BIXOLON

# Software Manual

# Label Artist-II

**Ver. 2.10**

http://www.bixolon.com

# Table of Contents

# Copyright



BIXOLON Co., Ltd. maintains ongoing efforts to enhance and upgrade the functions and quality of all our products.
In the following, product specifications and/or user manual content may be changed without prior notice.

# Caution

Some semiconductor devices are easily damaged by static electricity. You should turn the printer “OFF”, before you connect or remove the cables on the rear side, in order to guard the printer against the static electricity. If the printer is damaged by the static electricity, you should turn the printer “OFF”.

# 1. Operating System (OS) Environment

The following operating systems are supported for usage.

Microsoft Windows Server 2008 (32bit/64bit)
Microsoft Windows Server 2008R2 (64bit)
Microsoft Windows Server 2012 (64bit)
Microsoft Windows Server 2016 (64bit)
Microsoft Windows Server 2019 (64bit)
Microsoft Windows 7 (32bit/64bit)
Microsoft Windows 8 (32bit/64bit)
Microsoft Windows 10 (32bit/64bit)

# 2. Supported Printers

Label Artist-II is available for the following BIXOLON printers.

SLP-TX220/TX223, TX420/TX423, TX400/TX403, TX400R/TX403R
SLP-DX220/DX223, DX420/DX423
SLP-DL410/DL413
SLP-T400/T403
SLP-D220/D223, D420/D423
SRP-770II/770III/E770III
SPP-L3000/L310/L410
XT5-40/43/46
XD3-40d
XD3-40t
XL5-40CT/43CT
XD5-40d/43d
XD5-40t/43t
XM7-20/40R
SRP-S3000
XT3-40/43
BDP-T42/T43/D42/D43, BTP-42/43/46

# 3. Before Startup

The latest version of Label Artist-II is available for download at our website. (www.bixolon.com).

# 4. Installation & Uninstallation

## 4-1 Installation

1) Double-click the file Label Artist-II_Vx.x.x.exe.

※ Administrator privilege may be required to run the installation file.

2) Follow the instructions on the screen to complete the installation process.

“Microsoft Visual C++ 2008 SP1 Redistributable Package (x86)” may be required to run the installation file. When a pop-up message is appeared for the package installation, please click the “install” button.

## 4-2 Uninstallation

1) Open "Remove Programs" in the Control Panel.

2) Select Label Artist-II and click the "Remove" button to uninstall the Label Artist-II on your PC.

# 5. How to use

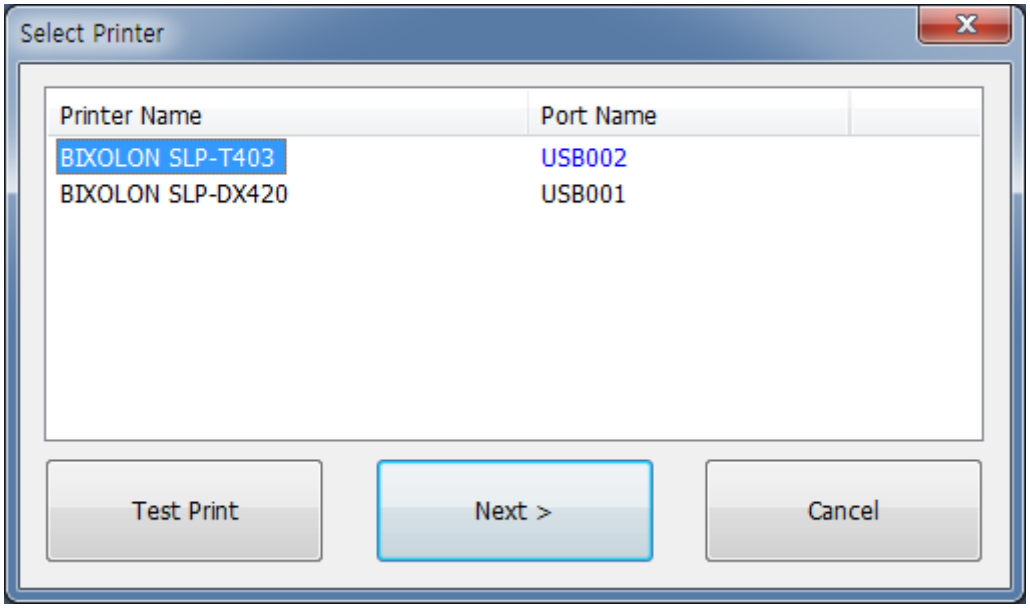
## 5-1 <File> Menu

### 5-1-1 New

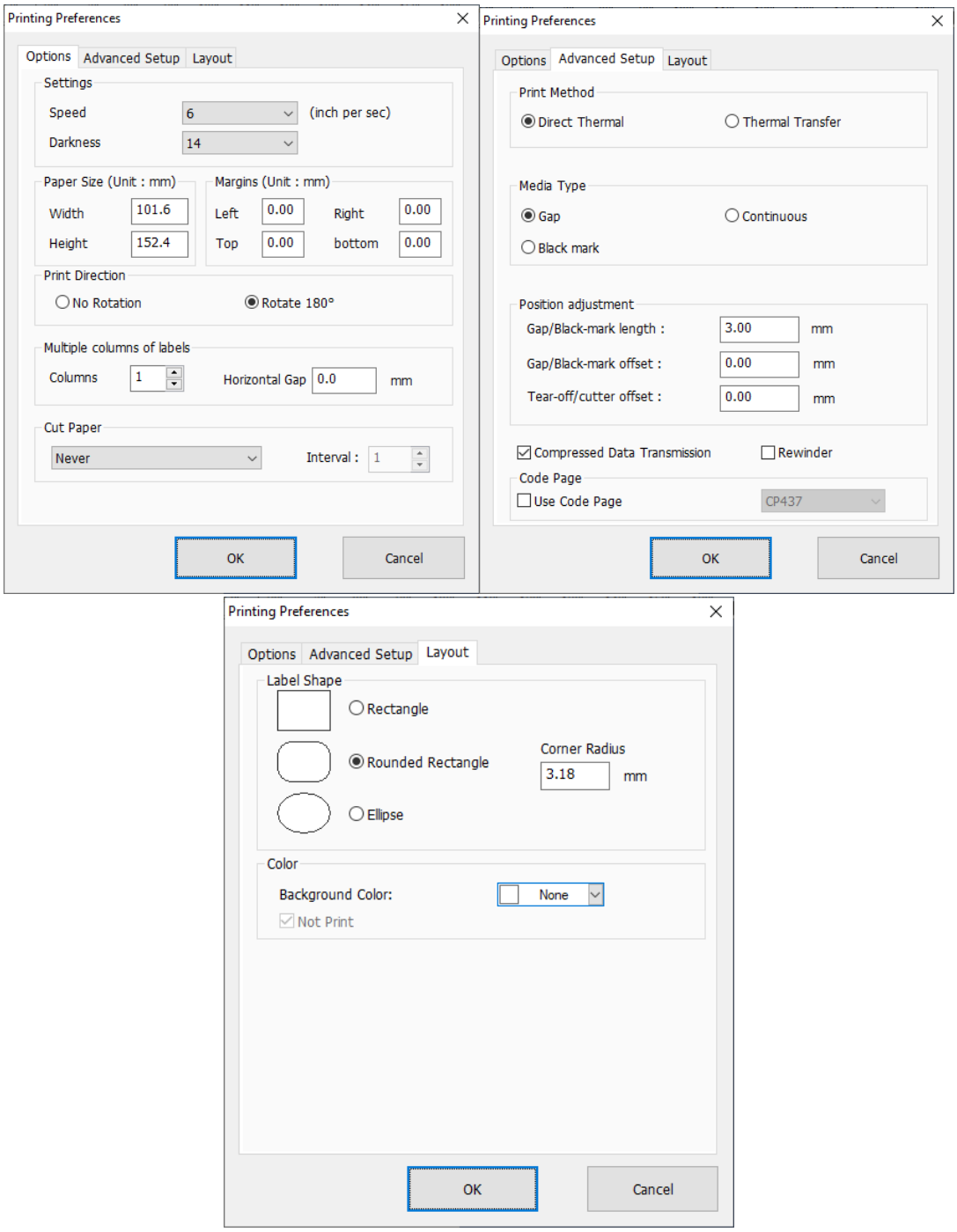
1) Select <New> from the <File> tab.

> Printer Driver must be installed in order to use Label Artist-II application.

2) The dialog windows appears as below to be able to select a printer driver.
Select the printer to use and click <Next>.

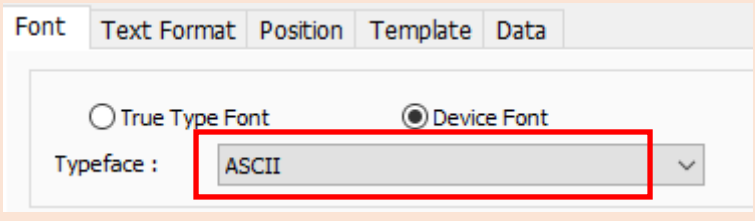


3) The <Printing Preferences> window opens as shown below.
Specify the printing options such as label size and click <OK>.

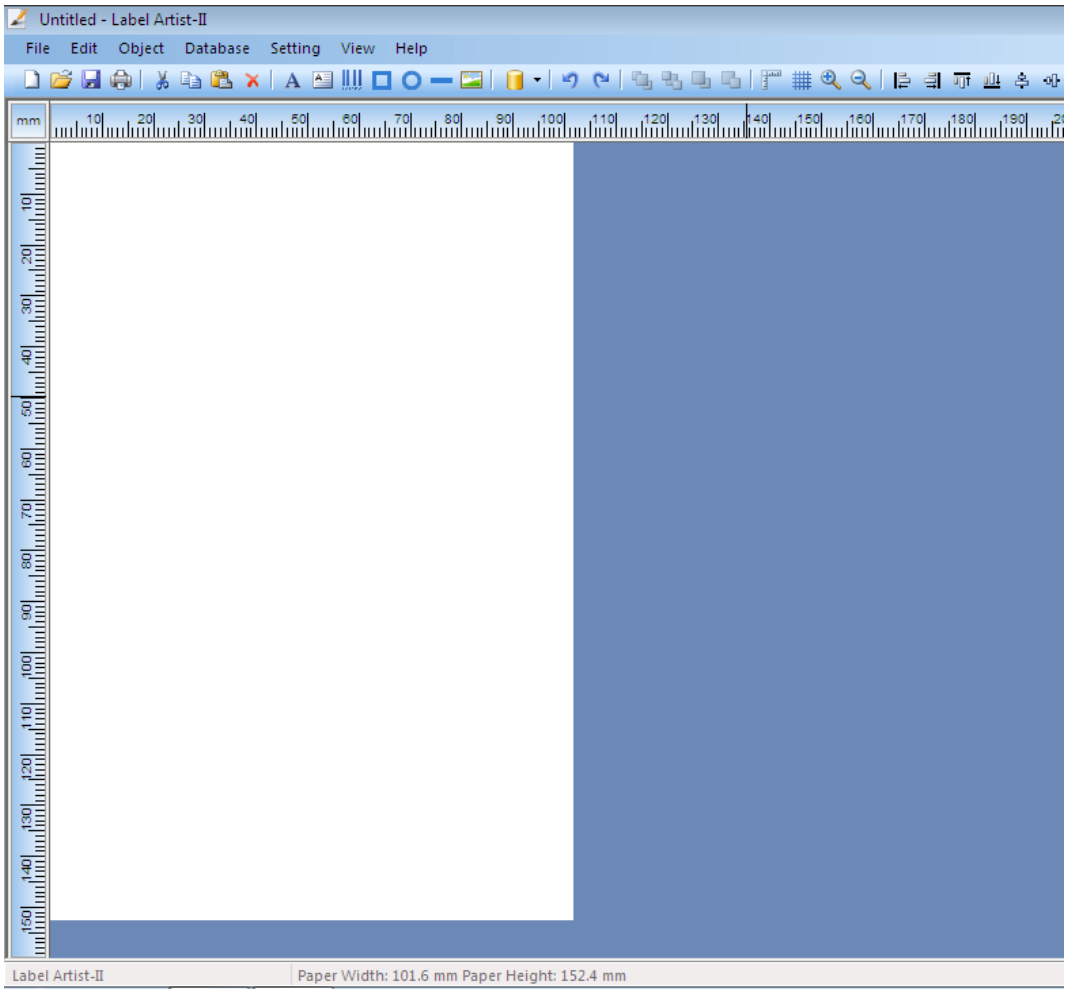



**Use Code Page**

The code page is applied when printing ASCII fonts among printer fonts.



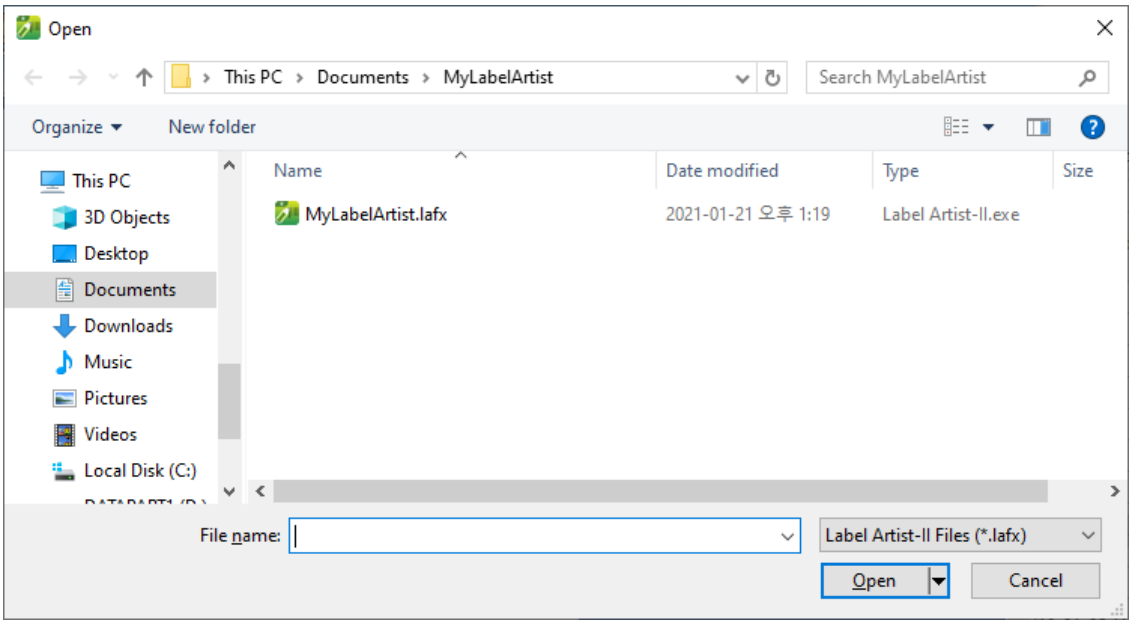
4) Once the preferences are set, new label is created.



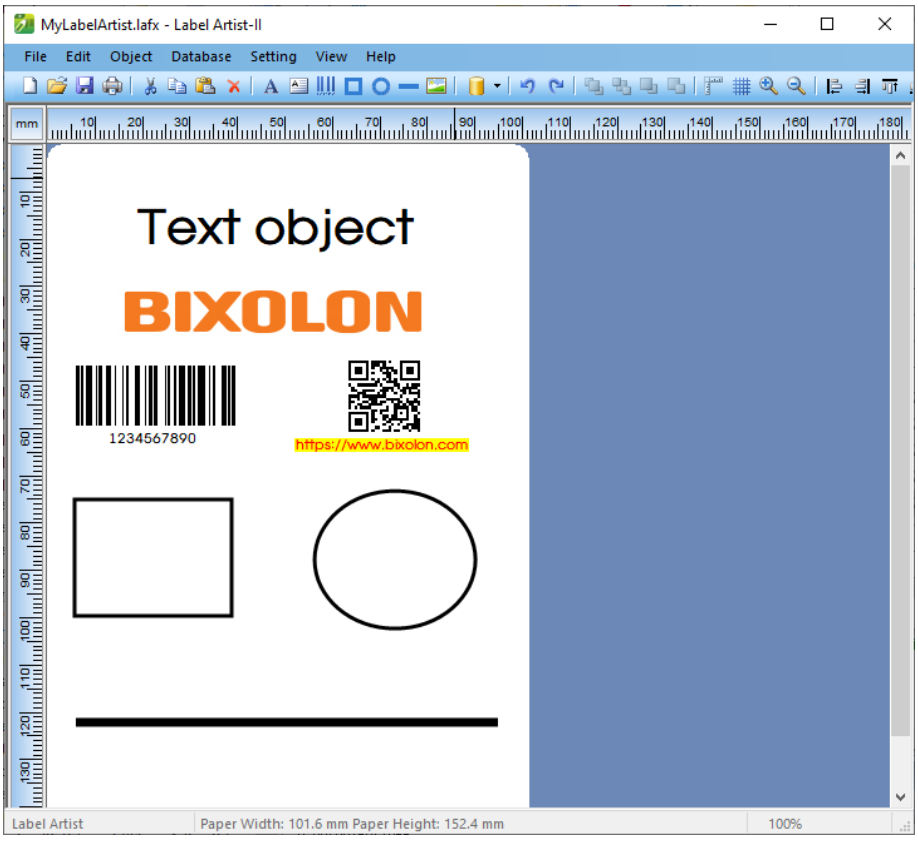
## 5-1-2 Open

1) Select <Open> from the <File> tab.

2) When the “Open” dialog is shown as below, select one of the files saved previously (*.lafx) and click [Open].



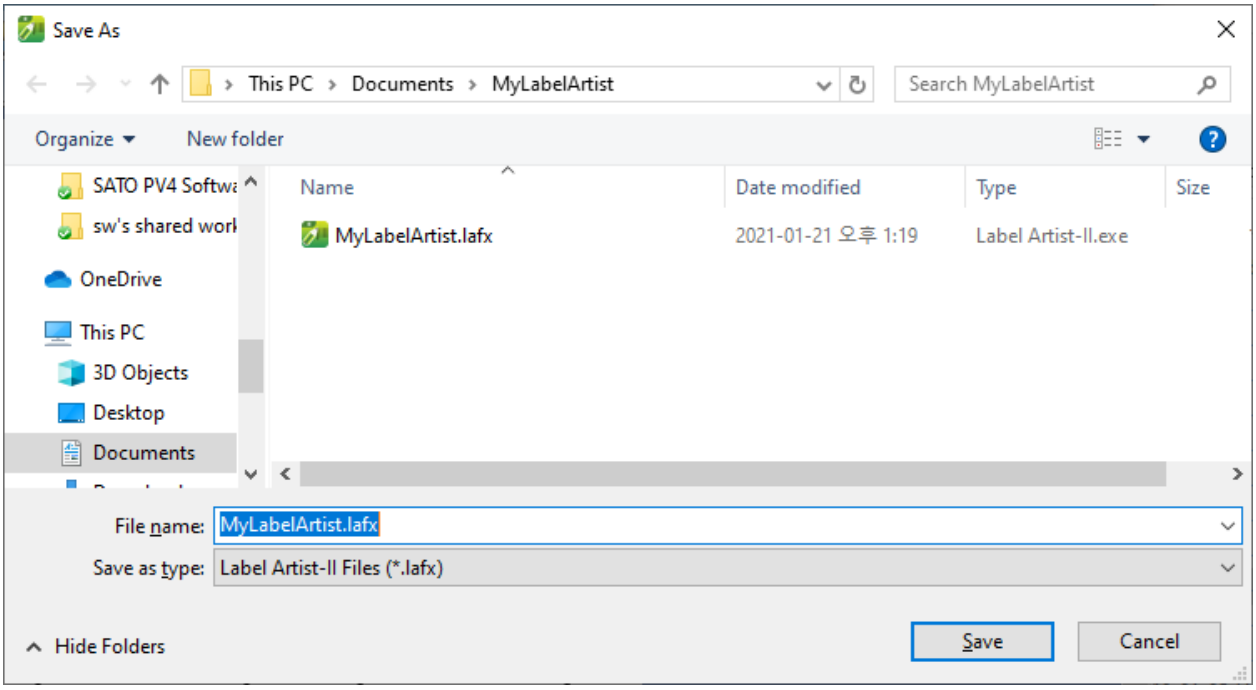

3) The content of the file is shown as below.



5-1-3 Save

1) Select <Save> from the <File> tab to save your progress.

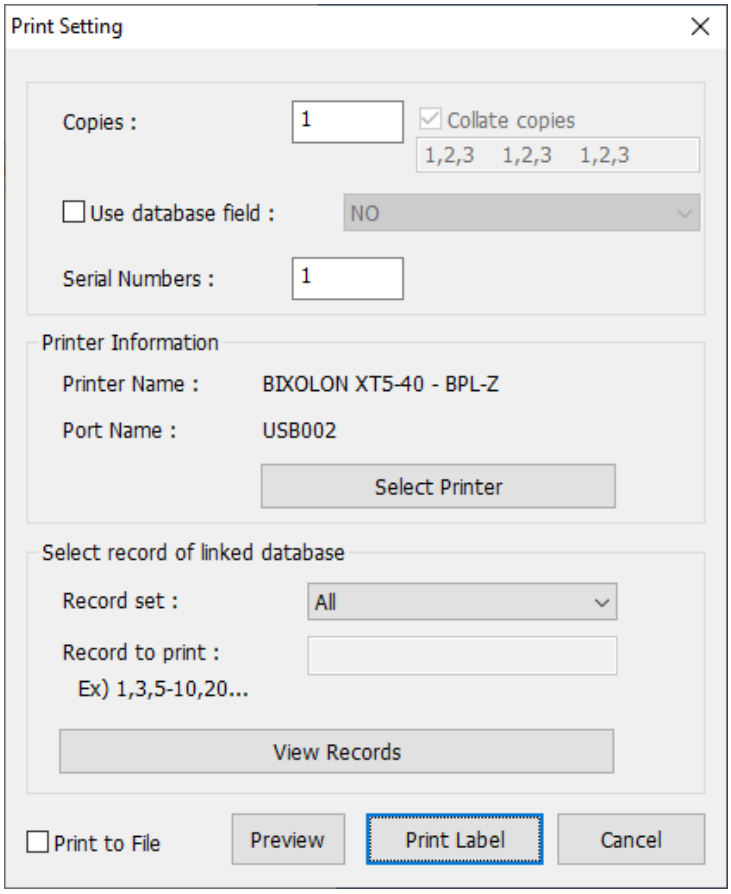
2) The dialog window opens as shown below when <Save As> is selected.
Specify the location and name of the file to be saved and click [Save] to save the label you created.



5-1-4 Print

1) Select <Print> from the <File> tab.

2) When the [Print Setting] window opens as below, specify the number of copies and click [Print Label] to start printing.




- Copies: Set the number of copies.
- Use database field: This can be used when printing objects connected to the database, and sets the number of copies based on the selected field.
- Serial Numbers: Set the printing range when using the auto counter.

**Collate copies**

The collate copies option is used for copies printing. When the option is checked all label is collated and print repeatedly. When the option is unchecked each label is printed repeatedly.

3) If there are any objects linked to the database, specify the record set to print the content of the database.
   - All: all rows are printed.
   - Selected: the selected rows are printed.

## 5-2 <Edit> Menu

### 5-2-1 Copy, Cut, Paste, Delete

1) Copy (Ctrl + C) 
   Copy the selected object (e.g. Text, Barcode, Line).

2) Cut (Ctrl + X) 
   Cut the selected object (e.g. Text, Barcode, Line).

3) Paste (Ctrl + V) 
   Paste the copied or cut object.

4) Delete 
   Delete the selected object.

### 5-2-2 Undo, Redo

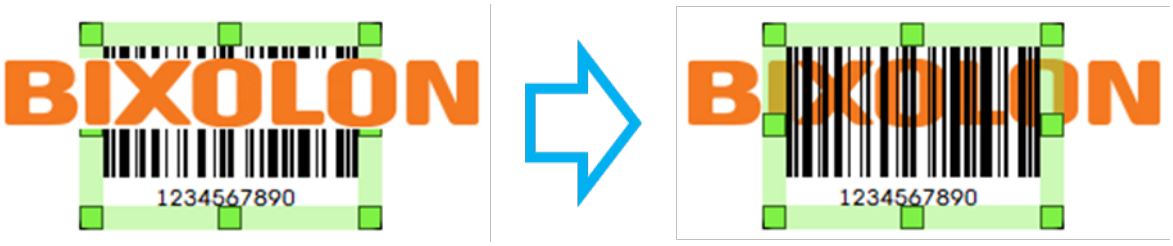
1) Undo (Ctrl + Z) 
   Undo the last action that you made on the label.

2) Redo (Ctrl + Y) 
   Redo the [Undo].

5-2-3 Order

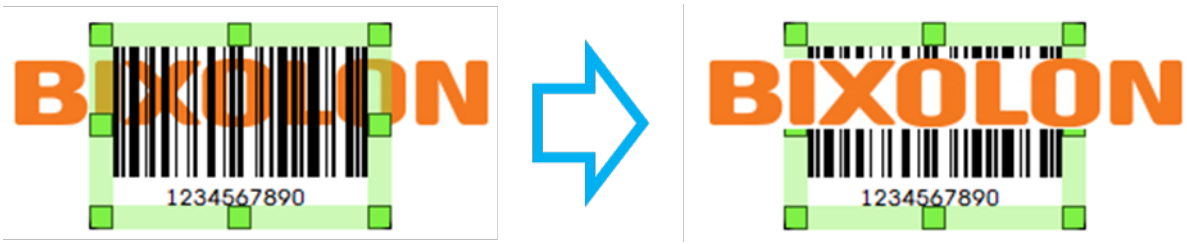
It indicates the order of drawing the objects. An object may be hidden behind.

1) Bring Forward
Reorder the selected object and the one that overlaps it.




2) Send Backward

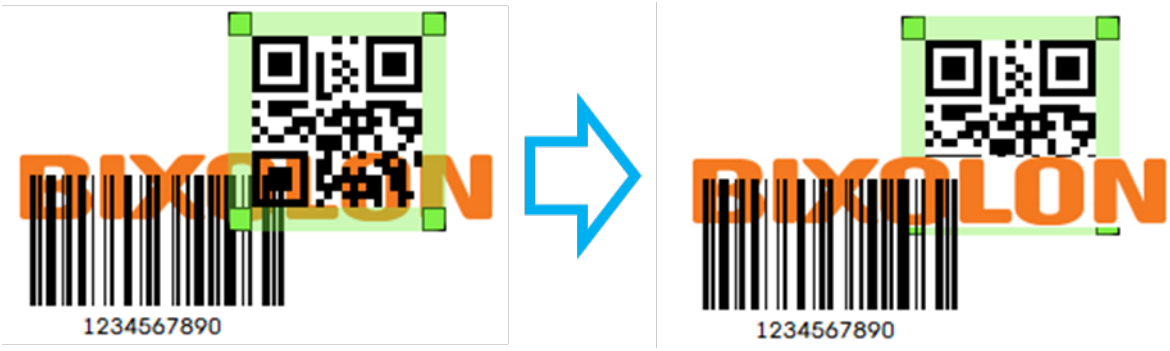
Reorder the selected object and the one that overlaps it.




3) Move To Front
Reorder the selected object and the one that overlaps iMove all the selected objects to the front-most level of any overlapping objects.

4) Send To Back

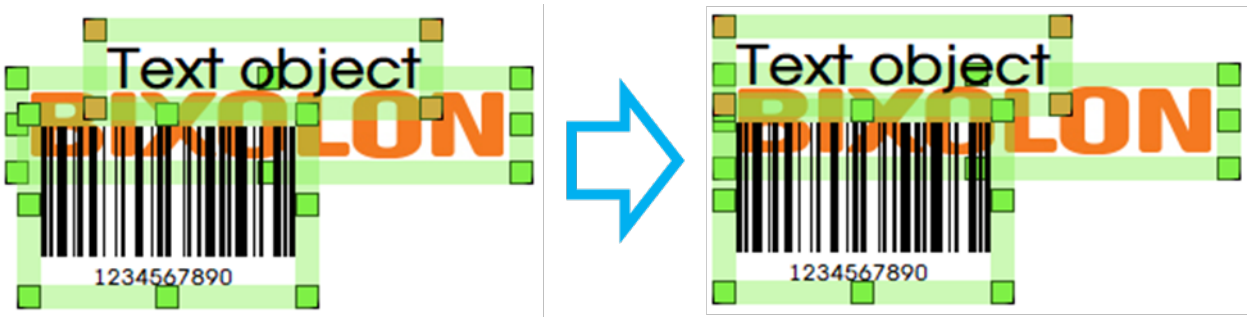
Move all the selected objects to the rear-most level of any overlapping objects.



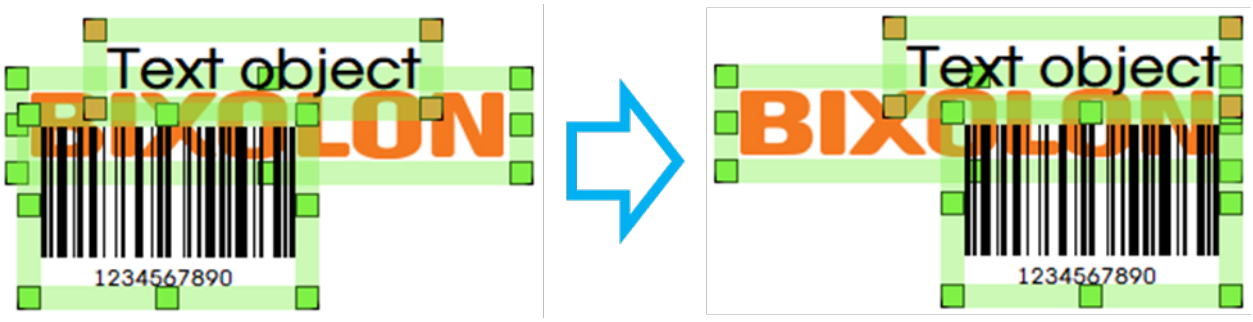
5-2-4 Align

"Align" allows you to line up objects to a specific object.
Use the Ctrl key to select multiple objects. The selected objects are aligned to the last selected object.

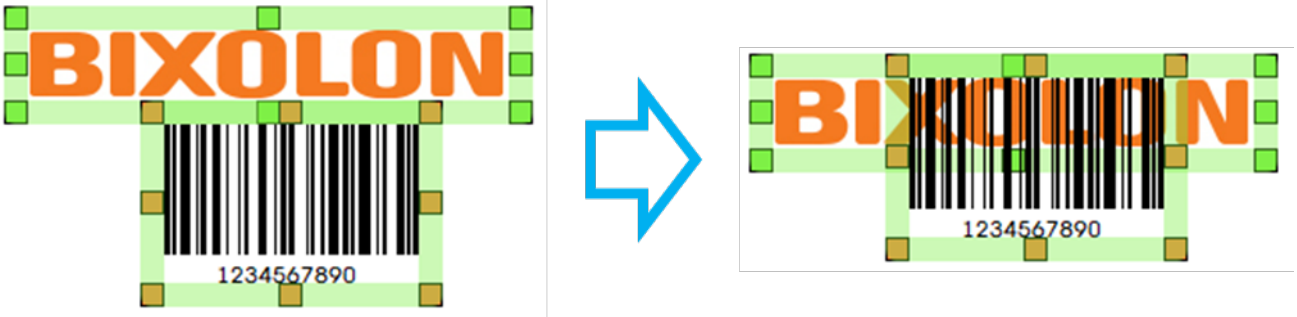
1) Align Left Sides
Align objects to the left side of the last selected object.




2) Align Right Sides
Align objects to the right side of the last selected object.

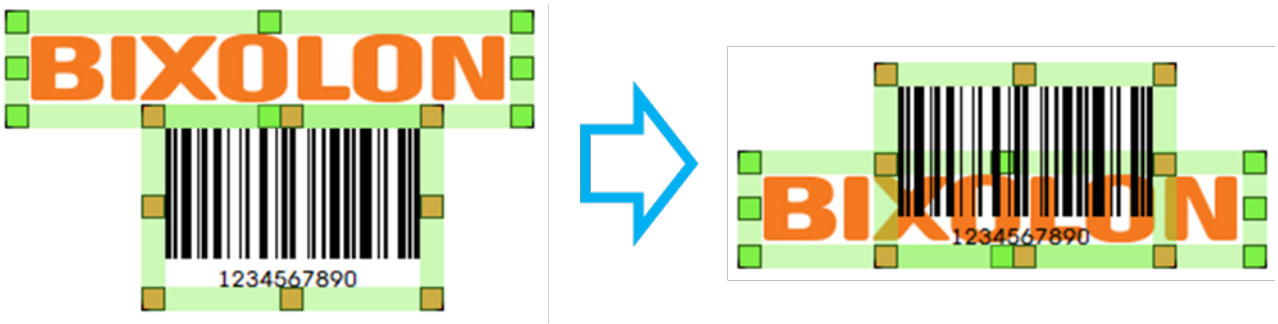



3) Align Top Sides
Align objects to the top of the last selected object.



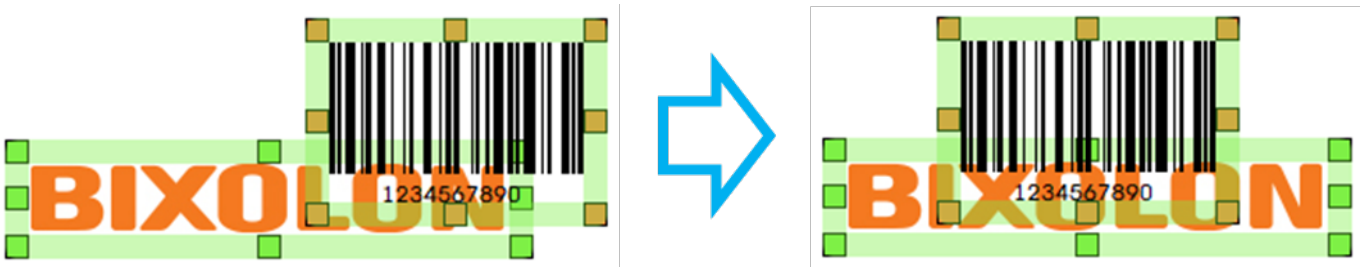
4) Align Bottom Sides

Align objects to the bottom of the last selected object.



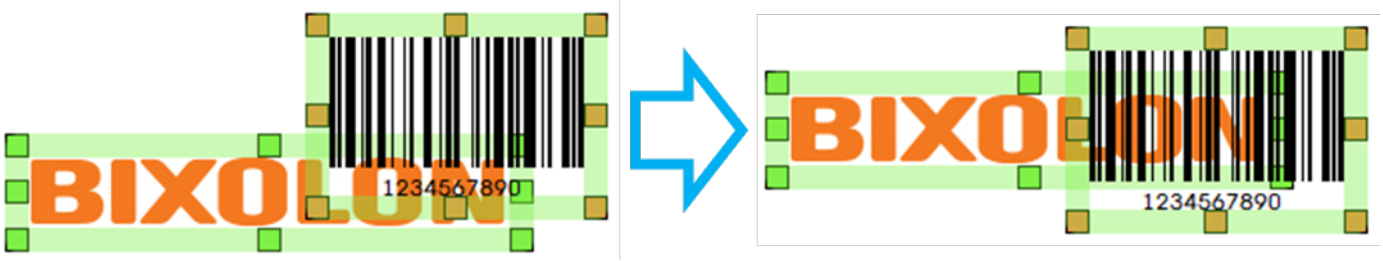

5) Center In Horizontal Region

Align the horizontal center of each object to the center of the last selected object.



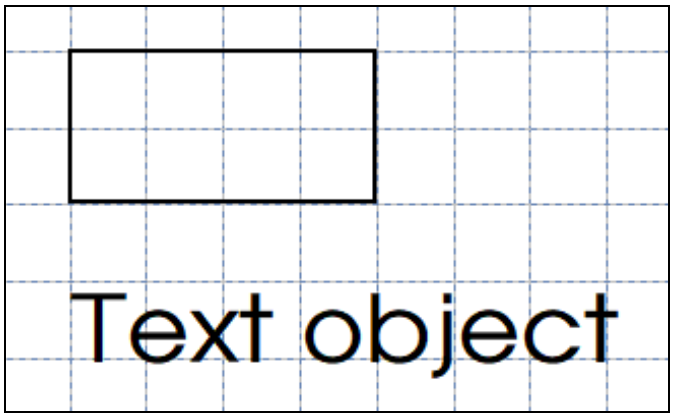

6) Center In Vertical Region

Align the vertical center of each object to the center of the last selected object.

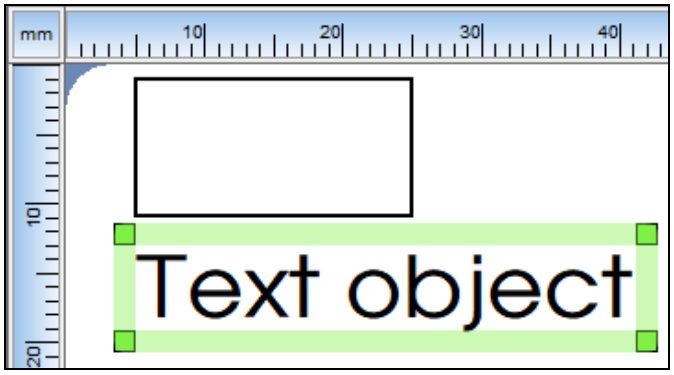


5-2-5 Snap To Grid/Ruler

1) Snap To Grid
When moving and resizing an object, it changes based on grid.




2) Snap To Ruler
When moving and resizing an object, it changes based on ruler.



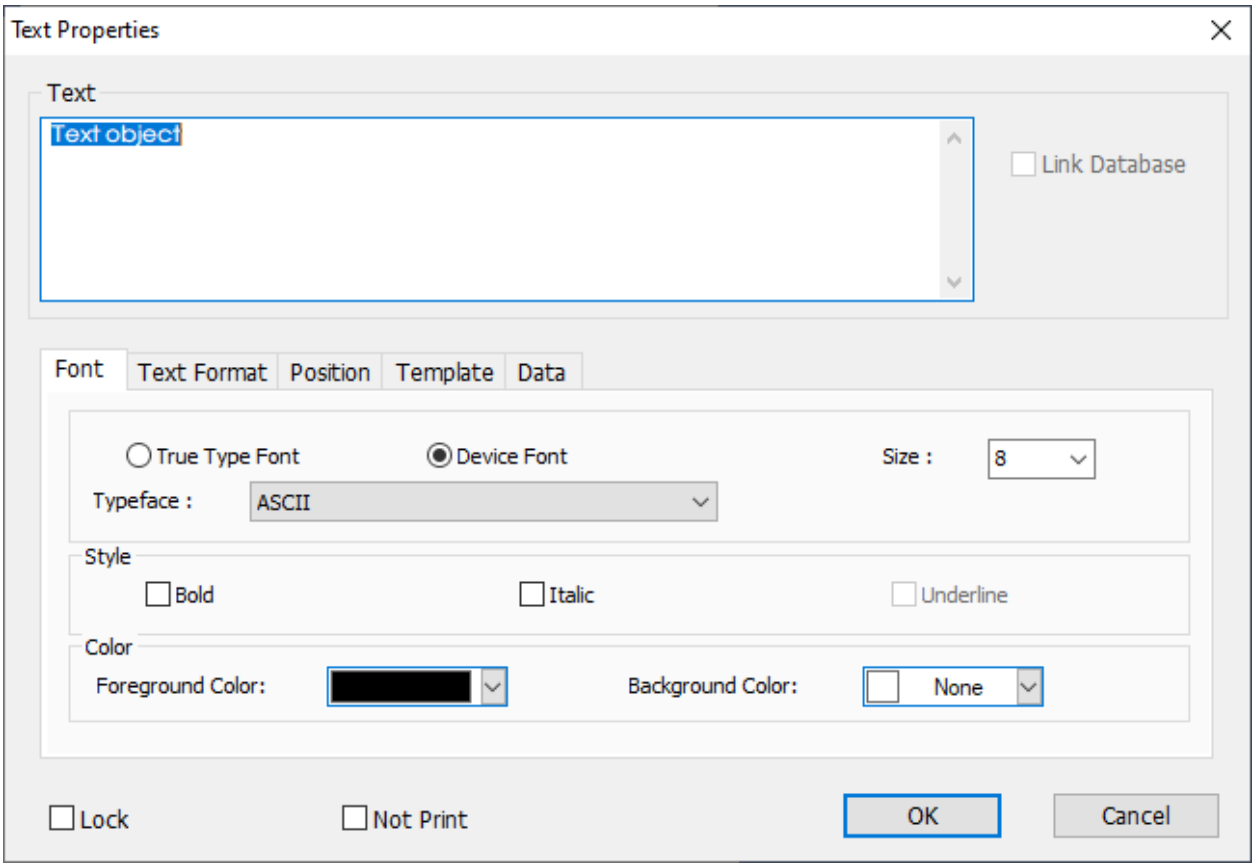
## 5-3 <Object> Menu

• Select objects (e.g. Text, Barcode and Line) to design.

### 5-3-1 Text

1) Select <Text A> from the <Object> tab and drag holding the left mouse button to create a text box with the default attributes.

2) Double-click on the text object to open a dialog window where you can change its attributes.



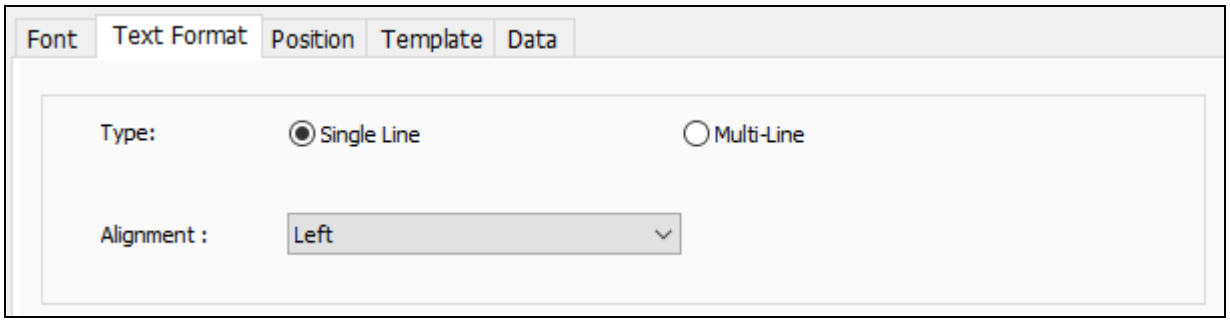
3) Set the font properties.

**Device Font**
The device font is hardware-resident fonts. It is possible to select as below types.

ASCII – alphanumeric and code page
KS5601 – Korean
BIG5 – Chinese( Traditional)
GB2312 – Chinese( Simplified)
Shift-JIS – Japanese
OCR-A - alphanumeric
OCR-B - alphanumeric

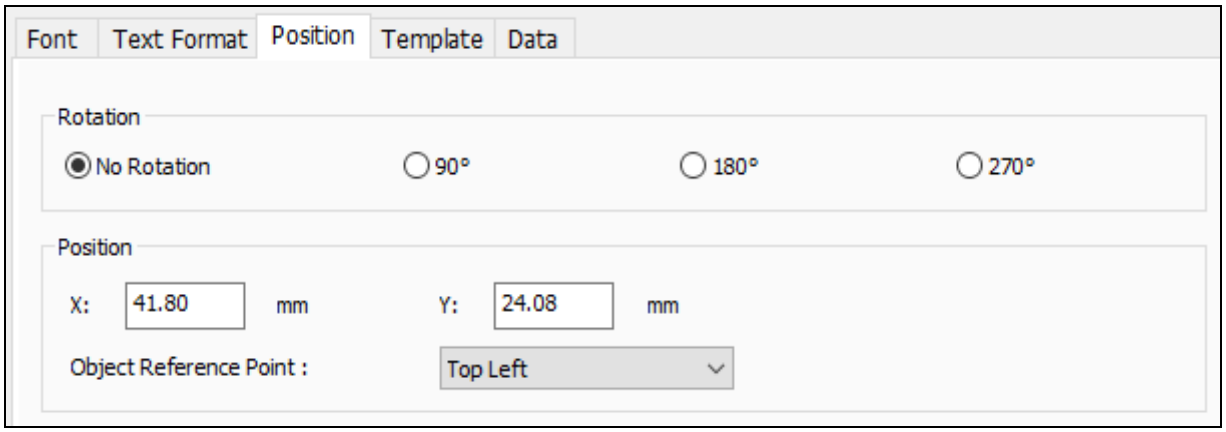
4) Set the text format.

- Single Line: Using single line text.
- Multi-Line: Using multi-line text.




When changing the text format, it is possible to switch to multi-line text object.

5) Set the position.




**Object Reference Point**

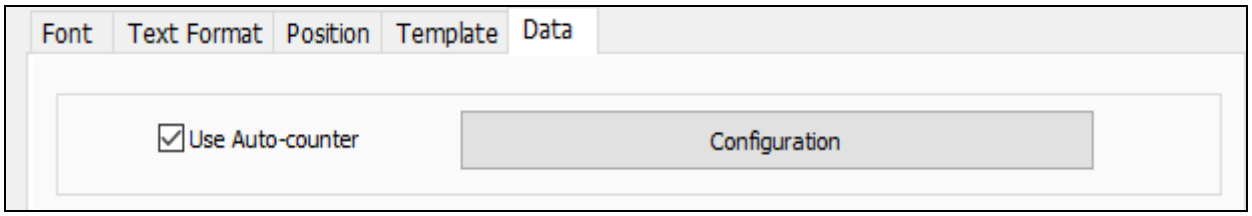
Reference position when moving and resizing due to object rotation and data change (auto counter, database connection, etc.)

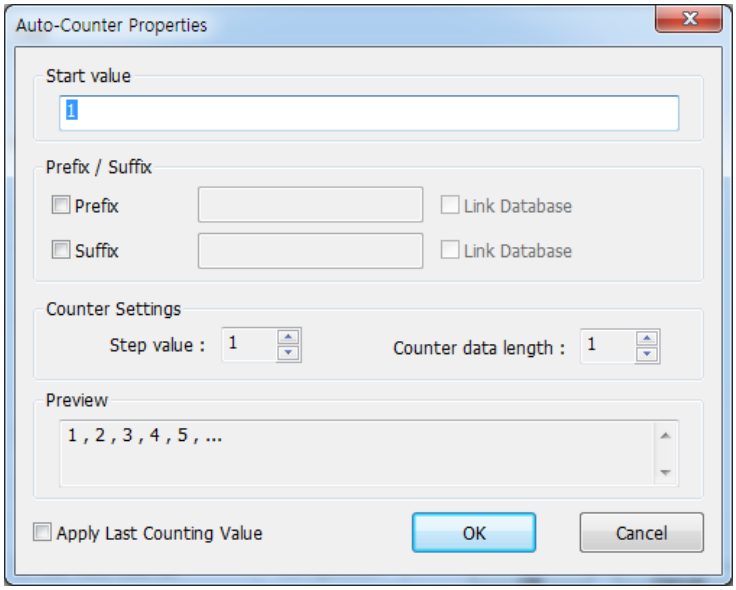
6) Use Template

When the device font is selected, the Template is able to use.
Please refer to 5-5 Template section for more understanding.

7) Use Auto-counter

Check “Use Auto-counter” to open the Auto-Counter Properties window as shown below. When “Auto-counter” is applied, the text changes to the [Start Value].




Check [Apply Last Counting Value] to change [Start Value] to the next value of the last counter set after printing.

## 5-3-2 Multi-line Text

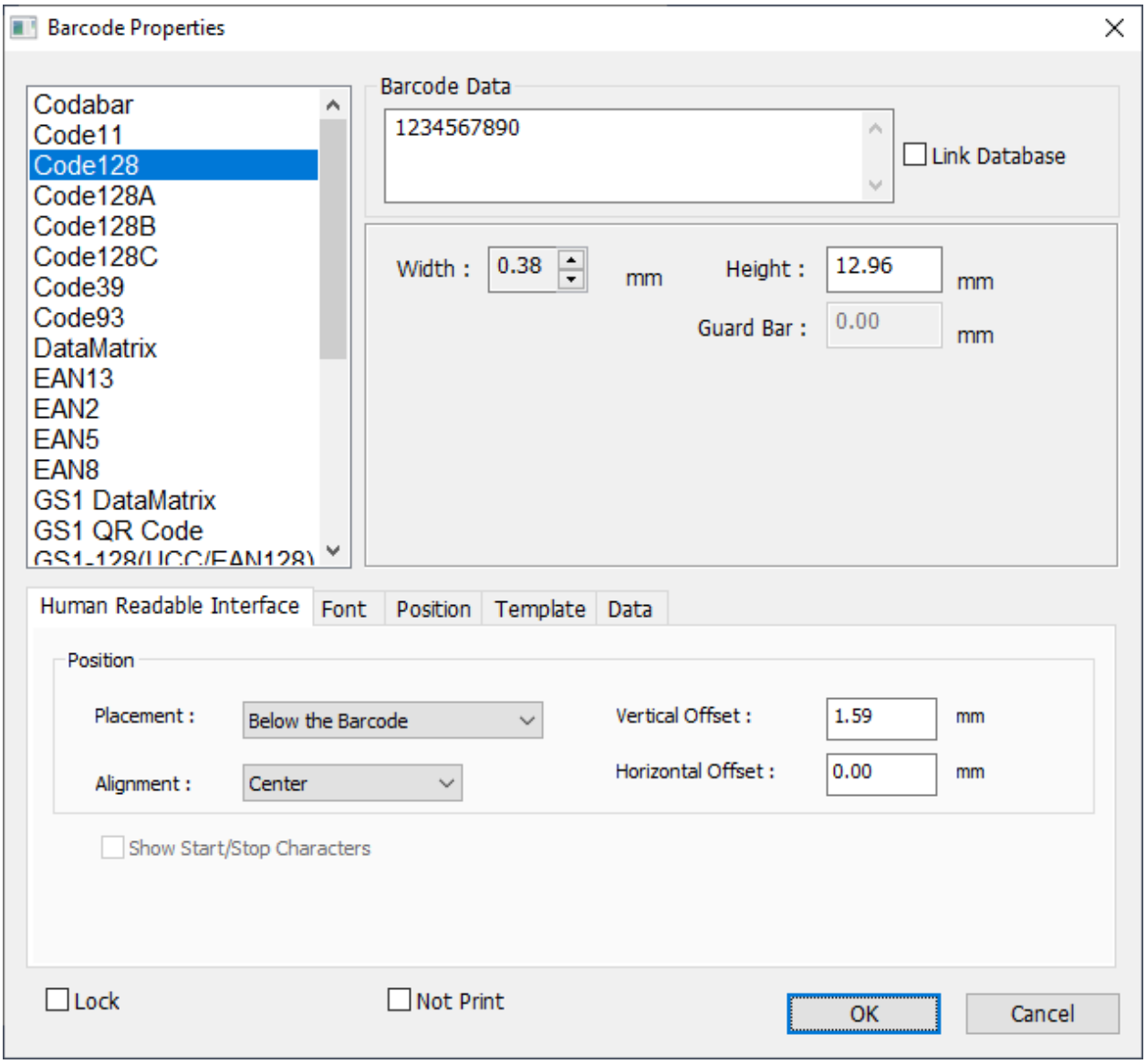
"Multi-line Text" allows to enter text in several lines.

When changing the text format, it is possible to switch to singe line text object.

5-3-3 Barcode

1) Select <Barcode> from the <Object> tab and drag holding the left mouse button to create a barcode with the default attributes.

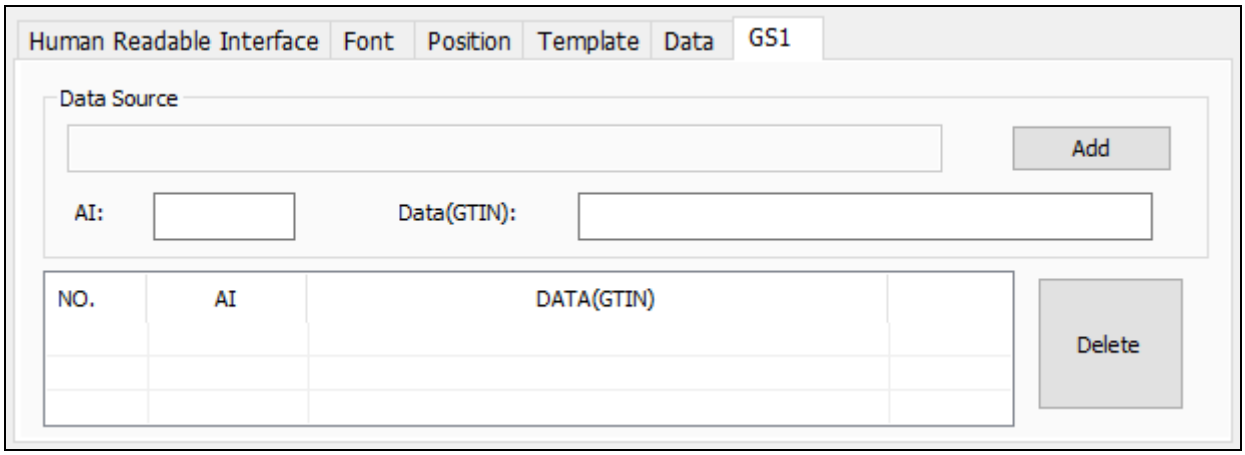
2) Double-click on the text object to open a dialog window where you can change its attributes.



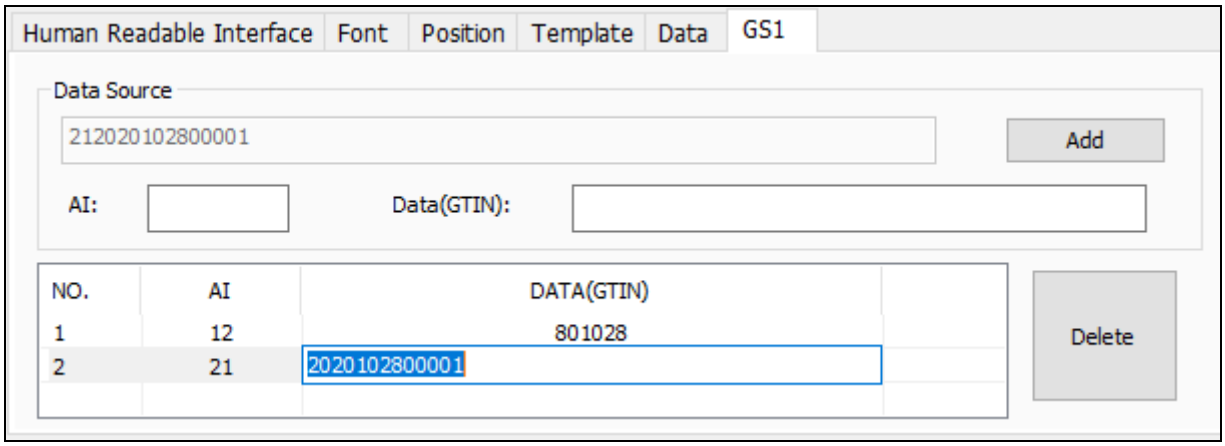

3) Set the barcode type and other attributes.

4) Set the HRI(Human Readable Interface).

5) Set the GS1 barcode.
The data of GS1 barcode is set in GS1 tab.

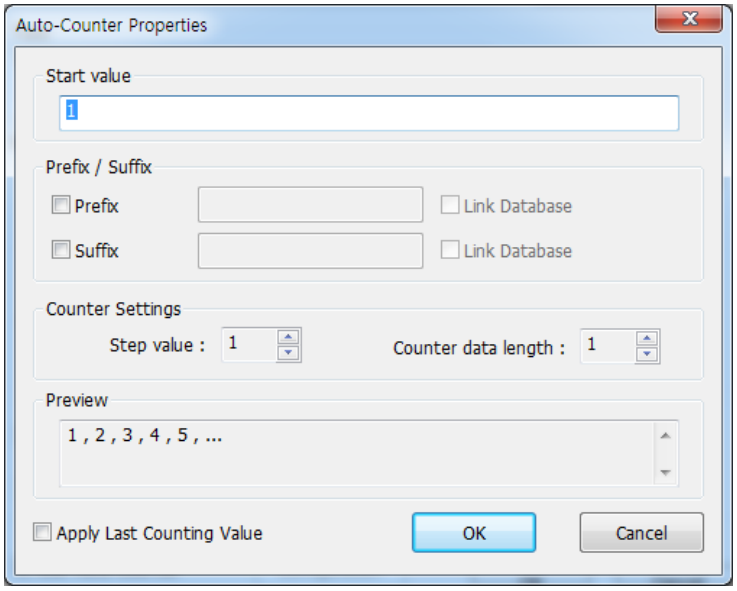



After entering AI and data, click the Add button.
You can change the item by double-clicking the item in the list.



6) Use Auto-counter

Check “Use Auto-counter” to open the Auto-Counter Properties window as shown below. When “Auto-counter” is applied, the text changes to random data.




Check [Apply Last Counting Value] to change [Start Value] to the next value of the last counter set after printing.

7) Use Template

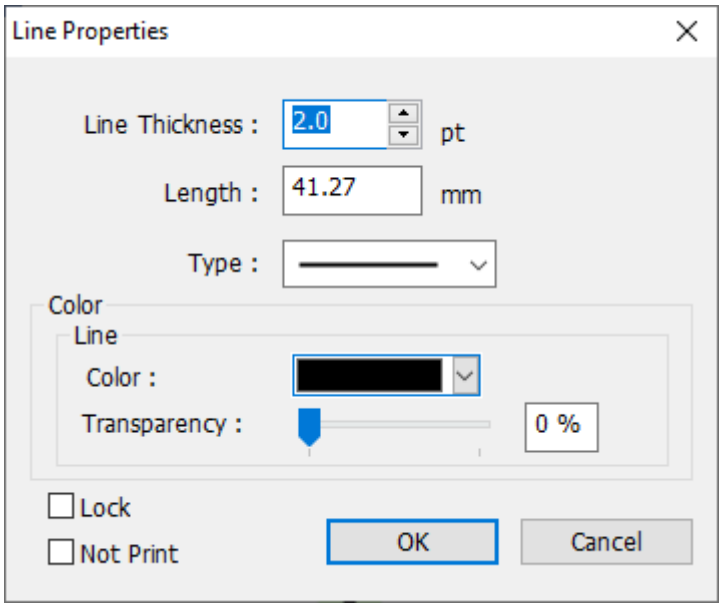
When the 1D Barcode is selected, the Template is able to use.
Please refer to 5-5 Template section for more understanding.

5-3-4 Line

1) Select <Line> from the <Object> tab and drag holding the left mouse button to draw a line.

2) Drag holding the Shift key to draw a diagonal or straight line.

3) Double-click on the object to open a dialog window where you can change its attributes.



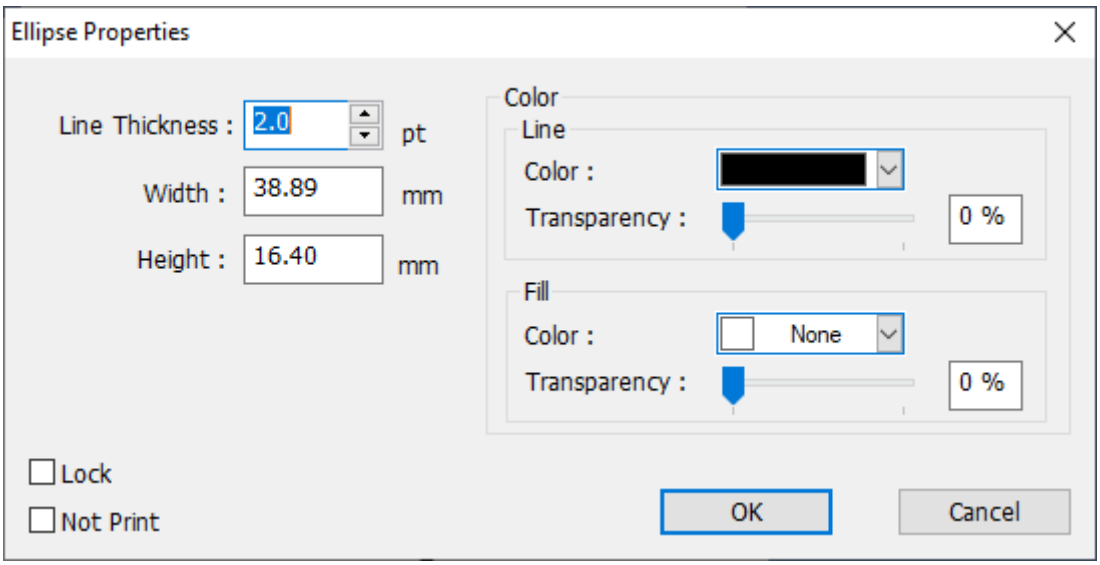

4) Set the line type and other attributes and click [OK] to apply the changes.

5-3-5 Rectangle, Ellipse

1) Select <Rectangle> or <Ellipse> from the <Object> tab and drag holding the left mouse button to draw a shape.

2) Drag holding the Shift key to draw a shape with the same width and height.

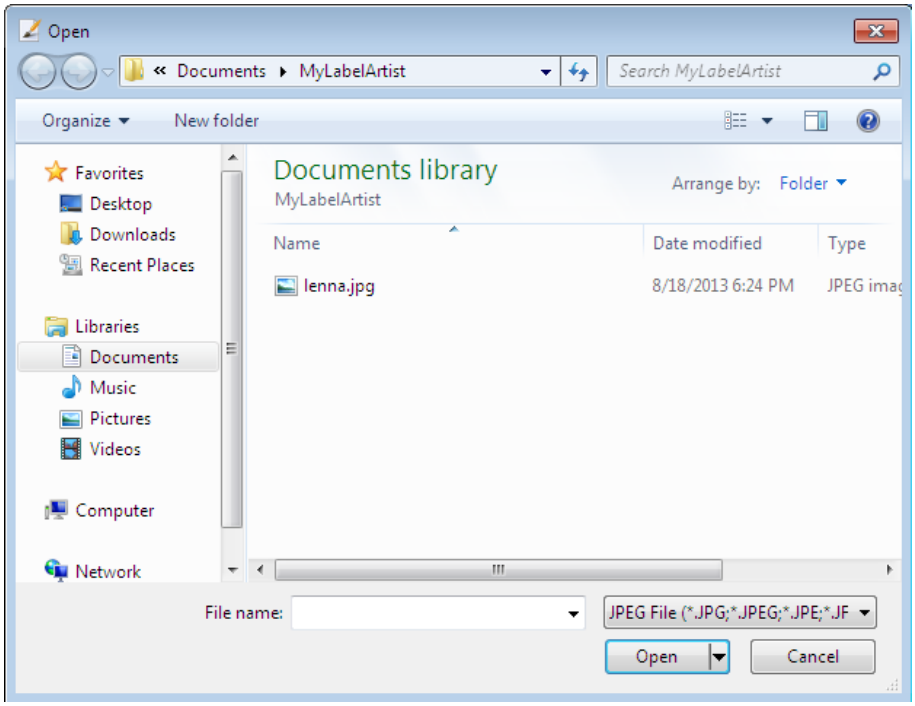
3) Double-click on the object to open a dialog window where you can change its attributes.




4) Set the line type and other attributes and click [OK] to apply the changes.

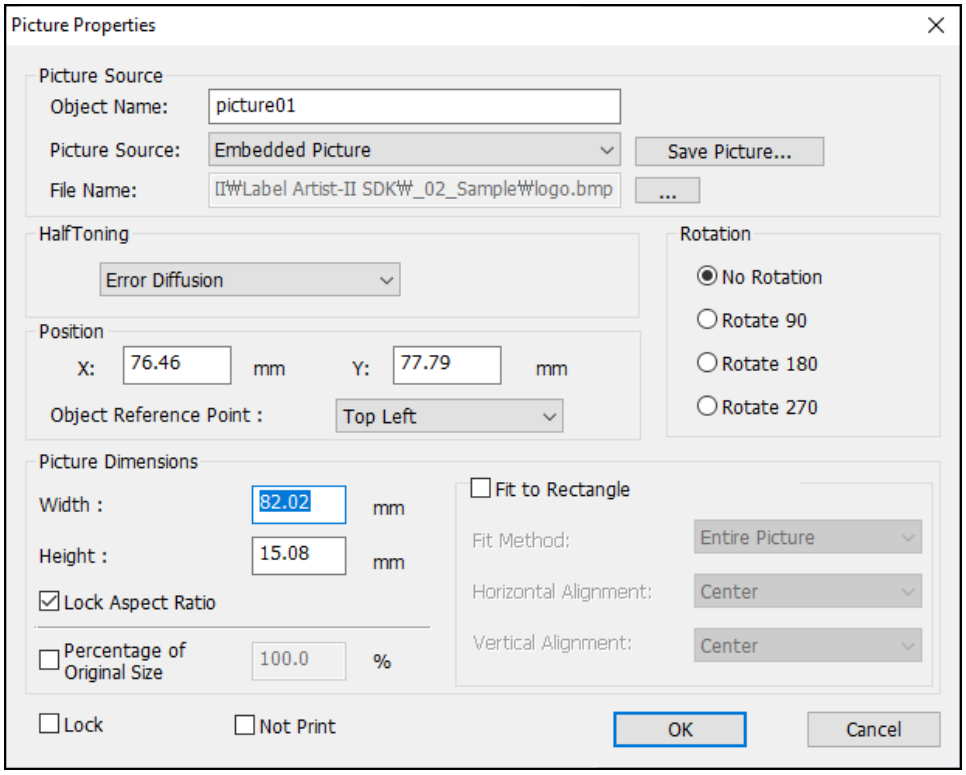
5-3-6 Picture

1) Select <Picture> from the <Object> tab to open a dialog window where you can select an image file.

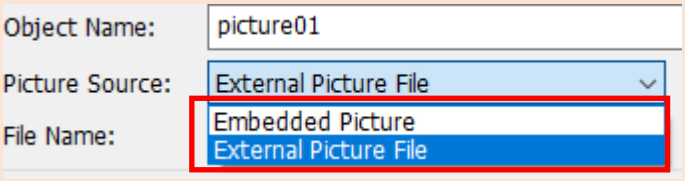



2) Select an image file, click “Open” and drag holding the left mouse button to place the image on the screen.

3) Double-click on the object to open a dialog window where you can change its attributes.




**Picture Source**



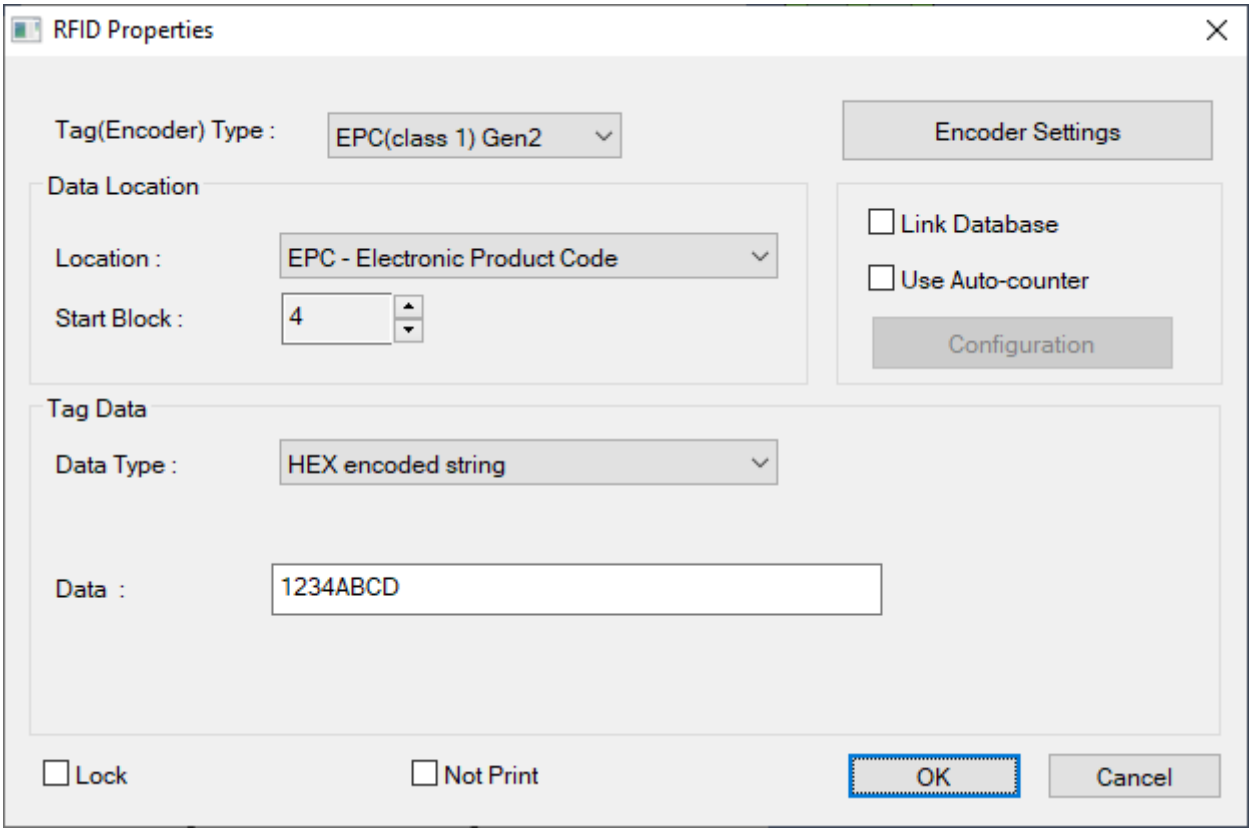

- **Embedded Picture**: When loading pictures, pictures are included in the label design file and printed without a separate reference when printing labels.
- **External Picture File**: The label design file contains the path of the picture so that when the label is printed, the picture of the path is referenced and printed.

4) Set the type and other attributes and click [OK] to apply the changes.

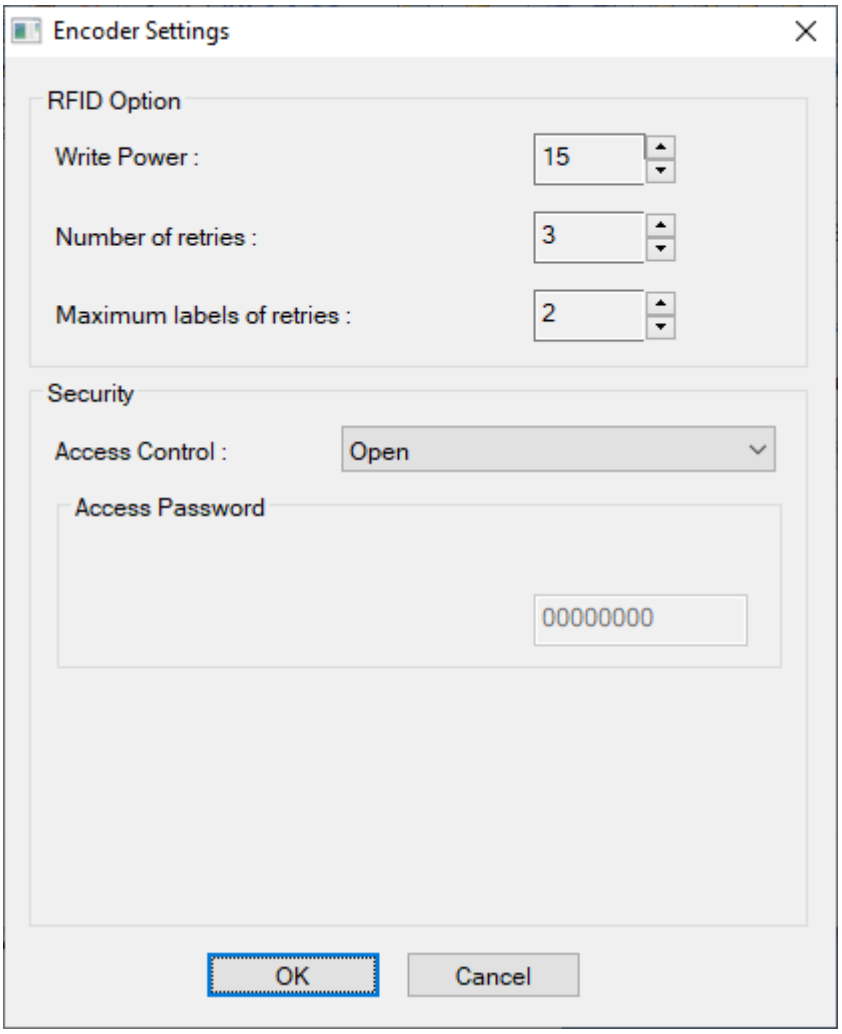
5-3-7 RFID

1) Select <RFID> from the <Object> tab to create RFID object as shown below.


2) Double-click on the object to open a dialog window where you can change its attributes.



3) Click the Encoder Settings button to select an encoder option.




RFID objects can be used when using a printer that supports RFID.

RFID supported model
: SLP-T400R/T403R, SLP-TX400R/TX403R, XT5-40NR/43NR/46NR, XD5-40tR/43tR

5-3-8 Misc.

Lock: Enable this feature in the dialog window where you can set the attributes of each object.

- The locked objects cannot be moved or resized.

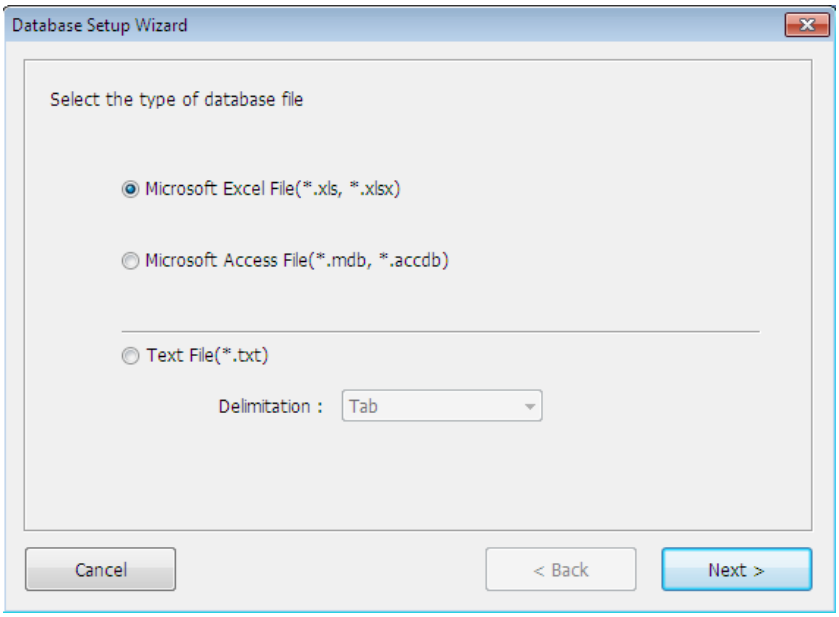
Not Print: Enable this feature in the dialog window where you can set the attributes of each object.

- The set objects cannot be printed.

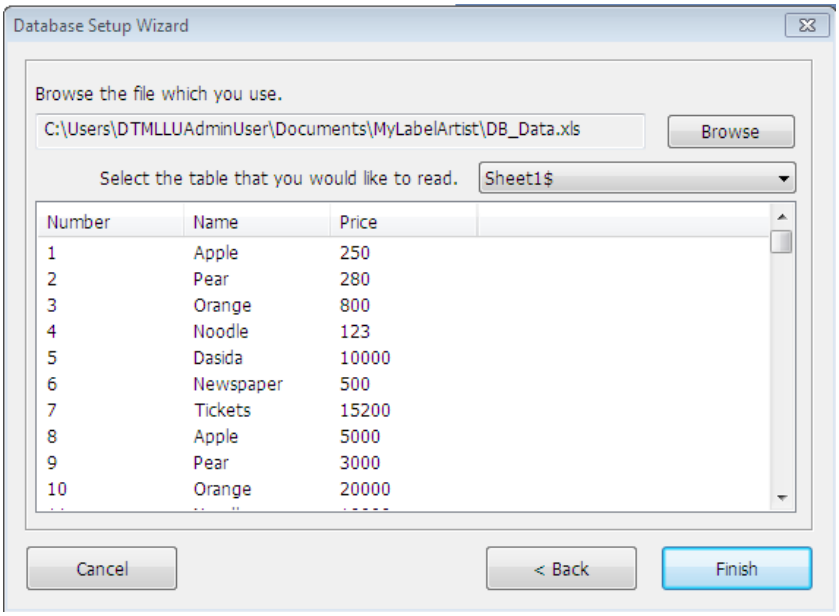
## 5-4 <Database> Menu

### 5-4-1 Connect Database

1) Select <Connect Database> from the <Database> tab.
Select the type of database to use and click “Next”.




2) Click “Browse” to select a database file as shown below.




3) Select a table and click “Finish” to link the database when designing a label.

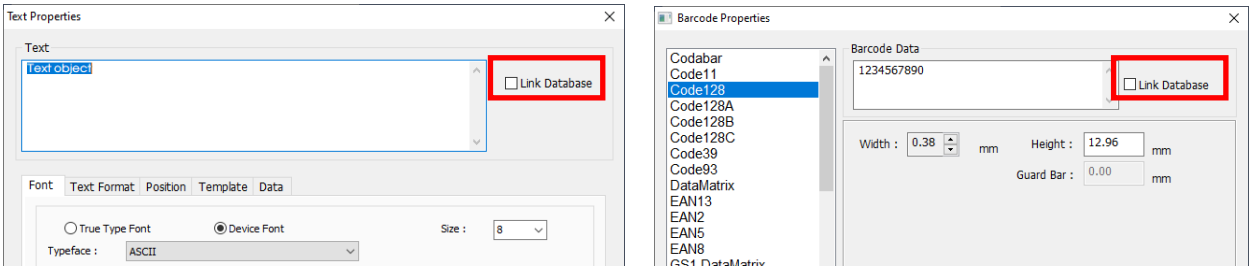
Make sure to disconnect the current database to use another one.

## 5-4-2 Disconnect Database

Select <Disconnect Database> from the <Database> tab to disconnect the database.

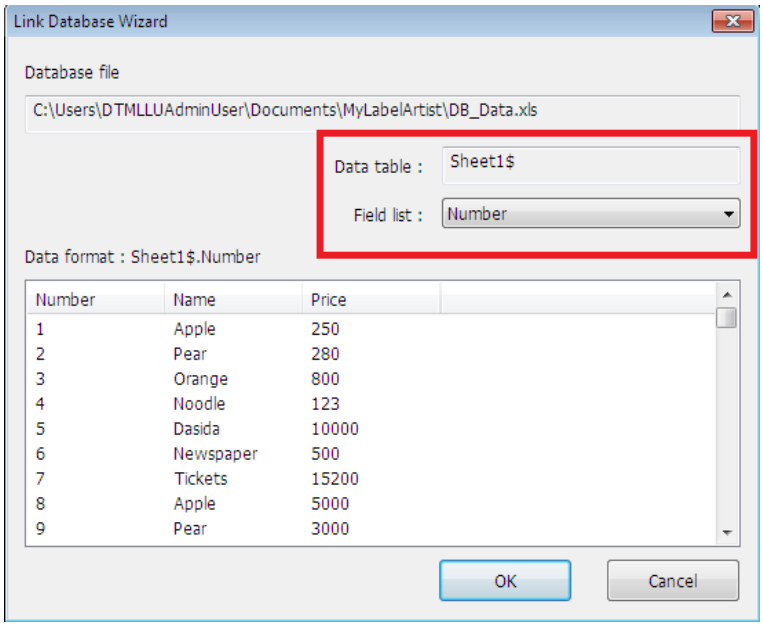
## 5-4-3 Link Database

1) Check “Link Database” in the <Text> or <Barcode> dialog window.




The check box is enabled to check when the <Connect Database> is finished on <Database> menu.

2) Select a field you want to use from the field list.




3) When a text object is linked to the database, the following message appears.
[WorkSheetName].[FieldName]
Ex) Sheet1$.Num

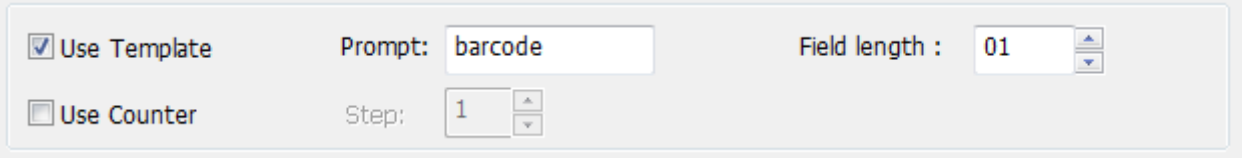
4) When a barcode object is linked to the database, a barcode with random data is created on the screen.

## 5-5 Template

• This section explains setting and exporting for BIXOLON Template and BIXOLON XML Enabled.

### 5-5-1 Template Selection

1D barcode and device font objects can be used template. Other objects as rectangle, picture, and line are inserted into template by background image.




1) Use Template

Please check the “Use Template” box and then write prompt and field length.

Prompt?
This text string is transmitted to host(PC) by serial interface in order to give information to host about the declared counter.

In case of “BIXOLON XML Enabled” type, the prompt is used for variable name.

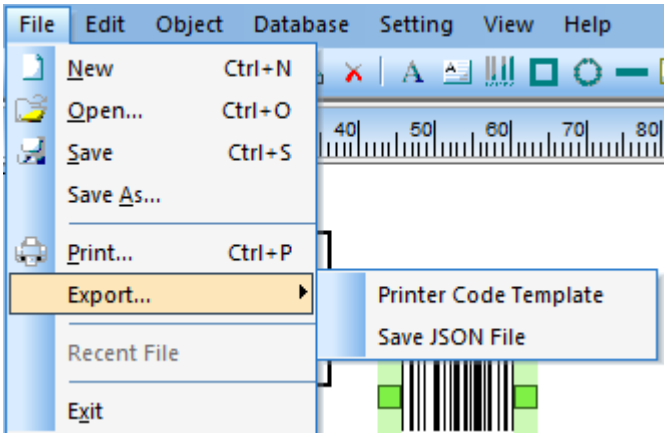
2) Use Counter

Please check the “Use Counter” box and then adjust the step value. Counter can be used up to 10 in one template.

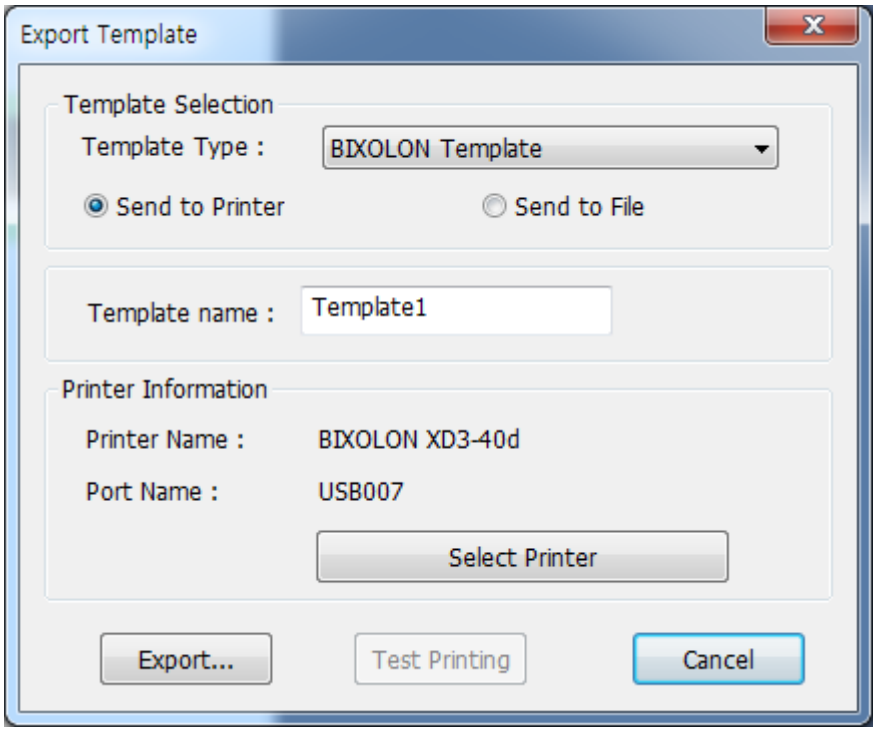
The “Use Counter” is enable for “BIXOLON Template” type.

5-5-2 Export Template

1) <File> - <Export> - <Printer Code Template> select

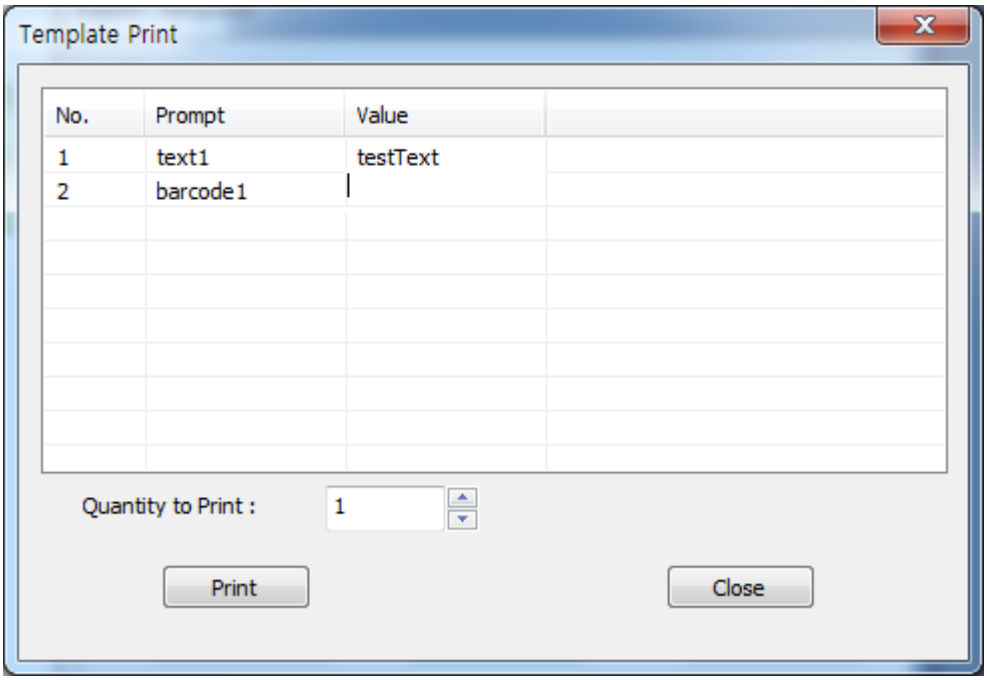



2) Template exporting
Select template type, transmission type and template name, and then click "Export..." button as below dialog.
When the exportation is finished, the "Test Printing" button is enabled.

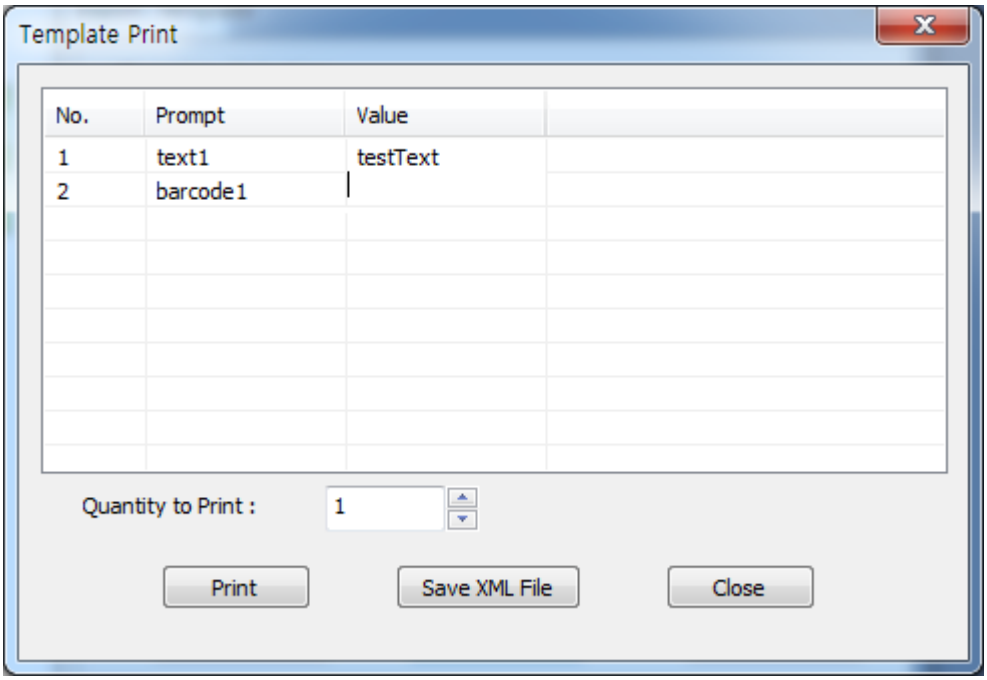



Template Type
- BIXOLON Template: Exporting a SLCS-based template form to the printer.
- BIXOLON XML Enabled: Exporting a XML enabled form to the printer.
- BIXOLON XML Enabled supported model T5-40/43/46

3) Test printing for “BIXOLON Template”
Insert text data into “Value” and quantity, and then click “Print” button.




4) Test printing for “BIXOLON XML Enabled”
Insert text data into “Value” and quantity, and then click “Print” button.




When “Save XML File” button is clicked, XML document is saved.

### 5-6 Save JSON file

• <File> - <Export> - <Save JSON file> menu can be saved JSON file can be loaded and print on Label Artist Mobile.

### 5-7 <View> Menu

1) Zoom In
   - Enlarges the view of the entire layout by 20%.

2) Zoom Out
   - Reduces the view of the entire layout by 20%.

3) 100% Size
   - Returns to the original size.

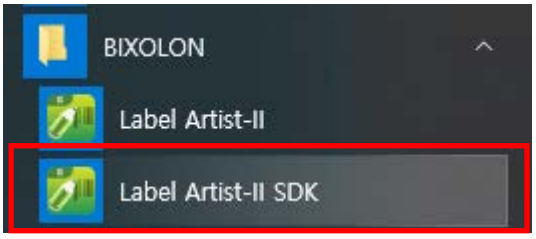
4) Fit to the height
   - Enlarge/reduce the view to fit the height of the screen.

5) Fit to the width
   - Enlarge/reduce the view to fit the width of the screen.

# 6. Label Artist-II SDK

Label Artist-II SDK is provided to automatically print Label Artist-II documents (.lafx) from customer applications. By designating text and barcode objects whose data changes as variables, the application supports the function of substituting the values of variables and printing them.

Label Artist-II SDK is located at <Start>-<Programs>-<BIXOLON>-<Label Artist-II SDK>.




The Label Artist-II SDK provides the C ++ libraries(Dll) and C # class.

-C# components

| File name | Description | Location |
|---|---|---|
| LabelArtistAPI.dll | x86 DLL file | _01_Bin folder |
| LabelArtistAPI_x64.dll | x64 DLL file | _01_Bin folder |
| LabelArtistAPI.cs | C# class | C# sample |

-C++ components

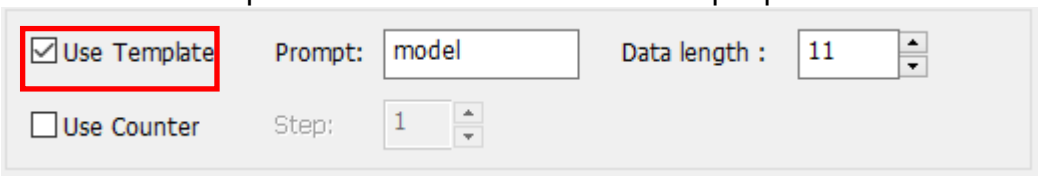
| File name | Description | Location |
|---|---|---|
| LabelArtistAPI.dll | x86 DLL file | _01_Bin folder |
| LabelArtistAPI.lib | x86 LIB file | _01_Bin folder |
| LabelArtistAPI_x64.dll | x64 DLL file | _01_Bin folder |
| LabelArtistAPI_x64.lib | x64 LIB file | _01_Bin folder |
| LabelArtistSDK.h | Header file | VC++ sample |

## 6-1 Variable setting

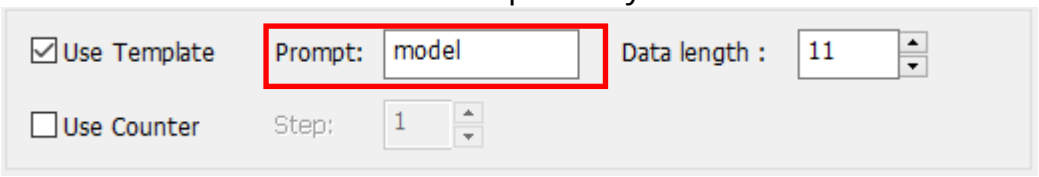
• In Label Artist-II, you can set text and barcode objects as variables. The variable name and initial value should be inputed.

The Variables can only be set up with text (single-line text) and barcode objects.

Select “Use Template” in the Text and Barcode properties.




Write the variable name to “Prompt” entry.




Variable names cannot be duplicated.

## 6-2 C# Programming

• Label Artist-II SDK provides C # sample written in Visual Studio 2015 (.NET 4.5) for C # programming. LabelArtist, a wrapping class that imports the C ++ library (LabelArtistAPI.dll), is provided. The LabelArtist class is defined in LabelArtistAPI.cs included in the C # sample.

To use LabelArtist class, DLL (LabelArtistAPI.dll, LabelArtistAPI_x64.dll) files must be included in package when application is distributed.

### 6-2-1 LabelArtist Properties

The properties of the LabelArtist class consist of a list of variables and return information for each method.

1) NamedSubStrings
   NamedSubStrings is collection of SubString containing variable information. If the OpenLabel method is called successfully, the list information is updated.

Properties of SubString class

| Properties | Description |
|---|---|
| Name | Variable name |
| Value | Variable value |

2) Return Information of Methods
These properties define the error values returned after methods are called. See the table below for error details.

| LabelArtist properties | Value | Description |
|---|---|---|
| BXL_SUCCESS | 0 | Method execution successful |
| BXL_ERR_SEARCH_FILE | -100 | File not found |
| BXL_ERR_SEARCH_PRINTER | -101 | Printer not found |
| BXL_ERR_SEARCH_OBJECT | -102 | Object not found |
| BXL_ERR_OPEN_FILE | -103 | File open failed |
| BXL_ERR_OPEN_PRINTER | -104 | Printer connection failed |
| BXL_ERR_SUPPORT_FILE | -105 | Unsupported file |
| BXL_ERR_SUPPORT_PRINTER | -106 | Unsupported printer |
| BXL_ERR_PARAMETER | -107 | Invalid input value (parameter) |
| BXL_ERR_ALREADY_OPEN | -108 | Already connected with printer |
| BXL_ERR_NOT_OPEN | -109 | No connection with the printer |
| BXL_ERR_VARIABLE_NAME | -110 | Invalid variable name used |
| BXL_ERR_PRINT_RESOURCE | -111 | Printer Resource Usage Failed |
| BXL_ERR_PRINT | -112 | Printing failed |

6-2-2 LabelArtist Methods

The LabelArtist class provides the functionality of opening Label Artist-II documents (.lafx), setting variable values, and printing.

1) OpenLabel
Open a Label Artist-II document and specify the printer to print. Update NamedSubStrings if OpenLabel method call is successful.

**[Syntax]**
public int OpenLabel(string filePath, string printerName)

**[Parameters]**
string filePath: Label Artist-II document(.lafx file) path
string printerName: Printer Driver Name(can be blank, Optional)

**[Returns]**
If the method succeeds, the return value is zero(0). Other cases, returns non-zero.

• If the printerName is empty, specify the printer to print in the followingorder.
1. Print to the printer specified in the input file (.lafx)
2. If the printer specified in the file is not on the system, print to the system default printer.
• Please use the **OpenLabelW** method when using Unicode strings.

2) CloseLabel
Close the currently opened Label Artist-II document.

**[Syntax]**
public int CloseLabel()

**[Parameters]**
None

**[Returns]**
If the method succeeds, the return value is zero(0). Other cases, returns non-zero.

3) SetNamedSubStringValue
Set the value of the variable.

**[Syntax]**
public int SetNamedSubStringValue(string name, string value)

**[Parameters]**
string name: Variable name
string value: Variable value

**[Returns]**
If the method succeeds, the return value is zero(0). Other cases, returns non-zero.

Please use the **SetNamedSubStringValueW** method when using Unicode strings.

4) Find
Get object information by object name.

**[Syntax]**
public DesignObject Find(string objectName)

**[Parameters]**
string objectName: Object name

**[Returns]**
the method succeeds, the return DesignObject reference. Other cases, returns null.

Properties of DesignObject class

| Properties | Description |
|---|---|
| X | Left position |
| Y | Top position |
| Width | Width of object |
| Height | Height of object |

5) PrintOut
Print opened Label Artist-II Document.

**[Syntax]**
public int PrintOut()

**[Parameters]**
None

**[Returns]**
If the method succeeds, the return value is zero(0). Other cases, returns non-zero.

6) Save
Changed object information is applied to Label Artist-II document and saved to file.

**[Syntax]**
public int Save()

**[Parameters]**
None

**[Returns]**
If the method succeeds, the return value is zero(0). Other cases, returns non-zero.

## 6-2-3 Example of using LabelArtist class

Use the LabelArtist class to describe how to program C #.

1) Get variables after opening Label Artist-II document
   Open a Label Artist-II document with the OpenLabel method and get the variable name and initial value into the NamedSubStrings Collection.

```
LabelArtist format = new LabelArtist();
int ret = format.OpenLabel(filePath, printerName);
if (ret == LabelArtist.BXL_SUCCESS)
{
    DataTable table = new DataTable();
    table.Columns.Add("Variable Name");
    table.Columns.Add("Variable Value");
        // Getting for variable name, value
    foreach (SubString laLabelVar in format.NamedSubStrings)
    {
        table.Rows.Add(laLabelVar.Name, laLabelVar.Value);
    }
}
```

2) Print document after setting variable value
   Set the value of the variable with the SetNamedSubStringValue method and print document with the PrintOut method.

```
DataRow[] rows = table.Select();
for (int i = 0; i < rows.Length; i++)
{
    format.SetNamedSubStringValue(rows[i]["Variable Name"].ToString(),
        rows[i]["Variable Value"].ToString());
}
format.PrintOut();
```

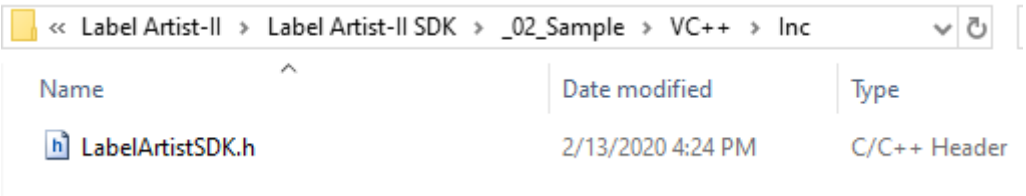
3) Save after changing object location

```
DesignObject obj = format.Find("picture01");
if(obj != null)
{
    obj.X += 3; //3mm
    obj.Y -= 2;
    obj.Width -= 5;
    obj.Height += 2;
    format.Save();    //Save
}
```

## 6-3 C++ Programming

• The Label Artist-II SDK provides VC++ samples written in Visual Studio 2015 for C++ programming. The VC++ sample is based on the Label Artist-II document Sample.lafx provided with the sample.


Please refer to the header file (LabelArtistSDK.h) included in the VC++ sample for the API and defined return values provided in the Label Artist-II SDK.



6-3-1 Label Artist-II SDK Methods

1) OpenLabel
Open a Label Artist-II document and specify the printer to print.

**[Syntax]**
long OpenLabel(LPCSTR filePath, LPCSTR szPrinterName = "")

**[Parameters]**
LPCSTR filePath: Label Artist-II document(.lafx file) path
LPCSTR szPrinterName: Printer Driver Name(can be blank, Optional)

**[Returns]**
If the method succeeds, the return value is zero(0). Other cases, returns non-zero.

• If the szPrinterName is empty, specify the printer to print in the following order.
1. int to the printer specified in the input file (.lafx)
2. the printer specified in the file is not on the system, print to the system default printer.
• Please use the **OpenLabelW** method when using Unicode strings.

2) CloseLabel
Close the currently opened Label Artist-II document.

**[Syntax]**
long CloseLabel()

**[Parameters]**
None

**[Returns]**
If the method succeeds, the return value is zero(0). Other cases, returns non-zero.

3) SaveLabel

Save the currently opened Label Artist-II document.

**[Syntax]**

long SaveLabel()

**[Parameters]**

None

**[Returns]**

If the method succeeds, the return value is zero(0). Other cases, returns non-zero.

4) GetVariableCount

Get the number of variables.

**[Syntax]**

long GetVariableCount(int& count)

**[Parameters]**

int& count: The number of variables in the document

**[Returns]**

If the method succeeds, the return value is zero(0). Other cases, returns non-zero.

5) GetVariableName

Gets the variable name corresponding to the location.

**[Syntax]**

long GetVariableName(LPCSTR pName, int pos = BXL_IN_ORDER)

**[Parameters]**

LPCSTR pName: Variable name (IN)

int pos: Variable value (OUT).

If pos was set BXL_IN_ORDER (-1), the call gets the variable names sequentially.

**[Returns]**

If the method succeeds, the return value is zero(0). Other cases, returns non-zero.

Please use the **GetVariableNameW** method when using Unicode strings.

6) GetVariableValue

Gets the variable value by variable name.

**[Syntax]**

long GetVariableValue(LPCSTR pName, LPCSTR pValue)

**[Parameters]**

LPCSTR pName: Variable name (OUT)

LPCSTR pValue: Variable value (IN)

**[Returns]**

If the method succeeds, the return value is zero(0). Other cases, returns non-zero.

Please use the **GetVariableValueW** method when using Unicode strings.

7) SetVariableValue
Set the value of the variable.

**[Syntax]**
long SetVariableValue(LPCSTR pName, LPCSTR pValue)

**[Parameters]**
LPCSTR pName: Variable name (OUT)
LPCSTR pValue: Variable value (OUT)

**[Returns]**
If the method succeeds, the return value is zero(0). Other cases, returns non-zero.

Please use the **SetVariableValueW** method when using Unicode strings.

8) FindObject
Find object information by object name.

**[Syntax]**
long FindObject(LPCSTR objectName, Object* obj);

**[Parameters]**
LPCSTR objectName: Variable name (OUT)
Object* obj: Object information(OUT)

**[Returns]**
If the method succeeds, the return value is zero(0). Other cases, returns non-zero.

Please use the **FindObjectW** method when using Unicode strings.

9) SetObject

Set the object information.

**[Syntax]**

long SetObject ()

**[Parameters]**

None

**[Returns]**

If the method succeeds, the return value is zero(0). Other cases, returns non-zero.

10)PrintOut

Print opened Label Artist-II Document.

**[Syntax]**

long PrintOut()

**[Parameters]**

None

**[Returns]**

If the method succeeds, the return value is zero(0). Other cases, returns non-zero.

6-3-2 Label Artist-II SDK Error Code table

The error code table below summarizes the values returned when calling the API provided by Label Artist-II SDK.

| Error Code | Value | Description |
|---|---|---|
| BXL_SUCCESS | 0 | Method execution successful |
| BXL_ERR_SEARCH_FILE | -100 | File not found |
| BXL_ERR_SEARCH_PRINTER | -101 | Printer not found |
| BXL_ERR_SEARCH_OBJECT | -102 | Object not found |
| BXL_ERR_OPEN_FILE | -103 | File open failed |
| BXL_ERR_OPEN_PRINTER | -104 | Printer connection failed |
| BXL_ERR_SUPPORT_FILE | -105 | Unsupported file |
| BXL_ERR_SUPPORT_PRINTER | -106 | Unsupported printer |
| BXL_ERR_PARAMETER | -107 | Invalid input value (parameter) |
| BXL_ERR_ALREADY_OPEN | -108 | Already connected with printer |
| BXL_ERR_NOT_OPEN | -109 | No connection with the printer |
| BXL_ERR_VARIABLE_NAME | -110 | Invalid variable name used |
| BXL_ERR_PRINT_RESOURCE | -111 | Printer Resource Usage Failed |
| BXL_ERR_PRINT | -112 | Printing failed |

## 6-3-3 Example of using Label Artist-II API

Describes how to program C ++ using the Label Artist-II API.

1) Open Label Artist-II document
   Open the Label Artist-II document with the OpenLabel method. If you use a Unicode string, use the OpenLabelW method.

```
long ret = OpenLabelW(m_filePath, m_printerName);
if (ret == BXL_SUCCESS) AfxMessageBox(_T("Open Success"));
else AfxMessageBox(_T("Open Fail"));
```

2) Variable value setting
   Retrieve the variable name with the GetVariableName method and set the variable value with the SetVariableValue method.

```
long ret;
TCHAR variableName[100];
CString name;

wmemset(variableName, 0, 100);

ret = GetVariableNameW(variableName);

if (ret != BXL_SUCCESS) return;
name.Format(_T("%s"), variableName);

while (name.GetLength())
{
	if (!name.Compare(_T("model"))) {
		SetVariableValueW(name, _T("SDK Printer"));
	}
	if (!name.Compare(_T("sn"))) {
		SetVariableValueW(name, _T("NUMBER2020010001"));
	}
	wmemset(variableName, 0, 100);
	GetVariableNameW(variableName);
	name.Format(_T("%s"), variableName);
}
```

The string (LPCSTR / LPCWSTR) used in the parameter to get the variable name and variable value must be managed (allocated and freed) by the application.

3) Adjust Object Position

Get the object information with the FindObject method and set the object with the SetObject method.

```
Object obj;
ret = FindObjectW(_T("picture01"),&obj);
if ((ret == BXL_SUCCESS)) {
        obj.x += 3;          //3mm
        obj.y -= 2;
        obj.width -= 5;
        obj.height += 2;
        SetObject(&obj);
}
```

4) Print

Print a Label Artist-II document with the PrintOut method.

```
long ret = PrintOut();
if (ret == BXL_SUCCESS) AfxMessageBox(_T("Print Success"));
else AfxMessageBox(_T("Print Fail"));
```

# Revision history

| Rev. | Date | Description |
|---|---|---|
| 2.00 | 2018-03-12 | New |
| 2.01 | 2018-05-18 | Modified model name |
| 2.02 | 2019-01-14 | SPP-L310/L410 Added,<br>Collate copies option Added |
| 2.03 | 2019-03-04 | XD3-40d Added |
| 2.04 | 2019-04-05 | Device Font selection, Template,<br>Save JSON File. |
| 2.05 | 2019-06-24 | XD3-40t, XL5-40/43, XD5-40d/43d Added,<br>BIXOLON XML Enabled,<br>Speed, Darkness settings Added |
| 2.06 | 2019-12-03 | XD5-40t/43t, XM7-40 Added,<br>XL5-40CT/43CT modified |
| 2.07 | 2020-03-11 | XM7-20 Added,<br>Change picture properties,<br>Label Artist-II SDK Added |
| 2.08 | 2020-11-10 | SRP-S3000 Added,<br>Change supported OS,<br>Code page settings added,<br>Change object settings and add RFID,<br>Added a setting to not print objects |
| 2.09 | 2021.01.27 | Replaced captured image |
| 2.10 | 2021-08-24 | XM7-40 RFID function added<br>XT3-40/43 model added<br>BDP-T42/T43/D42/D43, BTP-42/43/46 model added |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |