Windows Label SDK

PV3/PV4

BARCODE PRINTER

Ver. 1.01

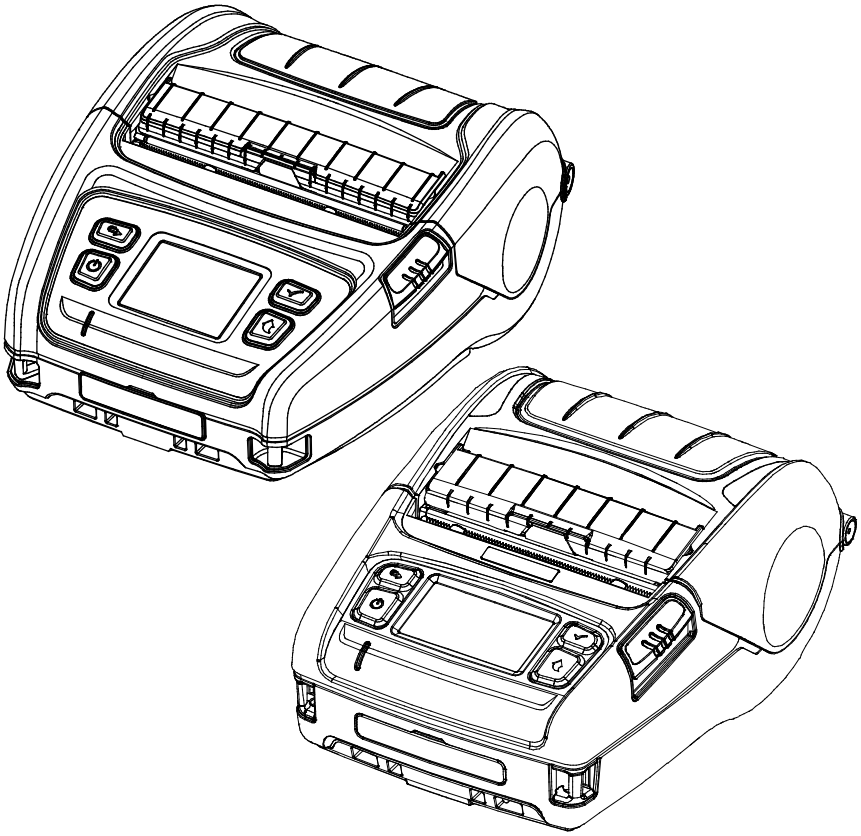

Table of Contents

1. Operating System (OS) Environment

The following operating systems are supported for usage.

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

2. Windows Label SDK Preparation

The Windows Label SDK is included in the enclosed CD, and the latest file version can be downloaded from SATO website. (http://www.satoworldwide.com/)

3. Supported Printers

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

Model	DPI	Max Printable Width	Supported Speed
PV3	203 dpi	576 dots	1,2,3,4,5ips
PV4	203 dpi	832 dots	1,2,3,4,5ips

4. 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)

5. 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.

5-1 ConnectPrinter

- Establishes a communication connection with the printer.

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

[Syntax]

```
int ConnectPrinter(
        int nInterface,
        LPCSTR szPortName,
        int nBaudRate,
        int nDataBits,
        int nParity,
        int nStopBits,
        int nFlowControl
)
```

[Parameters]

<table>
<tr><td>nInterface</td><td>The printer communication interface.
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>INF_SERIAL</td><td>0</td><td>Serial Printer</td></tr>
<tr><td>INF_USB</td><td>2</td><td>USB 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 or IP address for network.
* This parameter is only valid for serial 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 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>
<tr><td>nFlowControl</td><td>The flow control method for serial communication.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Hardware</td></tr>
<tr><td>2</td><td>None</td></tr>
</table>
* This parameter is only valid for Serial communication.</td></tr>
</table>

[Return Value]

Code	Value	Description
ERR_CODE_NO_ERROR	0	No error (Success to connect)
ERR_CODE_WAIT_FOR_LABEL_TAKEN	2	Waiting for paper to be taken
ERR_CODE_BOARD_OVER_HEAT	4	Board overheat
ERR_CODE_AUTO_SENSING	8	Gap Detection Error(Auto-sensing failure)
ERR_CODE_TPH_OVER_HEAT	16	Thermal Head(TPH) overheat.
ERR_CODE_MOTOR_OVER_HEAT	32	Motor overheat
ERR_CODE_COVER_OPEN	64	Cover Open
ERR_CODE_PAPER_EMPTY	128	Paper Empty
ERR_CODE_PAUSED_LABEL	256	Issued label is paused in peeler unit
ERR_CODE_PRINTING_LABEL	512	On printing label in image buffer
ERR_CODE_BUILDING_LABEL	1024	On building label to be printed in the buffer
ERR_CODE_UNABLE_CONNECT	-1	Unable to connect the printer
ERR_CODE_UNKNOWN	-2	Unknown Error

5-2 DisconnectPrinter

- Disconnects the communication with the printer.

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

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

[Parameters]
None

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

5-3 SetConfigOfPrinter

- 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 (PV3 range: 0 to 4, PV4 range: 0 to 3)
* The supported printing speed varies depending on the printer.</td></tr>
<tr><td>nDensity</td><td>Print density (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>FALSE (Fixed)</td></tr>
<tr><td>nCuttingPeriod</td><td>0 (Fixed)</td></tr>
<tr><td>bBackFeeding</td><td>TRUE (Fixed)</td></tr>
</table>

[Return Value]

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

5-4 SetPaper

- Sets the type, width and length value of the label to be printed.

Note

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

dpi		dot	inch	millimeter
dpi	203 dpi	7.992	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 * Max print width of PV3: 576 * Max print width of PV4: 864
nPaperLength	Label length in dots * Max print length: 8,000
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 Value]

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

5-5 SetCharacterset

• 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 select one of the values in the above table.**

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

5-6 GetPrinterDPI

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

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

[Parameters]
None

[Return Value]
203 (dpi)

5-7 ClearBuffer

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

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

[Parameters]
None

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

5-8 CheckStatus

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

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

[Parameters]
None

[Return Value]

Code	Value	Description
ERR_CODE_NO_ERROR	0	No error (Success to connect)
ERR_CODE_WAIT_FOR_LABEL_TAKEN	2	Wait for paper to be taken
ERR_CODE_BOARD_OVER_HEAT	4	Board overheat
ERR_CODE_AUTO_SENSING	8	Gap Detection Error(Auto-sensing failure)
ERR_CODE_TPH_OVER_HEAT	16	Thermal Head(TPH) overheat.
ERR_CODE_MOTOR_OVER_HEAT	32	Motor overheat
ERR_CODE_COVER_OPEN	64	Cover Open
ERR_CODE_PAPER_EMPTY	128	Paper Empty
ERR_CODE_PAUSED_LABEL	256	Issued label is paused in peeler unit
ERR_CODE_PRINTING_LABEL	512	On printing label in image buffer
ERR_CODE_BUILDING_LABEL	1024	On building label to be printed in the buffer
ERR_CODE_UNKNOWN	-2	Unknown Error

6. 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.

Note	- 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'.

6-1 PrintDeviceFont

- 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]

<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>nFontName</td><td>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 rowspan="3">Japanese Font (Shift JIS)</td></tr>
<tr><td>24 × 24 (ASCII 12×24)</td><td>106</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>
<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 Value]

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

6-2 PrintDeviceFontW

- 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>Horizontal position in dot unit</td></tr>
<tr><td>nVerticalPos</td><td>Vertical position in dot unit</td></tr>
<tr><td>nFontName</td><td>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></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>
<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 Value]

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

6-3 PrintVectorFont

• 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]

<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>
<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 Value]
If the method succeeds, the return value is TRUE.
If the method fails, the return value is FALSE.

6-4 PrintVectorFontW

- 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>
<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>
</table>

nTextDirection	Text direction Value: 0 = Left to right; 1 = Right to left
wszText	Text string to print represented by Unicode code point.

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

6-5 PrintTrueFont

- 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: ROTATE_0, Value: 0, Description: 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 Value]
If the method succeeds, the return value is TRUE.
If the method fails, the return value is FALSE.

6-6 PrintTrueFontW

• 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]

nHorizontalPos	Horizontal position in dot unit
nVerticalPos	Vertical position in dot unit
wszFontName	TTF (True type font) name represented by Unicode code point
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
bUnderline	Underline or not
wszText	Text string to print represented by Unicode code point
bDataCompression	Data compression or not

[Return Value]

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

6-7 Print1DBarcode

- 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>
<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_SIZE1</td><td>1</td><td>Below barcode (size: 1)</td></tr>
<tr><td>HRI_ABOVE_SIZE1</td><td>2</td><td>Above barcode (size: 1)</td></tr>
<tr><td>HRI_BELOW_SIZE2</td><td>3</td><td>Below barcode (size: 2)</td></tr>
<tr><td>HRI_ABOVE_SIZE2</td><td>4</td><td>Above barcode (size: 2)</td></tr>
<tr><td>HRI_BELOW_SIZE3</td><td>5</td><td>Below barcode (size: 3)</td></tr>
<tr><td>HRI_ABOVE_SIZE3</td><td>6</td><td>Above barcode (size: 3)</td></tr>
<tr><td>HRI_BELOW_SIZE4</td><td>7</td><td>Below barcode (size: 4)</td></tr>
<tr><td>HRI_ABOVE_SIZE4</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 Value]

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

6-8 PrintQRCode

- 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 Value]

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

6-9 PrintPDF417

• 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>
<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 Value]

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

6-10 PrintDataMatrix

• 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	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 Value]

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

6-11 PrintBitmapFile

- Prints a graphic image file at the specified coordinates.

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

[Syntax]

```
bool PrintBitmapFile (
        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 Value]

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

6-12 PrintBitmapFileW

- Prints a graphic image file at the specified coordinates.

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

[Syntax]

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

[Parameters]

nHorizontalPos	Horizontal position in dot unit
nVerticalPos	Vertical position in dot unit
wszImageFilename	File path represented by Unicode code point
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 / 16x16 Matrix
bDataCompression	Data compression or not

[Return Value]

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

6-13 PrintBlock

• 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]

nStartHorizontalPos	Horizontal start position in dot unit
nStartVerticalPos	Vertical start position in dot unit
nEndHorizontalPos	Horizontal end position in dot unit
nEndVerticalPos	Vertical end position in dot unit
nOption	<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>
nThickness	Thickness of SLOPE or BOX *** This parameter is only valid if 'nOption' is SLOPE or BOX.**

[Return Value]

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

6-14 PrintCircle

• Prints a circle at the specified coordinates.

[Syntax]

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

[Parameters]

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

[Return Value]

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

6-15 PrintDirect

• Sends custom data to the printer.

Note	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.

[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 Value]

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

6-16 Prints

- Starts printing the content of the printer buffer.

Note	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 Value]

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

Copyright

SATO maintains ongoing efforts to enhance and upgrade the functions and quality of all our products.
In the following, product specifications and/or user manual content may be changed without prior notice.

Caution

Some semiconductor devices are easily damaged by static electricity. You should turn the printer “OFF”, before you connect or remove the cables on the rear side, in order to guard the printer against the static electricity. If the printer is damaged by the static electricity, you should turn the printer “OFF”.

Revision History

Rev.	Date	Page	Description
1.00	13.11.19	-	New
1.01	21.04.20	-	Added PV4
		3	Change ‘Supported Speed’ of PV4