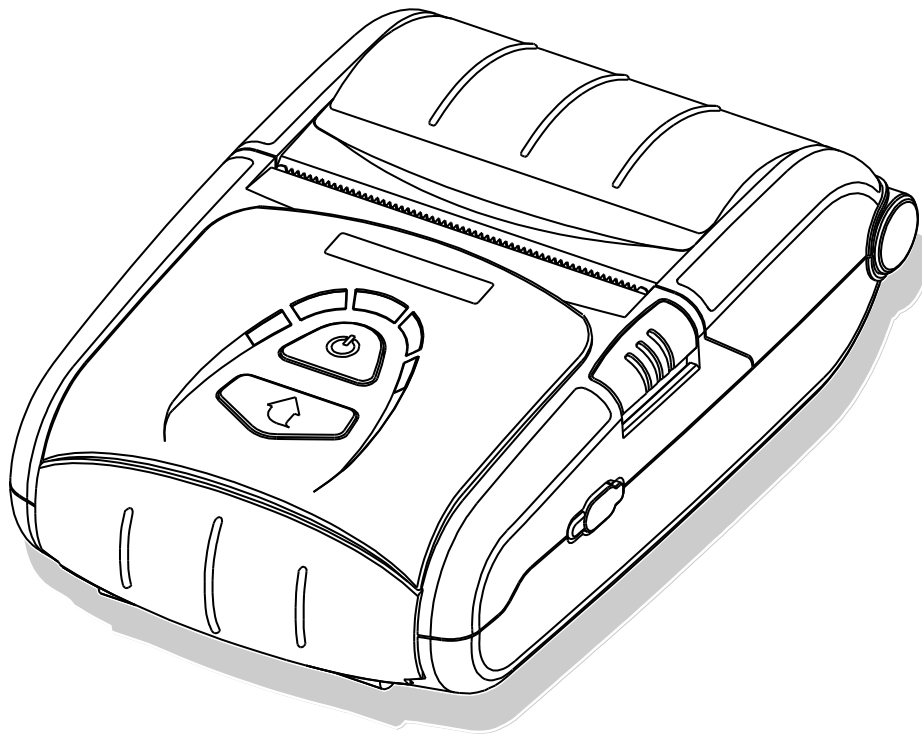# BIXOLON®

Windows Phone SDK API Reference Guide
Mobile Printer

Rev. 1.00

SPP-R210
SPP-R200II
SPP-R300
SPP-R400

■ Contents

1. About This Guide

This SDK Reference Guide documents the information about the dll files required for the development of applications for Windows OS.

At Bixolon, we continually strive to improve the performance and quality of our products. As a result, their specifications and user guides are subject to change without prior notice.

1-1 Supported O/S & Platform

- O/S
  • Windows Phone 8
  • Windows Phone 8.1

2. Property

The constant values used by the dll files provided are declared in BxlConst.cs.
The development environment is C#.

2-1 CharacterSet (LONG R/W)

The attribute which defines the printer's code page is set to BXL_CS_437 by default. Use
SetCharacterSet(), GetCharacterSet() to set new values or load the preset values.

The following code pages can be used:

Code	Value	Description
BXL_CS_PC437	0	Code page PC437
BXL_CS_KATAKANA	1	Katakana
BXL_CS_PC850	2	Code page PC850
BXL_CS_PC860	3	Code page PC860
BXL_CS_PC863	4	Code page PC863
BXL_CS_PC865	5	Code page PC865
BXL_CS_WPC1252	16	Code page WPC1252
BXL_CS_PC866	17	Code page PC866
BXL_CS_PC852	18	Code page PC852
BXL_CS_PC858	19	Code page PC858
BXL_CS_PC864	22	Code page PC864
BXL_CS_THAI42	23	Code page THAI42
BXL_CS_WPC1253	24	Code page WPC1253
BXL_CS_WPC1254	25	Code page WPC1254
BXL_CS_WPC1257	26	Code page WPC1257
BXL_CS_FARSI	27	Code page Farsi
BXL_CS_WPC1251	28	Code page WPC1251
BXL_CS_PC737	29	Code page PC737
BXL_CS_PC775	30	Code page PC775
BXL_CS_THAI14	31	Code page THAI14
BXL_CS_PC862	33	Code page PC862
BXL_CS_PC855	36	Code page PC855
BXL_CS_PC857	37	Code page PC857
BXL_CS_PC928	38	Code page PC928
BXL_CS_THAI16	39	Code page THAI16
BXL_CS_WPC1256	40	Code page PC1256
BXL_CS_PC1258	41	Code page PC1258
BXL_CS_KHMER	42	Code page KHMER
BXL_CS_PC1250	47	Code page PC1250
BXL_CS_USER	255	User set page

* Example

```
BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();
var ret = await bxlPrinter.ConnectToPrinterAsync("port info..");

……

bxlPrinter.SetCharacterSet(BXL_CS_PC850);

…...

int32 nCharSet;

nCharSet = bxlPrinter.GetCharacterSet();

……
```

2-2 International CharacterSet (LONG R/W)

The attribute which defines the printer's International character Set is set to BXL_ICS_USA by default. Use SetInterCharSet(), GetInterCharSet() to set new values or load the preset values.

The following International Character Set can be used:

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

* Example

```
BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();

var ret = await bxlPrinter.ConnectToPrinterAsync("port info..");

……

bxlPrinter.SetInterCharSet(BXL_ICS_SPAIN);

…...

int32 nCharSet;

nCharSet = bxlPrinter.GetInterCharSet();

……
```

2-3 State (LONG R)

This attribute sets the printer's state. It is read only and the printer's state can be verified and read by calling CheckPrinterStatusAsync. More than one State value can be set and each value can be checked through bitwise operations.

The following values represent the printer's state:

Code	Value	Description
BXL_STS_NORMAL	0	The printer is normal.
BXL_STS_PAPEREMPTY	1	The paper tray is empty.
BXL_STS_COVEROPEN	2	The paper tray cover is open.
BXL_STS_MSR_READY	8	MSR is in the Read Mode. Cannot print.
BXL_STS_ERROR	32	An error has occurred.
BXL_STS_NOT_OPEN	64	The printer is not open.

* Example

```
BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();

var ret = await bxlPrinter.ConnectToPrinterAsync("port info..");
……

Int32 status = await bxlPrinter.CheckPrinterStatusAsync();


if ((status & BXL_STS_PAPEREMPTY) == BXL_STS_PAPEREMPTY)
    ……
if ((status & BXL_STS_COVEROPEN) == BXL_STS_COVEROPEN)
    ……
…….
```

2-4 PowerValue (LONG R)

This attribute sets the printer's battery status. It is read only and set each time the battery status changes. The value can be read using CheckPowerAsync.

The following values represent the battery status:

Code	Value	Description
BXL_PWR_FULL	0	Battery status: 75% or higher
BXL_PWR_HIGH	1	Battery status: 50% or higher
BXL_PWR_MIDDLE	2	Battery status: 10% or higher
BXL_PWR_LOW	3	Battery status: less than 10%

* Example

```
BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();

var ret = await bxlPrinter.ConnectToPrinterAsync("port info..");

……

Int32 power = await bxlPrinter.CheckPowerAsync();

switch (power)
{
    case BXL_PWR_FULL:
        ……
        break;
    case BXL_PWR_HIGH:
        …….
        break;
    case BXL_PWR_MIDDLE:
        ……
        break;
    case BXL_PWR_LOW:
        ……
        break;
}

…….
```

3. Method

The development environment is C#.

3-1 ConnectToPrinterAsync

This function sets the connection to enable communication with the printer.

async Task<Int32> ConnectToPrinterAsync(string inputAddr)

[Parameters]

  * inputAddr
    This is [in] string Data. When connecting via Bluetooth interface, the MAC address of a
    device that is paired to a Windows phone is received while the IP address and port no.
    are received for WiFi connection.

    (Bluetooth: xx:xx:xx:xx:xx:xx, Wireless LAN: xxx.xxx.xxx.xxx:9100)

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns when the function succeeds.
BXL_NOT_OPENED	-101	The communication port cannot be opened.
BXL_CONNECT_ERROR	-105	The connection failed.
BXL_BAD_ARGUMENT	-108	The specified argument is incorrect.

* Example

```
BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();

// serial or bluetooth
var ret = await bxlPrinter.ConnectToPrinterAsync("xx:xx:xx:xx:xx:xx");
……

// WLAN
var ret = await bxlPrinter.ConnectToPrinterAsync ("xxx.xxx.xxx.xxx:9100");
……
```

3-2 ClosePrinter

This function terminates the connection with the printer. The printing and MSR-related features will no longer be available to use.

void ClosePrinter();

[Parameters]

   None

[Return Values]

   None

* Example

```
BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();
……

bxlPrinter.ClosePrinter();
```

3-3 LineFeed

This function performs line feeding to the extent of the integer value conveyed as a factor.

Int32 LineFeed (Int32 nFeed);

[Parameters]

  * nFeed
   [in] The number of lines fed in integer values is delivered as a factor.

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns when the function succeeds.
BXL_STATUS_ERROR	-103	Unable to print.
BXL_WRITE_ERROR	-300	Data transmission failed.

* Example

```
BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();

……

bxlPrinter.LineFeed(5);

……
```

3-4 PrintBarcode

This function supports one-dimensional and two-dimensional barcode printing.

async Task<Int32> PrintBarcode(
    byte[] data, Int32 barcodeType, Int32 height, Int32 width, Int32 alignment, Int32 textPosition)

    or

async Task<Int32> PrintBarcode(
    string data, Int32 barcodeType, Int32 height, Int32 width, Int32 alignment, Int32 textPosition)

[Parameters]

  * Data
    [in] This is byte array or string data which sends barcode data to print.

  * barcodeType
    [in] This defines the type of barcode and is defined in bxlconst.cs.

Barcode	Value	Data Count	Data Range
BXL_BCS_UPCA	101	11 ≤ n ≤ 12	48 ≤ data ≤ 57
BXL_BCS_UPCE	102	11 ≤ n ≤ 12	48 ≤ data ≤ 57
BXL_BCS_EAN13 BXL_BCS_JAN13	104 106	12 ≤ n ≤ 13	48 ≤ data ≤ 57
BXL_BCS_EAN8 BXL_BCS_JAN8	103 105	7 ≤ n ≤ 8	48 ≤ data ≤ 57
BXL_BCS_ITF	107	1 ≤ n ≤ 255 (even number)	48 ≤ data ≤ 57
BXL_BCS_CODABAR	108	1 ≤ n ≤ 255	48 ≤ data ≤ 57, 65 ≤ data ≤ 68, data =36,43,45,46,47,58
BXL_BCS_CODE39	109	1 ≤ n ≤ 255	48 ≤ data ≤ 57, 65 ≤ data ≤ 90, data =32,36,37,43,45,46,47
BXL_BCS_CODE93	110	1 ≤ n ≤ 255	0 ≤ data ≤ 127
BXL_BCS_CODE128	111	2 ≤ n ≤ 255	0 ≤ data ≤ 127
BXL_BCS_PDF417	200	2 ≤ n ≤ 928	0 ≤ data ≤ 255
BXL_BCS_QRCODE	202~3	2 ≤ n ≤ 928	0 ≤ data ≤ 255
BXL_BCS_DATAMATRIX	204	2 ≤ n ≤ 928	0 ≤ data ≤ 255
BXL_BCS_MAXICODE	205~6	2 ≤ n ≤ 928	0 ≤ data ≤ 255

  * Height
    [in] This value sets the height of the barcode and is in the unit of Dot. The value ranges
    1-255. Two-dimensional barcodes are not affected by it.

  * Width
    [in] This value sets the width of the barcode and ranges 2-7. If the printing size
    exceeds the space in the paper, the barcode may not be printed. Two-dimensional
    barcodes are not affected by it.

  * Alignment
    [in] This value sets the alignment of the barcode.

Code	Value	Description
BXL_ALIGNMENT_LEFT	0	Align to left
BXL_ALIGNMENT_CENTER	1	Align to center
BXL_ALIGNMENT_RIGHT	2	Align to right

* TextPosition

[in] This value sets the position of the barcode data. Two-dimensional barcodes only have the BXL_BC_TEXT_NONE value.

Code	Value	Description
BXL_BC_TEXT_NONE	0	Do not print barcode data.
BXL_BC_TEXT_ABOVE	1	Print barcode data above the barcode.
BXL_BC_TEXT_BELOW	2	Print barcode data below the barcode.

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns when the function succeeds.
BXL_WRITE_ERROR	-300	Data transmission failed.
BXL_BC_DATA_ERROR	-500	There is an error in the barcode data.
BXL_BC_NOT_SUPPORT	-501	The barcode type is not supported.

* Example

```
BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();
......
bxlPrinter.PrintBarcode("123456789012", BXL_BCS_UPCA, 50, 2, BXL_ALIGNMENT_LEFT,
            BXL_BC_TEXT_BELOW);

bxlPrinter.PrintBarcode ("123456789012", BXL_BCS_UPCE, 50, 2, BXL_ALIGNMENT_LEFT,
            BXL_BC_TEXT_BELOW);

bxlPrinter.PrintBarcode ("123456789012", BXL_BCS_EAN13, 50, 2, BXL_ALIGNMENT_LEFT,
            BXL_BC_TEXT_BELOW);

bxlPrinter.PrintBarcode ("123456789012", BXL_BCS_JAN13, 50, 2, BXL_ALIGNMENT_LEFT,
            BXL_BC_TEXT_BELOW);

bxlPrinter.PrintBarcode ("12345678", BXL_BCS_EAN8, 50, 2, BXL_ALIGNMENT_LEFT,
            BXL_BC_TEXT_BELOW);

bxlPrinter.PrintBarcode ("12345678", BXL_BCS_JAN8, 50, 2, BXL_ALIGNMENT_LEFT, BXL_BC_TEXT_BELOW);

bxlPrinter.PrintBarcode ("123456789012", BXL_BCS_Code39, 50, 2, BXL_ALIGNMENT_LEFT,
            BXL_BC_TEXT_BELOW);

bxlPrinter.PrintBarcode ("123456789012", BXL_BCS_ITF, 50, 2, BXL_ALIGNMENT_LEFT,
            BXL_BC_TEXT_BELOW);

bxlPrinter.PrintBarcode ("123456789012", BXL_BCS_Codabar, 50, 2, BXL_ALIGNMENT_LEFT,
            BXL_BC_TEXT_BELOW);

bxlPrinter.PrintBarcode ("123456789012", BXL_BCS_Code93, 50, 2, BXL_ALIGNMENT_LEFT,
            BXL_BC_TEXT_BELOW);

bxlPrinter.PrintBarcode ("123456789012", BXL_BCS_Code128, 50, 2, BXL_ALIGNMENT_LEFT,
            BXL_BC_TEXT_BELOW);

bxlPrinter.PrintBarcode ("123456789012", BXL_BCS_PDF417, 0, 2, BXL_ALIGNMENT_LEFT, 0);

bxlPrinter.PrintBarcode ("123456789012", BXL_BCS_QRCODE_MODEL1, 0, 2, BXL_ALIGNMENT_LEFT, 0);

bxlPrinter.PrintBarcode ("123456789012", BXL_BCS_QRCODE_MODEL2, 0, 2, BXL_ALIGNMENT_LEFT, 0);

bxlPrinter.PrintBarcode ("123456789012", BXL_BCS_DATAMATRIX, 0, 2, BXL_ALIGNMENT_LEFT, 0);

bxlPrinter.PrintBarcode ("123456789012", BXL_BCS_MAXICODE_MODE4, 0, 2, BXL_ALIGNMENT_LEFT, 0);
```

3-5 PrintText

This function supports text printing.

long PrintText(byte[] writeData, Int32 alignment, Int32 attribute, Int32 textSize);

or

long PrintText(string writeData, Int32 alignment, Int32 attribute, Int32 textSize);

[Parameters]

　* writeData
　　[in] This Unicode Data is terminated with null character and sends the barcode data to print.

　* alignment
　　[in] This value sets the alignment of the barcode.

Code	Value	Description
BXL_ALIGNMENT_LEFT	0	Align to left
BXL_ALIGNMENT_CENTER	1	Align to center
BXL_ALIGNMENT_RIGHT	2	Align to right

　* attribute
　　[in] This value sets the text attribute. More than one value of the text can be applied.

Code	Value	Description
BXL_FT_DEFAULT	0	Default value Font A, it prints using the default device font.
BXL_FT_FONTB	1	It sets to Font B.
BXL_FT_FONTC	16	It sets to Font C.
BXL_FT_BOLD	2	It adds the bold font style.
BXL_FT_UNDERLINE	4	It adds the 1-dot underline font style.
BXL_FT_UNDERTHICK	6	It adds the 2-dot underline font style.
BXL_FT_REVERSE	8	It adds the reverse font style.

 * textSize
   [in] This value sets the size of the text. Both width and height values can be set.

Code	Value	Description
BXL_TS_0WIDTH	0	Set it to scale width by x1.
BXL_TS_1WIDTH	16	Set it to scale width by x2.
BXL_TS_2WIDTH	32	Set it to scale width by x3.
BXL_TS_3WIDTH	48	Set it to scale width by x4.
BXL_TS_4WIDTH	64	Set it to scale width by x5.
BXL_TS_5WIDTH	80	Set it to scale width by x6.
BXL_TS_6WIDTH	96	Set it to scale width by x7.
BXL_TS_7WIDTH	112	Set it to scale width by x8.

Code	Value	Description
BXL_TS_0HEIGHT	0	Set it to scale height by x1.
BXL_TS_1HEIGHT	1	Set it to scale height by x2.
BXL_TS_2HEIGHT	2	Set it to scale height by x3.
BXL_TS_3HEIGHT	3	Set it to scale height by x4.
BXL_TS_4HEIGHT	4	Set it to scale height by x5.
BXL_TS_5HEIGHT	5	Set it to scale height by x6.
BXL_TS_6HEIGHT	6	Set it to scale height by x7.
BXL_TS_7HEIGHT	7	Set it to scale height by x8.

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns when the function succeeds.
BXL_STATUS_ERROR	-103	Unable to print.
BXL_WRITE_ERROR	-300	Data transmission failed.

* Example

```
BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();

……

bxlPrinter.PrintText("Bixolon Mobile Printer.\n", BXL_ALIGNMENT_LEFT,
                 BXL_FT_DEFAULT, BXL_TS_0WIDTH | BXL_TS_0HEIGHT);
……
```

3-6 PrintImage

This function prints image files. Supported formats vary with different platforms.

async Task<Int32> PrintImage (string fileName, Int32 Width, Int32 Alignment, Int32 Level);

[Parameters]

* fileName
  [in] This is the string for the entire path of an image file. JPG, BMP and GIF are
  supported.

* Width
  [in] This value sets the width of the image file to be converted. The value ranges from
  0 to 384, 576 and 832 for 2, 3 and 4-inch printer respectively.
  The size of the image is adjusted according to the following values:

Code	Value	Description
BXL_WIDTH_FULL	1	The image converts to fit a different size of paper according to the printer size.
BXL_WIDTH_NONE	2	The size of the image remains the same.

* Alignment
  [in] This value sets the alignment of the barcode.

Code	Value	Description
BXL_ALIGNMENT_LEFT	0	Align to left
BXL_ALIGNMENT_CENTER	1	Align to center
BXL_ALIGNMENT_RIGHT	2	Align to right

* Level
  [in] This value sets the color level of the image and ranges 0-100.

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns when the function succeeds.
BXL_STATUS_ERROR	-103	Unable to print.
BXL_WRITE_ERROR	-300	Data transmission failed.
BXL_BITMAPLOAD_ERROR	-400	Unable to read the image file.

* Example

```
BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();

……

bxlPrinter.PrintBitmap(strDirBmp, 300, BXL_ALIGNMENT_CENTER, 50);

……
```

3-7 CheckPrinterStatusAsync

This function checks the printer status (cover open, paper tray empty) and returns the status value.

async Task<Int32> CheckPrinterStatusAsync();

[Parameters]

　None

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns when the function succeeds.
BXL_STATUS_ERROR	-103	Unable to print.
BXL_READ_ERROR	-301	Data reception failed.
BXL_WRITE_ERROR	-300	Data transmission failed.

* Example

```
BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();
……

Int32 status = await bxlPrinter.CheckPrinterStatusAsync();
…...

if ((status & BXL_STS_PAPEREMPTY) == BXL_STS_PAPEREMPTY)
    ……
```

3-8 InitializePrinter

This function resets the current settings to the default values which are set when powered on.

Int32 InitializePrinter();

[Parameters]

  None

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns when the function succeeds.
BXL_STATUS_ERROR	-103	Unable to print.
BXL_WRITE_ERROR	-300	Data transmission failed.

* Example

BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();

……

bxlPrinter.InitializePrinter();

……

3-9 DirectIOAsync

This function can send and read user-defined data.

async Task<byte[]> DirectIOAsync (byte[] writeData, UInt32 readLen);

[Parameters]

　* byte[] writeData,
　　[in] This is the data to be sent to the printer.

　* UInt32 readLen
　　[in] The caller receives the size of the data to be read.

[Return Values]

　　The read data is returned if readLen is 1 or higher, otherwise, it returns null.

* Example

```
BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();

……

byte[] bytes = new byte[3] { 0x1d, 0x49, 67};
byte[] readBuf = new byte[20];

readBuf = await bxlPrinter.DirectIOAsync(bytes, 20);

……
```

3-10 MSRReadReady

This function switches the printer to the MSR Ready mode. The printing is disabled in the MSR Ready mode.
When BXL_SUCCESS is returned, the printer is considered in standby mode.

async Task<Int32> MSRReadReadyAsync();

[Parameters]

   None

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns when the function succeeds.
BXL_NOT_SUPPORT	-107	MSR Settings are incorrect.
BXL_MSRCHECK_ERROR	-110	Failed to check MSR.
BXL_WRITE_ERROR	-300	Data transmission failed.

* Example

```
BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();

……

Task<Int32> task = bxlPrinter.MsrReadReadyAsync();
await task;

if (BXL_SUCCESS != task.Result)
    return;

……
```

3-11 MSRReadCancel

This function deactivates the MSR Ready mode.

async Task<Int32> MSRReadCancel();

[Parameters]

   None

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns when the function succeeds.
BXL_WRITE_ERROR	-300	Data transmission failed.
BXL_MSR_NOTREADY	-602	The printer is not in the MSR READY mode.

* Example

BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();

……

await bxlPrinter.MSRReadCancel();

……

3-12 MSRReadTrack

This function retrieves the MSR data. If the MSR is in the Read Mode and BXL_MSR_DATAEMPTY is returned, it means the card has not been read by the MSR. In this case, scan the card with the MSR or use MSRReadCancel to deactivate the Read Mode.

Int32 MSRReadTrack (ref byte[] Data1, ref byte[] Data2, ref byte[] Data3);

[Parameters]

   * ref byte[] Data1
     [in, out] Retrieves MSR Data Track 1 to the buffer defined by the caller.

   * ref byte[] Data2
     [out, out] Retrieves MSR Data Track 2 to the buffer defined by the caller.

   * ref byte[] Data3
     [out, out] Retrieves MSR Data Track 3 to the buffer defined by the caller.

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns when the function succeeds.
BXL_MSR_DATAEMPTY	-603	No data has been read by MSR.

* Example

```
BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();
……

Task<Int32> task = bxlPrinter.MSRReadReadyAsync();
await task;

if (BXL_SUCCESS != task.Result)
    return;

byte[] track1 = new byte[120];
byte[] track2 = new byte[120];
byte[] track3 = new byte[120];

Int32 ret = bxlPrinter.MSRReadTrack(ref track1, ref track2, ref track3);

if (BXL_MSR_DATAEMPTY == ret)
    ……
else
    ……
```

3-13 MSRReadFullTrack

This function returns the entire MSR data. It returns error if it is not in the MSR Read Mode or no data is found. Use MSRReadCancel to deactivate the Read Mode.

Int32 MSRReadFullTrack (ref byte[] msrData, Int32 dataLen);

[Parameters]

  * ref byte[] msrData
    [in, out] Retrieves MSR Track Data to the buffer defined by the caller.

  * Int32 dataLen
    [in] Sends the size of the buffer defined by the caller.

[Return Values]

  If the function is called successfully, the entire MSR data is returned. If no MSR data is found, BXL_MSR_DATAEMPTY is returned instead. Each track data is classified as 0x1c.
  (Format: [track1 data]0x1c[track2 data]0x1c[track3 data]0x1c)

Code	Value	Description
BXL_SUCCESS	0	Returns when the function succeeds.
BXL_MSR_DATAEMPTY	-603	No data has been read by MSR.

*Example

```
BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();
……

Task<Int32> task = bxlPrinter.MSRReadReadyAsync();
await task;

if (BXL_SUCCESS != task.Result)
    return;

byte[] track = new byte[600];

Int32 ret = bxlPirnter.MSRReadFullTrack(track, track.length);

if (BXL_MSR_DATAEMPTY == ret)
    ……
else
    ……
……
```

3-14 SelectLabelMode

This function selects the Label/Receipt Mode.

Int32 SelectLabelMode(bool enable)

[Parameters]

  * bool enable
    [in] Sends whether the Label Mode is used or not.
    If the value is TRUE, the Label Mode is selected.

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns when the function succeeds.
BXL_STATUS_ERROR	-103	Unable to print.
BXL_WRITE_ERROR	-300	Data transmission failed.

* Example

```
BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();

……

// Select Label Mode
if (bxlPrinter.SelectLabelMode(true) != BXL_SUCCESS)
    return;

bxlPrinter.NextPrintPos();

// Select Receipt Mode
if (bxlPrinter.SelectLabelMode(false) != BXL_SUCCESS)
    return;

……
```

3-15 NextPosition

This function feeds the paper to the next printing position and is only active when it is in the Label Mode.

Int32 NextPosition();

[Parameters]

   None

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns when the function succeeds.
BXL_STATUS_ERROR	-103	Unable to print.
BXL_NOT_SUPPORT	-107	The printer is not in the Label Mode.
BXL_WRITE_ERROR	-300	Data transmission failed.

* Example

```
BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();

……

if (bxlPrinter.SelectLabelMode(true) != BXL_SUCCESS)
    return;

bxlPrinter.NextPosition();

……
```

3-16 AutoCalibration

This function performs Auto Calibration if the printer is in the Label Mode.

Int32 AutoCalibration()

[Parameters]

   None

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns when the function succeeds.
BXL_STATUS_ERROR	-103	Unable to print.
BXL_NOT_ SUPPORT	-107	The printer is not in the Label Mode.
BXL_WRITE_ERROR	-300	Data transmission failed.

* Example

```
BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();

……

if (bxlPrinter.SelectLabelMode(true) != BXL_SUCCESS)
    return;

bxlPrinter.AutoCalibration();

……
```

3-17 SelectPageMode

This function selects whether to use the Page Mode.

Int32 SelectPageMode(BOOL enable)

[Parameters]

* BOOL enable
   [in] Sends whether to use the Page Mode.
   If the value is TRUE, the Page Mode is selected.

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns when the function succeeds.
BXL_STATUS_ERROR	-103	Unable to print.
BXL_WRITE_ERROR	-300	Data transmission failed.

* Example

```
BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();

……

// Select Page Mode
if (bxlPrinter.SelectPageMode(true) != BXL_SUCCESS)
    return;

// Select Standard Mode
if (bxlPrinter.SelectPageMode(false) != BXL_SUCCESS)
    return;

……
```

3-18 FormFeed

This function prints all the data when it is in the Page Mode. After the printer finishes printing, it goes into the Standard Mode.

Int32 FormFeed(Int32 feedCount)

[Parameters]

  * long nFeed
    [in] The amount to be fed.

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns when the function succeeds.
BXL_STATUS_ERROR	-103	Unable to print.
BXL_NOT_ SUPPORT	-107	The printer is not in the Label Mode.
BXL_WRITE_ERROR	-300	Data transmission failed.

* Example

```
BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();

……

// Select Page Mode
if (bxlPrinter.SelectPageMode(true) != BXL_SUCCESS)
    return;

……

bxlPrinter.FormFeed(2);
```

3-19 SetPrintAreaInPM

This function sets the size and position of the printing area when it is in the Page Mode.

Int32 SetPrintAreaInPM (Int32 x, Int32 y, Int32 width, Int32 height)

[Parameters]

　* Int32　x
　　[in] x-coordinate of the printing area

　* Int32　y
　　[in] y-coordinate of the printing area

　* Int32　width
　　[in] width of the printing area

　* Int32　height
　　[in] height of the printing area

　　58mm wide paper: x = 0, y = 0, width = 384, height = 840

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns when the function succeeds.
BXL_STATUS_ERROR	-103	Unable to print.
BXL_NOT_ SUPPORT	-107	The printer is not in the Label Mode.
BXL_BAD_ARGUMENT	-108	The specified argument is incorrect.
BXL_WRITE_ERROR	-300	Data transmission failed.

* Example

```
BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();

……

// Select Page Mode
if (bxlPrinter.SelectPageMode(true) != BXL_SUCCESS)
    return;

bxlPrinter.SetPrintAreaInPM(0, 0, 416, 416);

……

bxlPrinter.FormFeed(2);
```

3-20 SetPrintDirectionInPM

This function sets the direction of printing when it is in the Page Mode.

Int32 SetPrintDirectionInPM (Int32 direction)

[Parameters]

  * Int32 direction

printDirection	Value	Printing Direction	Starting Position	Rotation
BXL_PD_LEFT_TO_RIGHT	48	Left -> Right	Top left	0 degree
BXL_PD_BOTTOM_TO_TOP	49	Bottom -> Top	Bottom left	270 degrees
BXL_PD_RIGHT_TO_LEFT	50	Right -> Left	Bottom right	180 degrees
BXL_PD_TOP_TO_BOTTOM	51	Top -> Bottom	Top right	90 degrees

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns when the function succeeds.
BXL_STATUS_ERROR	-103	Unable to print.
BXL_NOT_ SUPPORT	-107	The printer is not in the Label Mode.
BXL_BAD_ARGUMENT	-108	The specified argument is incorrect.
BXL_WRITE_ERROR	-300	Data transmission failed.

* Example

```
BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();

……

// Select Page Mode
if (bxlPrinter.SelectPageMode(true) != BXL_SUCCESS)
    return;

bxlPrinter.SetPrintAreaInPM(0, 0, 416, 416);
bxlPrinter.SetPrintDirectionInPM(BXL_PD_LEFT_TO_RIGHT);

……

bxlPrinter.FormFeed(2);
```

3-21 SetVerticalPositionInPM

This function sets the (horizontal) starting position for printing when it is in the Page Mode.

Int32 SetVerticalPositionInPM (Int32 position)

[Parameters]

　* Int32 position
　　[in] Sets the position to start printing.

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns when the function succeeds.
BXL_STATUS_ERROR	-103	Unable to print.
BXL_NOT_ SUPPORT	-107	The printer is not in the Label Mode.
BXL_BAD_ARGUMENT	-108	The specified argument is incorrect.
BXL_WRITE_ERROR	-300	Data transmission failed.

* Example

```
BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();

……

// Select Page Mode
if (bxlPrinter.SelectPageMode(true) != BXL_SUCCESS)
    return;

bxlPrinter.SetPrintAreaInPM(0, 0, 416, 416);
bxlPrinter.SetPrintDirectionInPM(BXL_PD_LEFT_TO_RIGHT);

bxlPrinter.SetVerticalPositionInPM(160);
bxlPrinter.SetHorizontalPositionInPM(40);
bxlPrinter.PrintText("Bixolon Mobile Printer.", 0, BXL_FT_DEFAULT, BXL_TS_0WIDTH |
BXL_TS_0HEIGHT);

……

bxlPrinter.PrintDataInPM();

bxlPrinter.FormFeed(2);
```

3-22 SetHorizontalPositionInPM

This function sets the starting position for printing.

Int32 SetHorizontalPositionInPM (Int32 position)

[Parameters]

  * Int32 position
    [in] Sets the position to start printing.

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns when the function succeeds.
BXL_STATUS_ERROR	-103	Unable to print.
BXL_NOT_ SUPPORT	-107	The printer is not in the Label Mode.
BXL_BAD_ARGUMENT	-108	The specified argument is incorrect.
BXL_WRITE_ERROR	-300	Data transmission failed.

* Example

```
BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();

……

// Select Page Mode
if (bxlPrinter.SelectPageMode(true) != BXL_SUCCESS)
    return;

bxlPrinter.SetPrintAreaInPM(0, 0, 416, 416);
bxlPrinter.SetPrintDirectionInPM(BXL_PD_LEFT_TO_RIGHT);

bxlPrinter.SetVerticalPositionInPM(160);
bxlPrinter.SetHorizontalPositionInPM(40);
bxlPrinter.PrintText("Bixolon Mobile Printer.", 0, BXL_FT_DEFAULT, BXL_TS_0WIDTH |
BXL_TS_0HEIGHT);

……

bxlPrinter.PrintDataInPM();

bxlPrinter.FormFeed(2);
```

3-23 PrintDataInPM

This function prints all the data stored in the buffer when it is in the Page Mode.

Int32 PrintDataInPM()

[Parameters]

　None

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns when the function succeeds.
BXL_STATUS_ERROR	-103	Unable to print.
BXL_NOT_ SUPPORT	-107	The printer is not in the Label Mode.
BXL_WRITE_ERROR	-300	Data transmission failed.

* Example

```
BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();

……

// Select Page Mode
if (bxlPrinter.SelectPageMode(true) != BXL_SUCCESS)
    return;

bxlPrinter.SetPrintAreaInPM(0, 0, 416, 416);
bxlPrinter.SetPrintDirectionInPM(BXL_PD_LEFT_TO_RIGHT);

bxlPrinter.SetVerticalPositionInPM(160);
bxlPrinter.SetHorizontalPositionInPM(40);
bxlPrinter.PrintText("Bixolon Mobile Printer.", 0, BXL_FT_DEFAULT, BXL_TS_0WIDTH |
BXL_TS_0HEIGHT);

……

bxlPrinter.PrintDataInPM();

bxlPrinter.FormFeed(2);
```

3-24 ScrPowerUp

This function turns on the SCR and retrieves the ATR (Answer To Reset) data and is only available to use with SPP-R210 SCR.

async Task<SCRData> ScrPowerUp ();

[Parameters]

   None

[Return Values]

   * SCRData class

```
class SCRData
{
     public byte[] ATR { get; set; }
     public UInt32 ATRLen { get; set; }
     public byte[] APDU { get; set; }
     public UInt32 APDULen { get; set; }
     public byte ResponseS { get; set; }
}
```

* Example

```
BxlWinPhoneSDK.ScrData scrData;

BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();

……

scrData = await bxlPrinter.ScrPowerUp();

if (0x00 != scrData.ResponseS)
    return;

……
```

3-25 ScrPowerDown

This function turns off the SCR and is only available to use with SPP-R210 SCR.

async Task<byte> ScrPowerDown ();

[Parameters]

　None

[Return Values]

　* ResponseS
　　[out] If the value is 0x00, it means that the SCR is successfully powered down,
　　otherwise, it fails. Please refer to the Command Manual for more information.

* Example

```
BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();

……

byte ResponseS = await bxlPrinter.ScrPowerDown();

if (0x00 != ResponseS)
    return;

……
```

3-26 ScrOperationMode

This function sets the operating mode and is only available to use with SPP-R210 SCR.

async Task<byte> ScrOperationMode (uint mode);

[Parameters]

  * UINT mode
   [in] Operating mode.

Code	Value	Description
BXL_SCR_MODE_APDU	0	APDU mode
BXL_SCR_MODE_TPDU	1	TPDU mode

[Return Values]

  * ResponseS
   [out] If the value is 0x00, it means that the SCR is successfully powered down, otherwise, it fails. Please refer to the Command Manual for more information.

* Example

```
BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();

……

byte ResponseS = await bxlPrinter.ScrOperationMode(BXL_SCR_MODE_APDU);

if (0x00 != ResponseS)
    return;

……
```

3-27 ScrExchangeAPDU

This function performs APDU/TPDU data communication and is only available to use with SPP-R210 SCR.

async Task<SCRData> ScrExchangeAPDU (byte[] cmdAPDU, uint cmdAPDULen);

[Parameters]

  * byte[] cmdAPDU
    [in] Command APDU data

  * UINT cmdAPDULen
    [in] Command APDU length

[Return Values]

  * SCRData class

```
class SCRData
{
    public byte[] ATR { get; set; }
    public UInt32 ATRLen { get; set; }
    public byte[] APDU { get; set; }
    public UInt32 APDULen { get; set; }
    public byte ResponseS { get; set; }
}
```

* Example

```
byte[] cmdAPDU = new byte[512];
BxlWinPhoneSDK.ScrData scrData;

BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();

……

scrData = await bxlPrinter.ScrExchangeAPDU(CmdAPDU, cmdAPDU.length);

if (0x00 != scrData.ResponseS)
    return;

……
```

3-28 ScrCheckStatus

This function checks the status of the smart card and is only available to use with SPP-R210 SCR.

async Task<SCRStatus> ScrCheckStatus ();

[Parameters]

   None

[Return Values]

   * SCRStatus class

```
class SCRStatus
{
      public byte status { get; set; }
      public byte ResponseS { get; set; }
}
```

* Example

```
BxlWinPhoneSDK.SCRStatus scrStatus;

BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();

……

scrStatus = bxlPrinter.ScrCheckStatus();

if (0x00 != scrStatus.ResponseS)
    return;
……
```

3-29 ScrSelectCard

This function selects the communication with Smart card, SAM1 or SAM2 and is only available to use with SPP-R210 SCR.

async Task<byte> ScrSelectCard (uint card);

[Parameters]

  * UINT card
    [in] smart card for communication

Code	Value	Description
BXL_SCR_SMARTCARD	48	Select Smart card for communication
BXL_SCR_SAM1	49	Select SAM1 for communication
BXL_SCR_SAM2	50	Select SAM2 for communication

[Return Values]

  * ResponseS
    [out] If the value is 0x00, it means that the SCR is successfully powered down, otherwise, it fails. Please refer to the Command Manual for more information.

* Example

```
BxlWinPhoneSDK bxlPrinter = new BxlWinPhoneSDK();

……

byte ResponseS = await bxlPrinter.ScrSelectCard(BXL_SCR_SMARTCARD);

if (0x00 != ResponseS)
    return;
……
```