



API Reference Guide

Windows POS SDK

Ver. 1.10

Table of Contents

1. Operating System (OS) Environment.....	3
2. Windows POS SDK Preparation	3
3. Supported Printers.....	4
4. Sample Program.....	4
5. Printer Methods	5
5-1 PrinterOpen	5
5-2 PrinterClose	6
5-3 LineFeed	6
5-4 SetCharacterSet	8
5-5 SetInterChrSet	9
5-6 PrintText	10
5-7 PrintTextW	12
5-8 PrintTextInImage	14
5-9 PrintTextInImageW	15
5-10 PrintBitmap	16
5-11 PrintBarcode	17
5-12 PrintQRCode	19
5-13 PrintPDF417	20
5-14 PrintDataMatrix	21
5-15 PrintGS1DataBar	22
5-16 PrintCompositeSymbology	23
5-17 GetPrinterCurrentStatus	24
5-18 InitializePrinter	25
5-19 DirectIO	25
5-20 CutPaper	26
5-21 OpenDrawer	26
5-22 WriteBuff	27
5-23 ReadBuff	27
5-24 BidiSetCallBack	28
5-25 BidiCancelCallBack	29
5-26 TransactionStart	29
5-27 TransactionEnd	30
5-28 SetPagemode	31
5-29 SetPagemodePrintArea	32
5-30 SetPagemodeDirection	32
5-31 SetPagemodePosition	33
6. KIOSK Presenter Methods	34
6-1 PaperEject	34
6-2 GetPresenterStatus	35
7. Customer Display Methods.....	36
7-1 ClearScreen	36
7-2 ClearLine	36
7-3 DisplayString	37
7-4 DisplayStringW	38
7-5 DisplayStringAtLine	39
7-6 DisplayStringAtLineW	40
7-7 StoreImageFile	41
7-8 DisplayImage	41
7-9 SetInternationalCharacterSet	42
8. Error Code Table	43

1. Operating System (OS) Environment

The following operating systems are supported for usage.

Microsoft® Windows XP SP3 (32bit)
Microsoft® Windows XP SP1 or later (64bit)
Microsoft Windows Server 2003 SP1 or later (32bit/64bit)
Microsoft Windows VISTA (32bit/64bit)
Microsoft Windows Server 2008 (32bit/64bit)
Microsoft Windows Server 2008R2 (64bit)
Microsoft Windows 7 (32bit/64bit)
Microsoft Windows 8 (32bit/64bit)
Microsoft Windows Server 2012 (64bit)
Microsoft Windows 10 (32bit/64bit)

2. Windows POS SDK Preparation

The Windows POS SDK is included in the enclosed CD, and the latest file version can be downloaded from BIXOLON website. (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-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-330II	180 dpi	512 dots
SRP-332II	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-340II	180 dpi	512 dots
SRP-342II	203 dpi	576 dots
SRP-275III	160 dpi	400 dots
BK3-3	203 dpi	576 dots
BK3-2	203 dpi	432 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 PrinterOpen

- Establishes a communication connection with the printer.



Note

- 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]

nInterface	The printer communication interface.		
	Code	Value	Description
	ISerial	0	Serial Printer
	IParallel	1	Parallel Printer
	IUsb	2	USB Printer
	ILan	3	Ethernet Printer
	IWLan	4	Wi-Fi Printer
	IBluetooth	5	Bluetooth Printer
szPortName	Port name for Serial, Bluetooth and Parallel, or IP address for network. * This parameter is only valid for Serial, parallel or network communication.		
nBaudRate	The baud rate for Serial communication, or port number for network. * This parameter is only valid for Serial, parallel or network communication.		
nDataBits	The number of bits in the bytes transmitted and received * This parameter is only valid for Serial communication.		
nParity	The parity scheme to be used for Serial communication. * This parameter is only valid for Serial communication.		
nStopBits	The number of stop bits to be used for Serial communication. * This parameter is only valid for Serial communication.		
nFlowControl	The flow control to be used for Serial communication. * This parameter is only valid for Serial communication.		

[Return Values]

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

*** Example**

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

5-2 PrinterClose

- Disconnects the communication with the printer.

**Note**

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

5-3 LineFeed

- Sends line-feeds commands to the printer.

[Syntax]

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

[Parameters]

Feed	Paper feed amount in lines. (Range: $1 \leq \text{Feed} \leq 255$)
------	---

[Return Values]

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

* Example

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

5-4 SetCharacterSet

- Sets the code table used for printing characters.

[Syntax]

```
long SetCharacterSet (
    long Codepage
);
```

[Parameters]

Codepage	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 IResult = PrinterOpen("portinfo...", 1000);
.....
SetCharacterSet(BXL_CS_WPC1252);
.....
```


5-5 SetInterChrSet

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

[Syntax]

```
long SetInterChrSet (  
    long InternationalCharacterSet  
);
```

[Parameters]

International CharacterSet	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 IResult = PrinterOpen("portinfo...", 1000);  
.....  
SetInterChrSet(BXL_ICS_USA);  
.....
```

5-6 PrintText

- Sends a character print command to the printer.

[Syntax]

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

[Parameters]

Data	Text data 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.
TextSize	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

[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 PrintTextW

- 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.
TextSize	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

CodePage	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 IResult = 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 PrintTextInImage

- 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 Alignment,
);
```

[Parameters]

Data	Text data to be printed.		
FontName	TTF (true type font) name.		
Italic	Italic or not		
Bold	Bold or not		
Underline	Underline or not		
FontSize	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 IResult = PrinterOpen("portinfo...", 1000);
.....
PrintTextInImage("ABCD", "Arial", FALSE, FALSE, FALSE, 15, BXL_ALIGNMENT_LEFT);
.....
```

5-9 PrintTextInImageW

- 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 Alignment,
);
```

[Parameters]

Data	Text data in UNICODE format to be printed.		
FontName	TTF (true type font) name in UNICOD format.		
Italic	Italic or not		
Bold	Bold or not		
Underline	Underline or not		
FontSize	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 IResult = PrinterOpen("portinfo...", 1000);
.....
PrintTextInImageW(L"ABCD", L "Arial", FALSE, FALSE, FALSE, 15, BXL_ALIGNMENT_LEFT);
.....
```

5-10 PrintBitmap

- Sends a graphic image command to the printer.

**Note**

Only BMP file format can be printed.

[Syntax]

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

[Parameters]

FileName	The full path of the image file.		
Width	The width of the image to be printed.		
	Code	Value	Description
	BXL_WIDTH_FULL	-1	Maximum width the printer can print.
	BXL_WIDTH_NONE	-2	No variation given to image size
Alignment	Alignment		
	Code	Value	Description
	BXL_ALIGNMENT_LEFT	0	Left-aligned
	BXL_ALIGNMENT_CENTER	1	Centered
	BXL_ALIGNMENT_RIGHT	2	Right-aligned
Level	The brightness level. (Range: $0 \leq \text{Level} \leq 100$)		
Dithering	Whether apply the dither option or not.		

[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 PrintBarcode

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

**Note**

This method is not supported on SRP-275III.

[Syntax]

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

[Parameters]

Data	Barcode data in ASCII format.			
Symbology	Code Symbology.			
	Symbology	Value	Length Limit	Value Range
	BXL_BCS_UPCA	101	$11 \leq n \leq 12$	$48 \leq \text{data} \leq 57$
	BXL_BCS_UPCE	102	$11 \leq n \leq 12$	$48 \leq \text{data} \leq 57$
	BXL_BCS_EAN13	104	$12 \leq n \leq 13$	$48 \leq \text{data} \leq 57$
	BXL_BCS_JAN13	106		
	BXL_BCS_EAN8	103	$7 \leq n \leq 8$	$48 \leq \text{data} \leq 57$
	□XL_BCS_JAN8	105		
	BXL_BCS_ITF	107	$1 \leq n \leq 255$ (even)	$48 \leq \text{data} \leq 57$
	BXL_BCS_CODABAR	108	$1 \leq n \leq 255$	$48 \leq \text{data} \leq 57$, $65 \leq \text{data} \leq 68$, data = 36,43,45,46,47,58
	BXL_BCS_CODE39	109	$1 \leq n \leq 255$	$48 \leq \text{data} \leq 57$, $65 \leq \text{data} \leq 90$, data = 32,36,37, 43, 45,46,47
	BXL_BCS_CODE93	110	$1 \leq n \leq 255$	$0 \leq \text{data} \leq 127$
	BXL_BCS_CODE128	111	$2 \leq n \leq 255$	$0 \leq \text{data} \leq 127$
	BXL_BCS_1D_GS1_128	112	$2 \leq n \leq 255$	$0 \leq \text{data} \leq 127$
	BXL_BCS_1D_GS1DATABAR_OMNIDIRECTION	113	$n = 13$	$48 \leq \text{data} \leq 57$
	BXL_BCS_1D_GS1DATABAR_TRUNCATED	114	$n = 13$	$48 \leq \text{data} \leq 57$
	BXL_BCS_1D_GS1DATABAR_LIMITED	115	$n = 13$	$48 \leq \text{data} \leq 57$
Height	The barcode height. (Range: $1 \leq \text{Height} \leq 255$)			
Width	The width or size of a 1d barcode. (Range: $2 \leq \text{Width} \leq 6$) * If the print width of the barcode exceeds the printing paper, barcodes may not be printed properly.			
Alignment	Alignment			
	Code	Value	Description	
	BXL_ALIGNMENT_LEFT	0	Left-aligned	
	BXL_ALIGNMENT_CENTER	1	Centered	
	BXL_ALIGNMENT_RIGHT	2	Right-aligned	

TextPosition	The HRI (Human Readable Interpretation) printing position of the 1d barcode.		
	Code	Value	Description
	BXL_BC_TEXT_NONE	0	Not Printed
	BXL_BC_TEXT_ABOVE	1	Above the barcode
	BXL_BC_TEXT_BELOW	2	Below the barcode
	BXL_BC_TEXT_BOTH	3	Above and Below the barcode

[Return Values]

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

*** Example**

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

5-12 PrintQRCode

- Sends a QR Code print command to the printer.

**Note**

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: $1 \leq \text{Size} \leq 8$)		
EccLevel	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 IResult = PrinterOpen("portinfo...", 1000);
.....
PrintQRCode("123456789012", BXL_QRCODE_MODEL_1, 4, _QRCODE_LEVEL_L,
BXL_ALIGNMENT_LEFT);
.....
```

5-13 PrintPDF417

- Sends a PDF417 print command to the printer.

**Note**

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
ColumnNumber	Number of columns (Range: 0 ≤ ColumnNumber ≤ 30)		
RowNumber	Number of rows (Range: 3 ≤ RowNumber ≤ 90)		
ModuleWidth	Module width (Range: 1 ≤ Width ≤ 4)		
ModuleHeight	Module height (Range: 2 ≤ Height ≤ 8)		
EccLevel	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 IResult = PrinterOpen("portinfo...", 1000);
.....
PrintPDF417("123456789012", BXL_PDF417_TYPE_1, 0, 0, 2, 2 BXL_PDF417_LEVEL_L,
BXL_ALIGNMENT_LEFT);
.....
```

5-14 PrintDataMatrix

- Sends a Data Matrix print command to the printer.



Note

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: $2 \leq \text{Size} \leq 3$)		
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 IResult = PrinterOpen("portinfo...", 1000);
.....
PrintDataMatrix("123456789012", 2, BXL_ALIGNMENT_LEFT);
.....
```

5-15 PrintGS1DataBar

- Sends a GS1 DataBar print command to the printer.

**Note**

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 DataBar Symbol		
	Code	Value	Description
	BXL_GS1DATABAR_STACKED	0	Stacked
	BXL_GS1DATABAR_STACKED_OMNIDIRECTIONAL	1	Stacked Omnidirectional
Size	Module size (Range: $1 \leq \text{Size} \leq 8$)		
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 IResult = PrinterOpen("portinfo...", 1000);
.....
PrintGS1DataBar("123456789012", BXL_GS1DATABAR_STACKED, 2, BXL_ALIGNMENT_LEFT);
.....
```

5-16 PrintCompositeSymbology

- Sends a Composite Symbology print command to the printer.

**Note**

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]

Data1d	1-dimensional Code data in ASCII format.		
Symbol1d	Type of straight line element		
	Code	Value	Description
	BXL_COMPOSITE_LINE_EAN8	65	EAN8
	BXL_COMPOSITE_LINE_EAN13	66	EAN13
	BXL_COMPOSITE_LINE_UPCA	67	UPC-A
	BXL_COMPOSITE_LINE_UPCE	69	UPC-E
	BXL_COMPOSITE_LINE_GS1DATABAR_OMNIDIRECTIONAL	70	GS1 DataBar Omnidirectional
	BXL_COMPOSITE_LINE_GS1DATABAR_TRUNCATED	71	GS1 DataBar Truncated
	BXL_COMPOSITE_LINE_GS1DATABAR_STACKED	72	GS1 DataBar Stacked
	BXL_COMPOSITE_LINE_GS1DATABAR_STACKED_OMNIDIRECTIONAL	73	GS1 DataBar Stacked Omnidirectional
	BXL_COMPOSITE_LINE_GS1DATABAR_LIMITED	74	GS1 DataBar Limited
	BXL_COMPOSITE_LINE_GS1DATABAR_EXPANDED	75	GS1 DataBar Expanded
	BXL_COMPOSITE_LINE_GS1_128	77	GS1-128
Data2d	2-dimensional Code data in ASCII format.		
Symbol2d	2-dimensional synthetic element		
	Code	Value	Description
	BXL_COMPOSITE_2D_AUTO	65	CC-A, CC-B or CC-C (AUTO)
	BXL_COMPOSITE_2D_CC_C	66	CC-C (Only GS1-128)
Size	Module size (Range: $1 \leq \text{Size} \leq 8$)		
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 IResult = PrinterOpen("portinfo...", 1000);
.....
PrintCompositeSymbology("123456789012", BXL_COMPOSITE_LINE_EAN8, "ABCDEFGHJIJ",
BXL_COMPOSITE_2D_AUTO, BXL_ALIGNMENT_LEFT);
.....
```

5-17 GetPrinterCurrentStatus

- 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.

*** Example**

```
long IResult = 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 InitializePrinter

- 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 DirectIO

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



Note

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.
DataSize	The size of buffer specified by the parameter 'Data'.
RecvBuffer	The buffer, which receives a response from the printer. * set this parameter as NULL if no need to receive a response.
RecvBufferSize	The size of the buffer specified by the parameter 'RecvBuffer'. * 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 IResult = PrinterOpen("portinfo...", 1000);
.....
CHAR cmd[3] = {0x10, 0x04, 0x02};
CHAR * readData;
UINT readLen = 0;
.....
IResult = DirectIO(cmd, sizeof(cmd), readData, readLen);
.....
```

5-20 CutPaper

- 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.

5-21 OpenDrawer

- Sends a cash drawer open command to the printer.

[Syntax]

long **OpenDrawer** (int PinNumber);



Note

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

[Parameters]

PinNumber	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 WriteBuff

- 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.
DataSize	The size of buffer specified by the parameter 'Data'
SizeSent	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 ReadBuff

- Reads data from the printer.

[Syntax]

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

[Parameters]

ReadBuffer	A pointer to the buffer that receives the data read from the printer.
NumberOfBytesToRead	Maximum data length to read from the printer.
DataSizeRead	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 BidiSetCallback

- 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.

[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 BidiCancelCallback

- 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 TransactionStart

- Enters 'Transaction' mode.



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

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

[Syntax]

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

[Parameters]

SendCompletedCheck	Whether or not to check sending data to the printer successfully.
TimeOut	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 'SendCompletedCheck' 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 SetPagemode

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

[Syntax]

```
long SetPagemode (
    long PageMode
);
```

[Parameters]

Pagemode	Enter or leave page mode		
	Code	Value	Description
	BXL_PRINTER_PAGEMODE_OUT	0	page mode in
	BXL_PRINTER_PAGEMODE_IN	1	page mode out and send the data

[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 SetPagemodePrintArea

- 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 SetPagemodeDirection

- 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 SetPagemodePosition

- 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.

6. KIOSK Presenter Methods

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

6-1 PaperEject

- 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 GetPresenterStatus

- 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) { ... }
.....
```

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. "PrinterOpen" method must be called before calling any other methods.

7-1 ClearScreen

- 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 ClearLine

- 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: $1 \leq nLine \leq 4$)
-------	---

[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 DisplayString

- 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 DisplayStringW

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

[Syntax]

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

[Parameters]

szData	Text data in UNICODE format.		
codepage	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

```
.....
DisplayStringW ("TEXT DATA", BXL_TEXT_ENCODE_PC437);
.....
```

7-5 DisplayStringAtLine

- 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: $1 \leq nLine \leq 4$)
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 DisplayStringAtLineW

- 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: $1 \leq nLine \leq 4$)		
szData	Text data in UNICODE format.		
codepage	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 StoreImageFile

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

**Note**

Only BMP file format can be stored.

[Syntax]

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

[Parameters]

FileName	The full path of the image file.
nIndex	The index number used to store image data. (Range: $1 \leq \text{nIndex} \leq 5$)

[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 DisplayImage

- Displays the image stored by "StoreImageFile" method at the specified coordinates.

[Syntax]

```
long DisplayImage (
    long nIndex,
    long nXPosition,
    long nYPosition);
```

[Parameters]

nIndex	The index number stored image data. (Range: $1 \leq \text{nIndex} \leq 5$)
nXPosition	X-coordinate. (Range: $0 \leq \text{nXPosition} < 160$)
nYPosition	Y-coordinate. (Range: $0 \leq \text{nYPosition} < 32$)

[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 SetInternationalCharacterSet

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

[Syntax]

```
long SetInternationalCharacterSet (
    long nInternationalCharacterSet
);
```

[Parameters]

nInternationalCharacterSet	Character set. (Range: $0 \leq \text{nInternationalCharacterSet} \leq 15$)		
	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

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".

Revision history

[illegible]