

5label

Integration Guide

V.1.1

Contents

- How to install 5label on Windows..... 3
- Quick Start..... 3
- Print at local printers using the Printserver 4
- Integration in ERP..... 5
 - Printserver batch interface..... 5
 - Web services 6
 - Get a preview image of ZPL or SBPL code 6
 - Print ZPL or SBPL code at network printer 7
 - Call a web service to include data into a template code 7

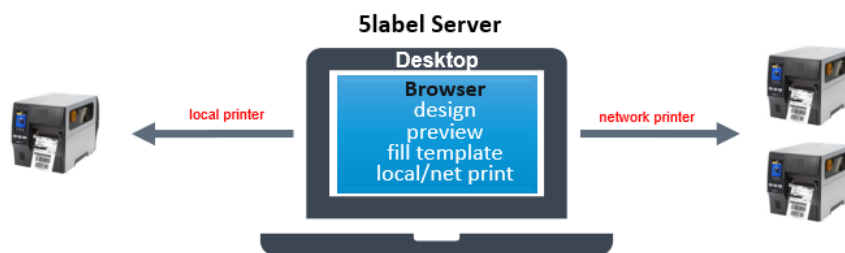
How to install 5label on Windows

Download the software from 5label.com as a zip file. Unzip the package in a directory of your choice on a local desktop or a server environment in the intranet or cloud.

5label consists of a Java **JRE** and a **Tomcat** application server.

The easiest way to get 5label running, is to install it at your desktop:

5label Server Desktop Installation



Quick Start

To start, execute the **StartServer.cmd**.

It will activate the tomcat and open the 5label webpage to check whether the label designer is running.

Printers defined at the server will be shown in the "print selection" box.

Design your first label by right clicking the preview image.

Use the Property.xml to define the label printers you might use.

```
<printers>
  <!-- use <printer>*</printer> to show all printers installed on the server -->
  <!-- use <printer>printername</printer> to show these printers in selection -->
  <printer>SATO WS408</printer>
  <printer>SATO CL4NX Plus 305dpi</printer>
  <printer>ZEBRA ZT230</printer>
</printers>
```

Watch the youtube videos

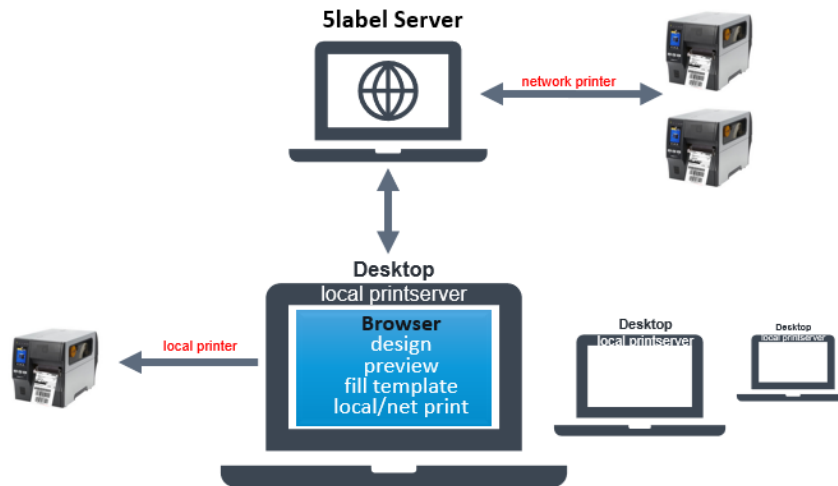
https://www.youtube.com/watch?v=WEw5QGmY0WY&list=PLkaoqwgSVJsz7Qw7J4L1dL_qQFkCi03mx

for an extensive introduction.

Print at local printers using the Printserver

To enable local desktops to fill label templates and print them at local printers, the printserver is used.

5label Server Cloud/Intranet Installation



Copy the **printserver** directory to a shared directory. This way, the local clients may start it by *sharedDir\StartPrintServer.cmd*.

Edit the *printserver.xml* to define all local desktops that intend to print using a local or server defined label printer. This you may do by editing the computer name.

Printserver.xml

```
<computer>
<name>DESKTOP-XXXXX</name>
  <!-- Open Printer Server, listening at myIP and myUDPPort -->
  <myip>127.0.0.1</myip>
  <!-- <protocol>udp</protocol> -->
  <!-- <myudpport>50089</myudpport> -->
  <!-- if protocol is tcp -->
  <protocol>tcp</protocol>
  <mytcpport>443</mytcpport>
  <printers>
    <printer>SATO WS408</printer>
    <printer>SATO CL4NX Plus 305dpi</printer>
    <printer>ZDesigner GK420t</printer>
  </printers>
  <!-- for batch processing -->
  <in>./in</in>
  <out>./out</out>
  <templates>./templates</templates>
</computer>
<computer>
  . . .
</computer>
```

Any defined local desktop requires **no further** local installations.

Integration in ERP

There are two kinds of interfaces to get preview images or to fill and print labels using 5label.

- Using the printserver, a file interface is offered. Just copy a csv or xml file into a defined directory. The printserver will catch it and will create a **preview** image or **print**. It will use the **template** directory to get the ZPL or SBP templates with variables in %% (%%city%%). If the template is not found in this directory, the printserver will call the 5label server via web service to get the template stored in the database.
- 5label offers **web services** to directly call 5label functions.

Printserver batch interface

In the Printserver directory you will find an **in** and an **out** and a **template** subdirectory. The **running printserver** will perform every file copied to the **in** directory.

The printserver offers the opportunity to run at local systems, without any connection to a cloud server!

The files in the in directory may contain csv files

```
template;language;printer;image;name;street;city;country;code
MyTemplate.zpl;ZPL;SATO WS408;image1.png;name1;street 19;city1;country1;1234567890
MyTemplate.zpl;ZPL;SATO WS408;image2.png;name2;street2;city2;country2;1234567891
MyTemplate.zpl;ZPL;SATO WS408;;name3;street3;city3;country3;1234567892
DataImport_SBPL;SBPL;SATO WS408;image4.png;name4;street4;city4;country4;1234567892
```

or

```
template;language;printer;image;dpi;widthpx;heightpx;fromName1;fromName2;fromAdr1;fromAdr2;fromAdr3;fromAdr4;fromAdr5;fromCountry;toName1;toName2;toAdr1;toAdr2;toAdr3;toAdr4;toAdr5;toCountry;idCode;routingCode;trackingNr;reference1;reference2
ShippingTemplate.zpl;ZPL;SATO WS408;addrlabel1.png;203;803;1213;Peter Parker;;39th Avenue;11368;NewYork Queens;;USA;Charly Brown;;512 Solway St;;Pittsburgh;PA 13442;;USA;23442556;67621923451;Z 225 3E3 66 2249 4521;ERP10072798;KU 2777541
ShippingTemplate.zpl;ZPL;SATO WS408;addrlabel2.png;203;803;1213;Charly Brown;;512 Solway St;;Pittsburgh;PA 13442;;USA;Peter Parker;;39th Avenue;11368;NewYork Queens;;USA;23442557;67621322445;Z 223 1A7 66 8235 3024;ERP10072245;KU 2267787
```

When the **printer** column is set the printserver will print the label to the given printer.

When the **image** column (and widthpx, heightpx and dpi) is set, an image will be stored at the out directory.

or xml files

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <row>
    <template>MyTemplate.zpl</template>
    <language>ZPL</language>
    <!-- print label at printer -->
    <printer>SATO WS408</printer>
    <!-- store the preview image in the out directory -->
    <image>
      <name>test1image1.png</name>
      <dpi>203</dpi>
      <widthpx>803</widthpx>
      <heightpx>1213</heightpx>
    </image>
    <data>
      <name>name1</name>
      <street>street1</street>
      <city>city1</city>
      <country>country1</country>
      <code>1234567890</code>
    </data>
  </row>
  <row>
    <template>MyTemplate.zpl</template>
    <language>ZPL</language>
    <image>
```

```

    <name>test1image2.png</name>
    <dpi>203</dpi>
    <widthpx>803</widthpx>
    <heightpx>1213</heightpx>
  </image>
  <data>
    <name>name2</name>
    <street>street2</street>
    <city>city2</city>
    <country>country2</country>
    <code>1234567899</code>
  </data>
</row>
</root>

```

The columns or tags template and language are mandatory.

For csv you may also use "" as the delimiter and , as the separator.

The file format must be UTF-8.

Web services

The 5label web server offers web services to get a preview image or print ZPL or SBPL code. If you use %%variables%% inside your code, you may include data in csv or xml format. The web service will automatically merge the data into the code before printing or generate a preview image.

Get a preview image of ZPL or SBPL code

Post: <http://localhost/5labelDesigner/FLServices>

Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<flabel>
  <preview>
    <dpi>203</dpi>
    <widthpx>803</widthpx>
    <heightpx>1213</heightpx>
    <language>ZPL</language>
    <returnformat>PNG</returnformat>
    <plcode><![CDATA[
      ^XA
      ^CFA,60
      ^FO250,100
      ^FDHello! ^FS
      ^CFA,30
      ^FO50,240^FDMy name^FS
      ^FO50,280^FDMy address^FS
      ^FO50,320^FDMy country^FS
      ^FO50,400^GB700,3,3^FS
      ^FO220,475
      ^BY4
      ^B2N,100,Y,N,N
      ^FD1234567890^FS
      ^FO58,645
      ^GB710,505,3,B,0
      ^XZ
    ]]></plcode>
  </preview>
</flabel>

```

Return:

```

<?xml version="1.0" encoding="UTF-8"?>
<flabel>
  <previewResponse>
    <png>iVBORw0KGgoAAAANSUHEUgAAAloAAAONCAIAAADRdVmeAABDxU1EQVR4Xu3de7BlaVkf4
      COIXKM0tygwqEMQLwU1TikSYgTLBlFG1KGxQ...

```

```
</png>
</previewResponse>
</flabel>
```

The <png> is Base64 coded.

Here's an example on how to encode a png in Java and write as a image:

```
byte[] png =
Base64.getDecoder().decode("iVBORw0KGgoAAAANSUhEUgAAAlAAAONCAIAADRdVmeAABDx...");
File fo = new File("myLabelImage.png");
final FileOutputStream fos = new FileOutputStream(fo);
fos.write(png);
fos.close();
```

Print ZPL or SBPL code at network printer

Post: <http://localhost/5labelDesigner/FLServices>

Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<flabel>
<print>
<dpi>203</dpi>
<widthpx>803</widthpx>
<heightpx>1213</heightpx>
<language>ZPL</language>
<printername>SATO WS408</printername>
<printjobname>DHLExpress</printjobname>
<plcode><![CDATA[
^XA
^CFA,60
^FO250,100
^FDHello! ^FS
^CFA,30
^FO50,240^FDMy name^FS
^FO50,280^FDMy address^FS
^FO50,320^FDMy country^FS
^FO50,400^GB700,3,3^FS
^FO220,475
^BY4
^B2N,100,Y,N,N
^FD1234567890^FS
^FO58,645
^GB710,505,3,B,0
^XZ
]]></plcode></print>
</flabel>
```

Return:

```
<?xml version="1.0" encoding="UTF-8"?>
<flabel>
<printResponse>
<printer>SATO WS408</printer>
<printjob>DHLExpress</printjob>
</printResponse>
</flabel>
```

Call a web service to include data into a template code

Example I (preview):

Post: <http://localhost/5labelDesigner/FLServices>

Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<flabel>
<preview>
  <dpi>203</dpi>
  <widthpx>803</widthpx>
  <heightpx>1213</heightpx>
  <language>ZPL</language>
  <returnformat>PNG</returnformat>
  <plcode><![CDATA[
    ^XA
    ^FX Filling variables out of data csv/XML
    ^CFA,30
    ^FO50,200^FD%%name%%^FS
    ^FO50,240^FD%%street%%^FS
    ^FO50,280^FD%%city%%^FS
    ^FO50,320^FD%%country%%^FS
    ^FO50,400^GB700,3,3^FS
    ^FO220,475
    ^BY4
    ^B2N,100,Y,N,N
    ^FD%%code%%^FS
    ^XZ
  ]]></plcode>
  <csvdata>
    name;street;city;country;code
    name1;street1;city1;country1;1234567890
    name2;street2;city2;country2;1234567891
    name3;street3;city3;country3;1234567892
  </csvdata>
</preview>
</flabel>

```

You will get back a png file which shows three labels containing data from name2 ... to name3

Example II (print):

Post: <http://localhost/5labelDesigner/FLServices>

Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<flabel>
<print>
  <printername>SATO WS408</printername>
  <language>ZPL</language>
  <plcode><![CDATA[
    ^XA
    ^FX Filling variables out of data csv/XML
    ^CFA,30
    ^FO50,200^FD%%name%%^FS
    ^FO50,240^FD%%street%%^FS
    ^FO50,280^FD%%city%%^FS
    ^FO50,320^FD%%country%%^FS
    ^FO50,400^GB700,3,3^FS
    ^FO220,475
    ^BY4
    ^B2N,100,Y,N,N
    ^FD%%code%%^FS
    ^XZ
  ]]></plcode>
  <xmldata><![CDATA[
    <root>
    <row>
      <name>name1</name>
      <street>street1</street>
      <city>city1</city>
      <country>country1</country>
      <code>1234567890</code>
    </row>
    <row>

```



```
    <name>name2</name>
    <street>street2</street>
    <city>city2</city>
    <country>country2</country>
    <code>1234567891</code>
  </row>
</root>
]]></xmldata>
</print>
</flabel>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<flabel>
  <printResponse>
    <printer>SATO WS408</printer>
    <printjob></printjob>
  </printResponse>
</flabel>
```

If an error occurs you will get back an error tag:

```
<?xml version="1.0" encoding="UTF-8"?>
<flabel>
  <printResponse>
    <error>
      <![CDATA[label could not be created!
      Exception: Printer not found: SAATO WS408]]>
    </error>
  </printResponse>
</flabel>
```