# API Reference Guide

# Windows POS SDK

## Ver. 1.01

# AP3

# Table of Contents

# 1. Operating System (OS) Environment

The following operating systems are supported for usage.

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

# 2. Supported Printers

| Product | Model | DPI | Max Printable Width |
|---|---|---|---|
| Thermal Printer | AP3 | 180 dpi | 512 dots |
| Thermal Printer | AP3-3 | 203 dpi | 576 dots |

# 3. Sample Program

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

Visual C# (Visual Studio 2008)
Visual C++ (Visual Studio 2008)

# 4. Printer Methods

This manual is based on the C ++ development environment and describes the various APIs needed to develop Windows applications that can control the printer. The methods and constant values disclosed in the provided are declared in the PrnAPIs.h and PrnConstants.h file.

## 4-1 PrinterOpen

• Establishes a communication connection with the printer.

| | |
|---|---|
| ⚠Note | - This method must be called first, than any other methods in this manual.<br>- 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]**

```
long PrinterOpen(
   int nInterface,
   LPCSTR szPortName,
   nBaudRate,
   nDataBits,
   nParity,
   nStopBits
);
```

**[Parameters]**

| | |
|---|---|
| nInterface | The printer communication interface.<br>Code / Value / Description:<br>_IF_SERIAL / 0 / Serial Printer<br>_IF_PARALLEL / 1 / Parallel Printer<br>_IF_USB / 3 / USB Printer<br>_IF_LAN / 4 / Ethernet Printer |
| szPortName | Port name for Serial and Parallel, or IP address for network.<br>*** This parameter is only valid for Serial, parallel or network communication.** |
| nBaudRate | The baud rate for Serial communication, or port number for network.<br>*** This parameter is only valid for Serial, parallel or network communication.** |
| nDataBits | The number of bits in the bytes transmitted and received<br>*** This parameter is only valid for Serial communication.** |
| nParity | The parity scheme to be used for Serial communication.<br>*** This parameter is only valid for Serial communication.** |
| nStopBits | The number of stop bits to be used for Serial communication.<br>*** This parameter is only valid for Serial communication.** |

**[Return Values]**

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

* Example

```
long lResult = PrinterOpen(_IF_SERIAL, “COM1”, 9600, 8, 0, 0);
//long lResult = PrinterOpen(_IF_USB, “”, 0, 0, 0, 0);
//long lResult = PrinterOpen(_IF_LAN,”192.168.1.1”, 9100, 0, 0, 0); // Network
```

## 4-2 PrinterClose

• Disconnects the communication with the printer.

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

**[Syntax]**
long **PrinterClose**();

**[Parameters]**
None

**[Return Values]**
If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
long lResult = PrinterOpen("portinfo…", 1000);
……
PrinterClose();
……
```

## 4-3 LineFeed

• Sends line-feeds commands to the printer.

**[Syntax]**
long **LineFeed** (
    int nFeed,
);

**[Parameters]**

| | |
|---|---|
| nFeed | Paper feed amount in lines. (Range: 1 ≤ nFeed ≤ 255) |

**[Return Values]**
If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
long lResult = PrinterOpen("portinfo…", 1000);
……
lResult = LineFeed(10);
……
```

## 4-4 SetCharacterSet

• Sets the code table used for printing characters.

**[Syntax]**

long **SetCharacterSet** (
  long Codpage
);

**[Parameters]**

Codepage: Code page used for encoding characters.

| Code | Value | Description |
|---|---|---|
| _CS_PC437 | 0 | USA, Standard Europe |
| _CS_PC850 | 2 | Multilingual Latin 1 |
| _CS_PC860 | 3 | Portuguese |
| _CS_PC863 | 4 | Canadian-French |
| _CS_PC865 | 5 | Nordic |
| _CS_WPC1252 | 16 | Latin 1 |
| _CS_PC866 | 17 | Cyrillic #2 |
| _CS_PC852 | 18 | Latin 2 |
| _CS_PC858 | 19 | Euro |
| _CS_PC862 | 21 | Hebrew DOS code |
| _CS_THAI42 | 23 | THAI 42 |
| _CS_WPC1253 | 24 | Greek |
| _CS_WPC1254 | 25 | Turkish |
| _CS_WPC1257 | 26 | Baltic |
| _CS_WPC1251 | 28 | Cyrillic |
| _CS_PC737 | 29 | Greek |
| _CS_PC775 | 30 | Baltic |
| _CS_THAI14 | 31 | THAI 14 |
| _CS_WPC1255 | 33 | Hebrew New Code |
| _CS_THAI11 | 34 | THAI 11 |
| _CS_THAI18 | 35 | THAI 18 |
| _CS_PC855 | 36 | Cyrillic |
| _CS_PC857 | 37 | Turkish |
| _CS_PC928 | 38 | Greek |
| _CS_THAI16 | 39 | THAI 16 |
| _CS_PC1258 | 41 | Vietnamese |
| _CS_PC1250 | 47 | Czech |
| _CS_LATIN9 | 48 | Latin 9 |
| _CS_TCVN | 49 | Vietnamese (TCVN) |
| _CS_TCVN_CAPITAL | 50 | Vietnamese (TCVN CAPITAL) |
| _CS_TCVN_VISCII | 51 | Vietnamese (VISCII) |

**[Return Values]**

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

* Example

```
long lResult = PrinterOpen("portinfo…", 1000);
……
SetCharacterSet(_CS_WPC1252);
……
```

## 4-5 SetInterChrSet

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

**[Syntax]**

long **SetInterChrSet** (
  long nInternationalCharacterSet
);

**[Parameters]**

nInternational CharacterSet

International character set.

| Code | Value | Description |
|---|---|---|
| _ICS_USA | 0 | USA code setting |
| _ICS_FRANCE | 1 | FRANCE code setting |
| _ICS_GERMANY | 2 | GERMANY code setting |
| _ICS_UK | 3 | UK code setting |
| _ICS_DENMARK1 | 4 | DENMARK1 code setting |
| _ICS_SWEDEN | 5 | SWEDEN code setting |
| _ICS_ITALY | 6 | ITALY code setting |
| _ICS_SPAIN | 7 | SPAIN code setting |
| _ICS_NORWAY | 9 | NORWAY code setting |
| _ICS_DENMARK2 | 10 | DENMARK 2 code setting |
| _ICS_SPAIN2 | 11 | SPAIN 2 code setting |
| _ICS_LATIN | 12 | LATIN AMERICA code setting |
| _ICS_KOREA | 13 | KOREA code setting |
| _ICS_SLOVENIA | 14 | SLOVENIA code setting |
| _ICS_CHINA | 15 | CHINA code setting |

**[Return Values]**

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

* Example

```
long lResult = PrinterOpen("portinfo…", 1000);
……
SetInterChrSet(_ICS_SPAIN);
……
```

## 4-6 PrintText

• Sends a character print command to the printer.

**[Syntax]**

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

**[Parameters]**

<table>
<tr><td>Data</td><td>Text data to be printed.<br>* Text data are encoded with the system default code page.</td></tr>
<tr><td>Alignment</td><td>Alignment
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>_ALIGNMENT_LEFT</td><td>0</td><td>Left-aligned</td></tr>
<tr><td>_ALIGNMENT_CENTER</td><td>1</td><td>Centered</td></tr>
<tr><td>_ALIGNMENT_RIGHT</td><td>2</td><td>Right-aligned</td></tr>
</table></td></tr>
<tr><td>Attribute</td><td>Text properties
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>_FT_DEFAULT</td><td>0</td><td>Font A (default font)</td></tr>
<tr><td>_FT_FONTB</td><td>1</td><td>Set to Font B</td></tr>
<tr><td>_FT_FONTC</td><td>2</td><td>Set to Font C</td></tr>
<tr><td>_FT_BOLD</td><td>2</td><td>Add Bold lettering attribute</td></tr>
<tr><td>_FT_UNDERLINE</td><td>4</td><td>Add Underline (1-dot thick)</td></tr>
<tr><td>_FT_UNDERTHICK</td><td>6</td><td>Add Underline (2-dot thick)</td></tr>
<tr><td>_FT_REVERSE</td><td>8</td><td>Add Reverse lettering attribute</td></tr>
</table></td></tr>
<tr><td>TextSize</td><td>Text size
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>_TS_0WIDTH</td><td>0</td><td>Width magnification set to x 1</td></tr>
<tr><td>_TS_1WIDTH</td><td>16</td><td>Width magnification set to x 2</td></tr>
<tr><td>_TS_2WIDTH</td><td>32</td><td>Width magnification set to x 3</td></tr>
<tr><td>_TS_3WIDTH</td><td>48</td><td>Width magnification set to x 4</td></tr>
<tr><td>_TS_4WIDTH</td><td>64</td><td>Width magnification set to x 5</td></tr>
<tr><td>_TS_5WIDTH</td><td>80</td><td>Width magnification set to x 6</td></tr>
<tr><td>_TS_6WIDTH</td><td>96</td><td>Width magnification set to x 7</td></tr>
<tr><td>_TS_7WIDTH</td><td>112</td><td>Width magnification set to x 8</td></tr>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>_TS_0HEIGHT</td><td>0</td><td>Height magnification set to x 1</td></tr>
<tr><td>_TS_1HEIGHT</td><td>1</td><td>Height magnification set to x 2</td></tr>
<tr><td>_TS_2HEIGHT</td><td>2</td><td>Height magnification set to x 3</td></tr>
<tr><td>_TS_3HEIGHT</td><td>3</td><td>Height magnification set to x 4</td></tr>
<tr><td>_TS_4HEIGHT</td><td>4</td><td>Height magnification set to x 5</td></tr>
<tr><td>_TS_5HEIGHT</td><td>5</td><td>Height magnification set to x 6</td></tr>
<tr><td>_TS_6HEIGHT</td><td>6</td><td>Height magnification set to x 7</td></tr>
<tr><td>_TS_7HEIGHT</td><td>7</td><td>Height magnification set to x 8</td></tr>
</table></td></tr>
</table>

**[Return Values]**

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

* Example

```
long lResult = PrinterOpen(“portinfo…”, 1000);
……
PrintText("TEXT\n", _ALIGNMENT_LEFT, _FT_DEFAULT, _TS_0WIDTH | _TS_0HEIGHT);
……
```

## 4-7 PrintTextW

• Sends a character print command to the printer.

**[Syntax]**

long **PrintTextW**(
  LPCWSTR Data,
  long Alignment,
  long Attribute,
  long TextSize,
  long CodePage
);

**[Parameters]**

| | |
|---|---|
| Data | Text data in UNICODE format to be printed. |
| Alignment | Alignment<br>Code / Value / Description<br>_ALIGNMENT_LEFT / 0 / Left-aligned<br>_ALIGNMENT_CENTER / 1 / Centered<br>_ALIGNMENT_RIGHT / 2 / Right-aligned |
| Attribute | Text properties<br>Code / Value / Description<br>_FT_DEFAULT / 0 / Font A (default font)<br>_FT_FONTB / 1 / Set to Font B<br>_FT_FONTC / 2 / Set to Font C<br>_FT_BOLD / 2 / Add Bold lettering attribute<br>_FT_UNDERLINE / 4 / Add Underline (1-dot thick)<br>_FT_UNDERTHICK / 6 / Add Underline (2-dot thick)<br>_FT_REVERSE / 8 / Add Reverse lettering attribute |
| Size | Text size<br>Code / Value / Description<br>_TS_0WIDTH / 0 / Width magnification set to x 1<br>_TS_1WIDTH / 16 / Width magnification set to x 2<br>_TS_2WIDTH / 32 / Width magnification set to x 3<br>_TS_3WIDTH / 48 / Width magnification set to x 4<br>_TS_4WIDTH / 64 / Width magnification set to x 5<br>_TS_5WIDTH / 80 / Width magnification set to x 6<br>_TS_6WIDTH / 96 / Width magnification set to x 7<br>_TS_7WIDTH / 112 / Width magnification set to x 8<br>Code / Value / Description<br>_TS_0HEIGHT / 0 / Height magnification set to x 1<br>_TS_1HEIGHT / 1 / Height magnification set to x 2<br>_TS_2HEIGHT / 2 / Height magnification set to x 3<br>_TS_3HEIGHT / 3 / Height magnification set to x 4<br>_TS_4HEIGHT / 4 / Height magnification set to x 5<br>_TS_5HEIGHT / 5 / Height magnification set to x 6<br>_TS_6HEIGHT / 6 / Height magnification set to x 7<br>_TS_7HEIGHT / 7 / Height magnification set to x 8 |

CodePage

Code page for encoding text data.

| Code | Value | Description |
|---|---|---|
| _CS_PC437 | 0 | USA, Standard Europe |
| _CS_PC850 | 2 | Multilingual Latin 1 |
| _CS_PC860 | 3 | Portuguese |
| _CS_PC863 | 4 | Canadian-French |
| _CS_PC865 | 5 | Nordic |
| _CS_WPC1252 | 16 | Latin 1 |
| _CS_PC866 | 17 | Cyrillic #2 |
| _CS_PC852 | 18 | Latin 2 |
| _CS_PC858 | 19 | Euro |
| _CS_PC862 | 21 | Hebrew DOS code |
| _CS_THAI42 | 23 | THAI 42 |
| _CS_WPC1253 | 24 | Greek |
| _CS_WPC1254 | 25 | Turkish |
| _CS_WPC1257 | 26 | Baltic |
| _CS_WPC1251 | 28 | Cyrillic |
| _CS_PC737 | 29 | Greek |
| _CS_PC775 | 30 | Baltic |
| _CS_THAI14 | 31 | THAI 14 |
| _CS_WPC1255 | 33 | Hebrew New Code |
| _CS_THAI11 | 34 | THAI 11 |
| _CS_THAI18 | 35 | THAI 18 |
| _CS_PC855 | 36 | Cyrillic |
| _CS_PC857 | 37 | Turkish |
| _CS_PC928 | 38 | Greek |
| _CS_THAI16 | 39 | THAI 16 |
| _CS_PC1258 | 41 | Vietnamese |
| _CS_PC1250 | 47 | Czech |
| _CS_LATIN9 | 48 | Latin 9 |
| _CS_TCVN | 49 | Vietnamese (TCVN) |
| _CS_TCVN_CAPITAL | 50 | Vietnamese (TCVN CAPITAL) |
| _CS_VISCII | 51 | Vietnamese (VISCII) |
| _CS_KS5601 | 949 | Korean |
| _CS_BIG5 | 950 | Chinese (BIG5) |
| _CS_GB2312 | 936 | Chinese (GB2312) |
| _CS_SHIFT_JIS | 932 | Japanese (Shift JIS) |

**[Return Values]**

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

* Example

```
long lResult = PrinterOpen(“portinfo…”, 1000);
……
PrintTextW(L"TEXT\n", _ALIGNMENT_LEFT, _FT_DEFAULT, _TS_0WIDTH |
_TS_0HEIGHT, _CS_PC437);
……
```

## 4-8 PrintBarcode

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

**[Syntax]**

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

**[Parameters]**

<table>
<tr><td>Data</td><td>Data in ASCII format.</td></tr>
<tr><td>Symbology</td><td>Code Symbology.
<table>
<tr><th>Symbology</th><th>Value</th><th>Length Limit</th><th>Value Range</th></tr>
<tr><td>_BCS_UPCA</td><td>101</td><td>11 ≤ n ≤12</td><td>48 ≤ data ≤ 57</td></tr>
<tr><td>_BCS_UPCE</td><td>102</td><td>11 ≤ n ≤12</td><td>48 ≤ data ≤ 57</td></tr>
<tr><td>_BCS_EAN13<br>_BCS_JAN13</td><td>104<br>106</td><td>12 ≤ n ≤13</td><td>48 ≤ data ≤ 57</td></tr>
<tr><td>_BCS_EAN8<br>_BCS_JAN8</td><td>103<br>105</td><td>7 ≤ n ≤ 8</td><td>48 ≤ data ≤ 57</td></tr>
<tr><td>_BCS_ITF</td><td>107</td><td>1 ≤ n ≤ 255<br>(even)</td><td>48 ≤ data ≤ 57</td></tr>
<tr><td>_BCS_CODABAR</td><td>108</td><td>1 ≤ n ≤ 255</td><td>48 ≤ data ≤ 57,<br>65 ≤ data ≤ 68,<br>Data = 36,43,45,46,47,58</td></tr>
<tr><td>_BCS_CODE39</td><td>109</td><td>1 ≤ n ≤ 255</td><td>48 ≤ data ≤ 57,<br>65 ≤ data ≤ 90,<br>data = 32,36,37,43,<br>45,46,47</td></tr>
<tr><td>_BCS_CODE93</td><td>110</td><td>1 ≤ n ≤ 255</td><td>0 ≤ data ≤ 127</td></tr>
<tr><td>_BCS_CODE128</td><td>111</td><td>2 ≤ n ≤ 255</td><td>0 ≤ data ≤ 127</td></tr>
</table></td></tr>
<tr><td>Height</td><td>The barcode height. (Range: 1 ≤ Height ≤ 255)</td></tr>
<tr><td>Width</td><td>The width or size of a 1d barcode. If the print width of the barcode exceeds the printing paper, barcodes may not be printed properly.<br>* The range of 1d barcode width is from 2 to 6.</td></tr>
<tr><td>Alignment</td><td>Alignment.
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>_ALIGNMENT_LEFT</td><td>0</td><td>Left-aligned</td></tr>
<tr><td>_ALIGNMENT_CENTER</td><td>1</td><td>Centered</td></tr>
<tr><td>_ALIGNMENT_RIGHT</td><td>2</td><td>Right-aligned</td></tr>
</table></td></tr>
<tr><td>TextPosition</td><td>The HRI (Human Readable Interpretation) printing position of the 1d barcode.
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>_BC_TEXT_NONE</td><td>0</td><td>Not Printed</td></tr>
<tr><td>_BC_TEXT_ABOVE</td><td>1</td><td>Above the barcode</td></tr>
<tr><td>_BC_TEXT_BELOW</td><td>2</td><td>Below the barcode</td></tr>
<tr><td>_BC_TEXT_BOTH</td><td>3</td><td>Above and Below the barcode</td></tr>
</table></td></tr>
</table>

**[Return Values]**
If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
long lResult = PrinterOpen("portinfo…", 1000);
……
PrintBarcode("123456789012", _BCS_CODE128, 50, 2, _ALIGNMENT_LEFT,
_BC_TEXT_BELOW);
;……
```

## 4-9 PrintQRCode

• Sends a QR Code print command to the printer.

**[Syntax]**

long **PrintQRCode**(
PCHAR Data,
long Model,
long Size,
long EccLevel,
long Alignment
);

**[Parameters]**

| | |
|---|---|
| Data | Code data in ASCII format. |
| Model | Model type of QR Code |

| Code | Value | Description |
|---|---|---|
| _QRCODE_MODEL_1 | 49 | MODEL 1 |
| _QRCODE_MODEL_2 | 50 | MODEL 2 |

| | |
|---|---|
| Size | Module size (Range: 1 ≤ Size ≤ 8) |
| EccLevel | Error Correction Level |

| Code | Value | Description |
|---|---|---|
| _QRCODE_ECC_LEVEL_L | 48 | LEVEL L |
| _QRCODE_ECC_LEVEL_M | 49 | LEVEL M |
| _QRCODE_ECC_LEVEL_Q | 50 | LEVEL Q |
| _QRCODE_ECC_LEVEL_H | 51 | LEVEL H |

Alignment

| Code | Value | Description |
|---|---|---|
| _ALIGNMENT_LEFT | 0 | Left-aligned |
| _ALIGNMENT_CENTER | 1 | Centered |
| _ALIGNMENT_RIGHT | 2 | Right-aligned |

**[Return Values]**

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

* Example

```
long lResult = PrinterOpen("portinfo…", 1000);
……
PrintQRCode("123456789012", _QRCODE_MODEL_1, 4, _QRCODE_LEVEL_L,
_ALIGNMENT_LEFT);
……
```

## 4-10 PrintPDF417

• Sends a PDF417 print command to the printer.

**[Syntax]**

long **PrintPDF417**(
PCHAR Data,
long Type,
long ColumnNumber,
long RowNumber,
long Width,
long Height,
long EccLevel,
long Alignment
);

**[Parameters]**

<table>
<tr><td>Data</td><td>Code data in ASCII format.</td></tr>
<tr><td>Type</td><td>PDF417 Type
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>_PDF417_TYPE_1</td><td>49</td><td>PDF417 Standard</td></tr>
<tr><td>_PDF417_TYPE_2</td><td>50</td><td>PDF417 Simplified</td></tr>
</table></td></tr>
<tr><td>ColumnNumber</td><td>Number of columns (Range: 0 ≤ ColumnNumber ≤ 30)</td></tr>
<tr><td>RowNumber</td><td>Number of rows (Range: 3 ≤ RowNumber ≤ 90)</td></tr>
<tr><td>ModuleWidth</td><td>Module width (Range: 1 ≤ Width ≤ 4)</td></tr>
<tr><td>ModuleHeight</td><td>Module height (Range: 2 ≤ Height ≤ 8)</td></tr>
<tr><td>EccLevel</td><td>Error Correction Level
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>_PDF417_ECC_LEVEL_0</td><td>48</td><td>Error Correction Level 0</td></tr>
<tr><td>_PDF417_ECC_LEVEL_1</td><td>49</td><td>Error Correction Level 1</td></tr>
<tr><td>_PDF417_ECC_LEVEL_2</td><td>50</td><td>Error Correction Level 2</td></tr>
<tr><td>_PDF417_ECC_LEVEL_3</td><td>51</td><td>Error Correction Level 3</td></tr>
<tr><td>_PDF417_ECC_LEVEL_4</td><td>52</td><td>Error Correction Level 4</td></tr>
<tr><td>_PDF417_ECC_LEVEL_5</td><td>53</td><td>Error Correction Level 5</td></tr>
<tr><td>_PDF417_ECC_LEVEL_6</td><td>54</td><td>Error Correction Level 6</td></tr>
<tr><td>_PDF417_ECC_LEVEL_7</td><td>55</td><td>Error Correction Level 7</td></tr>
<tr><td>_PDF417_ECC_LEVEL_8</td><td>56</td><td>Error Correction Level 8</td></tr>
</table></td></tr>
<tr><td>Alignment</td><td>Alignment
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>_ALIGNMENT_LEFT</td><td>0</td><td>Left-aligned</td></tr>
<tr><td>_ALIGNMENT_CENTER</td><td>1</td><td>Centered</td></tr>
<tr><td>_ALIGNMENT_RIGHT</td><td>2</td><td>Right-aligned</td></tr>
</table></td></tr>
</table>

**[Return Values]**

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

* Example

```
long lResult = PrinterOpen("portinfo…", 1000);
……
PrintPDF417("123456789012", _PDF417_TYPE_1, 0, 0, 2, 2 _PDF417_LEVEL_L,
_ALIGNMENT_LEFT);
……
```

## 4-11 PrintDataMatrix

• Sends a Data Matrix print command to the printer.

**[Syntax]**

long **PrintPDF417**(
  PCHAR Data,
  long Size,
  long Alignment
);

**[Parameters]**

| | |
|---|---|
| Data | Code data in ASCII format. |
| Size | Module size (Range: 2 ≤ Size ≤ 3) |
| Alignment | Alignment |

| Code | Value | Description |
|---|---|---|
| _ALIGNMENT_LEFT | 0 | Left-aligned |
| _ALIGNMENT_CENTER | 1 | Centered |
| _ALIGNMENT_RIGHT | 2 | Right-aligned |

**[Return Values]**

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

* Example

```
long lResult = PrinterOpen("portinfo…", 1000);
……
PrintDataMatrix("123456789012", 2, _ALIGNMENT_LEFT);
……
```

## 4-12 PrintBitmap

• Sends a graphic image command to the printer.

| ⚠Note | Only BMP file format can be printed. |
|---|---|

**[Syntax]**

long **PrintBitmap** (
  LPCSTR FileName,
  long Width,
  long Alignment,
  long Level,
  BOOL Dithering
);

**[Parameters]**

<table>
<tr><td>FileName</td><td>The full path of the image file.</td></tr>
<tr><td>Width</td><td>The width of the image to be printed.
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>_WIDTH_FULL</td><td>-1</td><td>Maximum width the printer can print.</td></tr>
<tr><td>_WIDTH_NONE</td><td>-2</td><td>No variation given to image size</td></tr>
</table></td></tr>
<tr><td>Alignment</td><td>Alignment
<table>
<tr><th>Code</th><th>Value</th><th>Description</th></tr>
<tr><td>_ALIGNMENT_LEFT</td><td>0</td><td>Left-aligned</td></tr>
<tr><td>_ALIGNMENT_CENTER</td><td>1</td><td>Centered</td></tr>
<tr><td>_ALIGNMENT_RIGHT</td><td>2</td><td>Right-aligned</td></tr>
</table></td></tr>
<tr><td>Level</td><td>The brightness level. (Range: 0 ≤ Level ≤ 100)</td></tr>
<tr><td>Dithering</td><td>Whether apply the dither option or not.</td></tr>
</table>

**[Return Values]**

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

* Example

```
long lResult = PrinterOpen("portinfo…", 1000);
……
PrintBitmap(strDirBmp, 300, _ALIGNMENT_CENTER, 50, TRUE);
……
```

## 4-13 GetPrinterCurrentStatus

• Gets the printer's current status value.

**[Syntax]**
long **GetPrinterCurrentStatus** ();

**[Parameters]**
None

**[Return Values]**
If the method succeeds, the return value is one of the following. Other cases, returns negative values.

| Printer Status | Value | Description |
|---|---|---|
| _STS_NORMAL | 0 | Normal. (No errors) |
| _STS_PAPEREMPTY | 1 | Paper Empty |
| _STS_COVEROPEN | 2 | Printer cover is open |
| _STS_CASHDRAWER_HIGH | 16 | Cash drawer Pin is in high state. |
| _STS_ERROR | 32 | Offline (Printer is disconnected with HOST) |

* Example

```
long lResult = PrinterOpen("portinfo…", 1000);
……
int status = GetPrinterCurrentStatus();
…...
if ((status & _STS_PAPEREMPTY) == _STS_PAPEREMPTY) { … }
if ((status & _STS_ COVEROPEN) == _STS_ COVEROPEN) { … }
if ((status & _STS_CASHDRAWER_HIGH) == _STS_CASHDRAWER_ HIGH) { … }
…….
```

## 4-14 InitializePrinter

• Reset all printer settings to the state they were in when the printer was powered on.

**[Syntax]**
long **InitializePrinter**();

**[Parameters]**
None.

**[Return Values]**
If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
long lResult = PrinterOpen(3, "192.168.1.1", 9100);
……
InitializePrinter();
……
```

## 4-15 DirectIO

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

| | |
|---|---|
| ⚠Note | If there are no methods you want to use, this method is useful to send custom data to the printer. For the information about the commands recognized by the printer, refer to the ESC/POS Programming manual. |

**[Syntax]**

long **DirectIO** (
  PCHAR Data,
  UINT DataSize,
  PCHAR RecvBuffer,
  UINT& RecvBufferSize
);

**[Parameters]**

| | |
|---|---|
| Data | The data to be sent. |
| DataSize | The size of buffer specified by the parameter 'Data'. |
| RecvBuffer | The buffer, which receives a response from the printer.<br>*** set this parameter as NULL if no need to receive a response.** |
| RecvBufferSize | The size of the buffer specified by the parameter 'RecvBuffer'.<br>*** This value also used as the size of a response.** |

**[Return Values]**

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

* Example

```
long lResult = PrinterOpen("portinfo…", 1000);
……
CHAR cmd[3] = {0x10, 0x04, 0x02};
CHAR * readData;
UINT readLen = 0;
……
lResult = DirectIO(cmd, sizeof(cmd), readData, readLen);
……
```

## 4-16 CutPaper

• Sends a paper cut commands with line feeds to the printer.

**[Syntax]**

long **CutPaper** ( );

**[Parameters]**

None

**[Return Values]**

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

## 4-17 OpenDrawer

• Sends a cash drawer open command to the printer.

**[Syntax]**
long **OpenDrawer** (int pinNumber);

| ⚠Note | Depending on the cash drawer type, the PIN number for opening the cash drawer is different. |
|---|---|

**[Parameters]**

| | |
|---|---|
| pinNumber | PIN Number. |

| Code | Value | Description |
|---|---|---|
| _CASHDRAWER_PIN2 | 0 | PIN Number 2 |
| _CASHDRAWER_PIN5 | 1 | PIN Number 5 |

**[Return Values]**
If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
……
OpenDrawer(_CASHDRAWER_PIN2);
OpenDrawer(_CASHDRAWER_PIN5);
……
```

## 4-18 BidiSetCallBack

• Sets the callback function which receives a status value of the printer.
The callback function is called whenever the status value of the printer changes.
The printer status value is one of the followings.

| Printer Status | Value | Description |
|---|---|---|
| _STS_NORMAL | 0 | Normal. (No errors) |
| _STS_PAPEREMPTY | 1 | Paper Empty |
| _STS_COVEROPEN | 2 | Printer cover is open |
| _STS_CASHDRAWER_HIGH | 16 | Cash drawer Pin is in high state. |
| _STS_ERROR | 32 | Offline (Printer is disconnected with HOST) |

**[Syntax]**
long **BidiSetCallback**(fnStatusCallback pBidiCallBackFn);

**[Parameters]**

| | |
|---|---|
| pBidiCallBackFn | A call back function pointer which receives printer status.<br>* typedef int (CALLBACK *fnStatusCallback)(int nStatus) |

**[Return Values]**
If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
long lResult = PrinterOpen(3, “192.168.1.1”, 9100);
……
int CALLBACK StatusCallBackMethod(int status){
  if ((status & _STS_PAPEREMPTY) == _STS_PAPEREMPTY) { … }
  if ((status & _STS_ COVEROPEN) == _STS_ COVEROPEN) { … }
  if ((status & _STS_CASHDRAWER_HIGH) == _STS_CASHDRAWER_HIGH) { … }
  if ((status & _STS_ERROR) == _STS_ERROR) { … }
}
……
BidiSetCallBack (StatusCallBackMethod);
……
```

## 4-19 BidiCancelCallBack

• Cancels notification of printer status values.

**[Syntax]**
long **BidiCancelCallback**();

**[Parameters]**
None.

**[Return Values]**
If the method succeeds, the return value is 0. Other cases, returns negative values.

* Example

```
……
BidiCancelCallback();
……
```

# Copyright



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

# Caution

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

# Revision history

| Rev. | Date | Page | Description |
|---|---|---|---|
| 1.00 | 18.07.18 | - | New |
| 1.01 | 25.04.19 | 3 | Information for the supported printer updated. |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |