



ZQ110

Mobile PrinterWindows SDK

API Reference Guide

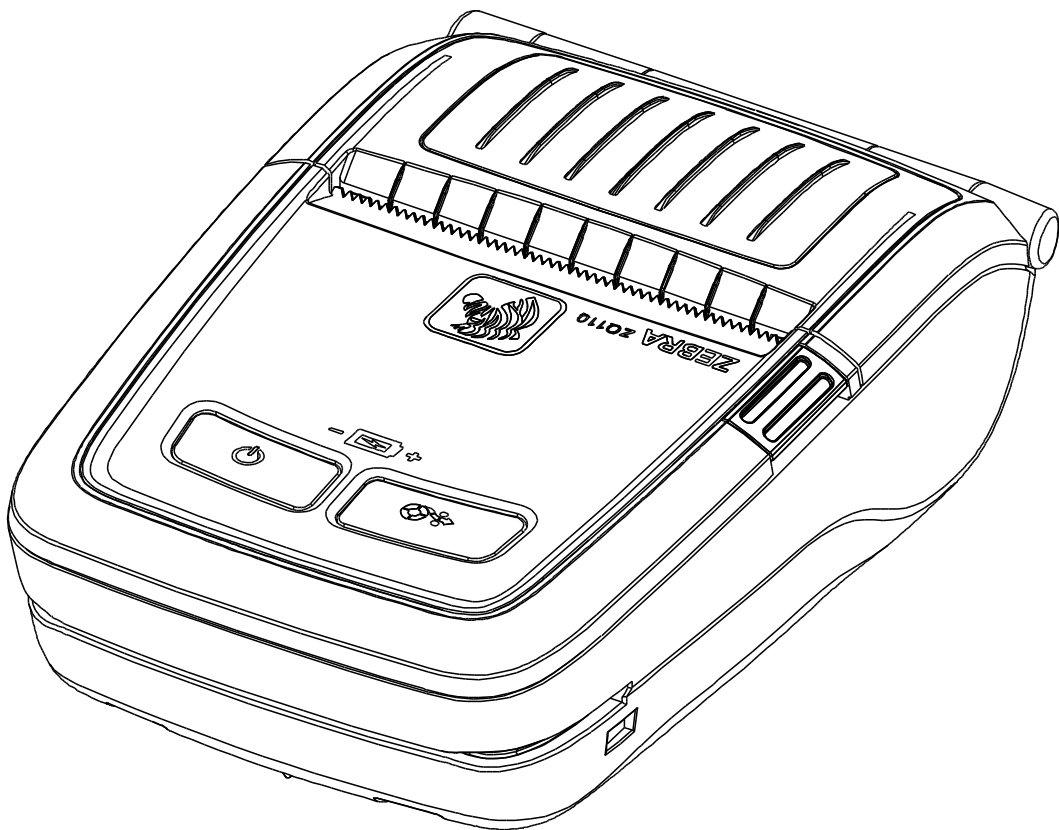


Table of Contents

1. MANUAL INFORMATION	6
1-1 SUPPORTED OPERATING SYSTEM (OS) & PLATFORM.....	6
1-2 SUPPORTING DEVICES.....	6
2. PROPERTIES	7
2-1 CHARACTERSET (LONG R/W)	7
2-2 INTERNATIONAL CHARACTERSET (LONG R/W)	8
2-3 STATE (LONG R).....	8
2-4 POWERVALUE (LONG R)	9
2-5 MsrMsgMode (BOOL W)	9
3. METHODS	10
3-1 PRINTEROPEN	10
3-2 PRINTERCLOSE.....	11
3-3 LINEFEED	11
3-4 PRINTBARCODE	12
3-5 PRINTTEXT	14
3-6 PRINTBITMAP	16
3-7 CHECKPRINTER	17
3-8 MSRREADREADY	17
3-9 MSRREADCANCEL	18
3-10 MSRREADTRACK	18
3-11 DIRECTIO	19
3-12 NEXTPRINTPOS	20
3-13 GETTRACK1	20
3-14 GETTRACK2	20
3-15 GETTRACK3	21
3-16 MSRREADFULLTRACK	21
3-17 MSRREAD1TRACK	22
3-18 MSRREAD2TRACK	22
3-19 MSRREAD3TRACK	23
4. MESSAGE	24
4-1 SETMSRMSGMODE(BOOL BVALUE).....	24
4-2 BXL_LPARAM_MSR.....	25

■ Proprietary Statements

This manual contains proprietary information for Zebra Technologies Corporation. It is intended solely for the information and use of parties operating and maintaining the equipment described herein. Such proprietary information may not be used, reproduced, or disclosed to any other parties for any other purpose without the expressed written permission of Zebra Technologies Corporation.

Product Improvements

Since continuous product improvement is a policy of Zebra Technologies Corporation, all specifications and signs are subject to change without notice.

FCC Compliance Statement

NOTE: This equipment has been tested and found to comply with the limits of a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference with radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and the receiver.
- Connect the equipment to an outlet or circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

WARNING: Exposure to Radio Frequency radiation. To conform to FCC RF exposure requirements this device shall be used in accordance with the operating conditions and instructions listed in this manual.

NOTE: This unit was tested with shielded cables on the peripheral devices. Shielded cables must be used with the unit to ensure compliance.

Changes or modifications to this unit not expressly approved by Zebra Technologies Corporation could void the user's authority to operate this equipment.

Canadian Compliance Statement

This Class B digital apparatus complies with Canadian ICES-003.

Cet appareil numérique de la classe B est conforme à la norme NMB-003 du Canada.

“IC:” before the equipment certification number signifies that the Industry Canada technical specifications were met. It does not guarantee that the certified product will operate to the user’s satisfaction.

Liability Disclaimer

Inasmuch as every effort has been made to supply accurate information in this manual, Zebra Technologies Corporation is not liable for any erroneous information or omissions. Zebra Technologies Corporation reserves the right to correct any such errors and disclaims liability resulting therefrom.

No Liability for Consequential Damage

In no event shall Zebra Technologies Corporation or anyone else involved in the creation, production, or delivery of the accompanying product (including hardware and software) be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or the results of use of or inability to use such product, even if Zebra Technologies Corporation has been advised of the possibility of such damages. Because some states do not allow the exclusion of liability for consequential or incidental damages, the above limitation may not apply to you.

Copyrights

The copyrights in this manual and the label print engine described therein are owned by Zebra Technologies Corporation. Unauthorized reproduction of this manual or the software in the label print engine may result in imprisonment of up to one year and fines of up to \$10,000 (17 U.S.C.506). Copyright violators may be subject to civil liability.

This product may contain ZPL®, ZPL II®, and ZebraLink™ programs; Element Energy Equalizer® Circuit; E3®; and AGFA fonts. Software © ZIH Corp. All rights reserved worldwide.

ZebraLink and all product names and numbers are trademarks, and Zebra, the Zebra logo, ZPL, ZPL II, Element Energy Equalizer Circuit, and E3 Circuit are registered trademarks of ZIH Corp. All rights reserved worldwide.

Monotype®, Intellifont® and UFST® are trademarks of Monotype Imaging, Inc. registered in the United States Patent and Trademark Office and may be registered in certain jurisdictions.

Andy™, CG Palacio™, CG Century Schoolbook™, CG Triumvirate™, CG Times™, Monotype Kai™, Monotype Mincho™ and Monotype Sung™ are trademarks of Monotype Imaging, Inc. and may be registered in some jurisdictions.

HY Gothic Hangul™ is a trademark of Hanyang Systems, Inc.

Angsana™ is a trademark of Unity Progress Company (UPC) Limited.

Andale®, Arial®, Book Antiqua®, Corsiva®, Gill Sans®, Sorts® and Times New Roman® are trademarks of The Monotype Corporation registered in the United States Patent and Trademark Office and may be registered in certain jurisdictions.

Century Gothic™, Bookman Old Style™ and Century Schoolbook™ are trademarks of The Monotype Corporation and may be registered in certain jurisdictions.

HGP Gothic B™ is a trademark of the Ricoh company, Ltd. and may be registered in some jurisdictions.

Univers™ is a trademark of Heidelberger Druckmaschinen AG, which may be registered in certain jurisdictions, exclusively licensed through Linotype Library GmbH, a wholly owned subsidiary of Heidelberger Druckmaschinen AG.

Futura® is a trademark of Bauer Types SA registered in the United States Patent and Trademark Office and may be registered in some jurisdictions.

TrueType® is a trademark of Apple Computer, Inc. registered in the United States Patent and Trademark Office and may be registered in certain jurisdictions.

All other product names are the property of their respective owners.

All other brand names, product names, or trademarks belong to their respective holders.

©2014 ZIH Corp.

1. Manual Information

This SDK Manual provides information on the dll files required for Windows application program development.

1-1 Supported Operating System (OS) & Platform

Supported OS:

- Microsoft Windows CE 5.x
- Microsoft Windows XP
- Microsoft Windows Vista
- Microsoft Windows 7
- Microsoft Windows 8

Supported Platform:

- Pocket PC 2002
- Pocket PC 2003
- Pocket PC 2003 second edition
- Windows Mobile 5
- Windows Mobile 6
- Custom Platform for Armv4i CPU
 - * Supported .Net Framework

1-2 Supporting Devices

This is a list of verified devices. If your PDA CPU type is Armv4 or Armv4i, it is compatible with the module driver.

- Bluebird BIP-6000
- Widely WF35
- HP iPAQ 212

2. Properties

The constant values used in the dll files provided are declared in the BxlConst.h file. The development environment was based on C++.

2-1 CharacterSet (LONG R/W)

As the attribute that defines the Code Page of the printer, the default setting is BXL_CS_437. Use the SetCharacterSet(), GetCharacterSet() functions to set values or recall previously set values.

The Code Pages that can be used are as follows:

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_LATIN9	48	Code page Latin 9
BXL_CS_USER	255	User set page

2-2 International CharacterSet (LONG R/W)

As the attribute that defines the International Character Set of the printer, the default setting is BXL_ICS_USA. Use the SetInterChrSet(), GetInterChrSet() functions to set values or recall previously set values.

The International Character Sets that can be used are as follows:

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

2-3 State (LONG R)

This attribute sets the printer State. It can only be read, and is automatically set when examining the printer state by using the CheckPrinter function. Note: State values can be set in duplication, each value can be confirmed via bit calculation. The values of this attribute can be recalled by using the GetState function.

Printer State values are as follows:

Code	Value	Description
BXL_STS_NORMAL	0	Printer status is normal
BXL_STS_PAPEREMPTY	1	No paper in printer
BXL_STS_COVEROPEN	2	The printer paper cover is open
BXL_STS_POWEROVER	4	The printer battery capacity is insufficient
BXL_STS_MSR_READY	8	Currently in MSR read mode, printing not possible
BXL_STS_PRINTING	16	Currently printing or transmitting data
BXL_STS_ERROR	32	An error has occurred
BXL_STS_NOT_OPEN	64	The printer is not open
BXL_STS_ERROR_OCCUR	128	An error has occurred

2-4 PowerValue (LONG R)

This attribute sets the battery capacity value of the printer. It can only be read, and is set every time the status of the printer battery changes. The values of this attribute can be recalled by using the GetPowerValue function.

Battery Capacity values are as follows:

Code	Value	Description
BXL_PWR_HIGH	0	Battery charge status: 95%
BXL_PWR_MIDDLE	1	Battery charge status: 85% or higher
BXL_PWR_LOW	2	Battery charge status: 50% or higher
BXL_PWR_SMALL	3	Battery charge status: 25% or higher
BXL_PWR_NOT	4	Battery charge status: 25% or lower

2-5 MsrMsgMode (BOOL W)

This function sets MSR Message capacity value of the printer. The default setting is FALSE. Use the SetMsrMsgMode() functions to set values. Default value is FALSE.

Battery Capacity values are as follows:

Code	Value	Description
TRUE	0	Disable MSR Message functions
FALSE	1	Enable MSR Message functions

3. Methods

The functions disclosed in the provided module are declared in the ExportFunc.h file. The development environment was based on C++.

3-1 PrinterOpen

In order to use the module, this function is enabled. For PDA types with different In/Out ports, port settings for each are required. In general, two-way communication is possible with just the strOutPortName setting.

```
long PrinterOpen(
    LPCTSTR strOutPortName,
    Long ITimeout
);
```

[Parameters]

* strOutPortName

[in] Unicode Data ending with null. The virtual Serial Port number and Baudrate of the currently connected device is conveyed for printing. If strOutPortName is NULL, strOutPortName sets to In/Out data processing (for example, COM4:19200).

* ITimeout

[in] Specifies the timeout, in milliseconds, used to calculate the total timeout period for open operations.

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_CREATE_ERROR	102	Failure to create communication target
BXL_NOT_OPENED	101	Unable to open communication port
BXL_STATUS_ERROR	103	Error occurred during status check
BXL_BAD_ARGUMENT	108	Incorrect argument specified
BXL_WRITE_ERROR	300	Failure to transmit data
BXL_READ_ERROR	301	Error in received value or reception failure

3-2 PrinterClose

This function disables the SDK module. Functions related to printing and MSR are no longer available for use.

```
long PrinterClose();
```

[Parameters]

None

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success

3-3 LineFeed

This function dictates line feeding to the integer value conveyed for printing.

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

[Parameters]

* nFeed

[in] The integer value conveys the number of lines for line feeding in the form of printing value.

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	101	Unable to open communication port
BXL_STATUS_ERROR	103	Error occurred during status check
BXL_WRITE_ERROR	300	Failure to transmit data

3-4 PrintBarcode

This function supports the 1-dimension and 2-dimension barcode printing function.

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

[Parameters]

- * Data
[in] ANSI Code Data ending with null. Barcode data to be printed are conveyed.
- * symbology
[in] Defines the barcode type. The barcode type is defined in BxlConst.h.

Code	Value	Data Number Limit	Data Value Range
BXL_BCS_UPCA	101	11 ≤ n ≤ 12	48 ≤ data ≤ 57
BXL_BCS_UPCE	102	11 ≤ n ≤ 12	48 ≤ data ≤ 57
BXL_BCS_EAN13	103	12 ≤ n ≤ 13	48 ≤ data ≤ 57
BXL_BCS_JAN13	104		
BXL_BCS_EAN8	105	7 ≤ n ≤ 8	48 ≤ data ≤ 57
BXL_BCS_JAN8	106		
BXL_BCS_Code39	107	1 ≤ n ≤ 255	48 ≤ data ≤ 57, 65 ≤ data ≤ 90, data =32,36,37,43,45,46,47
BXL_BCS_ITF	108	1 ≤ n ≤ 255 (even number)	48 ≤ data ≤ 57
BXL_BCS_Codabar	109	1 ≤ n ≤ 255	48 ≤ data ≤ 57, 65 ≤ data ≤ 68, data =36,43,45,46,47,58
BXL_BCS_Code93	110	1 ≤ n ≤ 255	0 ≤ data ≤ 127
BXL_BCS_Code128	111	2 ≤ n ≤ 255	0 ≤ data ≤ 127
BXL_BCS_PDF417	200	Refer to Command Manual	0 ≤ data ≤ 255
BXL_BCS_QRCODE	202~3		0 ≤ data ≤ 255
BXL_BCS_DATAMATRIX	204		0 ≤ data ≤ 255
BXL_BCS_GS1_xxxxx	50~64		0 ≤ data ≤ 255
BXL_BCS_AZTEC_xxxxx	210~212		0 ≤ data ≤ 255

- * Height
[in] This value sets the barcode height in Dot units. The value range is 1~255 at one dimensional barcodes and GS1 UCC/EAN-128 barcode or the range is 1~8 at GS1 DataBar barcode. The other two dimensional barcodes are not affected by this value.

*** Width**

[in] This value sets the barcode width according to a value range of 2~7 at one dimensional barcodes or the range is 1~8 at GS1 DataBar barcode and AZTEC barcode. If the print area of the barcode exceeds the printing paper, barcode printing may not be possible.

The other Two dimensional barcodes are not affected by this value.

*** Alignment**

[in] This value sets the barcode alignment.

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

*** Attribute**

- This value sets the text position at one dimensional barcodes.

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

Code	Value	Description
BXL_BC_TEXT_NONE	0	Barcode data not printed
BXL_BC_TEXT_ABOVE	1	Barcode data printed above barcode
BXL_BC_TEXT_BELOW	2	Barcode data printed below barcode

- This values sets height of separator between 2D and 1D barcode symbol at GS1 barcode. The value is 1 or 2.

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	101	Unable to open communication port
BXL_STATUS_ERROR	103	Error occurred during status check
BXL_WRITE_ERROR	300	Failure to transmit data
BXL_BC_DATA_ERROR	500	Error in barcode data
BXL_BC_NOT_SUPPORT	501	Barcode type not supported

3-5 PrintText

This function supports the text printing function.

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

[Parameters]

* Data

[in] Unicode Data ending with null. Conveys the barcode data to be printed.

* Alignment

[in] This value sets the barcode alignment.

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

* Attribute

[in] This value sets the text properties. The following values can be duplicated.

Code	Value	Description
BXL_FT_DEFAULT	0	Font A, print using the basic device font Basic setting value
BXL_FT_FONTB	1	Set to Font B
BXL_FT_BOLD	2	Adds Bold lettering attribute
BXL_FT_UNDERLINE	4	Adds Underlining attribute
BXL_FT_REVERSE	8	Adds Reverse lettering attribute

* TextSize

[in] This value sets the text size properties. The width and height values can be duplicated.

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]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	101	Unable to open communication port
BXL_STATUS_ERROR	103	Error occurred during status check
BXL_WRITE_ERROR	300	Failure to transmit data

3-6 PrintBitmap

This function prints image files. Only bmp files are supported.

```
long PrintBitmap (
LPCTSTR FileName,
long Width,
long Alignment,
long Level
);
```

[Parameters]

* FileName

[in] Expressed as a row of Unicode characters, the full path of the image file is set.

* Width

[in] This value is the width of the image file to be converted. The value range is 0~384. When setting the value as below, the image size is modified to fit conditions.

Code	Value	Description
BXL_WIDTH_FULL	-1	Set to a value of 384, image is fit to full size of paper
BXL_WIDTH_NONE	-2	No variation given to image size

* Alignment

[in] This value sets the barcode alignment.

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

* Level

[in] This value sets the color level of the image. The value range is 0~100.

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	101	Unable to open communication port
BXL_STATUS_ERROR	103	Error occurred during status check
BXL_WRITE_ERROR	300	Failure to transmit data
BXL_BITMAPLOAD_ERROR	400	Failure to read image file

3-7 CheckPrinter

This function checks the printer status and sets the printer status properties.

```
long CheckPrinter ();
```

[Parameters]

None

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	101	Unable to open communication port
BXL_READ_ERROR	301	Failure to receive data
BXL_WRITE_ERROR	300	Failure to transmit data

3-8 MsrReadReady

This function switches the printer to MSR Ready status. Printing is not possible when in ready status. After returning via BXL_SUCCESS and if the State property is BXL_STS_MSR_READY, the standby status is deemed normal.

```
long CheckPrinter ();
```

[Parameters]

None

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	101	Unable to open communication port
BXL_WRITE_ERROR	300	Failure to transmit data

3-9 MsrReadCancel

This function deactivates the MSR Ready status of the printer.

```
Long MsrReadCancel();
```

[Parameters]

None

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	101	Unable to open communication port
BXL_WRITE_ERROR	300	Failure to transmit data
BXLMSR_NOTREADY	602	Not in READY status
BXLMSR_FAILEDMODE	601	In automatic read mode

3-10 MsrReadTrack

This function recalls MSR data. If the MSR is in read mode and BXLMSR_DATAEMPTY is returned, the card has not yet been read in the MSR. Scanning the card in the MSR again may be tried, or the MsrReadCancel function may be used to cancel read mode.

```
Long MsrReadTrack (
    PCHAR Data1,
    PCHAR Data2,
    PCHAR Data3
);
```

[Parameters]

- * PCHAR Data1
[in, out] MSR Data Track 1 read from the buffer defined by the caller.
- * PCHAR Data2
[in, out] MSR Data Track 2 read from the buffer defined by the caller.
- * PCHAR Data3
[in, out] MSR Data Track 3 read from the buffer defined by the caller.

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	101	Unable to open communication port
BXL_WRITE_ERROR	300	Failure to transmit data
BXLMSR_DATAEMPTY	603	Not in READY status

3-11 DirectIO

This function can send and read data defined by the user.

```
Long DirectIO (  
    PCHAR Data,  
    UINT uiWrite,  
    PCHAR pRequet,  
    UINT uiRead  
);
```

[Parameters]

- * PCHAR Data,
[in] Data to be sent to the printer, data is ANSI code
- * UINT uiWrite
[in] Size of data to be sent to the printer
- * PCHAR pRequet
[in, out] Sets the values read from the buffered defined by the caller
- * UINT uiRead
[in, out] Converts the size of data to be read by the caller to the actual size of read data

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	101	Unable to open communication port
BXL_STATUS_ERROR	103	Error occurred during status check
BXL_WRITE_ERROR	300	Failure to transmit data
BXL_READ_ERROR	301	Error in received value or reception failure

3-12 NextPrintPos

If the label function is set, the paper is fed to the next printing position.

```
Long NextPrintPos ();
```

[Parameters]

* None

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	101	Unable to open communication port
BXL_STATUS_ERROR	103	Error occurred during status check
BXL_WRITE_ERROR	300	Failure to transmit data

3-13 GetTrack1

This function returns MSR Track1 data. If the MSR is in read mode and NULL is returned, the card has not yet been read by the MSR. Swipe the card again. MSRReadCancel function can be used to cancel read mode.

```
TCHAR * GetTrack1 ();
```

[Parameters]

* None

[Return Values]

If the function is successful, MSR Track1 data will be returned. If it has failed, the function will return NULL pointer.

3-14 GetTrack2

This function returns MSR Track2 data. If the MSR is in read mode and NULL is returned, the card has not yet been read by the MSR. Swipe the card again. MSRReadCancel function can be used to cancel read mode.

```
TCHAR * GetTrack2 ();
```

[Parameters]

* None

[Return Values]

If the function is successful, MSR Track2 data will be returned. If it has failed, the function will return NULL pointer.

3-15 GetTrack3

This function returns MSR Track3 data. If the MSR is in read mode and NULL is returned, the card has not yet been read by the MSR. Swipe the card again. MSRReadCancel function can be used to cancel read mode.

```
TCHAR * GetTrack3 ();
```

[Parameters]

* None

[Return Values]

If the function is successful, MSR Track2 data will be returned. If it has failed, the function will return NULL pointer.

3-16 MsrReadFullTrack

This function returns MSR Full Track data. If the MSR is in read mode and ERROR Code is returned, the card has not yet been read by the MSR. Swipe the card again. MSRReadCancel function can be used to cancel read mode.

```
long MsrReadFullTrack (PCHAR Track, UINT uiLength);
```

[Parameters]

* PCHAR Track

[in, out] MSR Track data read from the buffer defined by the caller.

If Printer sentinel mode is 0 or 1(None, Static), each Track, Track Data format set as below.

1,2,3 Track Format : [TRACK1 DATA] 0x1C [TRACK2 DATA] 0x1C [TRACK3 DATA]

1,3 Track Format : [TRACK1 DATA] 0x1C 0x1C [TRACK3 DATA]

If Printer sentinel mode is 2(Dynamic), Track Data format set as below.

1,2,3 Track Format : [TRACK1 DATA] [TRACK2 DATA] [TRACK3 DATA]

* UINT uiLength

[in] Size of Track buffer to be sent to the printer

[Return Values]

If the function is successful, MSR Full track data will be returned. If it has failed, the function will return error code and NULL pointer.

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	101	Unable to open communication port
BXL_NOT_SUPPORT	107	If sentinel mode set 2, it isn't support
BXLMSR_DATAEMPTY	603	Not in READY status

3-17 MsrRead1Track

This function returns MSR 1 Track data. If the MSR is in read mode and ERROR Code is returned, the card has not yet been read by the MSR. Swipe the card again. MSRReadCancel function can be used to cancel read mode.

long MsrRead1Track (PCHAR Track, UINT uiLength);

[Parameters]

- * PCHAR Track
[in, out] MSR Track data read from the buffer defined by the caller.
- * UINT uiLength
[in] Size of Track buffer to be sent to the printer

[Return Values]

If the function is successful, MSR 1 track data will be returned. If it has failed, the function will return error code and NULL pointer.

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	101	Unable to open communication port
BXL_NOT_SUPPORT	107	If sentinel mode set 2, it isn't support
BXLMSR_DATAEMPTY	603	Not in READY status

3-18 MsrRead2Track

This function returns MSR 2 Track data. If the MSR is in read mode and ERROR Code is returned, the card has not yet been read by the MSR. Strip the card with the MSR again. MSRReadCancel function can be used to cancel read mode.

long MsrRead2Track (PCHAR Track, UINT uiLength);

[Parameters]

- * PCHAR Track
[in, out] MSR Track data read from the buffer defined by the caller.
- * UINT uiLength
[in] Size of Track buffer to be sent to the printer

[Return Values]

If the function is successful, MSR 2 track data will be returned. If it has failed, the function will return error code and NULL pointer.

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	101	Unable to open communication port
BXL_NOT_SUPPORT	107	If sentinel mode set 2, it isn't support
BXLMSR_DATAEMPTY	603	Not in READY status

3-19 MsrRead3Track

This function returns MSR 3 Track data. If the MSR is in read mode and ERROR Code is returned, the card has not yet been read by the MSR. Swipe the card again. MSRReadCancel function can be used to cancel read mode.

long MsrRead3Track (PCHAR Track, UINT uiLength);

[Parameters]

- * PCHAR Track
[in, out] MSR Track data read from the buffer defined by the caller.
- * UINT uiLength
[in] Size of Track buffer to be sent to the printer

[Return Values]

If the function is successful, MSR 3 track data will be returned. If it has failed, the function will return error code and NULL pointer.

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	101	Unable to open communication port
BXL_NOT_SUPPORT	107	If sentinel mode set 2, it isn't support
BXLMSR_DATAEMPTY	603	Not in READY status

4. Message

If some events have occurred in printer, the SDK will send message through the “RegisterWindowMessage(Win32 API)”.

4-1 SetMsrMsgMode(BOOL bValue)

This function sets MSR Message capacity value of the printer. The default setting is FALSE. Use the SetMsrMsgMode() functions to set values. Default value is false.

Capacity values are as follows.

Code	Value	Description
TRUE	1	Enable MSR Message functions
FALSE	0	Disable MSR Message functions

* Example

```
long IResult;  
IResult = PrinterOpen("portinfo...", 1000);  
.....  
SetMsrMsgMode(TRUE);  
.....
```


4-2 BXL_LPARAM_MSR

This message will be posted when the printer reads MSR Track data from MSR sensor. IParam is set by the following data.

Value	Description
BXL_MSG_TRACK1	0x01
BXL_MSG_TRACK2	0x02
BXL_MSG_TRACK3	0x04

Example) BXL_MSG_TRACK1 + BXL_MSG_TRACK2 = 0x03

wParam = 0

* Example

```

long IResult;

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

.....

SetMsrMsgMode(TRUE);

.....

long lMsrMessage = RegisterWindowsMessage("_MSR_NOTIFY");
MSG msg;
int nRetry = 10;

memset(&message, 0, sizeof(message));

while(PeekMessage(&message, NULL, 0, 0, PM_REMOVE) && nRetry--)
{
    If(msg.message == lMsrMessage)
    {
        if (msg.lParam & BXL_MSG_TRACK1)
            // read track1 data

        if (msg.lParam & BXL_MSG_TRACK2)
            // read track2 data

        if (msg.lParam & BXL_MSG_TRACK3)
            // read track3 data

        break;
    }

    Sleep(100);
}

.....

```