BIXOLON

API Reference Guide

Windows POS SDK

Ver. 2.11

www.bixolon.com

Table of Contents

Copyright **4**

1. Operating System (OS) Environment **5**

2. Windows POS SDK Preparation **6**

3. Supported Printers **7**

4. Sample Program **9**

5. Printer Methods **10**

5-1 Printer Open 10
5-2 Printer Close 12
5-3 Line Feed 13
5-4 Set Character Set 14
5-5 Set Inter ChrSet 16
5-6 Print Text 17
5-7 Print Text W 19
5-8 Print Text In Image 22
5-9 Print Text In Image W 23
5-10 Print Bitmap 24
5-11 Print Barcode 25
5-12 Print QR Code 27
5-13 Print PDF417 29
5-14 Print Data Matrix 31
5-15 Print GS1 Data Bar 32
5-16 Print Composite Symbology 33
5-17 Get Printer Current Status 35
5-18 Initialize Printer 36
5-19 Direct IO 37
5-20 Cut Paper 38
5-21 Open Drawer 39
5-22 Write Buff 40
5-23 Read Buff 41
5-24 Bidi Set Call Back 42
5-25 Bidi Cancel Call Back 43
5-26 Transaction Start 44
5-27 Transaction End 45
5-28 Set Page mode 46
5-29 Set Page mode Print Area 47
5-30 Set Page mode Direction 48
5-31 Set Page mode Position 49

5-32 Get Statistics Data 49

6. KIOSK Presenter Methods 50
6-1 Paper Eject 50
6-2 Paper Eject Specify 51
6-3 Get Presenter Status 52
6-4 Paper Retract 53
6-5 Eject And Retract Paper 54

7. Customer Display Methods 55
7-1 Clear Screen 55
7-2 Clear Line 56
7-3 Display String 57
7-4 Display String W 58
7-5 Display String At Line 60
7-6 Display String At Line W 61
7-7 Store Image File 63
7-8 Display Image 64
7-9 Set International Character set 65

8. Error Code Table 66

Copyright

© BIXOLON Co., Ltd. All rights reserved.

This user manual and all property of the product are protected under copyright law.
It is strictly prohibited to copy, store, and transmit the whole or any part of the manual and any property of the product without the prior written approval of BIXOLON Co., Ltd.
The information contained herein is designed only for use with this BIXOLON product.
BIXOLON is not responsible for any direct or indirect damages, arising from or related to use of this information.

• The BIXOLON logo is the registered trademark of BIXOLON Co., Ltd.
• All other brand or product names are trademarks of their respective companies or organizations.

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 7 (32bit/64bit)
Microsoft Windows 8 (32bit/64bit)
Microsoft Windows 10 (32bit/64bit)
Microsoft Windows 11

2. Windows POS SDK Preparation

You can download the latest version of the Windows POS SDK from the BIXOLON homepage. (www.bixolon.com)

3. Supported Printers

The below table summarizes the supported printer models and specifications. The Windows POS SDK is available for the following BIXOLON printers.

Model	DPI	Max Printable Width
SRP-E300	180 dpi	512 dots
SRP-E302	203 dpi	576 dots
SRP-Q200	203 dpi	434 dots
SRP-Q300	180 dpi	512 dots
SRP-Q302	203 dpi	576 dots
SRP-Q300II	203 dpi	576 dots
SRP-QE300	180 dpi	512 dots
SRP-QE302	203 dpi	576 dots
SRP-S300	203 dpi	576 dots
SRP-380	180 dpi	512 dots
SRP-382	203 dpi	576 dots
SRP-383	300 dpi	864 dots
SRP-380plus	180 dpi	512 dots
SRP-382plus	203 dpi	576 dots
SRP-330II	180 dpi	512 dots
SRP-332II	203 dpi	576 dots
SRP-330III	180 dpi	512 dots
SRP-332III	203 dpi	576 dots
SRP-F310II	180 dpi	512 dots
SRP-F312II	203 dpi	576 dots
SRP-F313II	203 dpi	640 dots
SRP-350III	180 dpi	512 dots
SRP-352III	203 dpi	576 dots
SRP-350plusIII	180 dpi	512 dots
SRP-352plusIII	203 dpi	576 dots
SRP-350V	180 dpi	512 dots
SRP-352V	203 dpi	576 dots
SRP-350plusV	180 dpi	512 dots
SRP-352plusV	203 dpi	576 dots
SRP-340II	180 dpi	512 dots
SRP-342II	203 dpi	576 dots
SRP-275III	160 dpi	400 dots
SRP-B300	203 dpi	576 dots (42CPL: 546dots)
SRP-S320	203 dpi	576 dots
SRP-S3000	203 dpi	576 dots
SRP-S200	203 dpi	434 dots
BK3-3	203 dpi	576 dots
BK3-2	203 dpi	432 dots

BK5-3	203 dpi	576 dots
SPP-200	203 dpi	576 dots
SPP-300	203 dpi	576 dots
G30	203 dpi	576 dots
SRP-S300II	203 dpi	576 dots
SRP-G300	203 dpi	576 dots
BK5-4	203 dpi	768 dots
SRP-B300II	180 dpi	512 dots

4. Sample Program

Sample source files using the Windows POS SDK below are provided.

Visual C# (Visual Studio 2008)
Visual C++ (Visual Studio 2008)
Visual VB.NET (Visual Studio 2008)

5. Printer Methods

This manual is based on Visual C ++ development environment and describes the various APIs needed to develop Windows applications that can control the printer. The methods and constant values disclosed in the provided are declared in the BXLPApi.h and bxlconst.h file.

5-1 Printer Open

• Establishes a communication connection with the printer.

- This method must be called first, than any other methods in this manual.
- Only 1:1 communication with PC and a printer is supported.

[Syntax]

long **PrinterOpen**(
  int nInterface,
  LPCSTR szPortName,
  nBaudRate,
  nDataBits,
  nParity,
  nStopBits,
  nFlowControl
);

[Parameters]

<table>
<tr><td>nInterface</td><td>The printer communication interface.
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>ISerial</td><td>0</td><td>Serial Printer</td></tr>
<tr><td>IParallel</td><td>1</td><td>Parallel Printer</td></tr>
<tr><td>IUsb</td><td>2</td><td>USB Printer</td></tr>
<tr><td>ILan</td><td>3</td><td>Ethernet Printer</td></tr>
<tr><td>IWLan</td><td>4</td><td>Wi-Fi Printer</td></tr>
<tr><td>IBluetooth</td><td>5</td><td>Bluetooth Printer</td></tr>
</table></td></tr>
<tr><td>szPort Name</td><td>Port name for Serial, Bluetooth and Parallel, or IP address for network.
* This parameter is only valid for Serial, parallel or network communication.</td></tr>
<tr><td>nBaud Rate</td><td>The baud rate for Serial communication, or port number for network.
* This parameter is only valid for Serial, parallel or network communication.</td></tr>
<tr><td>nData Bits</td><td>The number of bits in the bytes transmitted and received
* This parameter is only valid for Serial communication.</td></tr>
<tr><td>nParity</td><td>The parity scheme to be used for Serial communication.
* This parameter is only valid for Serial communication.</td></tr>
<tr><td>nStop Bits</td><td>The number of stop bits to be used for Serial communication.
* This parameter is only valid for Serial communication.</td></tr>
<tr><td>nFlow Control</td><td>The flow control to be used for Serial communication.
* This parameter is only valid for Serial communication.</td></tr>
</table>

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
long lResult = PrinterOpen(0, “COM1”, 9600, 8, 0, 0, 0); // Serial
//long lResult = PrinterOpen(1, “LPT1”, 0, 0, 0, 0, 0); // Parallel
//long lResult = PrinterOpen(2, “”, 0, 0, 0, 0, 0); // USB
//long lResult = PrinterOpen(3,”192.168.1.1”, 9100, 0, 0, 0, 0); // Network (Ethernet, Wi-Fi)
```

5-2 Printer Close

• Disconnects the communication with the printer.



Call this method if you no longer need to communicate with the printer.

[Syntax]
long **PrinterClose**();

[Parameters]
None

[Return Values]
If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
long lResult = PrinterOpen("portinfo...", 1000);
......
PrinterClose();
......
```

5-3 Line Feed

• Sends line-feeds commands to the printer.

[Syntax]

long **LineFeed** (
  int Feed,
);

[Parameters]

Feed	Paper feed amount in lines. (Range: 1 ≤ Feed ≤ 255)

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
long lResult = PrinterOpen("portinfo...", 1000);
.....
lResult = LineFeed(10);
.....
```

5-4 Set Character Set

• Sets the code table used for printing characters.

[Syntax]

long **SetCharacterSet** (
  long Codepage
);

[Parameters]

Code page

Code page used for encoding characters.

Code	Value	Description
BXL_CS_PC437	0	USA, Standard Europe
BXL_CS_PC850	2	Multilingual Latin 1
BXL_CS_PC860	3	Portuguese
BXL_CS_PC863	4	Canadian-French
BXL_CS_PC865	5	Nordic
BXL_CS_WPC1252	16	Latin 1
BXL_CS_PC866	17	Cyrillic #2
BXL_CS_PC852	18	Latin 2
BXL_CS_PC858	19	Euro
BXL_CS_PC862	21	Hebrew DOS code
BXL_CS_THAI42	23	THAI 42
BXL_CS_WPC1253	24	Greek
BXL_CS_WPC1254	25	Turkish
BXL_CS_WPC1257	26	Baltic
BXL_CS_WPC1251	28	Cyrillic
BXL_CS_PC737	29	Greek
BXL_CS_PC775	30	Baltic
BXL_CS_THAI14	31	THAI 14
BXL_CS_WPC1255	33	Hebrew New Code
BXL_CS_THAI11	34	THAI 11
BXL_CS_THAI18	35	THAI 18
BXL_CS_PC855	36	Cyrillic
BXL_CS_PC857	37	Turkish
BXL_CS_PC928	38	Greek
BXL_CS_THAI16	39	THAI 16
BXL_CS_PC1258	41	Vietnamese (PC1258)
BXL_CS_PC1250	47	Czech
BXL_CS_LATIN9	48	Latin 9
BXL_CS_TCVN	49	Vietnamese (TCVN3)
BXL_CS_TCVN_CAPITAL	50	Vietnamese (TCVN3 CAPITAL)
BXL_CS_VISCII	51	Vietnamese (VISCII)

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
long lResult = PrinterOpen("portinfo…", 1000);
……
SetCharacterSet(BXL_CS_WPC1252);
……
```

5-5 Set Inter ChrSet

• Sets the character set used for printing some of international characters.

[Syntax]

long **SetInterChrSet** (
  long InternationalCharacterSet
);

[Parameters]

International Character Set	International character set.

Code	Value	Description
BXL_ICS_USA	0	USA code setting
BXL_ICS_FRANCE	1	FRANCE code setting
BXL_ICS_GERMANY	2	GERMANY code setting
BXL_ICS_UK	3	UK code setting
BXL_ICS_DENMARK1	4	DENMARK1 code setting
BXL_ICS_SWEDEN	5	SWEDEN code setting
BXL_ICS_ITALY	6	ITALY code setting
BXL_ICS_SPAIN	7	SPAIN code setting
BXL_ICS_JAPAN	8	JAPAN code setting
BXL_ICS_NORWAY	9	NORWAY code setting
BXL_ICS_DENMARK2	10	DENMARK 2 code setting
BXL_ICS_SPAIN2	11	SPAIN 2 code setting
BXL_ICS_LATIN	12	LATIN AMERICA code setting
BXL_ICS_KOREA	13	KOREA code setting
BXL_ICS_SLOVENIA	14	SLOVENIA code setting
BXL_ICS_CHINA	15	CHINA code setting

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
long lResult = PrinterOpen("portinfo...", 1000);
.....
SetInterChrSet(BXL_ICS_USA);
.....
```

5-6 Print Text

• Sends a character print command to the printer.

[Syntax]

long **PrintText**(
  LPCSTR Data,
  long Alignment,
  long Attribute,
  long TextSize
);

[Parameters]

<table>
<tr><td>Data</td><td>Text data to be printed.</td></tr>
<tr><td>Alignment</td><td>Alignment
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>BXL_ALIGNMENT_LEFT</td><td>0</td><td>Left-aligned</td></tr>
<tr><td>BXL_ALIGNMENT_CENTER</td><td>1</td><td>Centered</td></tr>
<tr><td>BXL_ALIGNMENT_RIGHT</td><td>2</td><td>Right-aligned</td></tr>
</table></td></tr>
<tr><td>Attribute</td><td>Text properties
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>BXL_FT_DEFAULT</td><td>0</td><td>Font A (default font)</td></tr>
<tr><td>BXL_FT_FONTB</td><td>1</td><td>Set to Font B</td></tr>
<tr><td>BXL_FT_BOLD</td><td>2</td><td>Add Bold lettering attribute</td></tr>
<tr><td>BXL_FT_UNDERLINE</td><td>4</td><td>Add Underline (1-dot thick)</td></tr>
<tr><td>BXL_FT_UNDERTHICK</td><td>6</td><td>Add Underline (2-dot thick)</td></tr>
<tr><td>BXL_FT_REVERSE</td><td>8</td><td>Add Reverse lettering attribute
* SRP-275III is not supported.</td></tr>
<tr><td>BXL_FT_FONTC</td><td>16</td><td>Set to Font C
* SRP-275III is not supported.</td></tr>
<tr><td>BXL_FT_RED_COLOR</td><td>64</td><td>Print in Red Color
* SRP-275III only supported.</td></tr>
</table></td></tr>
<tr><td>Text Size</td><td>Text size
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>BXL_TS_0WIDTH</td><td>0</td><td>Width magnification set to x1</td></tr>
<tr><td>BXL_TS_1WIDTH</td><td>16</td><td>Width magnification set to x2</td></tr>
<tr><td>BXL_TS_2WIDTH</td><td>32</td><td>Width magnification set to x3</td></tr>
<tr><td>BXL_TS_3WIDTH</td><td>48</td><td>Width magnification set to x4</td></tr>
<tr><td>BXL_TS_4WIDTH</td><td>64</td><td>Width magnification set to x5</td></tr>
<tr><td>BXL_TS_5WIDTH</td><td>80</td><td>Width magnification set to x6</td></tr>
<tr><td>BXL_TS_6WIDTH</td><td>96</td><td>Width magnification set to x7</td></tr>
<tr><td>BXL_TS_7WIDTH</td><td>112</td><td>Width magnification set to x8</td></tr>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>BXL_TS_0HEIGHT</td><td>0</td><td>Height magnification set to x1</td></tr>
<tr><td>BXL_TS_1HEIGHT</td><td>1</td><td>Height magnification set to x2</td></tr>
<tr><td>BXL_TS_2HEIGHT</td><td>2</td><td>Height magnification set to x3</td></tr>
<tr><td>BXL_TS_3HEIGHT</td><td>3</td><td>Height magnification set to x4</td></tr>
</table></td></tr>
</table>

		BXL_TS_4HEIGHT	4	Height magnification set to x5
		BXL_TS_5HEIGHT	5	Height magnification set to x6
		BXL_TS_6HEIGHT	6	Height magnification set to x7
		BXL_TS_7HEIGHT	7	Height magnification set to x8

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
long lResult = PrinterOpen(“portinfo…”, 1000);
……
PrintText("Bixolon Printer.\n", BXL_ALIGNMENT_LEFT, BXL_FT_DEFAULT,
BXL_TS_0WIDTH | BXL_TS_0HEIGHT);
……
```

5-7 Print Text W

• Sends a character print command to the printer.

[Syntax]

long **PrintTextW**(
LPCWSTR Data,
long Alignment,
long Attribute,
long TextSize,
long CodePage
);

[Parameters]

Data	Text data in UNICODE format to be printed.
Alignment	Alignment

Code	Value	Description
BXL_ALIGNMENT_LEFT	0	Left-aligned
BXL_ALIGNMENT_CENTER	1	Centered
BXL_ALIGNMENT_RIGHT	2	Right-aligned

Attribute — Text properties

Code	Value	Description
BXL_FT_DEFAULT	0	Font A (default font)
BXL_FT_FONTB	1	Set to Font B
BXL_FT_BOLD	2	Add Bold lettering attribute
BXL_FT_UNDERLINE	4	Add Underline (1-dot thick)
BXL_FT_UNDERTHICK	6	Add Underline (2-dot thick)
BXL_FT_REVERSE	8	Add Reverse lettering attribute * SRP-275III is not supported.
BXL_FT_FONTC	16	Set to Font C * SRP-275III is not supported.
BXL_FT_RED_COLOR	64	Print in Red Color * SRP-275III only supported.

Text Size — Text size

Code	Value	Description
BXL_TS_0WIDTH	0	Width magnification set to x1
BXL_TS_1WIDTH	16	Width magnification set to x2
BXL_TS_2WIDTH	32	Width magnification set to x3
BXL_TS_3WIDTH	48	Width magnification set to x4
BXL_TS_4WIDTH	64	Width magnification set to x5
BXL_TS_5WIDTH	80	Width magnification set to x6
BXL_TS_6WIDTH	96	Width magnification set to x7
BXL_TS_7WIDTH	112	Width magnification set to x8
Code	Value	Description
BXL_TS_0HEIGHT	0	Height magnification set to x1
BXL_TS_1HEIGHT	1	Height magnification set to x2
BXL_TS_2HEIGHT	2	Height magnification set to x3

	BXL_TS_3HEIGHT	3	Height magnification set to x4
	BXL_TS_4HEIGHT	4	Height magnification set to x5
	BXL_TS_5HEIGHT	5	Height magnification set to x6
	BXL_TS_6HEIGHT	6	Height magnification set to x7
	BXL_TS_7HEIGHT	7	Height magnification set to x8

Code Page

Code page for encoding text data.

Code	Value	Description
BXL_CS_PC437	0	USA, Standard Europe
BXL_CS_PC850	2	Multilingual Latin 1
BXL_CS_PC860	3	Portuguese
BXL_CS_PC863	4	Canadian-French
BXL_CS_PC865	5	Nordic
BXL_CS_WPC1252	16	Latin 1
BXL_CS_PC866	17	Cyrillic #2
BXL_CS_PC852	18	Latin 2
BXL_CS_PC858	19	Euro
BXL_CS_PC862	21	Hebrew DOS code
BXL_CS_THAI42	23	THAI 42
BXL_CS_WPC1253	24	Greek
BXL_CS_WPC1254	25	Turkish
BXL_CS_WPC1257	26	Baltic
BXL_CS_WPC1251	28	Cyrillic
BXL_CS_PC737	29	Greek
BXL_CS_PC775	30	Baltic
BXL_CS_THAI14	31	THAI 14
BXL_CS_WPC1255	33	Hebrew New Code
BXL_CS_THAI11	34	THAI 11
BXL_CS_THAI18	35	THAI 18
BXL_CS_PC855	36	Cyrillic
BXL_CS_PC857	37	Turkish
BXL_CS_PC928	38	Greek
BXL_CS_THAI16	39	THAI 16
BXL_CS_PC1258	41	Vietnamese (PC1258)
BXL_CS_PC1250	47	Czech
BXL_CS_LATIN9	48	Latin 9
BXL_CS_TCVN	49	Vietnamese (TCVN3)
BXL_CS_TCVN_CAPITAL	50	Vietnamese (TCVN3 CAPITAL)
BXL_CS_VISCII	51	Vietnamese (VISCII)
BXL_CS_KS5601	949	Korean
BXL_CS_BIG5	950	Chinese (BIG5)
BXL_CS_GB2312	936	Chinese (GB2312)
BXL_CS_SHIFT_JIS	932	Japanese (Shift JIS)

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
long lResult = PrinterOpen(“portinfo…”, 1000);
……
PrintTextW(L"Bixolon Printer.\n", BXL_ALIGNMENT_LEFT, BXL_FT_DEFAULT,
BXL_TS_0WIDTH | BXL_TS_0HEIGHT, BXL_CS_PC437);
……
```

5-8 Print Text In Image

• Using a true -type font, sends a character print command to the printer.

[Syntax]

long **PrintTextInImage** (
  LPCSTR Data,
  LPCSTR FontName,
  bool Italic,
  bool Bold,
  bool Underline,
  int FontSize,
  int Aligment,
);

[Parameters]

Data	Text data to be printed.
Font Name	TTF (true type font) name.
Italic	Italic or not
Bold	Bold or not
Underline	Underline or not
Font Size	Font Size (point)
Alignment	Alignment

Code	Value	Description
BXL_ALIGNMENT_LEFT	0	Left-aligned
BXL_ALIGNMENT_CENTER	1	Centered
BXL_ALIGNMENT_RIGHT	2	Right-aligned

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
long lResult = PrinterOpen("portinfo...", 1000);
.....
PrintTextInImage("ABCD", "Arial", FALSE, FALSE, FALSE, 15, BXL_ALIGNMENT_LEFT);
.....
```

5-9 Print Text In Image W

• Using a true -type font, sends a character print command to the printer.

[Syntax]

long **PrintTextInImage** (
LPCWSTR Data,
LPCWSTR FontName,
bool Italic,
bool Bold,
bool Underline,
int FontSize,
int Aligment,
);

[Parameters]

Data	Text data in UNICODE format to be printed.
Font Name	TTF (true type font) name in UNICOD format.
Italic	Italic or not
Bold	Bold or not
Underline	Underline or not
Font Size	Font Size (point)
Alignment	Alignment

Code	Value	Description
BXL_ALIGNMENT_LEFT	0	Left-aligned
BXL_ALIGNMENT_CENTER	1	Centered
BXL_ALIGNMENT_RIGHT	2	Right-aligned

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
long lResult = PrinterOpen("portinfo...", 1000);
.....
PrintTextInImageW(L"ABCD", L "Arial", FALSE, FALSE, FALSE, 15, BXL_ALIGNMENT_LEFT);
.....
```

5-10 Print Bitmap

• Sends a graphic image command to the printer.



Only BMP file format can be printed.

[Syntax]

long **PrintBitmap** (
  LPCSTR FileName,
  long Width,
  long Alignment,
  long Level,
  bool Dithering
);

[Parameters]

<table>
<tr><td>File Name</td><td>The full path of the image file.</td></tr>
<tr><td>Width</td><td>The width of the image to be printed.
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>BXL_WIDTH_FULL</td><td>-1</td><td>Maximum width the printer can print.</td></tr>
<tr><td>BXL_WIDTH_NONE</td><td>-2</td><td>No variation given to image size</td></tr>
</table></td></tr>
<tr><td>Alignment</td><td>Alignment
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>BXL_ALIGNMENT_LEFT</td><td>0</td><td>Left-aligned</td></tr>
<tr><td>BXL_ALIGNMENT_CENTER</td><td>1</td><td>Centered</td></tr>
<tr><td>BXL_ALIGNMENT_RIGHT</td><td>2</td><td>Right-aligned</td></tr>
</table></td></tr>
<tr><td>Level</td><td>The brightness level. (Range: 0 ≤ Level ≤ 100)</td></tr>
<tr><td>Dithering</td><td>Whether apply the dither option or not.</td></tr>
</table>

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
long lResult = PrinterOpen("portinfo...", 1000);
.....
PrintBitmap(strDirBmp, 300, BXL_ALIGNMENT_CENTER, 50, TRUE);
.....
```

5-11 Print Barcode

• Sends a 1d barcode print command to the printer.



This method is not supported on SRP-275III.

[Syntax]

long **PrintBarcode**(
PCHAR Data,
long Symbology,
long Height,
long Width,
long Alignment,
long TextPosition
);

[Parameters]

<table>
<tr><td>Data</td><td>Barcode data in ASCII format.</td></tr>
<tr><td>Symbology</td><td>Code Symbology.
<table>
<tr><th>Symbology</th><th>Value</th><th>Length Limit</th><th>Value Range</th></tr>
<tr><td>BXL_BCS_UPCA</td><td>101</td><td>11 ≤ n ≤12</td><td>48 ≤ data ≤ 57</td></tr>
<tr><td>BXL_BCS_UPCE</td><td>102</td><td>11 ≤ n ≤12</td><td>48 ≤ data ≤ 57</td></tr>
<tr><td>BXL_BCS_EAN13
BXL_BCS_JAN13</td><td>104
106</td><td>12 ≤ n ≤13</td><td>48 ≤ data ≤ 57</td></tr>
<tr><td>BXL_BCS_EAN8
BXL_BCS_JAN8</td><td>103
105</td><td>7 ≤ n ≤ 8</td><td>48 ≤ data ≤ 57</td></tr>
<tr><td>BX _BCS_ITF</td><td>107</td><td>1 ≤ n ≤ 255 (even)</td><td>48 ≤ data ≤ 57</td></tr>
<tr><td>BXL_BCS_CODABAR</td><td>108</td><td>1 ≤ n ≤ 255</td><td>48 ≤ data ≤ 57, 65 ≤ data ≤ 68, data = 36,43,45,46,47,58</td></tr>
<tr><td>BXL_BCS_CODE39</td><td>109</td><td>1 ≤ n ≤ 255</td><td>48 ≤ data ≤ 57, 65 ≤ data ≤ 90, data = 32,36,37, 43, 45,46,47</td></tr>
<tr><td>BXL_BCS_CODE93</td><td>110</td><td>1 ≤ n ≤ 255</td><td>0 ≤ data ≤ 127</td></tr>
<tr><td>BXL_BCS_CODE128</td><td>111</td><td>2 ≤ n ≤ 255</td><td>0 ≤ data ≤ 127</td></tr>
<tr><td>BXL_BCS_1D_GS1_128</td><td>112</td><td>2 ≤ n ≤ 255</td><td>0 ≤ data ≤ 127</td></tr>
<tr><td>BXL_BCS_1D_GS1DATABAR _OMNIDIRECTION</td><td>113</td><td>n = 13</td><td>48 ≤ data ≤ 57</td></tr>
<tr><td>BXL_BCS_1D_GS1DATABAR _TRUNCATED</td><td>114</td><td>n = 13</td><td>48 ≤ data ≤ 57</td></tr>
<tr><td>BXL_BCS_1D_GS1DATABAR _LIMITED</td><td>115</td><td>n = 13</td><td>48 ≤ data ≤ 57</td></tr>
</table></td></tr>
<tr><td>Height</td><td>The barcode height. (Range: 1 ≤ Height ≤ 255)</td></tr>
</table>

<table>
<tr><td>Width</td><td>The width or size of a 1d barcode. (Range: 2 ≤ Width ≤ 6)
* If the print width of the barcode exceeds the printing paper, barcodes may not be printed properly.</td></tr>
<tr><td>Alignment</td><td>Alignment
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>BXL_ALIGNMENT_LEFT</td><td>0</td><td>Left-aligned</td></tr>
<tr><td>BXL_ALIGNMENT_CENTER</td><td>1</td><td>Centered</td></tr>
<tr><td>BXL_ALIGNMENT_RIGHT</td><td>2</td><td>Right-aligned</td></tr>
</table></td></tr>
<tr><td>Text Position</td><td>The HRI (Human Readable Interpretation) printing position of the 1d barcode.
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>BXL_BC_TEXT_NONE</td><td>0</td><td>Not Printed</td></tr>
<tr><td>BXL_BC_TEXT_ABOVE</td><td>1</td><td>Above the barcode</td></tr>
<tr><td>BXL_BC_TEXT_BELOW</td><td>2</td><td>Below the barcode</td></tr>
<tr><td>BXL_BC_TEXT_BOTH</td><td>3</td><td>Above and Below the barcode</td></tr>
</table></td></tr>
</table>

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
long lResult = PrinterOpen("portinfo...", 1000);
.....
PrintBarcode("123456789012", BXL_BCS_CODE128, 50, 2, BXL_ALIGNMENT_LEFT,
BXL_BC_TEXT_BELOW);
.....
```

5-12 Print QR Code

• Sends a QR Code print command to the printer.



This method is not supported on SRP-275III.

[Syntax]

long **PrintQRCode**(
PCHAR Data,
long Model,
long Size,
long EccLevel,
long Alignment
);

[Parameters]

Data	Code data in ASCII format.
Model	Model type of QR Code

Code	Value	Description
BXL_QRCODE_MODEL_1	49	MODEL 1
BXL_QRCODE_MODEL_2	50	MODEL 2

Size	Module size (Range: 1 ≤ Size ≤ 8)
Ecc Level	Error Correction Level

Code	Value	Description
BXL_QRCODE_ECC_LEVEL_L	48	LEVEL L
BXL_QRCODE_ECC_LEVEL_M	49	LEVEL M
BXL_QRCODE_ECC_LEVEL_Q	50	LEVEL Q
BXL_QRCODE_ECC_LEVEL_H	51	LEVEL H

Alignment	Alignment

Code	Value	Description
BXL_ALIGNMENT_LEFT	0	Left-aligned
BXL_ALIGNMENT_CENTER	1	Centered
BXL_ALIGNMENT_RIGHT	2	Right-aligned

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
long lResult = PrinterOpen(“portinfo…”, 1000);
……
PrintQRCode(“123456789012”, BXL_QRCODE_MODEL_1, 4, _QRCODE_LEVEL_L,
BXL_ALIGNMENT_LEFT);
……
```

5-13 Print PDF417

• Sends a PDF417 print command to the printer.



This method is not supported on SRP-275III.

[Syntax]

long **PrintPDF417**(
PCHAR Data,
long Type,
long ColumnNumber,
long RowNumber,
long Width,
long Height,
long EccLevel,
long Alignment
);

[Parameters]

Data	Code data in ASCII format.
Type	PDF417 Type

Code	Value	Description
BXL_PDF417_TYPE_1	49	PDF417 Standard
BXL_PDF417_TYPE_2	50	PDF417 Simplified

Column Number	Number of columns (Range: 0 ≤ Column Number ≤ 30)
Row Number	Number of rows (Range: 3 ≤ Row Number ≤ 90)
Module Width	Module width (Range: 1 ≤ Width ≤ 4)
Module Height	Module height (Range: 2 ≤ Height ≤ 8)
Ecc Level	Error Correction Level

Code	Value	Description
BXL_PDF417_ECC_LEVEL_0	48	Error Correction Level 0
BXL_PDF417_ECC_LEVEL_1	49	Error Correction Level 1
BXL_PDF417_ECC_LEVEL_2	50	Error Correction Level 2
BXL_PDF417_ECC_LEVEL_3	51	Error Correction Level 3
BXL_PDF417_ECC_LEVEL_4	52	Error Correction Level 4
BXL_PDF417_ECC_LEVEL_5	53	Error Correction Level 5
BXL_PDF417_ECC_LEVEL_6	54	Error Correction Level 6
BXL_PDF417_ECC_LEVEL_7	55	Error Correction Level 7
BXL_PDF417_ECC_LEVEL_8	56	Error Correction Level 8

Alignment	Alignment

Code	Value	Description
BXL_ALIGNMENT_LEFT	0	Left-aligned
BXL_ALIGNMENT_CENTER	1	Centered
BXL_ALIGNMENT_RIGHT	2	Right-aligned

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
long lResult = PrinterOpen("portinfo…", 1000);
……
PrintPDF417("123456789012", BXL_PDF417_TYPE_1, 0, 0, 2, 2 BXL_PDF417_LEVEL_L,
BXL_ALIGNMENT_LEFT);
……
```

5-14 Print Data Matrix

• Sends a Data Matrix print command to the printer.



This method is not supported on SRP-275III and SRP-330II/332II.

[Syntax]

long **PrintDataMatrix**(
  PCHAR Data,
  long Size,
  long Alignment
);

[Parameters]

Data	Code data in ASCII format.
Size	Module size (Range: 2 ≤ Size ≤ 3)
Alignment	Alignment

Code	Value	Description
BXL_ALIGNMENT_LEFT	0	Left-aligned
BXL_ALIGNMENT_CENTER	1	Centered
BXL_ALIGNMENT_RIGHT	2	Right-aligned

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
long lResult = PrinterOpen("portinfo...", 1000);
.....
PrintDataMatrix("123456789012", 2, BXL_ALIGNMENT_LEFT);
.....
```

5-15 Print GS1 Data Bar

• Sends a GS1 Data Bar print command to the printer.

> This method is not supported on SRP-275III, SRP-330II/332II, SRP-350III/352III and SRP-340II/342III.

[Syntax]

long **PrintGS1DataBar**(
  PCHAR Data,
  long Symbology,
  long Size,
  long Alignment
);

[Parameters]

Data	Code data in ASCII format.
Symbology	GS1 Data Bar Symbol

Code	Value	Description
BXL_GS1DATABAR_STACKED	0	Stacked
BXL_GS1DATABAR_STACKED_OMNIDIRECTIONAL	1	Stacked Omnidirectional

Size	Module size (Range: 1 ≤ Size ≤ 8)
Alignment	Alignment

Code	Value	Description
BXL_ALIGNMENT_LEFT	0	Left-aligned
BXL_ALIGNMENT_CENTER	1	Centered
BXL_ALIGNMENT_RIGHT	2	Right-aligned

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
long lResult = PrinterOpen("portinfo...", 1000);
.....
PrintGS1DataBar("123456789012", BXL_GS1DATABAR_STACKED, 2, BXL_ALIGNMENT_LEFT);
.....
```

5-16 Print Composite Symbology

• Sends a Composite Symbology print command to the printer.



This method is not supported on SRP-275III, SRP-330II/332II, SRP-350III/352III and SRP-340II/342III.

[Syntax]

long **PrintCompositeSymbology**(
PCHAR Data1d,
long Symbol1d,
PCHAR Data2d,
long Symbol2d,
long Size,
long Alignment
);

[Parameters]

<table>
<tr><td>Data1d</td><td>1-dimensional Code data in ASCII format.</td></tr>
<tr><td>Symbol1d</td><td>Type of straight-line element
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>BXL_COMPOSITE_LINE_EAN8</td><td>65</td><td>EAN8</td></tr>
<tr><td>BXL_COMPOSITE_LINE_EAN13</td><td>66</td><td>EAN13</td></tr>
<tr><td>BXL_COMPOSITE_LINE_UPCA</td><td>67</td><td>UPC-A</td></tr>
<tr><td>BXL_COMPOSITE_LINE_UPCE</td><td>69</td><td>UPC-E</td></tr>
<tr><td>BXL_COMPOSITE_LINE_GS1DATABAR_OMNIDIRECTIONAL</td><td>70</td><td>GS1 Data Bar Omnidirectional</td></tr>
<tr><td>BXL_COMPOSITE_LINE_GS1DATABAR_TRUNCATED</td><td>71</td><td>GS1 Data Bar Truncated</td></tr>
<tr><td>BXL_COMPOSITE_LINE_GS1DATABAR_STACKED</td><td>72</td><td>GS1 Data Bar Stacked</td></tr>
<tr><td>BXL_COMPOSITE_LINE_GS1DATABAR_STACKED_OMNIDIRECTIONAL</td><td>73</td><td>GS1 Data Bar Stacked Omnidirectional</td></tr>
<tr><td>BXL_COMPOSITE_LINE_GS1DATABAR_LIMITED</td><td>74</td><td>GS1 DataBar Limited</td></tr>
<tr><td>BXL_COMPOSITE_LINE_GS1DATABAR_EXPANDED</td><td>75</td><td>GS1 Data Bar Expanded</td></tr>
<tr><td>BXL_COMPOSITE_LINE_GS1_128</td><td>77</td><td>GS1-128</td></tr>
</table></td></tr>
<tr><td>Data2d</td><td>2-dimensional Code data in ASCII format.</td></tr>
<tr><td>Symbol2d</td><td>2-dimensional synthetic element
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>BXL_COMPOSITE_2D_AUTO</td><td>65</td><td>CC-A, CC-B or CC-C (AUTO)</td></tr>
<tr><td>BXL_COMPOSITE_2D_CC_C</td><td>66</td><td>CC-C (Only GS1-128)</td></tr>
</table></td></tr>
<tr><td>Size</td><td>Module size (Range: 1 ≤ Size ≤ 8)</td></tr>
</table>

Alignment	Alignment

Code	Value	Description
BXL_ALIGNMENT_LEFT	0	Left-aligned
BXL_ALIGNMENT_CENTER	1	Centered
BXL_ALIGNMENT_RIGHT	2	Right-aligned

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
long lResult = PrinterOpen("portinfo…", 1000);
……
PrintCompositeSymbology("123456789012", BXL_COMPOSITE_LINE_EAN8, "ABCDEFGHIJ",
BXL_COMPOSITE_2D_AUTO, BXL_ALIGNMENT_LEFT);
……
```

5-17 Get Printer Current Status

• Gets the printer's current status value.

[Syntax]

long **GetPrinterCurrentStatus** ();

[Parameters]

None

[Return Values]

If the method succeeds, the return value is one of the following. Other cases, returns negative values.

Printer Status	Value	Description
BXL_STS_NORMAL	0	Normal. (No errors)
BXL_STS_PAPEREMPTY	1	Paper Empty
BXL_STS_COVEROPEN	2	Printer cover is open
BXL_STS_PAPER_NEAR_END	4	Paper is near-end
BXL_STS_CASHDRAWER_HIGH	16	Cash drawer Pin is in high state.
BXL_STS_ERROR	32	Offline (Printer is disconnected with HOST)
BXL_STS_NOT_OPEN	64	Not ready to start communication with the printer.
BXL_STS_BATTERY_LOW	128	Battery runs low
BXL_STS_PAPER_TO_BE_TAKEN	256	Printed receipt(label) presence is detected.
BXL_STS_CASHDRAWER_LOW	512	Cash drawer Pin is in low state.
BXL_STS_STACKER_FULL	16384	Stacker is full of paper
BXL_STS_STACKER_ERROR	32768	Error occurred in stacker

* Example

```
long lResult = PrinterOpen("portinfo...", 1000);
......
int status = GetPrinterCurrentStatatus();
......
if ((status & BXL_STS_PAPEREMPTY) == BXL_STS_PAPEREMPTY) { ... }
if ((status & BXL_STS_COVEROPEN) == BXL_STS_COVEROPEN) { ... }
if ((status & BXL_STS_CASHDRAWER_HIGH) == BXL_STS_CASHDRAWER_HIGH) { ... }
if ((status & BXL_STS_BATTERY_LOW) == BXL_STS_BATTERY_LOW) { ... }
if ((status & BXL_STS_PAPER_TO_BE_TAKEN) == BXL_STS_PAPER_TO_BE_TAKEN) { ... }
.......
```

5-18 Initialize Printer

• Reset all printer settings to the state they were in when the printer was powered on.

[Syntax]

long **InitializePrinter**();

[Parameters]

None

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
long lResult = PrinterOpen(3, “192.168.1.1”, 9100);
……
InitializePrinter();
……
```

5-19 Direct IO

• Sends user-defined data and receive a response from the printer.

This method is useful to send custom data to the printer.
For the information about the commands recognized by the printer, refer to the BXL/POS Programming manual. The manual can be downloaded from BIXOLON website.

[Syntax]

long **DirectIO** (
  PCHAR Data,
  UINT DataSize,
  PCHAR RecvBuffer,
  UINT& RecvBufferSize
);

[Parameters]

Data	The data to be sent.
Data Size	The size of buffer specified by the parameter 'Data'.
Recv Buffer	The buffer, which receives a response from the printer. *** set this parameter as NULL if no need to receive a response.**
Recv Buffer Size	The size of the buffer specified by the parameter 'Recv Buffer'. *** This value also used as the size of a response.**

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
long lResult = PrinterOpen("portinfo…", 1000);
……
CHAR cmd[3] = {0x10, 0x04, 0x02};
CHAR * readData;
UINT readLen = 0;
……
lResult = DirectIO(cmd, sizeof(cmd), readData, readLen);
……
```

5-20 Cut Paper

• Sends a paper cut commands with line feeds to the printer.

[Syntax]
long **CutPaper** ( );

[Parameters]
None

[Return Values]
If the method succeeds, the return value is 0. Other cases, returns negative values.



This method works with partial or full cuts, depending on the printer settings.

5-21 Open Drawer

• Sends a cash drawer open command to the printer.

[Syntax]

long **OpenDrawer** (int PinNumber);

Depending on the cash drawer type, the PIN number for opening the cash drawer is different.

[Parameters]

Pin Number	PIN Number.

Code	Value	Description
BXL_CASHDRAWER_PIN2	0	PIN Number 2
BXL_CASHDRAWER_PIN5	1	PIN Number 5

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
.....
OpenDrawer(BXL_CASHDRAWER_PIN2);
OpenDrawer(BXL_CASHDRAWER_PIN5);
.....
```

5-22 Write Buff

• Sends user-defined data to the printer.

[Syntax]

long **WriteBuff** (
  PBYTE Data,
  long DataSize,
  long& SizeSent,
);

[Parameters]

Data	The data to be sent to the printer.
Data Size	The size of buffer specified by the parameter 'Data'
Size Sent	A pointer to a value that receives the number of bytes of data that were sent to the printer.

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

5-23 Read Buff

• Reads data from the printer.

[Syntax]

long **ReadBuff** (
  PBYTE ReadBuffer,
  long NumberOfBytesToRead,
  long& DataSizeRead
);

[Parameters]

Read Buffer	A pointer to the buffer that receives the data read from the printer.
Number Of Bytes to Read	Maximum data length to read from the printer.
Data Size Read	A pointer to a value that receives the number of data read from the printer.

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

5-24 Bidi Set Call Back

• Sets the callback function which receives a status value of the printer. The callback function is called whenever the status value of the printer changes.
The printer status value is one of the followings.

Printer Status	Value	Description
BXL_STS_NORMAL	0	Normal. (No errors)
BXL_STS_PAPEREMPTY	1	Paper Empty
BXL_STS_COVEROPEN	2	Printer cover is open
BXL_STS_PAPER_NEAR_END	4	Paper is near-end
BXL_STS_CASHDRAWER_HIGH	16	Cash drawer Pin is in high state.
BXL_STS_ERROR	32	Offline (Printer is disconnected with HOST)
BXL_STS_NOT_OPEN	64	Not ready to start communication with the printer.
BXL_STS_BATTERY_LOW	128	Battery runs low
BXL_STS_PAPER_TO_BE_TAKEN	256	Printed receipt(label) presence is detected.
BXL_STS_CASHDRAWER_LOW	512	Cash drawer Pin is in low state.
BXL_STS_STACKER_FULL	16384	Stacker is full of paper
BXL_STS_STACKER_ERROR	32768	Error occurred in stacker

[Syntax]

long **BidiSetCallback**(fnStatusCallback pBidiCallBackFn);

[Parameters]

pBidiCallBackFn	A call back function pointer which receives printer status. *** typedef int (CALLBACK *fnStatusCallback)(int nStatus)**

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
long lResult = PrinterOpen(3, “192.168.1.1”, 9100);
……
int CALLBACK StatusCallBackMethod(int status){
  if ((status & BXL_STS_PAPEREMPTY) == BXL_STS_PAPEREMPTY) { … }
  if ((status & BXL_STS_COVEROPEN) == BXL_STS_COVEROPEN) { … }
  if ((status & BXL_STS_CASHDRAWER_HIGH) == BXL_STS_CASHDRAWER_HIGH) { … }
  if ((status & BXL_STS_BATTERY_LOW) == BXL_STS_BATTERY_LOW) { … }
  if ((status & BXL_STS_ERROR) == BXL_STS_ERROR) { … }
  if ((status & BXL_STS_PAPER_TO_BE_TAKEN) == BXL_STS_PAPER_TO_BE_TAKEN) { … }
}
……
BidiSetCallBack (StatusCallBackMethod);
……
```

5-25 Bidi Cancel Call Back

• Cancels notification of printer status values.

[Syntax]

long **BidiCancelCallback**();

[Parameters]

None

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
.....
BidiCancelCallback();
.....
```

5-26 Transaction Start

• Enters 'Transaction' mode.



This method sends the print data in logical units (receipt, label). Actual printing only starts when the 'Transaction End' method is called, so be sure to call the 'Transaction End' method.

[Syntax]
long **TransactionStart**

[Parameters]
None

[Return Values]
If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
TransactionStart();
.....
TransactionEnd(TRUE, 3000 /*milliseconds*/);
```

5-27 Transaction End

• Leaves 'Transaction' mode, and then sends print data in the buffer to start printing.

[Syntax]

long **TransactionEnd**(
  bool SendCompletedCheck,
  long TimeOut,
);

[Parameters]

Send Completed Check	Whether or not to check sending data to the printer successfully.
Time Out	Time in milliseconds used to verify that the data was successfully sent to the printer. *** Be careful to set this value because it can return a non-zero value depending on the print length and the communication speed used.** *** This parameter is only valid when 'Send Completed Check ' is TRUE.**

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
TransactionStart();
.....
TransactionEnd(TRUE, 3000 /*milliseconds*/);
```

5-28 Set Page mode

• Enters or Leaves 'Page' mode, and then sends the data in the buffer to the printer.

[Syntax]

long **SetPagemode** (
  long PageMode
);

[Parameters]

Page mode	Enter or leave page mode

Code	Value	Description
BXL_PRINTER_PAGEMODE_OUT	0	page mode out and send the data
BXL_PRINTER_PAGEMODE_IN	1	page mode in

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
int xPosition = 50;
int yPosition = 50;
BXLAPI.SetPagemode(BXLAPI.BXL_PRINTER_PAGEMODE_IN);
BXLAPI.SetPagemodePrintArea(0, 0, 512, 600);
BXLAPI.SetPagemodeDirection(BXLAPI.BXL_PRINTER_PD_RIGHT90);
BXLAPI.SetPagemodePosition(xPosition, yPosition);
BXLAPI.PrintBarcode("12345678", BXLAPI.BXL_BCS_CODE128, 100, 2,
BXLAPI.BXL_ALIGNMENT_CENTER, BXLAPI.BXL_BC_TEXT_NONE);
BXLAPI.SetPagemode(BXLAPI.BXL_PRINTER_PAGEMODE_OUT);
```

5-29 Set Page mode Print Area

• Sets the printing area of page mode.

[Syntax]

long **SetPagemodePrintArea** (
  long X,
  long Y,
  long Width,
  long Height,
);

[Parameters]

X	The starting X axis coordinate
Y	The starting Y axis coordinate
Width	Printing Width
Height	Printing Height

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

5-30 Set Page mode Direction

• Sets the printing direction in page mode.

[Syntax]

long **SetPagemodeDirection** (
  long Direction
);

[Parameters]

Direction	Print Direction		
	Code	Value	Description
	BXL_PRINTER_PD_0	0	left to right (upper left)
	BXL_PRINTER_PD_LEFT90	1	bottom to top (lower left)
	BXL_PRINTER_PD_180	2	right to left (lower right)
	BXL_PRINTER_PD_RIGHT90	3	top to bottom (upper right)

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

5-31 Set Page mode Position

• Sets the X and Y coordinates of the printing target in page mode.

[Syntax]

long **SetPagemodePosition** (
  long X,
  long Y
);

[Parameters]

X	x-coordinate, in dot unit, of the object (text, barcode, image) to be printed.
Y	y-coordinate, in dot unit, of the object (text, barcode, image) to be printed.

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

5-32 Get Statistics Data

• Gets the printer's statistics data.

[Syntax]

long **GetStatisticsData** (int dataType);

[Parameters]

dataType	Statistics data type to be checked.

[Return Values]

If the method succeeds, the return value is data value.
Other cases, returns negative values.

6. KIOSK Presenter Methods

Please note that APIs are only available to presenter that is attached at BK3-3 KIOSK printer.

6-1 Paper Eject

• Ejects paper

[Syntax]

long **PaperEject** (
  int option
);

[Parameters]

Option	Options for paper ejecting		
	Code	Value	Description
	BXL_EJT_NONE	0	Ejects paper all
	BXL_EJT_HOLD	1	Ejects paper with hold(Default)

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

6-2 Paper Eject Specify

• Ejects specified length paper

[Syntax]

long **PaperEjectSpecify** (
  int ejectLength
);

[Parameters]

Eject Length	Eject paper Length (Range: 1 ≤ Eject Length ≤ 8191)

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

6-3 Get Presenter Status

• Gets the printer's current status value

[Syntax]

long **GetPresenterStatus** ();

[Parameters]

None

[Return Values]

If the method succeeds, the return value is one of the following. Other cases, returns negative values.

Presenter Status	Value	Description
BXL_PRT_STS_NORMAL	0	Normal. (No errors)
BXL_PRT_STS_PAPER_NEAR_END	1	Paper is near-end
BXL_PRT_STS_PAPEREMPTY	4	Paper empty
BXL_PRT_STS_PAPER_IN	8	Paper is detected at presenter
BXL_PRT_STS_PAPER_JAM	128	Paper jam

* Example

```
long lResult = PrinterOpen("portinfo...", 1000);
......
int status = GetPresenterStatus ();
......
if (status & BXL_PRT_STS_PAPER_NEAR_END) { ... }
if (status & BXL_PRT_STS_PAPEREMPTY) { ... }
if (status & BXL_PRT_STS_PAPER_IN) { ... }
if (status & BXL_PRT_STS_PAPER_JAM) { ... }
.......
```

6-4 Paper Retract

• Retract paper

[Syntax]

long **PaperRetract** ();

[Parameters]

None

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

6-5 Eject And Retract Paper

• Eject the paper and retract it after a timeout.

[Syntax]

long **EjectAndRetractPaper**(int timeout);

[Parameters]

Time Out	Time in milliseconds used for automatic retraction after paper ejection.

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

7. Customer Display Methods

Please note that APIs related to customer display in this document are only available to BCD-3000 connected with SRP-Q300/Q302 printer. “Printer Open” method must be called before calling any other methods.

7-1 Clear Screen

• Clears all the displayed and then the cursor moves to the home position (left upper).

[Syntax]
long **ClearScreen**( );

[Parameters]

None

[Return Values]
If the method succeeds, the return value is 0. Other cases, returns negative values.

7-2 Clear Line

• Moves the cursor to the specified line, and then clear that line.

[Syntax]

long **ClearLine** (long nLine);

[Parameters]

nLine	The line number to clear. (Range: 1 ≤ nLine ≤ 4)

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
.....
ClearLine(1);          // 1st line
ClearLine(2);          // 2nd line
.....
```

7-3 Display String

• Displays characters at the current cursor position of the customer display.

[Syntax]

long **DisplayString** (
  LPCSTR szData
);

[Parameters]

szData	Text data in ASCII format.

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
……
DisplayString (“TEXT DATA”);
;……
```

7-4 Display String W

• Displays characters at the current cursor position of the customer display.

[Syntax]

long **DisplayString** (
  LPCWSTR szData,
  long codepage
);

[Parameters]

<table>
<tr><td>szData</td><td>Text data in UNICODE format.</td></tr>
<tr><td>Code page</td><td>Code page for encoding text data.
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>BXL_TEXT_ENCODE_PC437</td><td>437</td><td>USA, Standard Europe</td></tr>
<tr><td>BXL_TEXT_ENCODE_PC850</td><td>850</td><td>Multilingual Latin 1</td></tr>
<tr><td>BXL_TEXT_ENCODE_PC860</td><td>860</td><td>Portuguese</td></tr>
<tr><td>BXL_TEXT_ENCODE_PC863</td><td>863</td><td>Canadian-French</td></tr>
<tr><td>BXL_TEXT_ENCODE_PC865</td><td>865</td><td>Nordic</td></tr>
<tr><td>BXL_TEXT_ENCODE_WPC1252</td><td>1252</td><td>Latin 1</td></tr>
<tr><td>BXL_TEXT_ENCODE_PC866</td><td>866</td><td>Cyrillic #2</td></tr>
<tr><td>BXL_TEXT_ENCODE_PC852</td><td>852</td><td>Latin 2</td></tr>
<tr><td>BXL_TEXT_ENCODE_PC858</td><td>858</td><td>Euro</td></tr>
<tr><td>BXL_TEXT_ENCODE_PC862</td><td>862</td><td>Hebrew DOS code</td></tr>
<tr><td>BXL_TEXT_ENCODE_WPC1253</td><td>1253</td><td>Greek</td></tr>
<tr><td>BXL_TEXT_ENCODE_WPC1254</td><td>1254</td><td>Turkish</td></tr>
<tr><td>BXL_TEXT_ENCODE_WPC1257</td><td>1257</td><td>Baltic</td></tr>
<tr><td>BXL_TEXT_ENCODE_WPC1251</td><td>1251</td><td>Cyrillic</td></tr>
<tr><td>BXL_TEXT_ENCODE_PC737</td><td>737</td><td>Greek</td></tr>
<tr><td>BXL_TEXT_ENCODE_PC775</td><td>775</td><td>Baltic</td></tr>
<tr><td>BXL_TEXT_ENCODE_WPC1255</td><td>1255</td><td>Hebrew New Code</td></tr>
<tr><td>BXL_TEXT_ENCODE_PC855</td><td>855</td><td>Cyrillic</td></tr>
<tr><td>BXL_TEXT_ENCODE_PC857</td><td>857</td><td>Turkish</td></tr>
<tr><td>BXL_TEXT_ENCODE_PC928</td><td>928</td><td>Greek</td></tr>
<tr><td>BXL_TEXT_ENCODE_PC1258</td><td>1258</td><td>Vietnamese (PC1258)</td></tr>
<tr><td>BXL_TEXT_ENCODE_PC1250</td><td>1250</td><td>Czech</td></tr>
<tr><td>BXL_TEXT_ENCODE_TCVN</td><td>49</td><td>Vietnamese (TCVN3)</td></tr>
<tr><td>BXL_TEXT_ENCODE_TCVN_CAPITAL</td><td>50</td><td>Vietnamese (TCVN3 CAPITAL)</td></tr>
<tr><td>BXL_TEXT_ENCODE_VISCII</td><td>51</td><td>Vietnamese (VISCII)</td></tr>
</table></td></tr>
</table>

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
……
DisplayStringW (“TEXT DATA”, BXL_TEXT_ENCODE_PC437);
……
```

7-5 Display String At Line

• Displays characters at the specified line left-most of the customer display

[Syntax]

long **DisplayStringAtLine** (
  long nLine,
  LPCSTR szData
);

[Parameters]

nLine	Line number to display. (Range: 1 ≤ nLine ≤ 4)
szData	Text data in ASCII format.

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
.....
DisplayStringAtLine (1, "TEXT DATA"); // 1st line
DisplayStringAtLine (2, "TEXT DATA"); // 2nd line
;.....
```

7-6 Display String At Line W

• Displays characters at the specified line left-most of the customer display

[Syntax]

long **DisplayStringAtLine** (
  long nLine,
  LPCSTR szData,
  long codepage
);

[Parameters]

nLine	Line number to display. (Range: 1 ≤ nLine ≤ 4)
szData	Text data in UNICODE format.
Code page	Code page for encoding text data.

Code	Value	Description
BXL_TEXT_ENCODE_PC437	437	USA, Standard Europe
BXL_TEXT_ENCODE_PC850	850	Multilingual Latin 1
BXL_TEXT_ENCODE_PC860	860	Portuguese
BXL_TEXT_ENCODE_PC863	863	Canadian-French
BXL_TEXT_ENCODE_PC865	865	Nordic
BXL_TEXT_ENCODE_WPC1252	1252	Latin 1
BXL_TEXT_ENCODE_PC866	866	Cyrillic #2
BXL_TEXT_ENCODE_PC852	852	Latin 2
BXL_TEXT_ENCODE_PC858	858	Euro
BXL_TEXT_ENCODE_PC862	862	Hebrew DOS code
BXL_TEXT_ENCODE_WPC1253	1253	Greek
BXL_TEXT_ENCODE_WPC1254	1254	Turkish
BXL_TEXT_ENCODE_WPC1257	1257	Baltic
BXL_TEXT_ENCODE_WPC1251	1251	Cyrillic
BXL_TEXT_ENCODE_PC737	737	Greek
BXL_TEXT_ENCODE_PC775	775	Baltic
BXL_TEXT_ENCODE_WPC1255	1255	Hebrew New Code
BXL_TEXT_ENCODE_PC855	855	Cyrillic
BXL_TEXT_ENCODE_PC857	857	Turkish
BXL_TEXT_ENCODE_PC928	928	Greek
BXL_TEXT_ENCODE_PC1258	1258	Vietnamese (PC1258)
BXL_TEXT_ENCODE_PC1250	1250	Czech
BXL_TEXT_ENCODE_TCVN	49	Vietnamese (TCVN3)
BXL_TEXT_ENCODE_TCVN_CAPITAL	50	Vietnamese (TCVN3 CAPITAL)
BXL_TEXT_ENCODE_VISCII	51	Vietnamese (VISCII)

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
……
DisplayStringAtLineW (1, “TEXT DATA”, BXL_TEXT_ENCODE_PC437);    // 1st line
DisplayStringAtLineW (2, “TEXT DATA”, BXL_TEXT_ENCODE_PC437);    // 2nd line
;……
```

7-7 Store Image File

• Stores image data to the nonvolatile memory of customer display.



Only BMP file format can be stored.

[Syntax]

long **StoreImageFile** (
  LPCSTR FileName,
  long nIndex
);

[Parameters]

File Name	The full path of the image file.
nIndex	The index number used to store image data. (Range: 1 ≤ nIndex ≤ 5)

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
.....
StoreImageFile ("c:\images\filename1.bmp",   1);
//StoreImageFile ("c:\images\filename2.bmp",   2);
//StoreImageFile ("c:\images\filename3.bmp",   3);
//StoreImageFile ("c:\images\filename4.bmp",   4);
//StoreImageFile ("c:\images\filename5.bmp",   5);
;.....
```

7-8 Display Image

• Displays the image stored by “Store Image File” method at the specified coordinates.

[Syntax]

long **DisplayImage** (
 long nIndex,
 long nXPostion,
 long nYPostion);

[Parameters]

nIndex	The index number stored image data. (Range: 1 ≤ nIndex ≤ 5)
nXPosition	X-coordinate. (Range: 0 ≤ nXPosition < 160)
nYPosition	Y-coordinate. (Range: 0 ≤ nYPosition < 32)

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
.....
// displays image stored by the index number at screen (1,1)
DisplayImage (1, 1, 1);
//DisplayImage (2, 1, 1);
//DisplayImage (3, 1, 1);
//DisplayImage (4, 1, 1);
//DisplayImage (5, 1, 1);
;.....
```

7-9 Set International Character set

• Sets the character set used for displaying some of international characters.

[Syntax]

long SetInternationalCharacterset (
long nInternationalCharacterSet
);

[Parameters]

nInternational Character Set

Character set. (Range: 0 ≤ nInternational Character Set ≤ 15)

Code	Value	Description
BXL_ICS_USA	0	USA code setting
BXL_ICS_FRANCE	1	FRANCE code setting
BXL_ICS_GERMANY	2	GERMANY code setting
BXL_ICS_UK	3	UK code setting
BXL_ICS_DENMARK1	4	DENMARK1 code setting
BXL_ICS_SWEDEN	5	SWEDEN code setting
BXL_ICS_ITALY	6	ITALY code setting
BXL_ICS_SPAIN	7	SPAIN code setting
BXL_ICS_NORWAY	9	NORWAY code setting
BXL_ICS_DENMARK2	10	DENMARK 2 code setting
BXL_ICS_SPAIN2	11	SPAIN 2 code setting
BXL_ICS_LATIN	12	LATIN AMERICA code setting
BXL_ICS_KOREA	13	KOREA code setting
BXL_ICS_SLOVENIA	14	SLOVENIA code setting
BXL_ICS_CHINA	15	CHINA code setting

[Return Values]

If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
.....
SetInternationalCharacterset (BXL_ICS_USA);
.....
```

8. Error Code Table

The error code table below summarizes the values returned when calling APIs provided in this manual.

Code	Value	Description
BXL_SUCCESS	0	Success
BXL_CONNECT_FAIL	-100	Failed to connect the printer
BXL_NOT_OPENED	-101	Not connected with the printer
BXL_NOT_SUPPORT	-107	Not Supported API
BXL_INVALID_PARAM	-108	Invalid Parameter
BXL_NO_PRINT_DATA	-110	No data to print
BXL_COMPLETE_ERROR	-111	Failed to receive a response from the printer in transaction mode
BXL_NO_TRANSACTION_START	-112	Not in transaction mode
BXL_TEXT_ENCODING_ERROR	-120	Failed to encoding characters
BXL_PAGE_MODE_ALREADY_IN	-130	Already page mode in
BXL_NOT_IN_PAGE_MODE	-131	Not in page mode
BXL_WRITE_ERROR	-300	Failed to send data
BXL_READ_ERROR	-301	Failed to receive data
BXL_BMP_LOAD_ERROR	-400	Failed to load a bitmap file
BXL_BMP_BUFFER_SIZE_ERROR	-401	Bitmap buffer size error
BXL_BC_DATA_ERROR	-500	Barcode data format is not correct
BXL_BC_NOT_SUPPORT	-501	Not supported barcode

Revision history

Ver.	Date	Description
1.02	13-11-2017	New document style applied. Information for the supported Printer models added.
1.03	01-04-2018	Information for the supported Printer models added. The Information about 1d/2d codes added. NEW APIs are added (5-12, 5-13)
1.04	23-02-2018	- Information about the printer status is updated. Constant values for code page not supported are removed.
1.05	27-03-2018	- New methods are added : Bidi Set Call Back, Bidi Cancel Call Back - Descriptions are updated : Check Printer
1.06	05-06-2018	Most of contents of this manual updated.
1.07	22-08-2018	- API Added : Print QR Code, PrintPDF417, Print Data Matrix, PrintGS1DataBar, Print Composite Symbology, Print Text W, Transaction Start, Transaction End - API Removed : Check Printer, Get Stat - Descriptions for APIs below are updated : Print Bitmap, Print Barcode, Open Drawer, Display String W, Display String At Line W, Print Text, Print Text W, Display String Display String At Line W, Set Character Set
1.08	17-10-2018	-Information for the supported Printer models added. - API Added : Set Page mode, Set Page mode Print Area, Set Page mode Direction : Set Page mode Position
1.09	04-03-2019	-Information for the supported Printer models added. - API Added : Print Text In Image, Print Text In Image W
1.10	28-11-2019	-Information for the supported Printer models added. - API Added : Paper Eject, Get Presenter Status
2.00	17-09-2019	-Information for the supported OS. -Information for the supported Printer models added.
2.01	24-02-2021	-Information for the supported Printer models added.
2.02	23-05-2022	- Add model: SRP-330III / 332III - Change COPYRIGHT Position.
2.03	02-06-2022	- Add model: SRP-350V / 352V / 350plusV / 352plusV
2.04	22-07-2022	- Add model: SRP-380II / 382II

Ver.	Date	Description
2.05	17-03-2023	- Add model: BK5-3 / SPP-200 / G30 - Add API : Paper Eject Specify
2.06	10-10-2023	- Add model: SRP-Q300II
2.07	09-05-2024	- Add model: SRP-S300II
2.08	26-12-2024	- Renameing the printer model : SRP-S380II/382II -> SRP-S380plus/S382plus - Add model : SRP-G300 - Add API : PaperRetract, EjectAndRetractPaper
2.09	31-03-2025	- Add model: BK5-4
2.10	28-04-2025	- Add model: SRP-B300II
2.11	12-03-2026	- Add model: SPP-300 - Add printer status value : BXL_STS_STACKER_FULL, BXL_STS_STACKER_ERROR