In this case, you use integration system (NiceLabel Automation) installed on-premises to receive and process data from a business system. NiceLabel Automation exposes an API that you connect to using the Cloud trigger API from your cloud business application.

Besides the label printing, the API can also:

- Get a label preview (PDF, PNG or JPEG)
- Report live status of the label printer
- Generate a list of all labels in the document management system (DMS)
- Get a list of available printers
- Get a list of variables defined in the label
- Provide a print job (binary)

IMPORTANT! These instructions are based on the sample NiceLabel Automation configuration provided with the integration package. By no means are you required to use and follow these samples. You are free to modify the provided configuration or create one from scratch.

Executing Cloud trigger API

Each Cloud trigger is identified by a unique name. You must reference the unique name in each HTTP request to the Cloud trigger API. In the example below, you would replace the "unique_name_of_the_Cloud_trigger" with the actual unique name as defined in NiceLabel Automation configuration (.MISX file).

Each Cloud trigger defined in the configuration exposes one method. We use one trigger for PRINT method, another trigger for PREVIEW method, another trigger for PRINTERS method etc. The unique name tells Automation which trigger it must execute.

When you load the Automation configuration and deploy/activate it, the Automation registers the Cloud triggers (unique names) in the Label Cloud account. The triggers become accessible from the Cloud trigger API. When you execute the Cloud trigger API your Label Cloud account then knows where the registered Automation trigger runs and will forward the data to it.

You can have multiple on-premises NiceLabel Automation servers running the same configuration at the same time. In this case, we enable automatic round-robin load balancing.

URL	https://labelcloudapi.onnicelabel.com/api/CloudTrigger/<unique_name_of_the_Cloud_trigger>
-----	---

You must include two custom HTTP headers with each request:

Api-Version	v1
-------------	----

Ocp-Apim-Subscription-Key	<the key is generated when you register Cloud Trigger on Developer Portal>
---------------------------	--

You must use the POST method and include the JSON or XML data in the message body.

Registering as integrator

You have to register as a developer and get a subscription key to be able to execute the Cloud trigger API. This is a self-service operation that you can complete as soon as you are able to sign-in to your Label Cloud account and when you have set up the account in the [Developer Portal](#).

The process is described in the NiceLabel user guides and online help. Please see:

1. NiceLabel Control Center [user guide](#) for the Label Cloud. See chapter **4.4.2 Cloud Integrations**.
2. NiceLabel Automation [user guide](#). See chapter **6.2.8 Cloud trigger**.

API Methods

This topic describes the available API methods and the structure of messages transferred between your business application and NiceLabel Automation: the structure of the inbound request and structure of NiceLabel Automation's feedback.

This sample API defines two data structures that you can use, JSON and XML.

For more information about incoming data payloads see the chapter **Data structures**.

Method PRINTERS

This method retrieves the list of available printers that NiceLabel Automation can use. The list contains names of printer drivers installed on the computer, where NiceLabel Automation runs.

You must send in the empty data payload for the API to determine the type of response you need.

Request URL

`https://labelcloudapi.onnicelabel.com/api/CloudTrigger/Api-CloudIntegrationDemo-Printers`

JSON data payload

```
{}
```

XML data payload

```
<?xml version="1.0" encoding="utf-8"?>
```

Method PRINT

This method prints the label. The name of the label, its version, the printer name and all data to print in a label are provided in a data payload.

Request URL

`https://labelcloudapi.onnicelabel.com/api/CloudTrigger/Api-CloudIntegrationDemo-Print`

JSON data payload

```
{
  "FilePath": "/folder/label.nlbl",
  "FileVersion": "",
  "Quantity": "1",
  "Printer": " ZEBRA R-402",
  "PrinterSettings": "",
  "PrintJobName": "",
  "Variables": [
    {
      "Name": "FIELD1",
      "Value": "Value1"
    },
    {
      "Name": "FIELD2",
      "Value": "Value2"
    }
  ]
}
```

XML data payload

```
<?xml version="1.0" encoding="utf-8"?>
<PrintData>
  <FilePath>/folder/label.nlbl</FilePath>
  <FileVersion></FileVersion>
  <Quantity>1</Quantity>
  <Printer>ZEBRA R-402</Printer>
  <PrinterSettings></PrinterSettings>
  <PrintJobName></PrintJobName>
  <Variables>
    <Variable Name="FIELD1">Value1</Variable>
    <Variable Name="FIELD2">Value2</Variable>
  </Variables>
</PrintData>
```

Method LABELCATALOG

This method returns the list of available label templates in the specific folder (and subfolders) from the DMS.

Request URL

<https://labelcloudapi.onnicelabel.com/api/CloudTrigger/Api-CloudIntegrationDemo-LabelCatalog>

JSON data payload

```
{
  "CatalogRoot": "/folder/subfolder"
}
```

XML data payload

```
<?xml version="1.0" encoding="utf-8"?>
<PrintData>
  <CatalogRoot>/folder/subfolder</CatalogRoot>
</PrintData>
```

Method PREVIEW

This method returns the label preview in the required format. Similar to the PRINT method, you have to provide the label name, version, and data values. The supported preview formats are PDF, JPG and PNG.

Request URL

<https://labelcloudapi.onnicelabel.com/api/CloudTrigger/Api-CloudIntegrationDemo-Preview>

JSON data payload

```
{
  "FilePath": "/folder/label.nlbl",
  "FileVersion": "",
  "Quantity": "1",
  "Printer": "",
  "PrinterSettings": "",
  "PreviewFormat": "",
  "Variables": [
    {
      "FIELD1": "Value1",
      "FIELD2": "Value2"
    }
  ]
}
```

XML data payload

```
<?xml version="1.0" encoding="utf-8"?>
<PrintData>
  <FilePath>/folder/label.nlbl</FilePath>
  <FileVersion></FileVersion>
  <Quantity>1</Quantity>
  <Printer></Printer>
  <PrinterSettings></PrinterSettings>
  <PreviewFormat></PreviewFormat>
  <Variables>
    <Variable Name="FIELD1">Value1</Variable>
    <Variable Name="FIELD2">Value2</Variable>
  </Variables>
</PrintData>
```

Method PRINTJOB

This method returns the label converted in a print job for the target printer. For example, if the target printer is Zebra, the returned content is ZPL (Zebra Programming Language). Similar to the PRINT method, you have to provide the label name, version, and data values.

Request URL

<https://labelcloudapi.onnicelabel.com/api/CloudTrigger/Api-CloudIntegrationDemo-PrintJob>

JSON data payload

```
{
  "FilePath": "/folder/label.nlbl",
  "FileVersion": "",
  "Quantity": "1",
  "Printer": "",
  "PrinterSettings": "",
}
  "Variables": [
    {
      "FIELD1": "Value1",
      "FIELD2": "Value2"
    }
  ]
}
```

XML data payload

```
<?xml version="1.0" encoding="utf-8"?>
<PrintData>
  <FilePath>/folder/label.nlbl</FilePath>
  <FileVersion></FileVersion>
  <Quantity>1</Quantity>
  <Printer></Printer>
  <PrinterSettings></PrinterSettings>
  <Variables>
    <Variable Name="FIELD1">Value1</Variable>
    <Variable Name="FIELD2">Value2</Variable>
  </Variables>
</PrintData>
```

Method VARIABLES

This method returns a list of variables defined in the specified label. You might want to know what variables are defined in a label so you can assign values to them when executing PRINT method.

Request URL

<https://labelcloudapi.onnicelabel.com/api/CloudTrigger/Api-CloudIntegrationDemo-Variables>

JSON data payload

```
{
  "FilePath": "/folder/label.nlbl",
  "FileVersion": "",
}
```

XML data payload

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<PrintData>
  <FilePath>/folder/label.nlbl</FilePath>
  <FileVersion></FileVersion>
</PrintData>
```

Method PRINTERSTATUS

This method returns the live printer status as reported by the printer. Not all printers are capable of reporting their statuses.

Request URL

<https://labelcloudapi.onnicelabel.com/api/CloudTrigger/Api-CloudIntegrationDemo-PrinterStatus>

JSON data payload

```
{
  "Printer": "ZEBRA ZD420-300dpi ZPL",
}
```

XML data payload

```
<?xml version="1.0" encoding="utf-8"?>
<PrintData>
  <Printer>ZEBRA ZD420-300dpi ZPL</Printer>
</PrintData>
```

Data structures

The provided API configuration accepts two types of the data payloads: JSON and XML. The configuration automatically determines the type of the provided payload and will provide response using the same type.

Incoming data payloads

Below are the data structures with all possible fields. The field usage depends on the API method you intend to call. See the chapter API Methods for the list of fields applicable to particular API method.

Structure of the JSON payload


```
{
  "FilePath": "/folder/label.nlbl",
  "FileVersion": "",
  "Quantity": "1",
  "Printer": "",
  "PrinterSettings": "",
  "PrintJobName": "",
  "PreviewFormat": "",
  "CatalogRoot": "",
  "Variables": [
    {
      "Variable1": "Value1",
      "Variable2": "Value2"
    }
  ]
}
```

Structure of the XML payload

```
<?xml version="1.0" encoding="utf-8"?>
<PrintData>
  <FilePath>/folder/label.nlbl</FilePath>
  <FileVersion>1</FileVersion>
  <Quantity>1</Quantity>
  <Printer></Printer>
  <PrinterSettings></PrinterSettings>
  <PrintJobName></PrintJobName>
  <PreviewFormat></PreviewFormat>
  <CatalogRoot></CatalogRoot>
  <Variables>
    <Variable Name="Variable1">Value1</Variable>
    <Variable Name="Variable2">Value2</Variable>
  </Variables>
</PrintData>
```

Field descriptions

FIELD NAME	FIELD DESCRIPTION
FilePath	Contains name of the full path and label name from the Document Storage.
FileVersion	Contains version of the label file you want to use. If you use major/minor versioning scheme, you must provide the version as "0.1" or "1.1". If you use major versioning scheme, you must provide the version as "1" or "2". When you do not provide a value, the latest available revision is used. <small>NOTE: If you use approval workflows the latest approved version of the label is used.</small>
Quantity	Defines number of labels to print.
Printer	Defines name of the printer where labels print. This must be exact printer name as installed on Windows computer with NiceLabel Automation.

PrinterSettings	Provides the base64-encoded DEVMODE structure. Use this option to override the printer settings either saved in a label template or recalled from the printer driver. NOTE: You can create PowerForms application to configure printer settings and export them to DEVMODE. There is also the application that exports settings to DEVMODE from the printer driver. You can download it from NiceLabel website (see Downloads-Utilities and Support Software).
PrintJobName	Defines the name for job file that will display in a Windows Spooler. If omitted, the job name equals the name of the label file.
PreviewFormat	Specifies the format of the label preview. The available formats are PDF, PNG and JPEG. If you do not specify a format, PDF is provided.
CatalogRoot	Specifies the starting folder from which the label catalog creates. The catalog also includes labels from all subfolders. NOTE: If you request label catalog from a large Document Storage repository, it might take a while to complete the request.
Variables	This structure contains name-value pairs for the label variables. For each "name" you have to provide a matching "value". All name-value pairs are sent to the label and are used with the variables defined with the same "name". The value mapping takes place automatically based on the same names.

Feedback messages

The result of data processing in Automation is provided back to the business application in a synchronous response. The response type always matches the structure of the incoming data payload. When you send request formatted in JSON, you will receive JSON response. When you send request formatted in XML, you will receive XML response.

- **Successful execution.** If there is no error while processing the data, the business system receives a JSON or XML-formatted message. The structure of the response is dependent on the API method that you executed.
- **Erroneous execution.** If there is a processing error, such as "label not found", "printer not available", "wrong data for label objects", the trigger in NiceLabel Automation raises an error and sends details back in the trigger response.

The API provides feedback in the HTTP header and in the body of the message. Please note, this only occurs when you use the XML data payload. For JSON data payload, the response will contain HTML message without any error details (this issue will be addressed in one of the future NiceLabel releases).

In any case, you can see the detailed error log inside **NiceLabel Automation Manager**.

Structure of PrinterList

There is a **Printer** element for each printer.

FIELD NAME	FIELD DESCRIPTION
PrinterName	Name of the Windows printer driver as visible to NiceLabel Automation. The Windows user can change this name.
DriverName	Name of the Windows printer model (as defined in the printer driver). The Windows user cannot change this name.

Structure of Variables

There is a **Variable** element for each variable found in the label.

Also see the [Automation User Guide](#), chapter **Get Label Information**.

FIELD NAME	FIELD DESCRIPTION
Name	Contains variable name.
Description	Contains variable description.
DefaultValue	Contains default value as defined for the variable during the label design process.
Format	Contains the acceptable type of variable content (characters).
IsPrompted	Contains information whether the variable is prompted at print time or not.
PromptText	Contains text that prompts the user for value input.
CurrentValue	Contains the actual value that is used for printing.
IncrementType	Contains information, if the variable is defined as a counter or not. If identified as a counter, it tells what kind of counter it is.
IncrementStep	Contains information about the counter step. Counter value increments/decrements for this value on the next label.
IncrementCount	Contains information about the point of counter value incrementing/decrementing. Usually, the counter changes value on every label, but that can be changed.
Length	Contains maximum number of stored characters in a variable.
IsPickListEnabled	Contains information whether the user selects variable values from a pick list.
PickListValues	Contains the actual (selectable) pick list values.

Structure of LabelCatalog

There is a **Label** element for each label found in the folder (and subfolders) in the Document Storage.

FIELD NAME	FIELD DESCRIPTION
Name	Name of the label including the path from the document storage root. The server name and port number are not included, NiceLabel Automation knows the location of its DMS.
Width	Contains the label width in 1/1000 of millimeter.
Height	Contains the label height in 1/1000 of millimeter.
PrinterName	Contains printer name for which the label has been designed for. NOTE: You can specify another printer for printing. NiceLabel Automation adapts the label template for the new printer type.
Description	Contains the description as configured in the label.
Revisions	Contains comma-delimited list of revisions that are accessible to NiceLabel Automation. NOTE: Label visibility is determined by the role-based access control (RBAC). When NiceLabel Automation runs as member of the Operator profile (best practice), it can only access the approved label revisions.