BIXOLON

API Reference Guide

Windows Label SDK

Ver. 2.15

www.bixolon.com

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”.

Table of Contents

Copyright 2

1. Introduction 5

2. Operating System (OS) Environment 6

3. Windows Label SDK Preparation 7

4. Supported Printers 8

5. Sample Program 10

6. Basic API Reference 11
6-1 Connect Printer Ex 11
6-2 Connect Usb SN 13
6-3 Connect Usb SNW 14
6-4 Disconnect Printer 15
6-5 Set Config Of Printer 16
6-6 Set Paper 17
6-7 Set Character set 19
6-8 Get Printer DPI 21
6-9 Clear Buffer 22
6-10 Check Status 23
6-11 Check Status CP 24
6-12 Get Printer Information 25

7. Print API Reference 26
7-1 Print Device Font 27
7-2 Print Device Font W 29
7-3 Print Vector Font 31
7-4 Print Vector Font W 33
7-5 Print True Font 35
7-6 Print True Font W 36
7-7 Print 1D Barcode 37
7-8 Print QR Code 39
7-9 Print PDF417 41
7-10 Print Data Matrix 43
7-11 Print Image Lib 44
7-12 Print Image Lib W 45
7-13 Print Block 46
7-14 Print Circle 47
7-15 Print Direct 48
7-16 Prints 49

8. RFID API Reference **50**

8-1 RFID Trans Position 50
8-2 RFID Setup 51
8-3 RFID Write 52
8-4 RFID Password 53
8-5 RFID Lock 54

1. Introduction

BIXOLON is continually improving the functions and quality of products.

2. Operating System (OS) Environment

The following operating systems are supported for usage.

Microsoft Windows 7 (32bit/64bit)
Microsoft Windows 8 (32bit/64bit)
Microsoft Windows 8.1 (32bit/64bit)
Microsoft Windows 10 (32bit/64bit)
Microsoft Windows 11 (64bit)

3. Windows Label SDK Preparation

You can download the latest version of the Windows Label SDK from our website. (www.bixolon.com)

4. Supported Printers

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

Model	DPI	Max Printable Width	Speed Support	RFID Support
SLP-D220	203 dpi	432 dots	2.5,3,4,5,6ips	NO
SLP-D223	300 dpi	672 dots	2,2.5,3,4ips	NO
SLP-DX220	203 dpi	432 dots	3,4,5,6ips	NO
SLP-DX223	300 dpi	672 dots	2.5,3,4ips	NO
SLP-TX220	203 dpi	432 dots	3,4,5,6ips	NO
SLP-TX223	300 dpi	672 dots	2.5,3,4ips	NO
SLP-D420	203 dpi	864 dots	2.5,3,4,5,6,7ips	NO
SLP-D423	300 dpi	1248 dots	2,2.5,3,4ips	NO
SLP-DX420	203 dpi	864 dots	3,4,5,6,7ips	NO
SLP-DX423	300 dpi	1248 dots	3,4,5ips	NO
SLP-TX420	203 dpi	864 dots	3,4,5,6,7ips	NO
SLP-TX423	300 dpi	1248 dots	3,4,5ips	NO
SLP-T400	203 dpi	864 dots	2.5,3,4,5,6ips	NO
SLP-T403	300 dpi	1248 dots	2,2.5,3,4ips	NO
SLP-T400R	203 dpi	864 dots	2.5,3,4,5,6ips	YES
SLP-T403R	300 dpi	1248 dots	2,2.5,3,4ips	YES
SLP-TX400	203 dpi	864 dots	3,4,5,6,7ips	NO
SLP-TX403	300 dpi	1248 dots	3,4,5ips	NO
SLP-TX400(RFID)	203 dpi	864 dots	3,4,5,6,7ips	YES
SLP-TX403(RFID)	300 dpi	1248 dots	3,4,5ips	YES
SLP-DL410	203 dpi	864 dots	3,4,5ips	NO
SLP-DL413	300 dpi	1248 dots	3,4,5ips	NO
SPP-L3000	203 dpi	576 dots	1,2,3,4,5ips	NO
SPP-L310	203 dpi	576 dots	1,2,3,4,5ips	NO
SPP-L410	203 dpi	832 dots	1,2,3,4ips	NO
SRP-770	203 dpi	832 dots	2.5,3,4,5ips	NO
SRP-770II	203 dpi	832 dots	2.5,3,4,5ips	NO
SRP-770III	203 dpi	832 dots	3,4,5ips	NO
SRP-E770III	203 dpi	832 dots	3,4,5ips	NO
XT5-40(RFID)	203 dpi	832 dots	2 ~ 14ips	YES (Factory Option)
XT5-43(RFID)	300 dpi	1248 dots	2 ~ 10ips	
XT5-46(RFID)	600 dpi	2496 dots	2,3,4,5ips	
XD5-40d	203 dpi	864 dots	3,4,5,6,7ips	NO
XD5-43d	300 dpi	1248 dots	2,3,4,5ips	NO
XL5-40CT	203 dpi	864 dots	3,4,5,6ips	NO

XL5-43CT	300 dpi	1248 dots	2,3,4,5ips	NO
XD5-40t(RFID)	203 dpi	864 dots	3,4,5,6ips	YES (Factory Option)
XD5-43t(RFID)	300 dpi	1248 dots	2,3,4ips	
XM7-40(RFID)	203 dpi	832 dots	1,2,3,4,5ips	NO
XM7-20	203 dpi	384 dots	1,2,3,4,5ips	NO
SRP-S3000 _LABEL	203 dpi	576 dots	1,2,3,4,5ips	NO
XT3-40	203 dpi	864 dots	3,4,5,6,7,8ips	NO
XT3-43	300 dpi	1248 dots	2,3,4,5,6ips	NO
XF3-40	203 dpi	864 dots	3,4,5,6ips	NO
XM7-30	203 dpi	576 dots	1,2,3,4,5ips	NO
XQ-840II	203 dpi	864 dots	3,4,5,6ips	NO
XQ-843II	300 dpi	1248 dots	2,3,4,5ips	NO
XD5-40IId	203 dpi	864 dots	3,4,5,6,7,8ips	YES (Factory Option)
XD5-40IIt	203 dpi	864 dots	3,4,5,6,7,8ips	
XD5-43IId	300 dpi	1248 dots	2,3,4,5,6ips	
XD5-43IIt	300 dpi	1248 dots	2,3,4,5,6ips	
XT6-60	203dpi	1344 dots	2,3,4,5,6,7,8,9,10,11,12ips	Yes
XT6-63	300dpi	1984 dots	2,3,4,5,6,7,8,9,10 Ips	Yes
XM5-30	203dpi	576 dots	1,2,3,4,5,6 ips	Yes

5. Sample Program

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

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

6. Basic API Reference

This manual is based on the C ++ development environment and describes the various APIs needed to develop Windows applications that can control the printer.

6-1 Connect Printer Ex

• Establishes a communication connection with the printer.



- This method must be called first, than any other methods in this manual.
- Only 1:1 communication with PC and a printer is supported. If multiple printers are connected to the PC, communication may not be performed normally.

[Syntax]

int **ConnectPrinterEx**(
    int nInterface,
    LPCSTR szPortName,
    int nBaudRate,
    int nDataBits,
    int nParity,
    int nStopBits
)

[Parameters]

<table>
<tr><td>nInterface</td><td>The printer communication interface.
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>INF_SERIAL</td><td>0</td><td>Serial Printer</td></tr>
<tr><td>INF_PARALLEL</td><td>1</td><td>Parallel Printer</td></tr>
<tr><td>INF_USB</td><td>2</td><td>USB Printer</td></tr>
<tr><td>INF_ETHERNET</td><td>3</td><td>Ethernet Printer</td></tr>
<tr><td>INF_WIFI</td><td>4</td><td>Wi-Fi Printer</td></tr>
<tr><td>INF_BLUETOOTH</td><td>5</td><td>Bluetooth Printer</td></tr>
</table></td></tr>
<tr><td>szPortName</td><td>Port name for Serial, Bluetooth and Parallel, or IP address for network.
* This parameter is only valid for Serial, parallel or network communication.</td></tr>
<tr><td>nBaudRate</td><td>The baud rate for Serial communication, or port number for network.
* This parameter is only valid for Serial, parallel or network communication.</td></tr>
<tr><td>nDataBits</td><td>The number of bits in the bytes transmitted and received
* This parameter is only valid for Serial communication.</td></tr>
<tr><td>nParity</td><td>The parity scheme to be used for Serial communication.
* This parameter is only valid for Serial communication.</td></tr>
<tr><td>nStopBits</td><td>The number of stop bits to be used for Serial communication.
* This parameter is only valid for Serial communication.</td></tr>
</table>

[Return Values]

Code	Value	Description
ERR_CODE_NO_ERROR	0	Success to connect
ERR_CODE_CONNECT	71	No printers available to connect
ERR_CODE_GETNAME	72	Not supported printer
ERR_CODE_OFFLINE	73	The printer is in an error status

6-2 Connect Usb SN

• USB communication with the serial number set on the printer.

- This method must be called first, than any other methods in this manual.
- Only 1:1 communication with PC and a printer is supported. If multiple printers are connected to the PC, communication may not be performed normally.

[Syntax]

int **ConnectUsbSN**(LPCSTR szUsbSN)

[Parameter]

szUsbSN	Serial number

[Return Values]

If the method succeeds, the return value is TRUE.
If the method fails, the return value is FALSE.

6-3 Connect Usb SNW

• USB communication with the serial number set on the printer.



- This method must be called first, than any other methods in this manual.
- Only 1:1 communication with PC and a printer is supported. If multiple printers are connected to the PC, communication may not be performed normally.

[Syntax]

int **ConnectUsbSNW**(LPCWSTR wszUsbSN)

[Parameter]

wszUsbSN	Serial number set to Unicode

[Return Values]

Code	Value	Description
ERR_CODE_NO_ERROR	0	Success to connect
ERR_CODE_CONNECT	71	No printers available to connect
ERR_CODE_GETNAME	72	Not supported printer
ERR_CODE_OFFLINE	73	The printer is in an error status

6-4 Disconnect Printer

• Disconnects the communication with the printer.



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

[Syntax]
bool **DisconnectPrinter** (void);

[Parameters]
None

[Return Values]
If the method succeeds, the return value is TRUE.
If the method fails, the return value is FALSE.

6-5 Set Config Of Printer

• Sets the printer's print speed, density, orientation, back feed, and cutting settings.

[Syntax]

bool **SetConfigOfPrinter**(
int nSpeed,
int nDensity,
int nOrientation,
bool bAutoCut,
int nCuttingPeriod,
bool bBackFeeding
)

[Parameters]

<table>
<tr><td>nSpeed</td><td>Print speed to set (Range: 0 to 13)
* The supported printing speed varies depending on the printer.</td></tr>
<tr><td>nDensity</td><td>Print density to set (Range: 0 to 20)</td></tr>
<tr><td>nOrientation</td><td>Print direction
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>TOP2BOTTOM</td><td>0</td><td>Print from top to bottom</td></tr>
<tr><td>BOTTOM2TOP</td><td>1</td><td>Print from bottom to top</td></tr>
</table></td></tr>
<tr><td>bAutoCut</td><td>Cutting operation after printing a label.</td></tr>
<tr><td>nCuttingPeriod</td><td>Cutting period
* Only valid when ‘bAutoCut’ is TRUE.</td></tr>
<tr><td>bBackFeeding</td><td>TRUE (Fixed)</td></tr>
</table>

[Return Values]

If the method succeeds, the return value is TRUE.
If the method fails, the return value is FALSE.

6-6 Set Paper

• Sets the type and length and height value of the label to be printed.

All units of length/width/margin are dots, and set the appropriate values according to the printer resolution.

		dot	inch	millimeter
dpi	203 dpi	7.992	0.3937	1
	300 dpi	11.811	0.3937	1
	600 dpi	23.622	0.3937	1

[Syntax]

bool **SetPaper**(
int nHorizontalMargin,
int nVerticalMargin,
int nPaperWidth,
int nPaperLength,
int nMediaType,
int nOffset,
int nGapLengthORThicknessOfBlackmark
)

[Parameters]

nHorizontalMargin	Horizontal Margin in dots
nVerticalMargin	Vertical Margin in dots
nPaperWidth	Label Width in dots
nPaperLength	Label Length in dots
nMediaType	Media Type Code / Value / Description GAP / 0 / Gap CONTINUOUS / 1 / Continuous BLACKMARK / 2 / Black mark
nOffset	Offset length in dot between black mark(or gap) and perforation line.
nGapLengthORThicknessOfBlackmark	Gap length or thickness of black line in dots

[Return Values]
If the method succeeds, the return value is TRUE.
If the method fails, the return value is FALSE.

6-7 Set Character set

• Sets the international character set and code table used for printing characters. This method is used for encoding character string and some international character when encoding characters in 'PrintDeviceFont(W)' or 'PrintVectorFont(W)' method.

[Syntax]

bool **SetCharacterset**(
    int nInternationalCharacterSet,
    int nCodepage
)

[Parameters]

nInternationalCharacterSet

International character set

Code	Value	Description
ICS_USA	0	U.S.A
ICS_FRANCE	1	France
ICS_GERMANY	2	Germany
ICS_UK	3	U.K
ICS_DENMARK_I	4	Denmark I
ICS_SWEDEN	5	Sweden
ICS_ITALY	6	Italy
ICS_SPAIN_I	7	Spain I
ICS_NORWAY	8	Norway
ICS_DENMARK_II	9	Denmark II
ICS_JAPAN	10	Japan
ICS_SPAIN_II	11	Spain II
ICS_LATIN_AMERICA	12	Latin America
ICS_KOREA	13	Korea
ICS_SLOVENIA_CROATIA	14	Slovenia/Croatia
ICS_CHINA	15	China

nCodepage

Code page used for encoding characters.

Code	Value	Description
FCP_CP437	0	U.S.A
FCP_CP850	1	Latin 1
FCP_CP852	2	Latin 2
FCP_CP860	3	Portuguese
FCP_CP863	4	Canadian French
FCP_CP865	5	Nordic
FCP_CP1252	6	Latin 1
FCP_CP857	8	Turkish
FCP_CP737	9	Greek

	FCP_CP1250	10	Latin 2
	FCP_CP1253	11	Greek
	FCP_CP1254	12	Turkish
	FCP_CP855	13	Cyrillic
	FCP_CP862	14	Hebrew
	FCP_CP866	15	Cyrillic
	FCP_CP1251	16	Cyrillic
	FCP_CP1255	17	Hebrew
	FCP_CP928	18	Greek
	FCP_CP775	20	Baltic
	FCP_CP1257	21	Baltic
	FCP_CP858	22	Latin I + Euro
	*** For Chinese / Korean / Japanese, the characters are not encoded by the 'nCodepage' parameter, so choose one of the values in the above table.**		

[Return Values]
If the method succeeds, the return value is TRUE.
If the method fails, the return value is FALSE.

6-8 Get Printer DPI

• Gets the dpi (dots per inch) of the printer connected with the PC.

[Syntax]
int **GetPrinterDPI** (void)

[Parameters]
None

[Return Values]
203, 300 or 600 (dpi)

6-9 Clear Buffer

• Clears print buffer and be ready to make a new label.

[Syntax]

bool **ClearBuffer** (void);

[Parameters]

None

[Return Values]

If the method succeeds, the return value is TRUE.
If the method fails, the return value is FALSE.

6-10 Check Status

• Gets the current status of the printer connected with the PC.



If you open and then close the printer cover and call this method, you may get 'ERR_CODE_COVER_OPEN' status until you press one of the buttons on the front of the printer.

[Syntax]

int **CheckStatus** (void)

[Parameters]

None

[Return Values]

Result	Value	Description
ERR_CODE_NO_ERROR	0	Normal (no errors)
ERR_CODE_NO_PAPER	1	No Paper
ERR_CODE_COVER_OPEN	2	Cover Open
ERR_CODE_CUTTER_JAM	3	Cutter jammed
ERR_CODE_TPH_OVER_HEAT	4	TPH overheat
ERR_CODE_AUTO_SENSING	5	Auto-sensing failure (Gap detection error)
ERR_CODE_NO_RIBBON	6	Ribbon end error
ERR_CODE_BOARD_OVER_HEAT	11	Board overheat
ERR_CODE_MOTOR_OVER_HEAT	12	Motor overheat
ERR_CODE_WAIT_LABEL_TAKEN	13	Waiting for the label to be taken
ERR_CODE_UNKNOWN	99	Unknown error

6-11 Check Status CP

• Gets the current status of the printer connected with the PC.

If you open and then close the printer cover and call this method, you may get 'ERR_CODE_COVER_OPEN' status until you press one of the buttons on the front of the printer.

[Syntax]

int **CheckStatusCP**(void)

[Parameters]

None

[Return Values]

Result	Value	Description
ERR_CODE_NO_ERROR	0	Normal (no errors)
ERR_CODE_NO_PAPER	1	No Paper
ERR_CODE_COVER_OPEN	2	Cover Open
ERR_CODE_CUTTER_JAM	3	Cutter jammed
ERR_CODE_TPH_OVER_HEAT	4	TPH overheat
ERR_CODE_AUTO_SENSING	5	Auto-sensing failure (Gap detection error)
ERR_CODE_NO_RIBBON	6	Ribbon end error
ERR_CODE_PRINTING	9	Printing status
ERR_CODE_BOARD_OVER_HEAT	11	Board overheat
ERR_CODE_MOTOR_OVER_HEAT	12	Motor overheat
ERR_CODE_WAIT_LABEL_TAKEN	13	Waiting for the label to be taken
ERR_CODE_MAKE_LABEL	14	Configuring labels in image buffer
ERR_CODE_MAKE_PEELER	15	Printed labels are attached to the peeler
ERR_CODE_STANDBY_OR_RFID	16	RFID standby status
ERR_CODE_RFID_WRITNING	20	RFID writing
ERR_CODE_INVALID_PARAM	60	Invalid parameter
ERR_CODE_CONNECT	71	Connect Fail
ERR_CODE_GETNAME	72	Get name Fail
ERR_CODE_WRITE	74	Write Fail
ERR_CODE_BUFFER_FULL	90	Buffer Full status
ERR_CODE_UNKNOWN	99	Unknown error

6-12 Get Printer Information

• Get Printer Information



If you open and then close the printer cover and call this method, you may get 'ERR_CODE_COVER_OPEN' status until you press one of the buttons on the front of the printer.

[Syntax]

int **GetPrinterInformation** (int nOption, LPSTR pszInfo);

[Parameter]

Int nOption	0: ModelName 2: Firmware Version 4: Serial Number
LPSTR pszInfo	Printer Information

[Return value]

If the method succeeds, the return value is TRUE.
If the method fails, the return value is FALSE.

7. Print API Reference

The following APIs can be used to print text, barcodes, images, and figures on label paper. All printing APIs basically receive x, y coordinate values in dots unit as parameters. Note that the coordinates to be printed differ depending on the printer resolution.



- Call ‘Prints’ method last to start printing the content of the printer buffer.
- When printing character string using ‘PrintDeviceFont(W)’ or ‘PrintVectorFont(W)’, character encoding may be required to print character string correctly. To set the code page to use for character encoding, call the method 'SetCharacterset'.

7-1 Print Device Font

• Prints characters using the printer's built-in bitmap fonts at the specified coordinates.

[Syntax]

bool **PrintDeviceFont**(
int nHorizontalPos,
int nVerticalPos,
int nFontName,
int nHorizontalMulti,
int nVerticalMulti,
int nRotation,
bool bBold,
LPCSTR szText
)

[Parameters]

nHorizontalPos	Horizontal position in dot unit
nVerticalPos	Vertical position in dot unit
nFontName	Font Selection

Font Size	Value	Description
9 × 15	0	ASCII Font
12 × 20	1	
16 × 25	2	
19 × 30	3	
24 × 38	4	
32 × 50	5	
48 × 76	6	
22 × 34	7	
28 × 44	8	
37 × 58	9	
16 × 16 (ASCII 9×15)	97	Korean Font (KS5601)
24 × 24 (ASCII 12×24)	98	
20 × 20 (ASCII 12×20)	99	
26 × 26 (ASCII 16×30)	100	
20 × 26 (ASCII 16×30)	101	
38 × 38 (ASCII 22×34)	102	
24 × 24 (ASCII 12×24)	106	Japanese Font (Shift JIS)
24 × 24 (ASCII 12×24)	109	Chinese Font (GB2312)
24 × 24 (ASCII 12×24)	110	Chinese Font (BIG5)

<table>
<tr><td>nHorizontalMulti</td><td>Multiplier in horizontal direction (Range: 1 to 9)</td></tr>
<tr><td>nVerticalMulti</td><td>Multiplier in vertical direction (Range: 1 to 9)</td></tr>
<tr><td>nRotation</td><td>Rotation
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>ROTATE_0</td><td>0</td><td>No rotation</td></tr>
<tr><td>ROTATE_90</td><td>1</td><td>90 degrees</td></tr>
<tr><td>ROTATE_180</td><td>2</td><td>180 degrees</td></tr>
<tr><td>ROTATE_270</td><td>3</td><td>270 degrees</td></tr>
</table></td></tr>
<tr><td>bBold</td><td>Bold or not</td></tr>
<tr><td>szText</td><td>Text string to print represented by single or multibyte character set.</td></tr>
</table>

[Return Values]

If the method succeeds, the return value is TRUE.
If the method fails, the return value is FALSE.

7-2 Print Device Font W

• Prints characters using the printer's built-in bitmap fonts at the specified coordinates.

[Syntax]

```
bool PrintDeviceFontW(
        int nHorizontalPos,
        int nVerticalPos,
        int nFontName,
        int nHorizontalMulti,
        int nVerticalMulti,
        int nRotation,
        bool bBold,
        LPCWSTR wszText
)
```

[Parameters]

<table>
<tr><td>nHorizontalPos</td><td colspan="3">Horizontal position in dot unit</td></tr>
<tr><td>nVerticalPos</td><td colspan="3">Vertical position in dot unit</td></tr>
<tr><td>nFontName</td><td colspan="3">Font Selection
<table>
<tr><th>Font Size</th><th>Value</th><th>Description</th></tr>
<tr><td>9 × 15</td><td>0</td><td rowspan="9">ASCII Font</td></tr>
<tr><td>12 × 20</td><td>1</td></tr>
<tr><td>16 × 25</td><td>2</td></tr>
<tr><td>19 × 30</td><td>3</td></tr>
<tr><td>24 × 38</td><td>4</td></tr>
<tr><td>32 × 50</td><td>5</td></tr>
<tr><td>48 × 76</td><td>6</td></tr>
<tr><td>22 × 34</td><td>7</td></tr>
<tr><td>28 × 44</td><td>8</td></tr>
<tr><td>37 × 58</td><td>9</td><td rowspan="6">Korean Font (KS5601)</td></tr>
<tr><td>16 × 16 (ASCII 9×15)</td><td>97</td></tr>
<tr><td>24 × 24 (ASCII 12×24)</td><td>98</td></tr>
<tr><td>20 × 20 (ASCII 12×20)</td><td>99</td></tr>
<tr><td>26 × 26 (ASCII 16×30)</td><td>100</td></tr>
<tr><td>20 × 26 (ASCII 16×30)</td><td>101</td></tr>
<tr><td>38 × 38 (ASCII 22×34)</td><td>102</td><td></td></tr>
<tr><td>24 × 24 (ASCII 12×24)</td><td>106</td><td>Japanese Font (Shift JIS)</td></tr>
<tr><td>24 × 24 (ASCII 12×24)</td><td>109</td><td>Chinese Font (GB2312)</td></tr>
<tr><td>24 × 24 (ASCII 12×24)</td><td>110</td><td>Chinese Font (BIG5)</td></tr>
</table></td></tr>
</table>

<table>
<tr><td>nHorizontalMulti</td><td>Multiplier in horizontal direction (Range: 1 to 9)</td></tr>
<tr><td>nVerticalMulti</td><td>Multiplier in vertical direction (Range: 1 to 9)</td></tr>
<tr><td>nRotation</td><td>Rotation
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>ROTATE_0</td><td>0</td><td>No rotation</td></tr>
<tr><td>ROTATE_90</td><td>1</td><td>90 degrees</td></tr>
<tr><td>ROTATE_180</td><td>2</td><td>180 degrees</td></tr>
<tr><td>ROTATE_270</td><td>3</td><td>270 degrees</td></tr>
</table></td></tr>
<tr><td>bBold</td><td>Bold or not</td></tr>
<tr><td>wszText</td><td>Text string to print represented by Unicode code point.</td></tr>
</table>

[Return Values]
If the method succeeds, the return value is TRUE.
If the method fails, the return value is FALSE.

7-3 Print Vector Font

• Prints characters using the printer's built-in vector fonts at the specified coordinates.

[Syntax]

```
bool PrintVectorFont(
        int nHorizontalPos,
        int nVerticalPos,
        LPCSTR szFontSelection,
        int nFontWidth,
        int nFontHeight,
        LPCSTR szRightSideCharSpacing,
        bool bBold,
        bool bReversePrinting,
        bool bItalic,
        int nRotation,
        LPCSTR szTextAlignment,
        int nTextDirection,
        LPCSTR szText
)
```

[Parameters]

nHorizontalPos	Horizontal position in dot unit
nVerticalPos	Vertical position in dot unit
szFontSelection	Font Selection

Code	Value	Description
ASCII	“U”	ASCII Font
KS5601	“K”	Korean Font (KS5601)
BIG5	“B”	Chinese Font (BIG5)
GB2312	“G”	Chinese Font (GB2312)
ShiftJIS	“J”	Japanese Font (Shift JIS)

nFontWidth	Font width in dot unit
nFontHeight	Font height in dot unit
szRightSideCharSpacing	Right-side character spacing in dot unit
bBold	Bold or not
bReversePrinting	Reverse or not
bItalic	Italic or not
nRotation	Rotation

Code	Value	Description
ROTATE_0	0	No rotation
ROTATE_90	1	90 degrees
ROTATE_180	2	180 degrees
ROTATE_270	3	270 degrees

<table>
<tr><td>szTextAlignment</td><td>Text alignment<table><tr><th>Value</th><th>Description</th></tr><tr><td>“L”</td><td>Left Alignment</td></tr><tr><td>“R”</td><td>Right Alignment</td></tr><tr><td>“C”</td><td>Center Alignment</td></tr></table></td></tr>
<tr><td>nTextDirection</td><td>Text direction<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Left to right</td></tr><tr><td>1</td><td>Right to left</td></tr></table></td></tr>
<tr><td>szText</td><td>Text string to print represented by single or multibyte character set.</td></tr>
</table>

[Return Values]
If the method succeeds, the return value is TRUE.
If the method fails, the return value is FALSE.

7-4 Print Vector Font W

• Prints characters using the printer's built-in vector fonts at the specified coordinates.

[Syntax]

```
bool PrintVectorFontW(
        int nHorizontalPos,
        int nVerticalPos,
        LPCSTR szFontSelection,
        int nFontWidth,
        int nFontHeight,
        LPCSTR szRightSideCharSpacing,
        bool bBold,
        bool bReversePrinting,
        bool bItalic,
        Int nRotation,
        LPCSTR szTextAlignment,
        int nTextDirection,
        LPCWSTR wszText
)
```

[Parameters]

<table>
<tr><td>nHorizontalPos</td><td>Horizontal position in dot unit</td></tr>
<tr><td>nVerticalPos</td><td>Vertical position in dot unit</td></tr>
<tr><td>szFontSelection</td><td>Font Selection
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>ASCII</td><td>“U”</td><td>ASCII Font</td></tr>
<tr><td>KS5601</td><td>“K”</td><td>Korean Font (KS5601)</td></tr>
<tr><td>BIG5</td><td>“B”</td><td>Chinese Font (BIG5)</td></tr>
<tr><td>GB2312</td><td>“G”</td><td>Chinese Font (GB2312)</td></tr>
<tr><td>ShiftJIS</td><td>“J”</td><td>Japanese Font (Shift JIS)</td></tr>
</table></td></tr>
<tr><td>nFontWidth</td><td>Font width in dot unit</td></tr>
<tr><td>nFontHeight</td><td>Font height in dot unit</td></tr>
<tr><td>szRightSideCharSpacing</td><td>Right-side character spacing in dot unit</td></tr>
<tr><td>bBold</td><td>Bold or not</td></tr>
<tr><td>bReversePrinting</td><td>Reverse or not</td></tr>
<tr><td>bItalic</td><td>Italic or not</td></tr>
<tr><td>nRotation</td><td>Rotation
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>ROTATE_0</td><td>0</td><td>No rotation</td></tr>
<tr><td>ROTATE_90</td><td>1</td><td>90 degrees</td></tr>
<tr><td>ROTATE_180</td><td>2</td><td>180 degrees</td></tr>
<tr><td>ROTATE_270</td><td>3</td><td>270 degrees</td></tr>
</table></td></tr>
</table>

<table>
<tr><td>szTextAlignment</td><td>Text alignment<table><tr><th>Value</th><th>Description</th></tr><tr><td>“L”</td><td>Left Alignment</td></tr><tr><td>“R”</td><td>Right Alignment</td></tr><tr><td>“C”</td><td>Center Alignment</td></tr></table></td></tr>
<tr><td>nTextDirection</td><td>Text direction<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Left to right</td></tr><tr><td>1</td><td>Right to left</td></tr></table></td></tr>
<tr><td>wszText</td><td>Text string to print represented by Unicode code point.</td></tr>
</table>

[Return Values]

If the method succeeds, the return value is TRUE.
If the method fails, the return value is FALSE.

7-5 Print True Font

• Prints characters using Windows TTF(True type font) at the specified coordinates.

[Syntax]

```
bool PrintTrueFont (
        int nHorizontalPos,
        int nVerticalPos,
        LPCSTR szFontName
        int nFontSize,
        int nRotation,
        bool bItalic,
        bool bBold,
        bool bUnderline,
        LPCSTR szText,
        bool bDataCompression
)
```

[Parameters]

nHorizontalPos	Horizontal position in dot unit
nVerticalPos	Vertical position in dot unit
szFontName	TTF (True type font) name
nFontSize	Font size in point unit
nRotation	Rotation Code / Value / Description ROTATE_0 / 0 / No rotation ROTATE_90 / 1 / 90 degrees ROTATE_180 / 2 / 180 degrees ROTATE_270 / 3 / 270 degrees
bItalic	Italic or not
bBold	Bold or not
Underline	Underline or not
szText	Text string to print represented by single or multibyte character set.
bDataCompression	Data compression or not

[Return Values]

If the method succeeds, the return value is TRUE.
If the method fails, the return value is FALSE.

7-6 Print True Font W

• Prints characters using Windows TTF(True type font) at the specified coordinates.

[Syntax]

```
bool PrintTrueFontW (
        int nHorizontalPos,
        int nVerticalPos,
        LPCWSTR wszFontName
        int nFontSize,
        int nRotation,
        bool bItalic,
        bool bBold,
        bool bUnderline
        LPCWSTR wszText,
bool bDataCompression)
```

[Parameters]

<table>
<tr><td>nHorizontalPos</td><td>Horizontal position in dot unit</td></tr>
<tr><td>nVerticalPos</td><td>Vertical position in dot unit</td></tr>
<tr><td>wszFontName</td><td>TTF (True type font) name represented by Unicode code point</td></tr>
<tr><td>nFontSize</td><td>Font size in point unit</td></tr>
<tr><td>nRotation</td><td>Rotation
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>ROTATE_0</td><td>0</td><td>No rotation</td></tr>
<tr><td>ROTATE_90</td><td>1</td><td>90 degrees</td></tr>
<tr><td>ROTATE_180</td><td>2</td><td>180 degrees</td></tr>
<tr><td>ROTATE_270</td><td>3</td><td>270 degrees</td></tr>
</table></td></tr>
<tr><td>bItalic</td><td>Italic or not</td></tr>
<tr><td>bBold</td><td>Bold or not</td></tr>
<tr><td>bUnderline</td><td>Underline or not</td></tr>
<tr><td>wszText</td><td>Text string to print represented by Unicode code point</td></tr>
<tr><td>bDataCompression</td><td>Data compression or not</td></tr>
</table>

[Return Values]

If the method succeeds, the return value is TRUE.
If the method fails, the return value is FALSE.

7-7 Print 1D Barcode

• Prints an one-dimensional bar code at the specified coordinates.

[Syntax]

```
bool Print1DBarcode (
        int nHorizontalPos,
        Int nVerticalPos,
        int nBarcodeType,
        int nNarrowBarWidth,
        int int nWideBarWidth,
        int nBarcodeHeight,
        int nRotation,
        int nHRI,
        LPCSTR szData
)
```

[Parameters]

<table>
<tr><td>nHorizontalPos</td><td>Horizontal position in dot unit</td></tr>
<tr><td>nVerticalPos</td><td>Vertical position in dot unit</td></tr>
<tr><td>nBarcodeType</td><td>Bar code Symbology
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>CODE39</td><td>0</td><td>Code39</td></tr>
<tr><td>CODE128</td><td>1</td><td>Code128</td></tr>
<tr><td>I2OF5</td><td>2</td><td>Interleaved 2 of 5</td></tr>
<tr><td>CODABAR</td><td>3</td><td>Codabar</td></tr>
<tr><td>CODE93</td><td>4</td><td>Code93</td></tr>
<tr><td>UPC_A</td><td>5</td><td>UPC-A</td></tr>
<tr><td>UPC_E</td><td>6</td><td>UPC-E</td></tr>
<tr><td>EAN13</td><td>7</td><td>EAN13</td></tr>
<tr><td>EAN8</td><td>8</td><td>EAN8</td></tr>
<tr><td>UCC_EAN128</td><td>9</td><td>UCC/EAN128</td></tr>
</table></td></tr>
<tr><td>nNarrowBarWidth</td><td>Width of the narrow bar in dot unit</td></tr>
<tr><td>nWideBarWidth</td><td>Width of the wide bar in dot unit</td></tr>
<tr><td>nBarcodeHeight</td><td>Bar code height in dot unit</td></tr>
<tr><td>nRotation</td><td>Rotation
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>ROTATE_0</td><td>0</td><td>No rotation</td></tr>
<tr><td>ROTATE_90</td><td>1</td><td>90 degrees</td></tr>
<tr><td>ROTATE_180</td><td>2</td><td>180 degrees</td></tr>
<tr><td>ROTATE_270</td><td>3</td><td>270 degrees</td></tr>
</table></td></tr>
</table>

<table>
<tr><td>nHRI</td><td>Printing HRI (Human Readable Interpretation) characters or not
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>HRI_NOT_PRINT</td><td>0</td><td>Not printed</td></tr>
<tr><td>HRI_BELOW_SIZE 1</td><td>1</td><td>Below barcode (size: 1)</td></tr>
<tr><td>HRI_ABOVE_SIZE 1</td><td>2</td><td>Above barcode (size: 1)</td></tr>
<tr><td>HRI_BELOW_SIZE 2</td><td>3</td><td>Below barcode (size: 2)</td></tr>
<tr><td>HRI_ABOVE_SIZE 2</td><td>4</td><td>Above barcode (size: 2)</td></tr>
<tr><td>HRI_BELOW_SIZE 3</td><td>5</td><td>Below barcode (size: 3)</td></tr>
<tr><td>HRI_ABOVE_SIZE 3</td><td>6</td><td>Above barcode (size: 3)</td></tr>
<tr><td>HRI_BELOW_SIZE 4</td><td>7</td><td>Below barcode (size: 4)</td></tr>
<tr><td>HRI_ABOVE_SIZE 4</td><td>8</td><td>Above barcode (size: 4)</td></tr>
</table></td></tr>
<tr><td>szData</td><td>Barcode data represented by single or multibyte character set.</td></tr>
</table>

[Return Values]
If the method succeeds, the return value is TRUE.
If the method fails, the return value is FALSE.

7-8 Print QR Code

• Prints a QR code at the specified coordinates.

[Syntax]

```
bool PrintQRCode (
        int nHorizontalPos,
        int nVerticalPos,
        int nModel,
        int nECCLevel,
        int nQRCodeSize,
        int nRotation,
        LPCSTR szData
)
```

[Parameters]

<table>
<tr><td>nHorizontalPos</td><td>Horizontal position in dot unit</td></tr>
<tr><td>nVerticalPos</td><td>Vertical position in dot unit</td></tr>
<tr><td>nModel</td><td>Model selection
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>QRMODEL_1</td><td>1</td><td>Model 1</td></tr>
<tr><td>QRMODEL_2</td><td>2</td><td>Model 2</td></tr>
</table></td></tr>
<tr><td>nECCLevel</td><td>Error Correction Level
<table>
<tr><th>Code</th><th>Value</th><th>Recovery Rate</th></tr>
<tr><td>QRECCLEVEL_L</td><td>1</td><td>7%</td></tr>
<tr><td>QRECCLEVEL_M</td><td>2</td><td>15%</td></tr>
<tr><td>QRECCLEVEL_Q</td><td>3</td><td>25%</td></tr>
<tr><td>QRECCLEVEL_H</td><td>4</td><td>30%</td></tr>
</table></td></tr>
<tr><td>nQRCodeSize</td><td>QR Code size (Range: 1 to 9)</td></tr>
<tr><td>nRotation</td><td>Rotation
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>ROTATE_0</td><td>0</td><td>No rotation</td></tr>
<tr><td>ROTATE_90</td><td>1</td><td>90 degrees</td></tr>
<tr><td>ROTATE_180</td><td>2</td><td>180 degrees</td></tr>
<tr><td>ROTATE_270</td><td>3</td><td>270 degrees</td></tr>
</table></td></tr>
<tr><td>szData</td><td>QR Code data represented by single or multibyte character set.</td></tr>
</table>

[Return Values]
If the method succeeds, the return value is TRUE.
If the method fails, the return value is FALSE.

7-9 Print PDF417

• Prints a PDF417 code at the specified coordinates.

[Syntax]

```
bool PrintPDF417 (
        int nHorizontalPos,
        int nVerticalPos,
        int nMaxRow,
        int nMaxCol,
        int nECCLevel,
        int nDataType,
        bool bHRI,
        int nOriginPoint,
        int nModuleWidth,
        int nBarHeight,
        int nRotation,
        LPCSTR szData
)
```

[Parameters]

<table>
<tr><td>nHorizontalPos</td><td>Horizontal position in dot unit</td></tr>
<tr><td>nVerticalPos</td><td>Vertical position in dot unit</td></tr>
<tr><td>nMaxRow</td><td>Maximum Row Count (Range: 3 to 90)</td></tr>
<tr><td>nMaxCol</td><td>Maximum Column Count (Range: 1 to 30)</td></tr>
<tr><td>nECCLevel</td><td>Error Correction Level (Range: 0 to 8)</td></tr>
<tr><td>nDataType</td><td>Data Compression Method
<table>
<tr><th>Code</th><th>Value</th><th>Data compression</th></tr>
<tr><td>PDF417_TEXT_TYPE</td><td>0</td><td>2 characters per codeword</td></tr>
<tr><td>PDF417_NUMERIC_TYPE</td><td>1</td><td>2.93 characters per codeword</td></tr>
<tr><td>PDF417_BINARY_TYPE</td><td>2</td><td>1.2 bytes per codeword</td></tr>
</table></td></tr>
<tr><td>bHRI</td><td>Printing HRI (Human Readable Interface) or not.</td></tr>
<tr><td>nOriginPoint</td><td>Barcode Origin point (0 or 1)
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Center of barcode</td></tr>
<tr><td>1</td><td>Upper left corner of barcode</td></tr>
</table></td></tr>
<tr><td>nModuleWidth</td><td>Module Width (Range: 2 to 9)</td></tr>
<tr><td>nBarHeight</td><td>Bar Height (Range: 4 to 99)</td></tr>
</table>

<table>
<tr><td>nRotation</td><td>Rotation<table><tr><th>Code</th><th>Value</th><th>Description</th></tr><tr><td>ROTATE_0</td><td>0</td><td>No rotation</td></tr><tr><td>ROTATE_90</td><td>1</td><td>90 degrees</td></tr><tr><td>ROTATE_180</td><td>2</td><td>180 degrees</td></tr><tr><td>ROTATE_270</td><td>3</td><td>270 degrees</td></tr></table></td></tr>
<tr><td>szData</td><td>PDF417 data represented by single or multibyte character set.</td></tr>
</table>

[Return Values]

If the method succeeds, the return value is TRUE.
If the method fails, the return value is FALSE.

7-10 Print Data Matrix

• Prints a Data matrix at the specified coordinates.

[Syntax]

```
bool PrintDataMatrix (
        int nHorizontalPos,
        int nVerticalPos,
        int nSize,
        bool bReverse,
        int nRotation,
        LPCSTR szData
)
```

[Parameters]

nHorizontalPos	Horizontal position in dot unit
nVerticalPos	Vertical position in dot unit
nSize	Data Matrix Size (Range: 1 to 4)
bReverse	Reverse printing or not
nRotation	Rotation Code / Value / Description ROTATE_0 / 0 / No rotation ROTATE_90 / 1 / 90 degrees ROTATE_180 / 2 / 180 degrees ROTATE_270 / 3 / 270 degrees
szData	Data Matrix data represented by single or multibyte character set.

[Return Values]

If the method succeeds, the return value is TRUE.
If the method fails, the return value is FALSE.

7-11 Print Image Lib

• Prints a graphic image file at the specified coordinates.



Only files with a bitmap (* .BMP) file extension can be printed.

[Syntax]

```
bool PrintImageLib (
        int nHorizontalPos,
        int nVerticalPos,
        LPCSTR szImageFilename,
        int nDither,
        bool bDataCompression
)
```

[Parameters]

nHorizontalPos	Horizontal position in dot unit
nVerticalPos	Vertical position in dot unit
szImageFilename	File path represented by single or multibyte character set.
nDither	Dithering Algorithm

Code	Value	Dithering Algorithm
DITHER_NONE	-1	No Dithering Algorithm
DITHER_1	0	Error Diffusion
DITHER_2	1	Ordered dither
DITHER_3	6	Stevenson-Arce Algorithm
DITHER_4	7	Bayer matrix[16x16 ordered dither]

bDataCompression	Data compression or not

[Return Values]

If the method succeeds, the return value is TRUE.
If the method fails, the return value is FALSE.

7-12 Print Image Lib W

• Prints a graphic image file at the specified coordinates.



Only files with a bitmap (* .BMP) file extension can be printed.

[Syntax]

```
bool PrintImageLibW (
        int nHorizontalPos,
        int nVerticalPos,
        LPCWSTR wszImageFilename,
        int nDither,
        bool bDataCompression
)
```

[Parameters]

<table>
<tr><td>nHorizontalPos</td><td>Horizontal position in dot unit</td></tr>
<tr><td>nVerticalPos</td><td>Vertical position in dot unit</td></tr>
<tr><td>wszImageFilename</td><td>File path represented by Unicode code point</td></tr>
<tr><td>nDither</td><td>Dithering Algorithm
<table>
<tr><th>Code</th><th>Value</th><th>Dithering Algorithm</th></tr>
<tr><td>DITHER_NONE</td><td>-1</td><td>No Dithering Algorithm</td></tr>
<tr><td>DITHER_1</td><td>0</td><td>Error Diffusion</td></tr>
<tr><td>DITHER_2</td><td>1</td><td>Ordered dither</td></tr>
<tr><td>DITHER_3</td><td>6</td><td>Stevenson-Arce Algorithm</td></tr>
<tr><td>DITHER_4</td><td>7</td><td>16x16 Matrix</td></tr>
</table></td></tr>
<tr><td>bDataCompression</td><td>Data compression or not</td></tr>
</table>

[Return Values]

If the method succeeds, the return value is TRUE.
If the method fails, the return value is FALSE.

7-13 Print Block

• Prints a line, block, box or slope (oblique line) at the specified coordinates.

[Syntax]

```
bool PrintBlock (
        int nStartHorizontalPos,
        int nStartVerticalPos,
        int nEndHorizontalPos,
        int nEndVerticalPos,
        int nOption,
        int nThickness
)
```

[Parameters]

<table>
<tr><td>nStartHorizontalPos</td><td>Horizontal start position in dot unit</td></tr>
<tr><td>nStartVerticalPos</td><td>Vertical start position in dot unit</td></tr>
<tr><td>nEndHorizontalPos</td><td>Horizontal end position in dot unit</td></tr>
<tr><td>nEndVerticalPos</td><td>Vertical end position in dot unit</td></tr>
<tr><td>nOption</td><td>
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>LINE_OVER_WRITING</td><td>0</td><td>Line Overwriting</td></tr>
<tr><td>LINE_EXCLUSIVE_OR</td><td>1</td><td>Line Exclusive OR</td></tr>
<tr><td>LINE_DELETE</td><td>2</td><td>Line Exclusive Delete</td></tr>
<tr><td>SLOPE</td><td>3</td><td>Slope (oblique line)</td></tr>
<tr><td>BOX</td><td>4</td><td>Box</td></tr>
</table>
</td></tr>
<tr><td>nThickness</td><td>Thickness of SLOPE or BOX
* This parameter is only valid if ‘nOption’ is SLOPE or BOX.</td></tr>
</table>

[Return Values]

If the method succeeds, the return value is TRUE.
If the method fails, the return value is FALSE.

7-14 Print Circle

• Prints a circle at the specified coordinates.

[Syntax]

```
bool PrintCircle (
        int nStartHorizontalPos,
        int nStartVerticalPos,
        int nCircleSize,
        int nEndVerticalPos,
        int nMultiplier
)
```

[Parameters]

<table>
<tr><td>nStartHorizontalPos</td><td>Horizontal start position in dot unit</td></tr>
<tr><td>nStartVerticalPos</td><td>Vertical start position in dot unit</td></tr>
<tr><td>nCircleSize</td><td>Circle size (Range: 1 to 6)
<table>
<tr><th>Code</th><th>Value</th><th>Width x Height</th></tr>
<tr><td>CIRCLE_SIZE_1</td><td>1</td><td>40 x 40 in dot unit</td></tr>
<tr><td>CIRCLE_SIZE_2</td><td>2</td><td>56 x 56 in dot unit</td></tr>
<tr><td>CIRCLE_SIZE_3</td><td>3</td><td>72 x 72 in dot unit</td></tr>
<tr><td>CIRCLE_SIZE_4</td><td>4</td><td>88 x 88 in dot unit</td></tr>
<tr><td>CIRCLE_SIZE_5</td><td>5</td><td>104 x 104 in dot unit</td></tr>
<tr><td>CIRCLE_SIZE_6</td><td>6</td><td>168 x 168 in dot unit</td></tr>
</table></td></tr>
<tr><td>nMultiplier</td><td>Multiplier (Range: 1 to 4)</td></tr>
</table>

[Return Values]

If the method succeeds, the return value is TRUE.
If the method fails, the return value is FALSE.

7-15 Print Direct

• Sends custom data to the printer.



If there are no methods you want, this method is useful to send custom data to the printer. For the information about commands recognized by the printer, refer to the SLCS Programming manual. This manual can be downloaded from BIXOLON website.

[Syntax]

```
bool PrintDirect (
        LPCSTR szUserDefinedCmd,
        bool bAddCrLf
)
```

[Parameters]

szUserDefinedCmd	Custom text string represented by single or multibyte character set.
bAddCrLf	Adding CR(Carriage return) and LF(Line Feed) or not

[Return Values]

If the method succeeds, the return value is TRUE.
If the method fails, the return value is FALSE.

7-16 Prints

• Starts printing the content of the printer buffer.



This method must be called last, than any other methods in this manual.

[Syntax]

```
bool Prints (
        int nLabelSet,
        int nCopiesOfEachLabel
)
```

[Parameters]

nLabelSet	The number of label sets (Range: 1 to 65535)
nCopiesOfEachLabel	The number of copies of each label (Range: 1 to 65535)

[Return Values]

If the method succeeds, the return value is TRUE.
If the method fails, the return value is FALSE.

8. RFID API Reference

The following APIs are available only for printers equipped with RFID modules. Please refer to page 4 of the manual to check if RFID is supported.

8-1 RFID Trans Position

• Set the coding position(read/write position of the transponder) of the RFID label on the printer and printing

[Syntax]

bool RFIDTransPosition (int transPosition)

[Parameters]

transPosition	vertical position: Default value(320)

[Return Values]

If the method succeeds, the return value is TRUE.
If the method fails, the return value is FALSE.

8-2 RFID Setup

• Sets the RFID transponder type, number of coding(write/read) retries, number of labels upon retry, and sending/receiving power.

[Syntax]

```
bool RFIDSetup (
        int nRFIDType,
        int nNumberOfRetries,
        int nNumberOfLabel,
        int nRadioPower
)
```

[Parameters]

<table>
<tr><td>nRFIDType</td><td>RFID Transponder Type
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>NONE</td><td>0</td><td>None</td></tr>
<tr><td>ISO18000_6TYPEA</td><td>1</td><td>ISO 18000-6 Type A</td></tr>
<tr><td>ISO18000_6TYPEB</td><td>2</td><td>ISO 18000-6 Type B</td></tr>
<tr><td>EPC_CLASS0</td><td>3</td><td>EPC Class 0</td></tr>
<tr><td>EPC_CLASS1</td><td>4</td><td>EPC Class 1</td></tr>
<tr><td>EPC_CLASS1_GENERATION2</td><td>5</td><td>EPC Class Generation 2</td></tr>
</table></td></tr>
<tr><td>nNumberOfRetries</td><td>The number of coding retries upon coding failure (Write/Read Retries) (Range: 1 to 10)</td></tr>
<tr><td>nNumberOfLabel</td><td>The number of labels upon retry following RFID label writing failure (Range: 1 to 5)</td></tr>
<tr><td>nRadioPower</td><td>Sending/Receiving power adjustment (Range: 0 to 30)</td></tr>
</table>

[Return Values]

If the method succeeds, the return value is TRUE.
If the method fails, the return value is FALSE.

8-3 RFID Write

• Writes data on RFID labels.

[Syntax]

```
bool RFIDWrite (
        int nDataType,
        int nStartingBlockNumber,
        int nWriteByte,
        LPCSTR szData
)
```

[Parameters]

<table>
<tr><td>nDataType</td><td>Data type
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>RFID_ASCII</td><td>0</td><td>ASCII Format</td></tr>
<tr><td>RFID_HEXADECIMAL</td><td>1</td><td>Hexadecimal Format</td></tr>
</table></td></tr>
<tr><td>nStartingBlockNumber</td><td>Starting Block Number (Range: 4 to 10)</td></tr>
<tr><td>nWriteByte</td><td>The number of bytes for writing. The number of bytes must be designated in units of 2 bytes.</td></tr>
<tr><td>szData</td><td>Text string in ASCII format or hexadecimal format</td></tr>
</table>

[Return Values]

If the method succeeds, the return value is TRUE.
If the method fails, the return value is FALSE.

8-4 RFID Password

• Sets the RFID access password and kill password.

[Syntax]

```
bool RFIDPassword (
        LPCSTR szOldAccessPwd,
        LPCSTR szOldKillPwd,
        LPCSTR szNewAccessPwd,
        LPCSTR szNewKillPwd
)
```

[Parameters]

OldAccessPwd	8 characters in hexadecimal format (4 bytes for old Access Password)
OldKillPwd	8 characters in hexadecimal format (4 bytes for old Kill Password)
NewAccessPwd	8 characters in hexadecimal format (4 bytes for new Access Password)
NewKillPwd	8 characters in hexadecimal format (4 bytes for new Kill Password)

[Return Values]

If the method succeeds, the return value is TRUE.
If the method fails, the return value is FALSE.

8-5 RFID Lock

• Locks kill, access, and EPC data via the access password.



This method must be called after 'RFIDPassword' method.

[Syntax]
bool RFIDLock (void)

[Parameters]
None

[Return Values]
If the method succeeds, the return value is TRUE.
If the method fails, the return value is FALSE.

Revision history

Ver.	Date	Description
2.00	12-03-2018	New
2.01	16-05-2018	Information of supported printers modified.
2.02	18-01-2019	- Information of supported printers modified. - Info missing to 'SetPaper' API added.
2.03	11-02-2019	- Information of supported printers added. (XD3-40d) - Corrected typos
2.04	21-06-2019	- Information of supported printers added. (XD3-40t, XL5-40, XL5-43, XD5-40d, XD5-43d) - PrintTrueFont, PrintTrueFontW API Modifying 'phrase
2.05	02-10-2019	- Information of supported printers added. - Description changed for following methods. : SetConfigOfPrinter, ConnectPrinterEx : CheckStatus : PrintDeviceFont, PrintDeviceFontW
2.06	18-11-2019	Information of supported printers modified. (XD5-40t, XD5-43t, XM7-40) Add method : RFIDTransPosition
2.07	28-02-2020	- Information of supported printers added : XM7-20
2.08	06-11-2020	- Information of supported OS modified - Information of supported RFID printers added - Information of supported SRP-S3000_LABEL added
2.09	24-08-2021	- Information of supported printers added (XT3-40, XT3-43) - Information of supported printers modified (XM7-40R RFID option added)
2.10	13-04-2022	- Information of supported printers added (XF3-40) - Information of supported printers added (XM7-30)
2.11	22-09-2022	- CheckStatusCP function added
2.12	08-03-2023	- CheckStatusCP status added
2.13	31-01-2024	- ConnectUsbSN function added - ConnectUsbSNW function added - Information of supported printers added (XQ-840II, XQ-843II)
2.14	18-10-2024	- Information of supported printers added (XD5-40IId/40IIt/43IId/43IIt)
2.15	10-03-2025	- Information of supported printers added (XT6-60/63, XM5-30)
