SBARCO SDK MANUAL

SDK API

This manual provides software programmer to use the API easily. When programmer wants their own applications to link with SBARCO printer, SBARCO company provides those dlls to help them. So, you can refer this manual for your big help.

Features:

- Access, Excel, VBA, VB6, VB2005, VC6, VC2005, Delphi6, CShap Examples
- Single Line Text, MultiLine Paragraph (Horizontal Alignment)
- True-Type-Font Unicode Ready
- 1-D Barcode, 2-D Barcode (QR Code, PDF-417, Maxicode, Data Matrix)
- Sent to USB, RS232, LPT, Ethernet, File or Pipe
- Demand printer to do something
- Get serial number, dpi, emulation, etc…
- Auto color to monochrome
- Support all SBARCO printer
- Support Standalone Form Access

Ver. 1.22 2013/12/12

SBSDK_API

Content

SDK MATERIAL

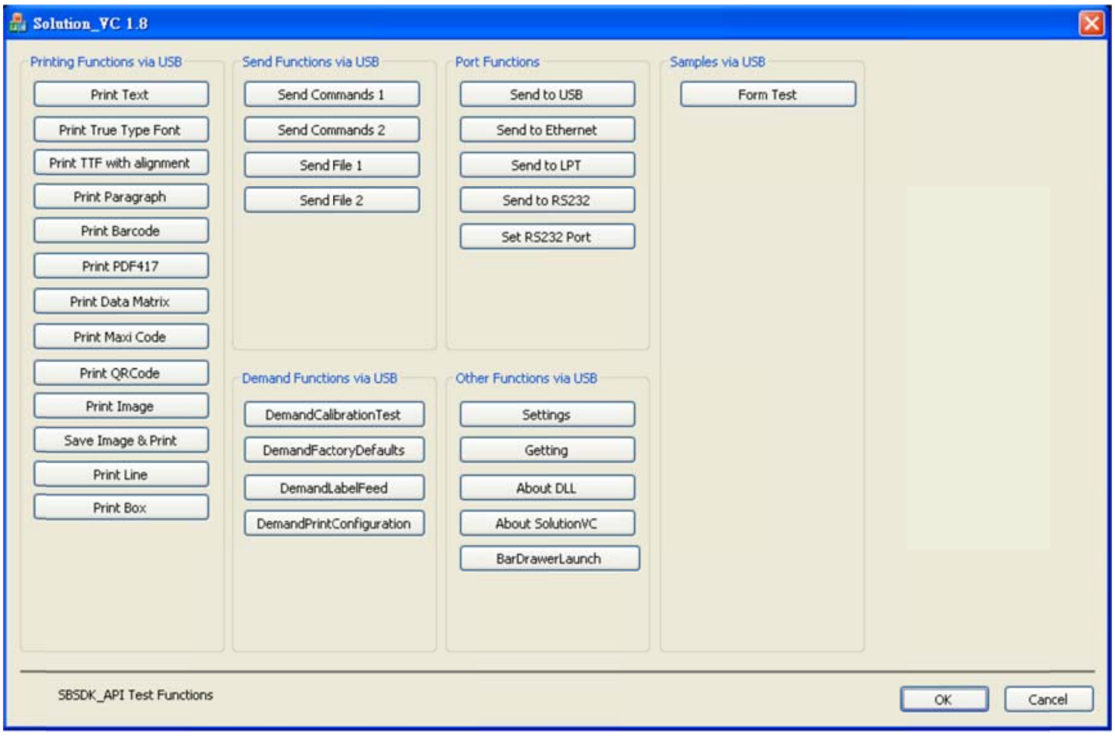
We support two solutions for VB, VC, CShap and Delphi now. Programmer can find them in directory of “sbarco_sdk”. Those solutions have had many examples for every function for programmer can easy to use them. So after implement, you must copy the “SBSDK_API.dll” and “NECI.dll“ files to your own application directory.

Path			
sbarco_sdk	Solution_VB\\		
	Solution_VC\\	BIN_900\\	Release\\
		BIN_800\\	Release\\
	Solution_Delphi6\\		
	Solution_CShap\\	SBARCO_SDK\bin\\	Debug\\
	SBSDK_API.dll		
	NECI.dll		

Solution_VC

This solution is made from Visual C++ 2005 and Visual C++ 2008.

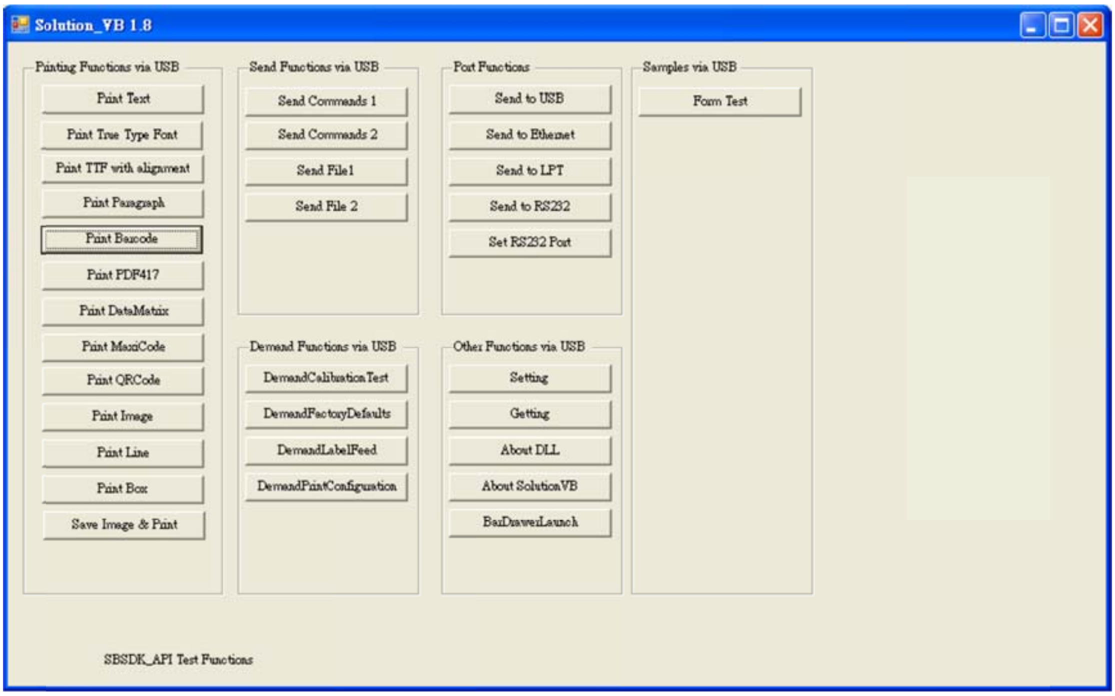
- Workspace file: Solution_VC\ solution_vc_2005.sln or solution_vc_2008.sln



Solution_VB

This solution is made form Visual C++ 2005

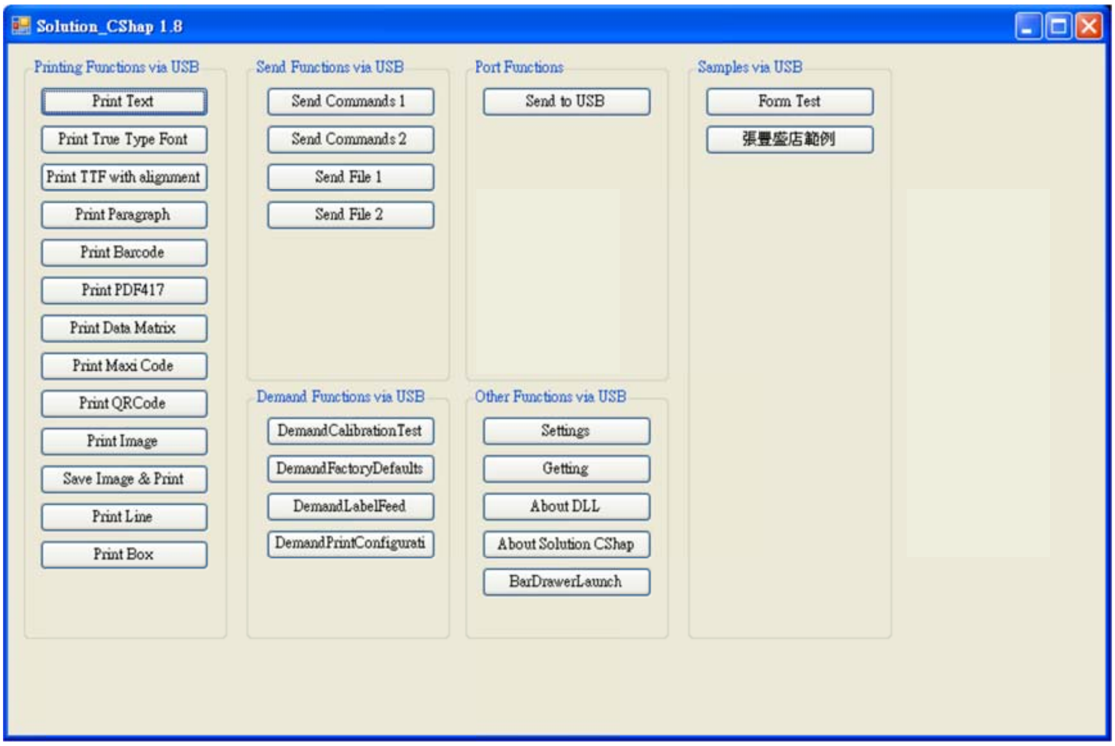
- Workspace file: Solution_VB\ Solution_VB_2005.sln



Solution_CShap_2008

This solution is made from CShap

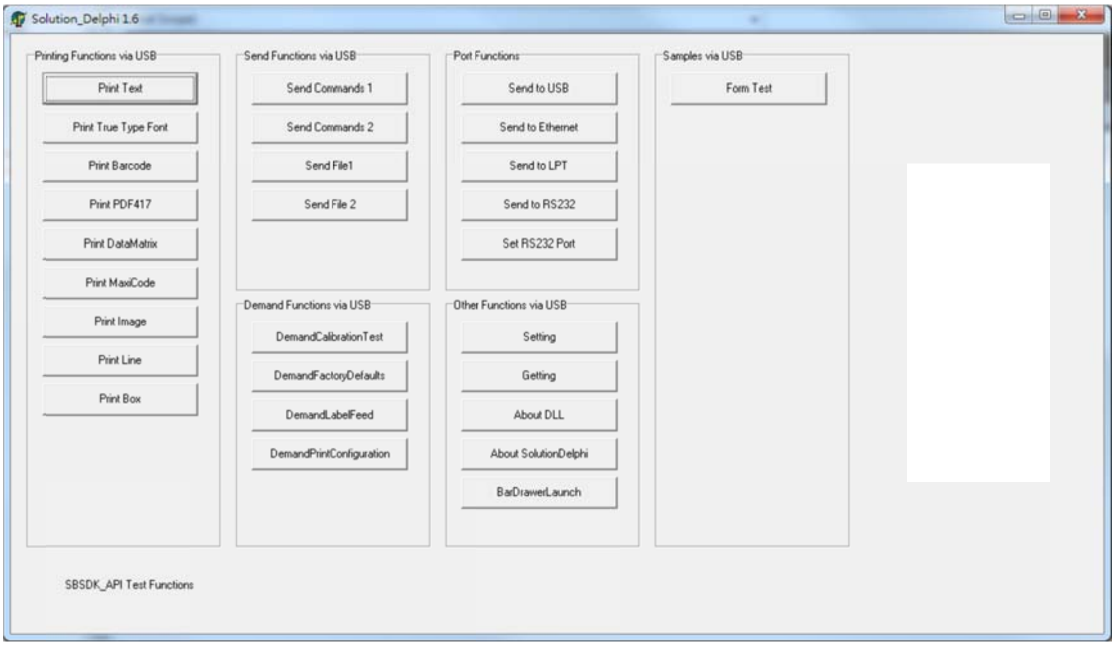
- Workspace file: Solution_CShap\ solution_cshap_2008.sln



Solution_Delphi

This solution is made from Delphi6

- Workspace file: Solution_Delphi6\ Project1.dpr



HISTORY

2010/08/09 Ver:1.0	First Release
2010/11/11 Ver:1.1	Add BarDrawer Functions
2010/11/23 Ver:1.2	Remove the rectangle of PrintImage function. The rectangle is for debuging.
2011/05/26 Ver:1.4	Fixed bug for Tear-Off Mode of SetAfterPrint Function.
2011/06/09 Ver:1.5	Function of PortOpen added an option of name pipe for sharing printer.
2011/07/04 Ver:1.6	Add function for SetReferencePoint, GetReferencePoint.
2011/07/25 Ver:1.7	Modify TPH width from 200 to 203 dpi.
	Fixed USB enum bug for one of longer USB register name.
	Add function for Y offset of TPH.
2011/08/31 Ver:1.8	Add Form Functions for FormSaveByFileA, FormSaveByFileW, FormCheck, FormExecute,
	FormParameterData and FormPrint.
	Add a FormTest sample in VC and VB.
2011/09/08 Ver:1.9	Add Ethernet communication for PortEnumCount(), PortEnumGet() and PortOpen() Functions.
2011/10/03 Ver:1.10	Add Delphi Sample Codes.
	Fixed RS-232 default baud rate value from 115200 to 9600.
	Fixed RS-232 communication bug in Win7
	Fixed GetPrintDistance() bug in Dot Mode.
	Change the DllAbout() return value from integer to double.
2011/12/08 Ver:1.11	OpenPort() removed the query action.
	PrintLabel() add a command for saving data to RAM.
	When the print fails, the program automatically clears the memory of the print content.
	If the program execution path exists file "DLLTOFILE.txt", the program will automatically print the contents to "DLLTOFILE.txt".
	Add ImageSave(), ImagePrint(), ImageDelete() functions for not storing image for many times.
2012/1/31 Ver:1.12	Add PrintTTF_AlignmentA, PrintTTF_AlignmentW for True Type Font Text with horizontal alignment.
	Add PrintParagraphA, PrintParagraphW for True Type Font paragraph with horizontal alignement.
	Add PrintQRcode for QR code.
	Add CShap Sample Codes.
2013/1/16 Ver:1.17	Add unicode functions for PrintTextA, PrintTextW, PrintBarcodeA, PrintBarcodeW, PrintPDF417A, PrintPDF417W, PrintDataMatrixA, PrintDataMatrixW, PrintMaxiCodeA, PrintMaxiCodeW, PrintQRcodeA, PrintQRcodeW.
	Fixed PortOpen ("192.168.0.2"); / / TCP / IP using print two or more, will be unable to continue printing problems.

2013/6/24 Ver:1.19	Add PrintImageW, Image_SaveW to support Unicode string input.
2013/6/24 Ver:1.20	Fixed USB printing will not be fully finished printing issue.
2013/6/25 Ver:1.21	Add PrintImageW() and ImageSaveW() functions.
	Fixed "Parser_PrintText", "Parser_PrintBarcode", "Parser_PrintPDF417", "Parser_PrintDataMatrix", "Parser_PrintMaxiCode", "Parser_PrintQRcode", "Parser_PrintImage", "Parser_PrintTrueTypeFontA", "Parser_PrintTTF_AlignmentA", "Parser_PrintParagraphA", "Parser_LinePrint_TTF_Print", "Parser_LinePrint_QRCode_Print", "Parser_LinePrint_1DBar_Print" function can retention problems double quotes.
	Parser_LinePrint_QRCode_Print() add data string from unicode to ASCIIfunction.
2013/12/12 Ver:1.22	Add GetScaleData() function.

I WANT TO…

List

Print over Pallet (LPT)
Print over Serial (RS-232)
Print over USB
Print over Ethernet
Print over File
Print over a name pipe
Send files or data to the printer
Make some demands
Get some data from printer
Check this library
Get Scale Data from RS-232 to link with scale.

Example: Direct a port (VC)

```
PortOpen("LPT1");

SetMeasurement(1, 0); // inchs

PrintText(1.1f, 0.2f, 0, 1, 1, 1, 0, 17, "Print from LPT");

PrintLabel(1, 1, 0);

PortClose();
```

Example: Enumerate LPT Ports (VC)

```
int nPortCount = PortEnumCount(2); // Get LPT count

char acPortBuffer[24];

if(nPortCount > 0)

{

    PortEnumGet(2, 0, acPortBuffer); // Get name of number 0 of LPT port

    PortOpen(acPortBuffer);

    SetMeasurement(1, 0); // inchs

    PrintText(1.1f, 0.2f, 0, 1, 1, 1, 0, 16, "Print from LPT");

    PrintLabel(1, 1, 0);

    PortClose();

}
```

See Also

Port Functions, Printing Functions, SetMeasurement

Print over Serial (RS-232)

Example: Direct a port (VC)

```
PortSetupSerial(115200, 8, 0, 1, 1); // 115200 bps, 8 data-bits, no parity, 1 stop bit, hardware

PortOpen("COM1");

flowing control.

SetMeasurement(1, 0); // inchs

PrintText(1.1f, 0.2f, 0, 1, 1, 1, 0, 17, "Print from RS232");

PrintLabel(1, 1, 0);

PortClose();
```

Example: Enumerate RS232 Ports (VC)

```
int nPortCount = PortEnumCount(1); // Get RS232 count

char acPortBuffer[24];

if(nPortCount > 0)

{

	PortEnumGet(1, 0, acPortBuffer); // Get name of number 1 of RS232 port

	PortSetupSerial(115200, 8, 0, 1, 1); // 115200 bps, 8 data-bits, no parity, 1 stop bit, hardware

	PortOpen(acPortBuffer);

	SetMeasurement(1, 0); // inchs

	PrintText(1.1f, 0.2f, 0, 1, 1, 1, 0, 16, "Print from RS232");

	PrintLabel(1, 1, 0);

	PortClose();

}
```

See Also

Port Functions, Printing Functions, SetMeasurement

Example: Enumerate USB Ports (VC)

```
int nPortCount = PortEnumCount(0); // Get USB count

char acPortBuffer[24];

if(nPortCount > 0)

{

        PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

        PortOpen(acPortBuffer);

        SetMeasurement(1, 0); // inchs

        PrintText(1.1f, 0.2f, 0, 1, 1, 1, 0, 14, "Print from USB");

        PrintLabel(1, 1, 0);

        PortClose();

}
```

See Also

Port Functions, Printing Functions, SetMeasurement

Example: Enumerate Ethernet IP Addresses (VC)

```
int nPortCount = PortEnumCount(3); // Get Ethernet IP Address count
char acPortBuffer[24];
if(nPortCount > 0)
{
    PortEnumGet(3, 0, acPortBuffer); // Get IP Address of number 0 of Ethernet
    PortOpen(acPortBuffer);
    SetMeasurement(1, 0); // inchs
    PrintText(1.1f, 0.2f, 0, 1, 1, 1, 0, 14, "Print from Ethernet");
    PrintLabel(1, 1, 0);
    PortClose();
}
```

See Also

Port Functions, Printing Functions, SetMeasurement

Example (VC)

```
PortOpen("c:\\SBAPI-Out.txt");

SetMeasurement(1, 0); // inchs

PrintText(1.1f, 0.2f, 0, 1, 1, 1, 0, 17, "Print from file");

PrintLabel(1, 1, 0);

PortClose();
```

See Also

Port Functions, Printing Functions, SetMeasurement

Example (VC)

```
PortOpen("\\\\TESTPC\\SbarcoT4-usb02");
SetMeasurement(1, 0); // inchs
PrintText(1.1f, 0.2f, 0, 1, 1, 1, 0, 17, "Print from file");
PrintLabel(1, 1, 0);
PortClose();
```

See Also

Port Functions, Printing Functions, SetMeasurement

Example (VC)

```
PortOpen("LPT1");

int nDataLen = SendFileA("testcmd.txt");

//int nDataLen = SendFileW(_T("testcmd.txt"));

//int nDataLen = SendFileA("open");

//int nDataLen = SendFileW(_T("open"));

CString strMessage;

strMessage.Format(_T("Data length = %d"), nDataLen);

AfxMessageBox(strMessage);

PortClose();
```

See Also

Port Functions, SendFileA, SendFileW

Example: Calibration Test (VC)

```
PortOpen("LPT1");

DemandCalibrationTest(); // demand the printer to run a calibration test.

PortClose();
```

Example: Print Configuration (VC)

```
PortOpen("LPT1");

DemandPrintConfiguration(); // demand the printer to print out a label with printer
configuration.

PortClose();
```

See Also

Port Functions, Demand Functions

Example: Get Printer Emulation (VC)

```
PortSetupSerial(115200, 8, 0, 1, 1); // 115200 bps, 8 data-bits, no parity, 1 stop bit,
hardware
PortOpen("COM1"); // not for LPT port
flowing control.
if(GetEmulationType() == 100)
        AfxMessageBox(_T("PEPL Emulation"));
else
        AfxMessageBox(_T("Can't get emulation from printer"));
PortClose();
```

Example: Get Printer Serial Number (VC)

```
char acBuffer[32];
CString strTemp;
PortSetupSerial(115200, 8, 0, 1, 1); // 115200 bps, 8 data-bits, no parity, 1 stop bit,
hardware
PortOpen("COM1"); // not for LPT port
flowing control.
if(GetSerialNumber(acBuffer) > 0)
{
        strTemp = _T("Serial Number: ");
        strTemp += acBuffer;
        AfxMessageBox(strTemp);
}
else
        AfxMessageBox(_T("Can't get serial number"));
PortClose();
```

See Also

Port Functions, Getting Functions

Example (VC)

```
double dbVersion = DllAbout(TRUE); // 1=display dialog, 0=only return version

CString strVersion;

strVersion.Format(_T("DLL Version: %.1f"), dbVersion);

AfxMessageBox(strVersion);
```

See Also

Port Functions, Other Functions

Example (VC)

```
double dbVersion = DllAbout(TRUE); // 1=display dialog, 0=only return version

CString strVersion;

strVersion.Format(_T("DLL Version: %.1f"), dbVersion);

AfxMessageBox(strVersion);
```

See Also

Port Functions, Other Functions

PORT FUNCTIONS

Name	**Description**
PortClose	This function can close computer port which is opened by “PortOpen”.
PortEnumCount	This function can enumerate count of port.
PortEnumGet	This function can get port name from computer.
PortOpen	This function can open port.
PortSetupSerial	Defines the control setting for a serial communications device.

PortClose

This function can close computer port which is opened by “PortOpen”.

C++

```
extern "C" __declspec(dllimport) void _stdcall PortClose();
```

Visual Basic

```
Private Declare Sub PortClose Lib "SBSDK_API.DLL" ()
```

Delphi

```
Procedure PortClose ();stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern void PortClose();
```

Return Value

Parameters

Remarks

Example

```
char acPortBuffer[40];

int nPortCount = PortEnumCount(0); // Get USB count

PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

if(PortOpen(acPortBuffer) == 0)

{

        AfxMessageBox(_T("Can't open port!"));

        return;

}


SetAfterPrint (1, 0); // Enable Tear-Off mode after printing.

PrintLabel(0, 0, 0); // Only send data to printer without printing

PortClose();
```

See Also

Port Functions, PortOpen

PortEnumCount

This function can enumerate count of port.

C++

```
extern "C" __declspec(dllimport) int _stdcall PortEnumCount(int nPortType);
```

Visual Basic

```
Private Declare Function PortEnumCount Lib "SBSDK_API.DLL" (ByVal nPortType As Long) As Long
```

Delphi

```
function PortEnumCount (nPortType:integer):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern int PortEnumCount(int nPortType);
```

Return Value

The return value is 0~999 enumerated devices.

Parameters

- *nPortType*

  Port type.

Value	Description
0	USB
1	RS232
2	LPT
3	TCP/IP (Ethernet)

Remarks

The “PortEnumGet” and “PortEnumCount” both functions must use together.

Example (VC)

```
 int nPortCount = PortEnumCount(0); // Get USB count

char acPortBuffer[24];

if(nPortCount > 0)

{

        PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

        PortOpen(acPortBuffer);

        SetMeasurement(1, 0); // inchs

        PrintText(1.1f, 0.2f, 0, 1, 1, 1, 0, 14, "Print from USB");

        PrintLabel(1, 1, 0);

        PortClose();
}
```

See Also

Port Functions, PortEnumGet

PortEnumGet

This function can get port name from computer.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall PortEnumGet(int nPortType, int nItem, char* pBuffer);
```

Visual Basic

```
Private Declare Function PortEnumGet Lib "SBSDK_API.DLL" (ByVal nPortType As Long, ByVal nItem As Long, ByVal pBuffer As String) As Long
```

Delphi

```
function PortEnumGet (nPortType:integer; nItem:integer; pBuffer:Pchar):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static unsafe extern bool PortEnumGet(int nPortType, int nItem, StringBuilder pBuffer);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0.

Parameters

- *nPortType*

  Port type.

Value	Description
0	USB
1	RS232
2	LPT
3	TCP/IP (Ethernet)

- *nItem*

  Port number.

- *pBuffer*

  Buffer of getting port name.

Remarks

The “PortEnumGet” and “PortEnumCount” both functions must use together.

Example (VC)

```
int nPortCount = PortEnumCount(1); // Get RS232 count

char acPortBuffer[24];

if(nPortCount > 0)

{
        PortEnumGet(1, 0, acPortBuffer); // Get name of number 0 of USB port
        PortOpen(acPortBuffer);
        SetMeasurement(1, 0); // inchs
        PrintText(1.1f, 0.2f, 0, 1, 1, 1, 0, 16, "Print from RS232");
        PrintLabel(1, 1, 0);
        PortClose();
    }
```

See Also

Port Functions, PortEnumCount

PortOpen

This function can open port.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall PortOpen(char* pPortName);
```

Visual Basic

```
Private Declare Function PortOpen Lib "SBSDK_API.DLL" (ByVal pPortName As String) As Long
```

Delphi

```
function PortOpen (pPortName:pchar):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static unsafe extern bool PortOpen(StringBuilder pPortName);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0.

Parameters

- *pPortName*

  Port name to open.

Type	Value
USB	Use Enum function
RS232	COMx
LPT	LPTx
TCP/IP	XXX.XXX.XXX.XXX
File	File name with path
Pipe	\\ComputerName\PrinterShareName

Remarks

Example

```
char acPortBuffer[40];

int nPortCount = PortEnumCount(0); // Get USB count

PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

if(PortOpen(acPortBuffer) == 0)

{

        AfxMessageBox(_T("Can't open port!"));

        return;

}
```

```
SetAfterPrint (1, 0); // Enable Tear-Off mode after printing.

PrintLabel(0, 0, 0); // Only send data to printer without printing

PortClose();
```

See Also

Port Functions, PortEnumCount, PortClose

PortSetupSerial

Defines the control setting for a serial communications device.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall PortSetupSerial(int nBaudRate, int nDataBits, int
nParity, int nStopBits, int nFlowControl);
```

Visual Basic

```
Private Declare Function
```

Delphi

```
function PortSetupSerial (nBaudRate:integer; nDataBits:integer; nParity:integer; nStopBits:integer;
nFlowControl:integer):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool PortSetupSerial(int nBaudRate, int nDataBits, int nParity, int nStopBits, int
nFlowControl);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0.

Parameters

- *nBaudRate*

  The baud rate at which the communications device operates. This member can be an actual baud rate value, or one of the following indexes.

Value	Description
1200	1200 bps
2400	2400 bps
4800	4800 bps
9600	9600 bps (Printer Default)
19200	19200 bps
38400	38400 bps
57600	57600 bps
115200	115200 bps

- *nDataBits*

  The number of bits in the bytes transmitted and received. Range: 7 or 8 (Printer Default).

- *nParity*

  The parity scheme to be used. This member can be one of the following values.

Value	Description
0	No parity (Printer Default)
1	Odd parity
2	Even parity

- *nStopBits*

The number of stop bits to be used. This member can be one of the following values.

Value	Description
1	1 stop bit (Printer Default)
2	2 stop bits

- *nFlowControl*

Communication flowing control.

Value	Description
0	None
1	Hardware (Printer Default)
2	Xon/Xoff

Remarks

Example

```
PortSetupSerial(115200, 8, 0, 1, 1); // 115200 bps, 8 data-bits, no parity, 1 stop bit,
hardware

PortOpen("COM1");

SetMeasurement(1, 0); // inchs

PrintText(1.1f, 0.2f, 0, 1, 1, 1, 0, 17, "Print from RS232");

PrintLabel(1, 1, 0);

PortClose();
```

See Also

Port Functions

PRINTING FUNCTIONS

Name	Description
ImageDelete	Use an ID string to delete a image of printer.
ImagePrint	Use ID name to call printer image to print out.
ImageSave	Send an image file to printer and save with ID string.
PrintBarcode	This function can print a linear barcode.
PrintBox	This function can print a box.
PrintDataMatrix	This function can print a DataMatrix barcode.
PrintLine	This function can print a line.
PrintImage	Send an image file to printer and print out it with monochrome.
PrintMaxiCode	This function can print a MaxiCode barcode.
PrintParagraphA, PrintParagraphW	PrintParagraphA can print a paragraph with true type font. PrintParagraphW can print a Unicode parpgraph with true type font.
PrintPDF417	This function can print a PDF-417 barcode.
PrintQRcode	This function can print a QR code barcode.
PrintText	This function can print a text with printer internal font.
PrintTrueTypeFontA, PrintTrueTypeFontW	PrintTrueTypeA can print a normal text with true type font. PrintTrueTypeW can print a Unicode text with true type font.
PrintTTF_AlignmentA, PrintTTF_AlignmentW	PrintTTF_AlignmentA can print a true type font text with alignment.. PrintTTF_AlignmentW can print a true type font Unicode text with alignment.
PrintLabel	This function can end functions of printing and setting to start printer to print patterns.

ImageDelete

Use an ID string to delete a image of printer.

C++

```
BOOL Image_Delete(char* pIDString);
```

Visual Basic

```
Private Declare Function Image_Delete Lib "SBSDK_API.DLL" (ByVal pIDString As String) As Integer
```

Delphi

```
function Image_Delete (pIDString:string):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool Image_Delete(string pIDString);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Parameters

- *pIDString*

  Image ID name.

Remarks

Example (Visual Basic)

```
' Open USB port
Dim bResult As Integer
Dim nPortCount As Integer
Dim acPortBuffer As String
acPortBuffer = New String(vbNullChar, 24)
nPortCount = PortEnumCount(0) ' Get USB count
bResult = PortEnumGet(0, 0, acPortBuffer) ' Get name of number 0 of USB port
bResult = PortOpen(acPortBuffer)
If bResult = 0 Then bResult = MsgBox("Can't open port!", 0, "Solution VB") : Exit Sub

bResult = SetMeasurement(1, 0) ' inchs
bResult = SetMedia_LabelWithGaps(4, 3, 0.16, 0) ' W=4 inch, H=3Inch, GapLen=32 dots, GapOffset=0;

bResult = Image_Save(1.0F, 1.0F, 0, 0, "TestImage.jpg", "IMAGE_01") ' nWidth=-1(Original Size), nHeight=-1(Original Size), bErrorDiffusion=0

' print image with the ID specified
bResult = Image_Print(3.0F, 0.2F, "IMAGE_01")
bResult = Image_Print(1.0F, 0.2F, "IMAGE_01")
bResult = PrintLabel(1, 1, 0)
Call PortClose()
```

See Also

Printing Functions, SetMeasurement, ImageSave, ImagePrint

ImagePrint

Use ID name to call printer image to print out.

C++

```
Image_Print(float fX, float fY ,char* pIDString);
```

Visual Basic

```
Private Declare Function Image_Print Lib "SBSDK_API.DLL" (ByVal fX As Single, ByVal fY As Single,
ByVal pIDString As String) As Integer
```

Delphi

```
function Image_Print (fX:Single; fY:Single; pIDString:string):integer;stdcall;external
'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool Image_Print(float fX, float fY, string pIDString);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Parameters

- *fX*

  Horizontal start position (X) in library measurement.

- *fY*

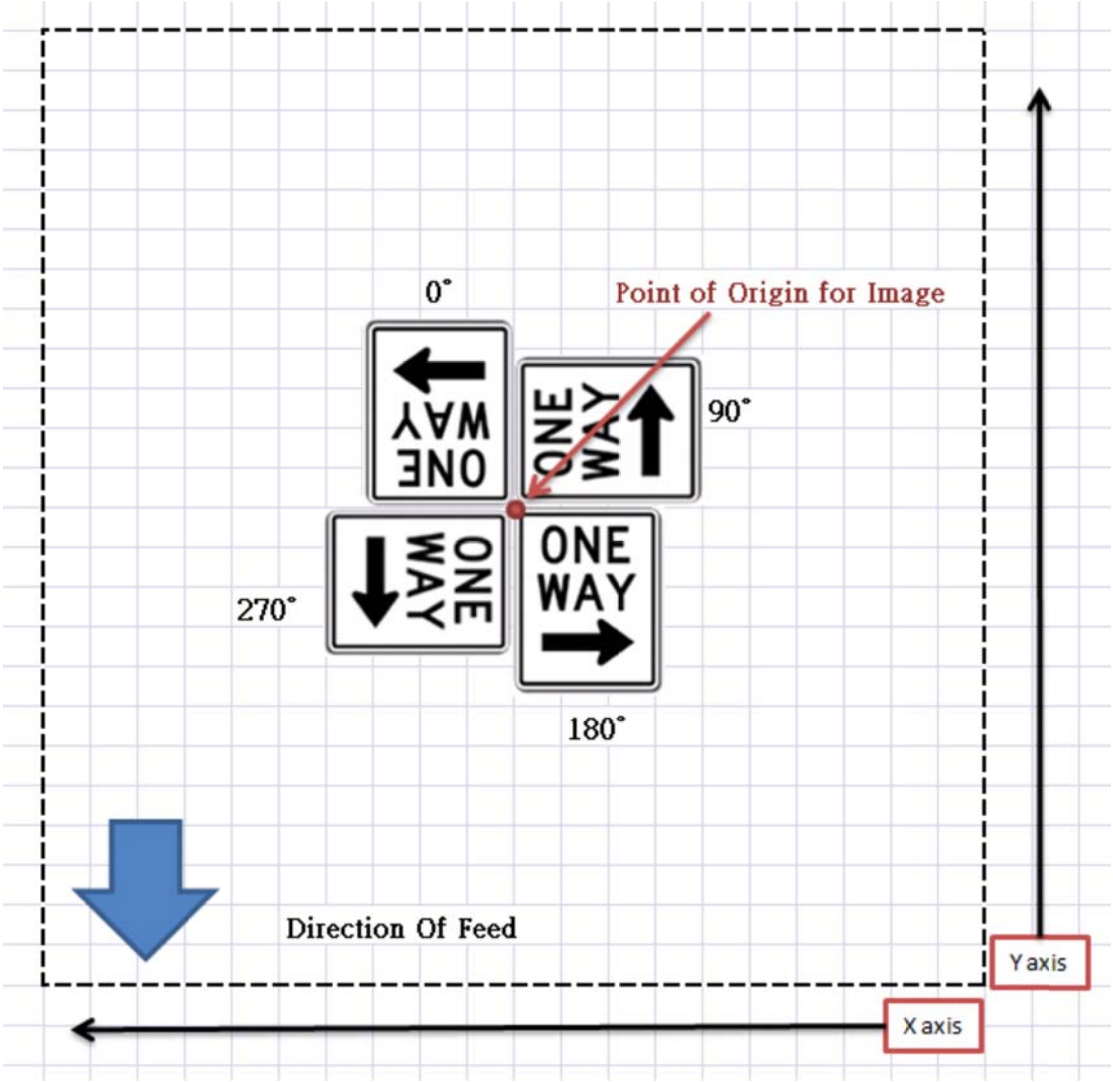
  Vertical start position (Y) in library measurement.

- *pIDString*

  Image ID name.

Remarks

Printing drawing.



Example (Visual Basic)

```
' Open USB port
Dim bResult As Integer
Dim nPortCount As Integer
Dim acPortBuffer As String
acPortBuffer = New String(vbNullChar, 24)
nPortCount = PortEnumCount(0) ' Get USB count
bResult = PortEnumGet(0, 0, acPortBuffer) ' Get name of number 0 of USB port
bResult = PortOpen(acPortBuffer)
If bResult = 0 Then bResult = MsgBox("Can't open port!", 0, "Solution VB") : Exit Sub

bResult = SetMeasurement(1, 0) ' inchs
bResult = SetMedia_LabelWithGaps(4, 3, 0.16, 0) ' W=4 inch, H=3Inch, GapLen=32 dots,
GapOffset=0;

bResult = Image_Save(1.0F, 1.0F, 0, 0, "TestImage.jpg", "IMAGE_01") ' nWidth=-1(Original
Size), nHeight=-1(Original Size), bErrorDiffusion=0

' print image with the ID specified
bResult = Image_Print(3.0F, 0.2F, "IMAGE_01")
bResult = Image_Print(1.0F, 0.2F, "IMAGE_01")
bResult = PrintLabel(1, 1, 0)
Call PortClose()
```

See Also

Printing Functions, SetMeasurement, ImageSave, ImageDelete

ImageSave

Send an image file to printer and save with ID string.

C++

```
BOOL Image_Save(float fWidth, float fHeight, int nRotation, BOOL bErrorDiffusion, char* pFileName,
char* pIDString);
```

Visual Basic

```
Private Declare Function Image_Save Lib "SBSDK_API.DLL" (ByVal fWidth As Single, ByVal fHeight
As Single, ByVal nRotation As Integer, ByVal bErrorDiffusion As Integer, ByVal pFileName As String,
ByVal pIDString As String) As Integer
```

Delphi

```
function Image_Save (fWidth:Single; fHeight:Single; nRotation:integer; bErrorDiffusion:integer;
pFileName:string; pIDString:string):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool Image_Save(float fWidth, float fHeight, int nRotation, int bErrorDiffusion,
string pFileName, string pIDString);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Parameters

➢ *fWidth*

Image Width in library measurement. If set -1, it will be original size.

➢ *fHeight*

Image Height in library measurement. If set -1, it will be original size.

➢ *nRotation*

Value	Description
0	No rotation
1	90 degrees
2	180 degrees
3	270 degrees

➢ *bErrorDiffusion*

If set 1, this function will decreases color to monochrome with error diffusion mode.

➢ *pFileName*

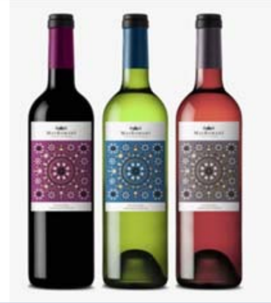
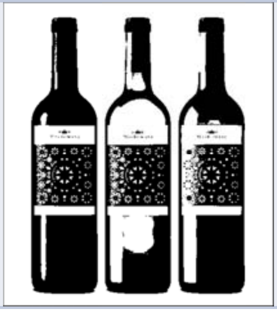
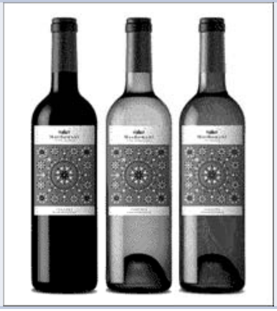
Image file name with path. It supports gif, jpg, jif, jpeg, png, tif, tiff, ico, bmp, dib and pcx formats.

➢ *pIDString*

Image ID name.

Remarks

Print out color Image.

Original image	Decrease color to monochrome	Decrease color to monochrome with error-diffusion
		

Example (Visual Basic)

```
' Open USB port
Dim bResult As Integer
Dim nPortCount As Integer
Dim acPortBuffer As String
acPortBuffer = New String(vbNullChar, 24)
nPortCount = PortEnumCount(0) ' Get USB count
bResult = PortEnumGet(0, 0, acPortBuffer) ' Get name of number 0 of USB port
bResult = PortOpen(acPortBuffer)
If bResult = 0 Then bResult = MsgBox("Can't open port!", 0, "Solution VB") : Exit Sub

bResult = SetMeasurement(1, 0) ' inchs
bResult = SetMedia_LabelWithGaps(4, 3, 0.16, 0) ' W=4 inch, H=3Inch, GapLen=32 dots, GapOffset=0;

bResult = Image_Save(1.0F, 1.0F, 0, 0, "TestImage.jpg", "IMAGE_01") ' nWidth=-1(Original Size), nHeight=-1(Original Size), bErrorDiffusion=0

' print image with the ID specified
bResult = Image_Print(3.0F, 0.2F, "IMAGE_01")
bResult = Image_Print(1.0F, 0.2F, "IMAGE_01")
bResult = PrintLabel(1, 1, 0)
Call PortClose()
```

See Also

Printing Functions, SetMeasurement, ImageDelete, ImagePrint

PrintBarcode

This function can print a linear barcode.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall PrintBarcode(float fX, float fY, float fBarHeight, int
nRotation, int nBarType, int nNarrowBar, int nWideBar, BOOL bHuman, int nDataLen, char* pData);
```

Visual Basic

```
Private Declare Function PrintBarcode Lib "SBSDK_API.DLL" (ByVal fX As Single, ByVal fY As Single,
ByVal fBarHeight As Single, ByVal nRotation As Long, ByVal nBarType As Long, ByVal nNarrowBar
As Long, ByVal nWideBar As Long, ByVal bHuman As Long, ByVal nDataLen As Long, ByVal pData
As String) As Long
```

Delphi

```
function PrintBarcode (fX:Single; fY:Single; fBarHeight:Single; nRotation:integer; nBarType:integer;
nNarrowBar:integer; nWideBar:integer; bHuman:integer; nDataLen:integer;
pData:string):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool PrintBarcode(float fX, float fY, float fBarHeight, int nRotation, int nBarType,
int nNarrowBar, int nWideBar, int bHuman, int nDataLen, string pData);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Parameters

- *fX*

  Horizontal start position (X) in library measurement.

- *fY*

  Vertical start position (Y) in library measurement.

- *fBarHeight*

  Barcode height in library measurement.

- *nRotation*

Value	Description
0	No rotation
1	90 degrees
2	180 degrees
3	270 degrees

- *nBarType*

nBarType	Description	nNarrowBar	nWideBar
1	Code 39 - Standard or Extended	1-10	2-30
2	Code 39 with check digit	1-10	0
3	Code 93	1-10	0
4	Code 128 UCC Serial Shipping Container Code	1-10	0

5	Code 128	1-10	0
6	Code 128 mode A	1-10	0
7	Code 128 mode B	1-10	0
8	Code 128 mode C	1-10	0
10	Codabar	1-10	2-30
11	EAN/JAN-8	2-4	0
12	EAN/JAN-8 with 2 digit add-on	2-4	0
13	EAN/JAN-8 with 5 digit add-on	2-4	0
14	EAN/JAN-13	2-4	0
15	EAN/JAN-13 with 2 digit add-on	2-4	0
16	EAN/JAN-13 with 5 digit add-on	2-4	0
18	Interleaved 2 of 5	1-10	2-30
19	Interleaved 2 of 5 with mode 10 check digit	1-10	2-30
20	Interleaved 2 of 5 with human readable check digit	1-10	2-30
24	EAN/UCC Code 128	1-10	0
25	UPC-A	2-4	0
26	UPC-A with 2 digit add-on	2-4	0
27	UPC-A with 5 digit add-on	2-4	0
28	UPC-E	2-4	0
29	UPC-E with 2 digit add-on	2-4	0
30	UPC-E with 5 digit add-on	2-4	0
31	UPC Interleaved 2 of 5	1-10	2-30

- *nNarrowBar*

  Narrow bar width in dots. (See above table)

- *nWideBar*

  Wide bar width in dots. Acceptable values are 2-30. (See above table)

- *bHuman*

  Print human readable code. 1=Enable, 0=Disable.

- *nDataLen*

  Length of printing out data.

- *pData*

  Print out data.

Remarks

Example

char acPortBuffer[40];

```
int nPortCount = PortEnumCount(0); // Get USB count

PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

if(PortOpen(acPortBuffer) == 0)

{

        AfxMessageBox(_T("Can't open port!"));

        return;

}


SetMeasurement(1, 0); // inchs
PrintBarcode(1.0f, 1.0f, 0.3f, 0, 1, 1, 2, TRUE, 5, "12345"); //fBarHeight=0.3,
nRotation=0, nBarType=Code39, nNarrowBar=1, nWideBar=2, bHuman=Enabl
PrintLabel(1, 1, 0);
PortClose();
```

See Also

Printing Functions, SetMeasurement

PrintBox

This function can print a box.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall PrintBox(float fX1, float fY1, int nThickness, float
fX2, float fY2);
```

Visual Basic

```
Private Declare Function PrintBox Lib "SBSDK_API.DLL" (ByVal fX1 As Single, ByVal fY1 As Single,
ByVal nThickness As Long, ByVal fX2 As Single, ByVal fY2 As Single) As Long
```

Delphi

```
function PrintBox (fX1:Single; fY1:Single; nThickness:integer; fX2:Single;
fY2:Single):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool PrintBox(float fX1, float fY1, int nThickness, float fX2, float fY2);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Parameters

- *fX1*

  Horizontal start position (X) in library measurement.

- *fY1*

  Vertical start position (Y) in library measurement.

- *nThickness*

  Line thickness in dots.

- *fX2*

  Horizontal end position (X) in library measurement.

- *fY2*

  Vertical end position (Y) in library measurement.

Remarks

Example

```
char acPortBuffer[40];

int nPortCount = PortEnumCount(0); // Get USB count

PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

if(PortOpen(acPortBuffer) == 0)

{

        AfxMessageBox(_T("Can't open port!"));
        return;
    }


    SetMeasurement(1, 0); // inchs
    PrintBox(0.5f, 0.5f, 12, 1.0f, 1.0f); // (0.5, 0.5) to (1, 1) inch with 12 dots thickness
    PrintLabel(1, 1, 0);
    PortClose();
```

See Also

Printing Functions, SetMeasurement

PrintDataMatrix

This function can print a DataMatrix barcode.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall PrintDataMatrix(float fX, float fY, int nRotation, int
nColumns, int nRows, int nSquare, BOOL bInverseImage, int nDataLen, char* pData);
```

Visual Basic

```
Private Declare Function PrintDataMatrix Lib "SBSDK_API.DLL" (ByVal fX As Single, ByVal fY As
Single, ByVal nRotation As Long, ByVal nColumns As Long, ByVal nRows As Long, ByVal nSquare As
Long, ByVal bInverseImage As Long, ByVal nDataLen As Long, ByVal pData As String) As Long
```

Delphi

```
function PrintDataMatrix (fX:Single; fY:Single; nRotation:integer; nColumns:integer; nRows:integer;
nSquare:integer; bInverseImage:integer; nDataLen:integer; pData:string):integer;stdcall;external
'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool PrintDataMatrix(float fX, float fY, int nRotation, int nColumns, int nRows, int
nSquare, int bInverseImage, int nDataLen, string pData);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Parameters

- *fX*

  Horizontal start position (X) in library measurement.

- *fY*

  Vertical start position (Y) in library measurement.

- *nRotation*

Value	Description
0	No rotation
1	90 degrees
2	180 degrees
3	270 degrees

- *nColumns*

  Number of columns to encode.

- *nRows*

  Number of rows to encode.

- *nSquare*

  Sets the minimum square data module size used for encoding data. Range: 1~40

- *bInverseImage*

  Selects an inverse image of the barcode. 1=inverse, 0=normal.

- *nDataLen*

  Length of printing out data.

- *pData*

  Print out data.

Remarks

Example

```
int nRotation = 0;
int nColumns = 10;
int nRows = 10;
int nSquare = 4;
BOOL bInverseImage = FALSE;
int nDataLen = 5;
char *pData = "12345";

char acPortBuffer[40];
int nPortCount = PortEnumCount(0); // Get USB count
PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port
if(PortOpen(acPortBuffer) == 0)
{
        AfxMessageBox(_T("Can't open port!"));
        return;
}

SetMeasurement(1, 0); // inchs

PrintDataMatrix(0.8f, 0.8f, nRotation, nColumns, nRows, nSquare, bInverseImage,
nDataLen, pData);

PrintLabel(1, 1, 0);
PortClose();
```

See Also

Printing Functions, SetMeasurement

PrintLine

This function can print a line.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall PrintLine(float fX, float fY, float fHoriLength, float
fVertLength, int nDrawMethod);
```

Visual Basic

```
Private Declare Function PrintLine Lib "SBSDK_API.DLL" (ByVal fX As Single, ByVal fY As Single,
ByVal fHoriLength As Single, ByVal fVertLength As Single, ByVal nDrawMethod As Long) As Long
```

Delphi

```
function PrintLine (fX:Single; fY:Single; fHoriLength:Single; fVertLength:Single;
nDrawMethod:integer):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool PrintLine(float fX, float fY, float fHoriLength, float fVertLength, int
nDrawMethod);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Parameters

- *fX*

  Horizontal start position (X) in library measurement.

- *fY*

  Vertical start position (Y) in library measurement.

- *fHoriLength*

  Horizontal length in library measurement.

- *fVertLength*

  Vertical length in library measurement.

- *nDrawMethod*

Value	Description
0	Line Draw Black
1	Line Draw White
2	Line Draw Exclusive OR

Remarks

Example

```
char acPortBuffer[40];

int nPortCount = PortEnumCount(0); // Get USB count

PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port
```

```
if(PortOpen(acPortBuffer) == 0)
{
        AfxMessageBox(_T("Can't open port!"));
        return;
}

SetMeasurement(0, 0); // dots
PrintLine(50, 100, 200, 5, 0); // Black Line
PrintLine(100, 50, 5, 200, 0); // Black Line
PrintLine(150, 50, 5, 200, 1); // White Line
PrintLine(200, 50, 5, 200, 2); // Exclusive OR Line
PrintLabel(1, 1, 0);
PortClose();
```

See Also

Printing Functions, SetMeasurement

PrintImage

Send an image file to printer and print out it with monochrome.

C++

```
BOOL PrintImage(float fX, float fY, float fWidth, float fHeight, int nRotation, BOOL bErrorDiffusion,
char* pFileName);
```

Visual Basic

```
Private Declare Function PrintImage Lib "SBSDK_API.DLL" (ByVal fX As Single, ByVal fY As Single,
ByVal fWidth As Single, ByVal fHeight As Single, ByVal nRotation As Long, ByVal bErrorDiffusion As
Long, ByVal pFileName As String) As Long
```

Delphi

```
function PrintImage (fX:Single; fY:Single; fWidth:Single; fHeight:Single; nRotation:integer;
bErrorDiffusion:integer; pFileName:string):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool PrintImage(float fX, float fY, float fWidth, float fHeight, int nRotation, int
bErrorDiffusion, string pFileName);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Parameters

- *fX*

  Horizontal start position (X) in library measurement.

- *fY*

  Vertical start position (Y) in library measurement.

- *fWidth*

  Image Width in library measurement. If set -1, it will be original size.

- *fHeight*

  Image Height in library measurement. If set -1, it will be original size.

- *nRotation*

Value	Description
0	No rotation
1	90 degrees
2	180 degrees
3	270 degrees

- *bErrorDiffusion*

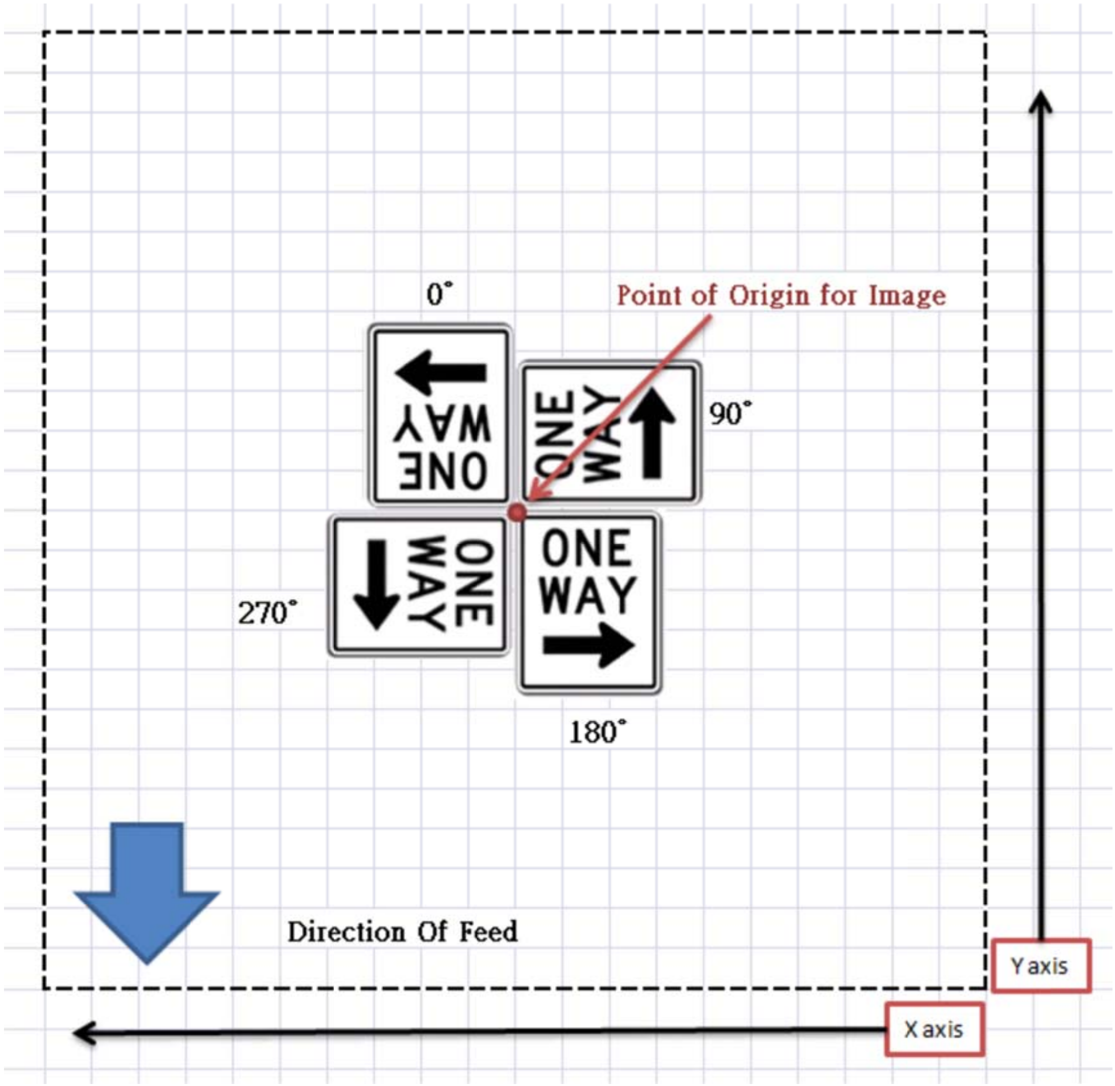
  If set 1, this function will decreases color to monochrome with error diffusion mode.

- *pFileName*

  Image file name with path. It supports gif, jpg, jif, jpeg, png, tif, tiff, ico, bmp, dib and pcx formats.

Remarks

Printing drawing.



Print out color Image.

Original image	Decrease color to monochrome	Decrease color to monochrome with error-diffusion

Example

```
char acPortBuffer[40];

int nPortCount = PortEnumCount(0); // Get USB count

PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

if(PortOpen(acPortBuffer) == 0)

{

        AfxMessageBox(_T("Can't open port!"));

        return;

}


SetMeasurement(1, 0); // inchs
PrintImage(0.5f, 0.2f, -1, -1, 0, 1, "TestImage.jpg"); // nWidth=-1(Original Size),
nHeight=-1(Original Size), bErrorDiffusion=1
PrintImage(3.0f, 0.2f, 1.0f, 1.0f, 0, 0, "TestImage.jpg"); // nWidth=-1(Original Size),
nHeight=-1(Original Size), bErrorDiffusion=1
PrintLabel(1, 1, 0);
PortClose();
```

See Also

Printing Functions, SetMeasurement

PrintMaxiCode

This function can print a MaxiCode barcode.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall PrintMaxiCode(float fX, float fY, int nRotation, int
nMode, int nDataLen, char* pData);
```

Visual Basic

```
Private Declare Function PrintMaxiCode Lib "SBSDK_API.DLL" (ByVal fX As Single, ByVal fY As
Single, ByVal nRotation As Long, ByVal nMode As Long, ByVal nDataLen As Long, ByVal pData As
String) As Long
```

Delphi

```
function PrintMaxiCode (fX:Single; fY:Single; nRotation:integer; nMode:integer; nDataLen:integer;
pData:string):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool PrintMaxiCode(float fX, float fY, int nRotation, int nMode, int nDataLen,
string pData);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Parameters

- *fX*

  Horizontal start position (X) in library measurement.

- *fY*

  Vertical start position (Y) in library measurement.

- *nRotation*

Value	Description
0	No rotation
1	90 degrees
2	180 degrees
3	270 degrees

- *nMode*

  Mode Selection.

Value	Description
0	Automatic selection mode 2 or 3
2	Mode 2
3	Mode 3
4	Mode 4
6	Mode 6

- *nDataLen*

Length of printing out data.

- *pData*

  Print out data.

Remarks

Example

```
char acPortBuffer[40];
int nPortCount = PortEnumCount(0); // Get USB count
PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port
if(PortOpen(acPortBuffer) == 0)
{
    AfxMessageBox(_T("Can't open port!"));
    return;
}

SetAfterPrint (1, 0); // Enable Tear-Off mode after printing.
SetMeasurement(1, 200); // inches, 200 dpi printer
PrintBarcode(1.0f, 1.0f, 0.3f, 0, 1, 1, 2, TRUE, 5, "12345"); //fBarHeight=0.3,
nRotation=0, nBarType=Code39, nNarrowBar=1, nWideBar=2, bHuman=Enabl
PortClose();
```

See Also

Printing Functions, SetMeasurement

PrintParagraphA, PrintParagraphW

PrintParagraphA can print a paragraph with true type font.

PrintParagraphW can print a Unicode parpgraph with true type font.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall PrintParagraphA(float fX, float fY, float fRangeWidth,
int nAlignment, int nFontH, int nFontW, int nRotation, int nStyle, BOOL bUnderline, BOOL
bWhitOnBlack, char* pFaceName, char* pText);

extern "C" __declspec(dllimport) BOOL _stdcall PrintParagraphW(float fX, float fY, float
fRangeWidth, int nAlignment, int nFontH, int nFontW, int nRotation, int nStyle, BOOL bUnderline,
BOOL bWhitOnBlack, TCHAR* ptcFaceName, TCHAR* ptcText);
```

Visual Basic

```
Private Declare Function PrintParagraphA Lib "SBSDK_API.DLL" (ByVal fX As Single, ByVal fY As
Single, ByVal fRangeWidth As Single, ByVal nAlignment As Integer , ByVal nFontH As Long, ByVal
nFontW As Long, ByVal nRotation As Long, ByVal nStyle As Long, ByVal bUnderline As Long, ByVal
bWhitOnBlack As Long, ByVal pFaceName As String, ByVal pText As String) As Long
```

Delphi

```
function PrintParagraphA (fX:Single; fY:Single; fRangeWidth:Single; nAlignment: integer;
nFontH:integer; nFontW:integer; nRotation:integer; nStyle:integer; bUnderline:integer;
bWhitOnBlack:integer; pFaceName:string; pText:string):integer;stdcall;external 'SBSDK_API.DLL';

function PrintParagraphW (fX:Single; fY:Single; fRangeWidth:Single; nAlignment: integer;
nFontH:integer; nFontW:integer; nRotation:integer; nStyle:integer; bUnderline:integer;
bWhitOnBlack:integer; pFaceName:WideString; pText:WideString):integer;stdcall;external
'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool PrintParagraphA(float fX, float fY, float fRangeWidth, int nAlignment, int
nFontH, int nFontW, int nRotation, int nStyle, int bUnderline, int bWhitOnBlack, string pFaceName,
string pText);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Parameters

- *fX*

  Horizontal start position (X) in library measurement.

- *fY*

  Vertical start position (Y) in library measurement.

- *fRangeWidth*

  The width of the printable area in horizontal alignment. (By measurement)

- *nAlignment*

Value	Description
0	Right Side
1	Centered
2	Left Side

- *nFontH*

  Font height in point size. (like as office word)

- *nFontW*

  Font average width. If set it -1, it will use default width.

- *nRotation*

Value	Description
0	No rotation
1	90 degrees
2	180 degrees
3	270 degrees

- *nStyle*

Value	Description
0	Regular
1	Italic
2	Bold
3	Bold and Italic

- *bUnderline*

  Specifies an underlined font if set to 1.

- *bWhitOnBlack*

  Specifies a black background and white text font if set to 1.

- *pFaceName*

  Specifies a null-terminated string that specifies the typeface name of the font.

- *pText*

  Print out data.

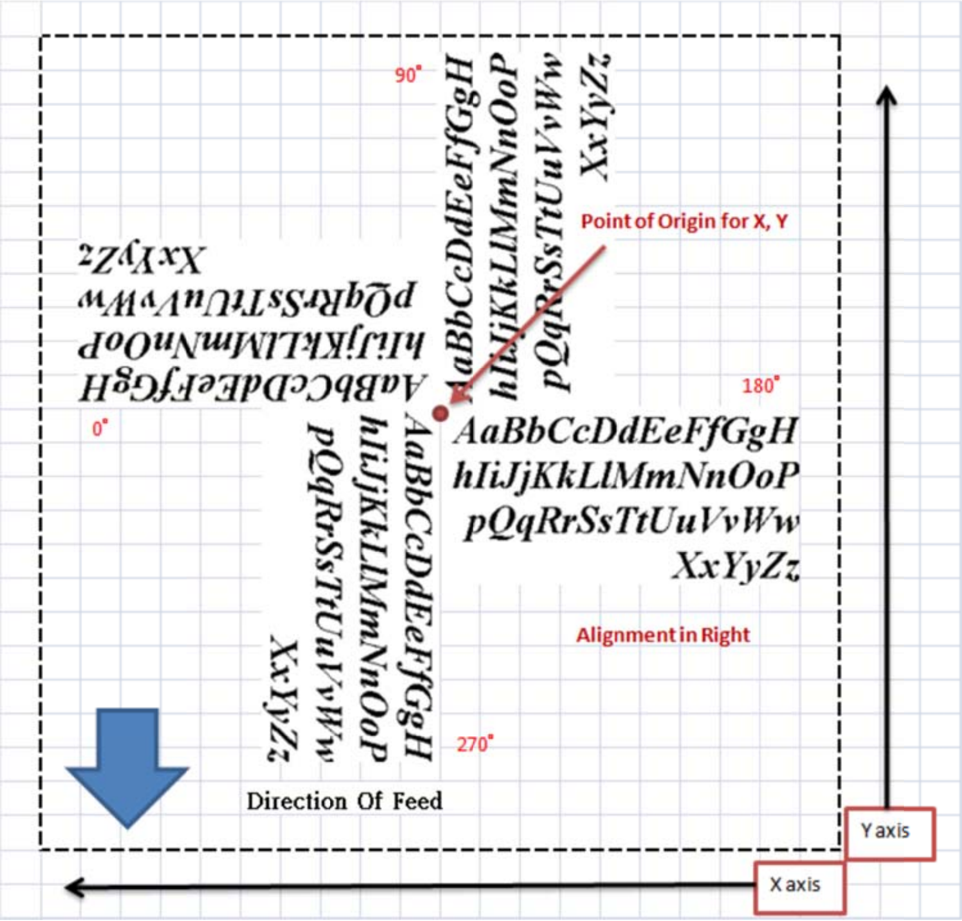
- *ptcText*

  Print out data that supports Unicode.

Remarks

- **If you want to use Unicode in VB, please chose the PrintParagraphW () function.**

- Printing drawing.



Example

```
char acPortBuffer[40];

int nPortCount = PortEnumCount(0); // Get USB count

PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

if(PortOpen(acPortBuffer) == 0)

{

        AfxMessageBox(_T("Can't open port!"));

        return;

}


SetMeasurement(1, 0); // inchs

SetMedia_LabelWithGaps(4, 3, 0.16f, 0); // W=4 inch, H=3Inch, GapLen=32 dots,
GapOffset=0;

SetPrinter(3, 8, 1); // nSpeed=3, nDarkness=8, bDirectThermal=1;
```

```
PrintParagraphA(2.0f, 1.5f, 1.5, 2, 12, -1, 0, 3, 0, 0, "Times New Roman",
"AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz"); // nFontH=12,
nFontW=-1, nRotation=0, nStyle=3(Bold and Italic), bUnderline=0, bWhitOnBlack=1
PrintParagraphA(2.0f, 1.5f, 1.5, 2, 12, -1, 1, 3, 0, 0, "Times New Roman",
"AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz"); // nFontH=12,
nFontW=-1, nRotation=0, nStyle=3(Bold and Italic), bUnderline=0, bWhitOnBlack=1
PrintParagraphA(2.0f, 1.5f, 1.5, 2, 12, -1, 2, 3, 0, 0, "Times New Roman",
"AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz"); // nFontH=12,
nFontW=-1, nRotation=0, nStyle=3(Bold and Italic), bUnderline=0, bWhitOnBlack=1
PrintParagraphA(2.0f, 1.5f, 1.5, 2, 12, -1, 3, 3, 0, 0, "Times New Roman",
"AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz"); // nFontH=12,
nFontW=-1, nRotation=0, nStyle=3(Bold and Italic), bUnderline=0, bWhitOnBlack=1
PrintLabel(1, 1, 0);
PortClose();
```

See Also

Printing Functions, SetMeasurement

PrintPDF417

This function can print a PDF-417 barcode.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall PrintPDF417(float fX, float fY, int nRotation, int nCorrectionLevel, BOOL bCompression, int nModuleWidth, int nBarHeight, int nMaxColumns, int nMaxRows, int nTruncated, int nDataLen, char* pData)
```

Visual Basic

```
Private Declare Function PrintPDF417 Lib "SBSDK_API.DLL" (ByVal fX As Single, ByVal fY As Single, ByVal nRotation As Long, ByVal nCorrectionLevel As Long, ByVal bCompression As Long, ByVal nModuleWidth As Long, ByVal nBarHeight As Long, ByVal nMaxColumns As Long, ByVal nMaxRows As Long, ByVal nTruncated As Long, ByVal nDataLen As Long, ByVal pData As String) As Long
```

Delphi

```
function PrintPDF417 (fX:Single; fY:Single; nRotation:integer; nCorrectionLevel:integer; bCompression:integer; nModuleWidth:integer; nBarHeight:integer; nMaxColumns:integer; nMaxRows:integer; nTruncated:integer; nDataLen:integer; pData:string):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool PrintPDF417(float fX, float fY, int nRotation, int nCorrectionLevel, int bCompression, int nModuleWidth, int nBarHeight, int nMaxColumns, int nMaxRows, int nTruncated, int nDataLen, string pData);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Parameters

- *fX*

  Horizontal start position (X) in library measurement.

- *fY*

  Vertical start position (Y) in library measurement.

- *nRotation*

Value	Description
0	No rotation
1	90 degrees
2	180 degrees
3	270 degrees

- *nCorrectionLevel*

  Sets error correction level Error Correction codewords per symbol, Values: 0~8

- *bCompression*

  Selects data compaction (compression) method, Values: 0=Auto-encoding, 1=Binary-mode

- *nModuleWidth*

Maximum print width in dots, Range: 2~9

- *nBarHeight*

  Set bar height (in dots), Range:4~99

- *nMaxColumns*

  Maximum column count, Range: 1~30

- *nMaxRows*

  Maximum row count, Range: 4~90 (When data is one character, the value must be started from 4.

- *nTruncated*

  Truncated flag - legal values: 0=not truncated, 1=truncated

- *nDataLen*

  Length of printing out data.

- *pData*

  Print out data.

Remarks

Example

```
int nCorrectionLevel = 1;
BOOL bCompression = 0;
int nModuleWidth = 1;
int nBarHeight = 4;
int nMaxColumns = 1;
int nMaxRows = 10;
int nTruncated = 1;
int nDataLen = 5;
char *pData = "12345";

char acPortBuffer[40];
int nPortCount = PortEnumCount(0); // Get USB count
PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port
if(PortOpen(acPortBuffer) == 0)
{
    AfxMessageBox(_T("Can't open port!"));
    return;
}
```

```
SetMeasurement(1, 0); // inchs

PrintPDF417(0.2f, 0.2f, 0, nCorrectionLevel, bCompression, nModuleWidth, nBarHeight,
nMaxColumns, nMaxRows, nTruncated, nDataLen, pData); // nRotation=0,

PrintLabel(1, 1, 0);
PortClose();
```

See Also

Printing Functions, SetMeasurement

PrintQRcode

This function can print a QR code barcode.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall PrintQRcode(float fX, float fY, int nRotation, int nErrorCorrection, int nModel, int nMaskNumber, int nPixel, int nDataLen, char* pData);
```

Visual Basic

```
Private Declare Function PrintQRcode Lib "SBSDK_API.DLL" (ByVal fX As Single, ByVal fY As Single, ByVal nRotation As Integer, ByVal nErrorCorrection As Integer, ByVal nModel As Integer, ByVal nMaskNumber As Integer, ByVal nPixel As Integer, ByVal nDataLen As Integer, ByVal pData As String) As Integer
```

Delphi

```
function PrintQRcode (fX:Single; fY:Single; nRotation:integer; nErrorCorrection:integer; nModel:integer; nMaskNumber:integer; nPixel:integer; nDataLen:integer; pData:string):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool PrintQRcode(float fX, float fY, int nRotation, int nErrorCorrection, int nModel, int nMaskNumber, int nPixel, int nDataLen, string pData);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Parameters

- *fX*

  Horizontal start position (X) in library measurement.

- *fY*

  Vertical start position (Y) in library measurement.

- *nRotation*

Value	Description
0	No rotation
1	90 degrees
2	180 degrees
3	270 degrees

- *nErrorCorrection*

  Use the Reed–Solomon error correction algorithm with four error correction levels. The higher the error correction level, the less storage capacity. The following table lists the approximate error correction capability at each of the four levels:

Value	Description
0	L - Recovery 7%
1	M - Recovery 15%
2	Q - Recovery 25%

3	H - Recovery 30%

- *nModel*

  Sets QR code mode, normally the value is 1 or 2, Values: 0~8.

- *nMaskNumber*

  Set the mask number of QR code. Values: 0 ~ 7

- *nPixel*

  The QR code base dots. Value: 1 ~ 20

- *nDataLen*

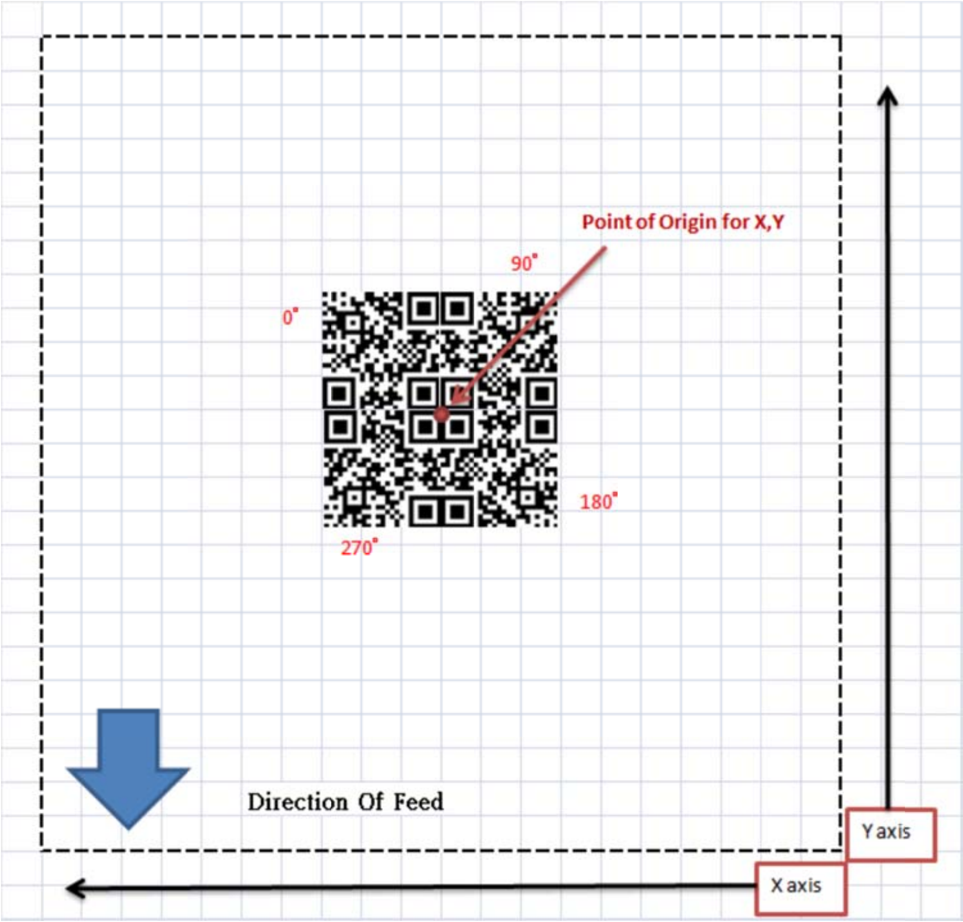
  Length of printing out data.

- *pData*

  Print out data.

Remarks

Printing drawing.



Example

```
int nErrorCorrection = 0;
int nModel = 2;
int nMaskNumber = 0;
int nPixel = 4;
int nDataLen = 5;
char *pData = "12345";
char acPortBuffer[40];
int nPortCount = PortEnumCount(0); // Get USB count
PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port
if(PortOpen(acPortBuffer) == 0)
{
    AfxMessageBox(_T("Can't open port!"));
    return;
}

SetMeasurement(1, 0); // inchs
SetMedia_LabelWithGaps(4.f, 3.0f, 0.16f, 0); // W=4 inch, H=3Inch, GapLen=32 dots, GapOffset=0;
SetPrinter(3, 8, 1); // nSpeed=3, nDarkness=8, bDirectThermal=1;
PrintQRcode(2.0f, 1.5f, 0, nErrorCorrection, nModel, nMaskNumber, nPixel, nDataLen, pData);
PrintQRcode(2.0f, 1.5f, 1, nErrorCorrection, nModel, nMaskNumber, nPixel, nDataLen, pData);
PrintQRcode(2.0f, 1.5f, 2, nErrorCorrection, nModel, nMaskNumber, nPixel, nDataLen, pData);
PrintQRcode(2.0f, 1.5f, 3, nErrorCorrection, nModel, nMaskNumber, nPixel, nDataLen, pData);
PrintLabel(1, 1, 0);
PortClose();
```

See Also

Printing Functions, SetMeasurement

PrintText

This function can print a text with printer internal font.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall PrintText(float fX, float fY, int nRotation, int
nFontID, int nFontHor, int nFontVer, BOOL bReverse, int nDataLen, char* pData);
```

Visual Basic

```
Private Declare Function PrintText Lib "SBSDK_API.DLL" (ByVal fX As Single, ByVal fY As Single,
ByVal nRotation As Long, ByVal nFontID As Long, ByVal nFontHor As Long, ByVal nFontVer As Long,
ByVal bReverse As Long, ByVal nDataLen As Long, ByVal pData As String) As Long
```

Delphi

```
function PrintText (fX:Single; fY:Single; nRotation:integer; nFontID:integer; nFontHor:integer;
nFontVer:integer; bReverse:integer; nDataLen:integer; pData:string):integer;stdcall;external
'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool PrintText(float fX, float fY, int nRotation, int nFontID, int nFontHor, int
nFontVer, int bReverse, int nDataLen, string pData);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Parameters

- *fX*

  Horizontal start position (X) in library measurement.

- *fY*

  Vertical start position (Y) in library measurement.

- *nRotation*

Value	Description
0	No rotation
1	90 degrees
2	180 degrees
3	270 degrees

- *nFontID*

Value	Description
1	20.3 cpi, 6 pts, (8 x 12 dots)
2	16.9 cpi, 7 pts, (10 x 16 dots)
3	14.5 cpi, 10 pts, (12 x 20 dots)
4	12.7 cpi, 12 pts, (14 x 24 dots)
5	5.6 cpi, 24 pts, (32 x 48 dots)
6	Numeric Only, (14 x 19 dots)
7	Numeric Only, (14 x 19 dots)

97~122	Soft font

- *nFontHor*

Horizontal multiplier expands the text horizontally. Values: 1-8.

- *nFontVer*

Vertical multiplier expands the text vertically. Values: 1-9.

- *bReverse*

Value	Description
0	Normal
1	Reverse Image

- *nDataLen*

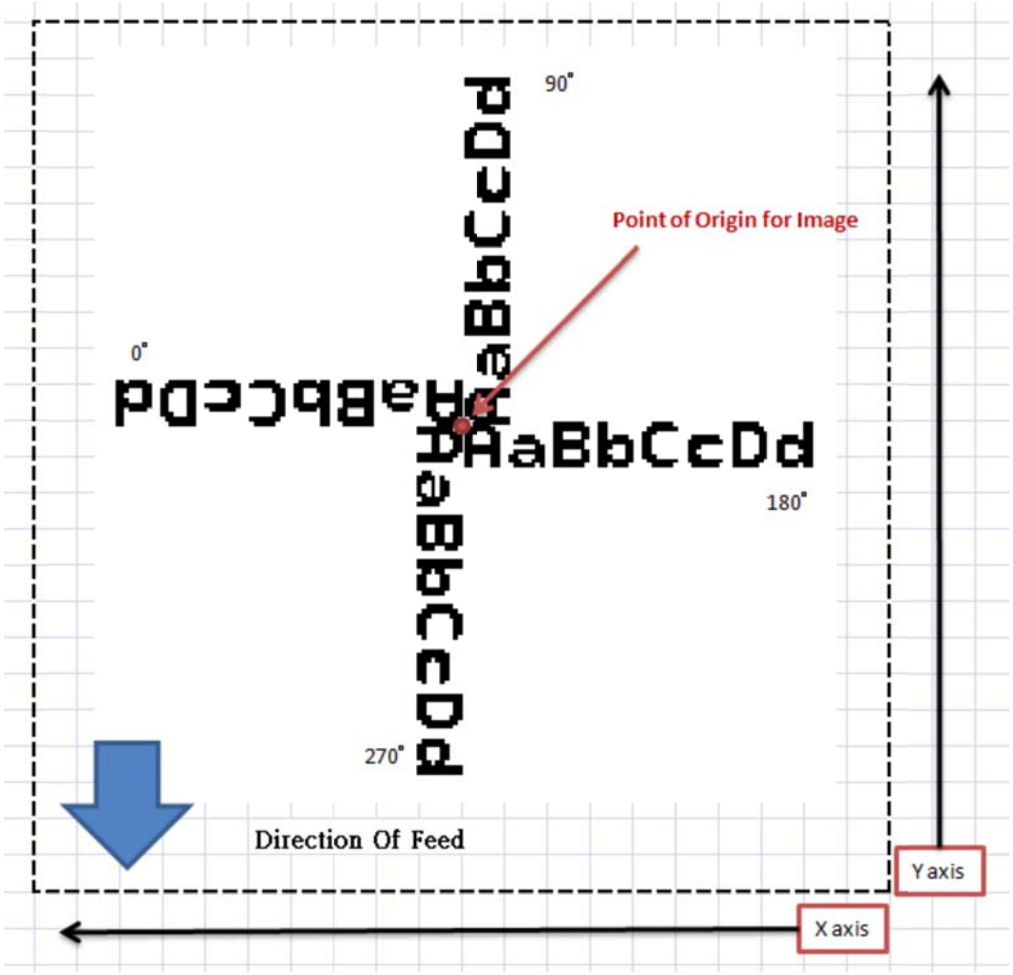
Length of printing out data.

- *pData*

Print out data.

Remarks

Printing drawing.



Example (VC)

```
char acPortBuffer[40];

int nPortCount = PortEnumCount(0); // Get USB count

PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

if(PortOpen(acPortBuffer) == 0)

{

        AfxMessageBox(_T("Can't open port!"));

        return;

}


SetMeasurement(1, 200); // inches, 200 dpi printer

PrintText(2.1f, 1.1f, 0, 1, 1, 1, 0, 8, “AaBbCcDd”); // X=2.1, Y=1.1 (inches), No rotation,
Font1, Hor Expand=1, Ver Expand=1, normal image

PrintLabel(0, 0, 0); // Only send data to printer without printing

PortClose();
```

See Also

Printing Functions, SetMeasurement

PrintTrueTypeFontA, PrintTrueTypeFontW

PrintTrueTypeA can print a normal text with true type font.

PrintTrueTypeW can print a Unicode text with true type font.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall PrintTrueTypeFontA(float fX, float fY, int nFontH, int
nFontW, int nRotation, int nStyle, BOOL bUnderline, BOOL bWhitOnBlack, char* pFaceName, char*
pText);

extern "C" __declspec(dllimport) BOOL _stdcall PrintTrueTypeFontW(float fX, float fY, int nFontH, int
nFontW, int nRotation, int nStyle, BOOL bUnderline, BOOL bWhitOnBlack, TCHAR* ptcFaceName,
TCHAR* ptcText);
```

Visual Basic

```
Private Declare Function PrintTrueTypeFontA Lib "SBSDK_API.DLL" (ByVal fX As Single, ByVal fY As
Single, ByVal nFontH As Long, ByVal nFontW As Long, ByVal nRotation As Long, ByVal nStyle As
Long, ByVal bUnderline As Long, ByVal bWhitOnBlack As Long, ByVal pFaceName As String, ByVal
pText As String) As Long
```

Delphi

```
function PrintTrueTypeFontA (fX:Single; fY:Single; nFontH:integer; nFontW:integer;
nRotation:integer; nStyle:integer; bUnderline:integer; bWhitOnBlack:integer; pFaceName:string;
pText:string):integer;stdcall;external 'SBSDK_API.DLL';

function PrintTrueTypeFontW (fX:Single; fY:Single; nFontH:integer; nFontW:integer;
nRotation:integer; nStyle:integer; bUnderline:integer; bWhitOnBlack:integer;
pFaceName:WideString; pText:WideString):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool PrintTrueTypeFontA(float fX, float fY, int nFontH, int nFontW, int nRotation,
int nStyle, int bUnderline, int bWhitOnBlack, string pFaceName, string pText);

        [DllImport("SBSDK_API.dll")]
        static extern bool PrintTrueTypeFontW(float fX, float fY, int nFontH, int nFontW, int nRotation,
int nStyle, int bUnderline, int bWhitOnBlack, string ptcFaceName, string ptcText);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Parameters

- *fX*

  Horizontal start position (X) in library measurement.

- *fY*

  Vertical start position (Y) in library measurement.

- *nFontH*

  Font height in point size. (like as office word)

- *nFontW*

  Font average width. If set it -1, it will use default width.

- *nRotation*

Value	Description
0	No rotation
1	90 degrees
2	180 degrees
3	270 degrees

- *nStyle*

Value	Description
0	Regular
1	Italic
2	Bold
3	Bold and Italic

- *bUnderline*

Specifies an underlined font if set to 1.

- *bWhitOnBlack*

Specifies a black background and white text font if set to 1.

- *pFaceName*

Specifies a null-terminated string that specifies the typeface name of the font.

- *pText*

Print out data.

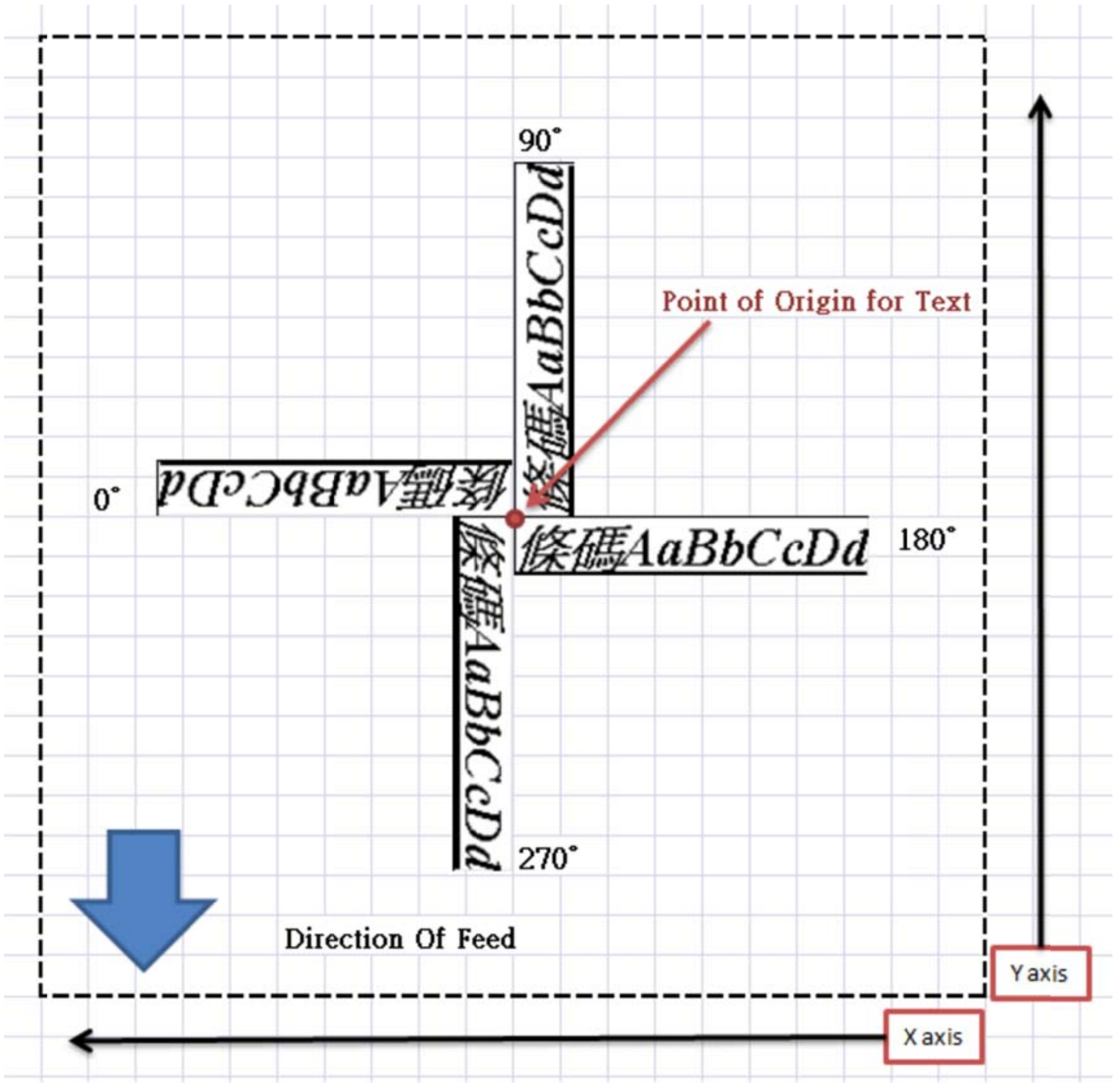
- *ptcText*

Print out data that supports Unicode.

Remarks

- **If you want to use Unicode in VB, please chose the PrintTrueTypeA() function.**

- Printing drawing.



Example

```
char acPortBuffer[40];

int nPortCount = PortEnumCount(0); // Get USB count

PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

if(PortOpen(acPortBuffer) == 0)

{

    AfxMessageBox(_T("Can't open port!"));

    return;

}


SetMeasurement(1, 0); // inchs
```

```
PrintTrueTypeFontA(1.0f, 1.0f, 12, -1, 0, 3, 0, 1, "Times New Roman", "AaBbCcDd"); // nFontH=12, nFontW=-1, nRotation=0, nStyle=3(Bold and Italic), bUnderline=0, bWhitOnBlack=1

PrintTrueTypeFontW(1.0f, 2.0f, 12, -1, 0, 3, 1, 0, _T("Times New Roman"), _T("AaBbCcDd")); // nFontH=12, nFontW=-1, nRotation=0, nStyle=3(Bold and Italic), bUnderline=1, bWhitOnBlack=0

PrintLabel(1, 1, 0);

PortClose();
```

See Also

Printing Functions, SetMeasurement

PrintTTF_AlignmentA, PrintTTF_AlignmentW

PrintTTF_AlignmentA can print a true type font text with alignment..

PrintTTF_AlignmentW can print a true type font Unicode text with alignment.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall PrintTTF_AlignmentA(float fX, float fY, float
fRangeWidth, int nAlignment, int nFontH, int nFontW, int nRotation, int nStyle, BOOL bUnderline,
BOOL bWhitOnBlack, char* pFaceName, char* pText);

extern "C" __declspec(dllimport) BOOL _stdcall PrintParagraphW(float fX, float fY, float
fRangeWidth, int nAlignment, int nFontH, int nFontW, int nRotation, int nStyle, BOOL bUnderline,
BOOL bWhitOnBlack, TCHAR* ptcFaceName, TCHAR* ptcText);
```

Visual Basic

```
Private Declare Function PrintTTF_AlignmentA Lib "SBSDK_API.DLL" (ByVal fX As Single, ByVal fY As
Single, ByVal fRangeWidth As Single, ByVal nAlignment As Integer, ByVal nFontH As Long, ByVal
nFontW As Long, ByVal nRotation As Long, ByVal nStyle As Long, ByVal bUnderline As Long, ByVal
bWhitOnBlack As Long, ByVal pFaceName As String, ByVal pText As String) As Long
```

Delphi

```
function PrintTTF_AlignmentA (fX:Single; fY:Single; fRangeWidth:Single; nAlignment: integer;
nFontH:integer; nFontW:integer; nRotation:integer; nStyle:integer; bUnderline:integer;
bWhitOnBlack:integer; pFaceName:string; pText:string):integer;stdcall;external 'SBSDK_API.DLL';

function PrintTTF_AlignmentW (fX:Single; fY:Single; fRangeWidth:Single; nAlignment: integer;
nFontH:integer; nFontW:integer; nRotation:integer; nStyle:integer; bUnderline:integer;
bWhitOnBlack:integer; pFaceName:WideString; pText:WideString):integer;stdcall;external
'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool PrintTTF_AlignmentA(float fX, float fY, float fRangeWidth, int nAlignment, int
nFontH, int nFontW, int nRotation, int nStyle, int bUnderline, int bWhitOnBlack, string pFaceName,
string pText);

        [DllImport("SBSDK_API.dll")]
        static extern bool PrintTTF_AlignmentW(float fX, float fY, float fRangeWidth, int nAlignment, int
nFontH, int nFontW, int nRotation, int nStyle, int bUnderline, int bWhitOnBlack, string ptcFaceName,
string ptcText);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Parameters

- *fX*

  Horizontal start position (X) in library measurement.

- *fY*

  Vertical start position (Y) in library measurement.

- *fRangeWidth*

  The width of the printable area in horizontal alignment. (By measurement)

- *nAlignment*

Value	Description

0	Right Side
1	Centered
2	Left Side

- *nFontH*

  Font height in point size. (like as office word)

- *nFontW*

  Font average width. If set it -1, it will use default width.

- *nRotation*

Value	**Description**
0	No rotation
1	90 degrees
2	180 degrees
3	270 degrees

- *nStyle*

Value	**Description**
0	Regular
1	Italic
2	Bold
3	Bold and Italic

- *bUnderline*

  Specifies an underlined font if set to 1.

- *bWhitOnBlack*

  Specifies a black background and white text font if set to 1.

- *pFaceName*

  Specifies a null-terminated string that specifies the typeface name of the font.

- *pText*

  Print out data.

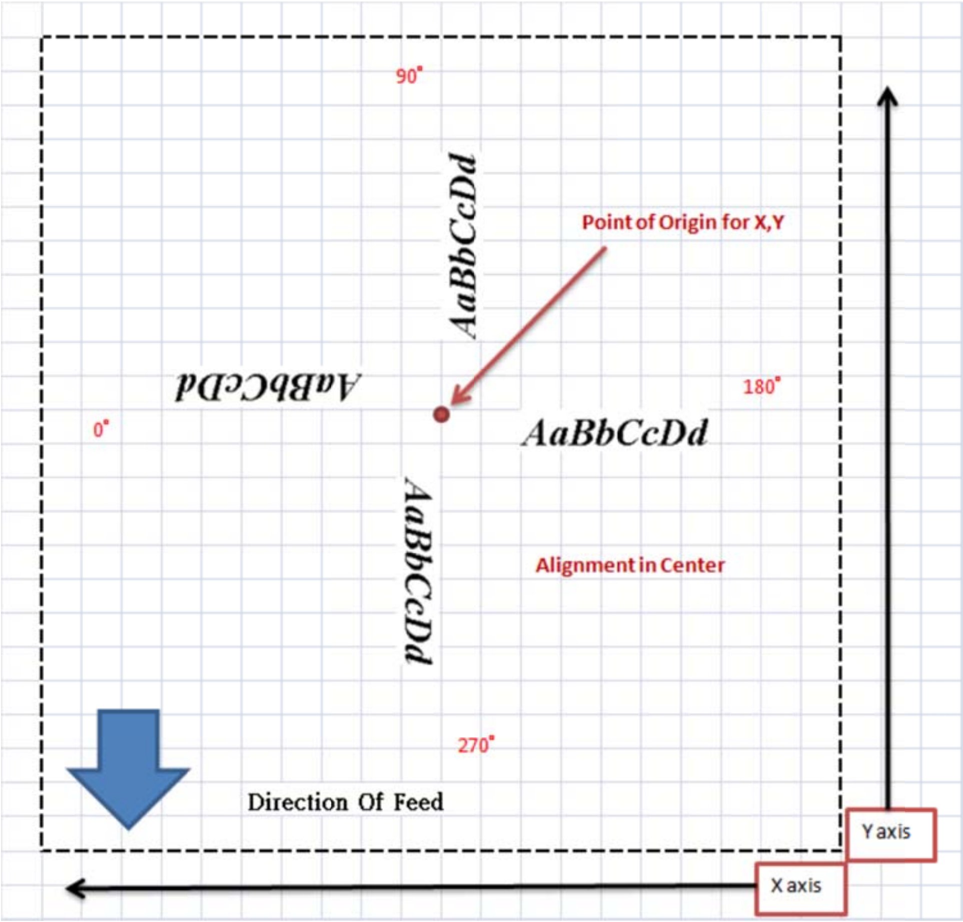
- *ptcText*

  Print out data that supports Unicode.

Remarks

- **If you want to use Unicode in VB, please chose the PrintTrueTypeA() function.**

- Printing drawing.



Example

```
char acPortBuffer[40];
int nPortCount = PortEnumCount(0); // Get USB count
PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port
if(PortOpen(acPortBuffer) == 0)
{
        AfxMessageBox(_T("Can't open port!"));
        return;
}

SetMeasurement(1, 0); // inchs
SetMedia_LabelWithGaps(4, 3, 0.16f, 0); // W=4 inch, H=3Inch, GapLen=32 dots, GapOffset=0;
SetPrinter(3, 8, 1); // nSpeed=3, nDarkness=8, bDirectThermal=1;
```

```
PrintTTF_AlignmentA(2.0f, 1.5f, 1.5, 2, 12, -1, 0, 3, 0, 0, "Times New Roman", "AaBbCcDd"); // nFontH=12, nFontW=-1, nRotation=0, nStyle=3(Bold and Italic), bUnderline=0, bWhitOnBlack=0
PrintTTF_AlignmentA(2.0f, 1.5f, 1.5, 2, 12, -1, 1, 3, 0, 0, "Times New Roman", "AaBbCcDd"); // nFontH=12, nFontW=-1, nRotation=1, nStyle=3(Bold and Italic), bUnderline=0, bWhitOnBlack=0
PrintTTF_AlignmentA(2.0f, 1.5f, 1.5, 2, 12, -1, 2, 3, 0, 0, "Times New Roman", "AaBbCcDd"); // nFontH=12, nFontW=-1, nRotation=2, nStyle=3(Bold and Italic), bUnderline=0, bWhitOnBlack=0
PrintTTF_AlignmentA(2.0f, 1.5f, 1.5, 2, 12, -1, 3, 3, 0, 0, "Times New Roman", "AaBbCcDd"); // nFontH=12, nFontW=-1, nRotation=3, nStyle=3(Bold and Italic), bUnderline=0, bWhitOnBlack=0
PrintLabel(1, 1, 0);
PortClose();
```

See Also

Printing Functions, SetMeasurement

PrintLabel

This function can end functions of printing and setting to start printer to print patterns.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall PrintLabel(int nSet, int nCopy, BOOL bReverse);
```

Visual Basic

```
Private Declare Function PrintLabel Lib "SBSDK_API.DLL" (ByVal nSet As Long, ByVal nCopy As
Long, ByVal bReverse As Long) As Long
```

Delphi

```
function PrintLabel (nSet:integer; nCopy:integer; bReverse:integer):integer;stdcall;external
'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool PrintLabel(int nSet, int nCopy, int bReverse);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Parameters

- *nSet*

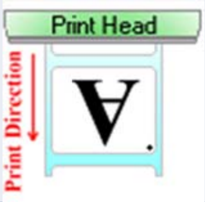
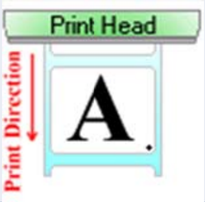
  Number of label sets. Range: 1 to 65535

- *nCopy*

  Number of copies of each label. Range: 1 to 65535

- *bReverse*

  Set the print orientation.

Value	= 0	= 1
Description	Normal 	Reversed 

Remarks

Example

```
char acPortBuffer[40];

int nPortCount = PortEnumCount(0); // Get USB count

PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

if(PortOpen(acPortBuffer) == 0)

{
        AfxMessageBox(_T("Can't open port!"));
        return;
    }

    SetAfterPrint (1, 0); // Enable Tear-Off mode after printing.
    PrintLabel(0, 0, 0); // Only send data to printer without printing
    PortClose();
```

See Also

Printing Functions, SetMeasurement

FORM FUNCTIONS

Name	Description
FormCheck	This function can check the form name from the printer to confirm that the form name already exists in the printer.
FormExecute	This function can perform printer form.
FormParameterData	This function can send the data of form parameter to the printer.
FormPrint	This function can send label set and label copies to printer to print the form.
FormSaveByFileA, FormSaveByFileW	This function can save form file to printer.

FormCheck

This function can check the form name from the printer to confirm that the form name already exists in the printer.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall FormCheck (char* pFormName);
```

Visual Basic

```
Private Declare Function FormSaveByFileA Lib "SBSDK_API.DLL" (ByVal pFileName As String) As Integer
```

Delphi

```
function FormCheck (pFormName: string):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool FormCheck(string pFormName);
```

Return Value

Returns TRUE if the form name is identical with printer's form name.

Parameters

- *pFormName*

  Form name.

Example (VC)

Please refer the example.

See Also

Form Functions, Port Functions

FormExecute

This function can perform printer form.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall FormExecute (char* pFormName);
```

Visual Basic

```
Private Declare Function FormExecute Lib "SBSDK_API.DLL" (ByVal pFormName As String) As Integer
```

Delphi

```
function FormExecute (pFormName: string):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool FormExecute(string pFormName);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Parameters

- *pFormName*

  Form name.

Example (VC)

```
char acPortBuffer[40];

int nPortCount = PortEnumCount(0); // Get USB count

PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

if(PortOpen(acPortBuffer) == 0)

{

        AfxMessageBox(_T("Can't open port!"));

        return;

}

//===== Send form file to printer

if(!FormCheck("TRUNCATION"))

{
        int nDataLen = FormSaveByFileA("Sample_Truncation_Form.prn");

        if(nDataLen == 0)

        {

            AfxMessageBox(_T("Can't open file!"));

            PortClose();

            return;

        }

    }

    //===== Send Parameter Data to printer

    FormExecute("TRUNCATION"); // Send command to printer to execute form by form name
"TRUNCATION".

    FormParameterData(“FF12345678”);

    FormParameterData(“RR87654321”);

    FormPrint(1,1);

    PortClose();

    AfxMessageBox(_T("--- Test End ---"));
```

See Also

Form Functions, Port Functions

FormParameterData

This function can send the data of form parameter to the printer.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall FormParameterData (char* pData);
```

Visual Basic

```
Private Declare Function FormParameterData Lib "SBSDK_API.DLL" (ByVal pData As String) As
Integer
```

Delphi

```
function FormParameterData (pData: string):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool FormParameterData(string pData);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Parameters

- *pData*

  Form parameter data.

Example (VC)

Please refer the example.

See Also

Form Functions, Port Functions

FormPrint

This function can send label set and label copies to printer to print the form.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall FormPrint (int nSet, int nCopy);
```

Visual Basic

```
Private Declare Function FormPrint Lib "SBSDK_API.DLL" (ByVal nSet As Integer, ByVal nCopy As
Integer) As Integer
```

Delphi

```
function FormPrint (nSet:integer; nCopy:integer):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool FormPrint(int nSet, int nCopy);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Parameters

- *nSet*

  Number of label sets. Range: 1 to 65535

- *nCopy*

  Number of copies of each label. Range: 1 to 65535

Example (VC)

Please refer the example.

See Also

Form Functions, Port Functions

FormSaveByFileA, FormSaveByFileW

This function can save form file to printer.

C++

```
extern "C" __declspec(dllimport) int _stdcall FormSaveByFileA(char* pFileName);
extern "C" __declspec(dllimport) int _stdcall FormSaveByFileW(TCHAR* ptcFileName);
```

Visual Basic

```
Private Declare Function FormSaveByFileA Lib "SBSDK_API.DLL" (ByVal pFileName As String) As
Integer
Private Declare Function FormSaveByFileW Lib "SBSDK_API.DLL" (ByVal ptcFileName As String) As
Integer
```

Delphi

```
function FormSaveByFileA (pFileName:string):integer;stdcall;external 'SBSDK_API.DLL';
function FormSaveByFileW (ptcFileName:WideString):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern int FormSaveByFileA(string pFileName);

[DllImport("SBSDK_API.dll")]
        static extern int FormSaveByFileW(string ptcFileName);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Parameters

- *pFileName*

  Form file name with path.

- *ptcFileName*

  Form file name with path that supports Unicode.

Example (VC)

Please refer the example.

See Also

Form Functions, Port Functions

DEMAND FUNCTIONS

Name	Description
DemandCalibrationTest	This function can demand the printer to run a calibration test.
DemandFactoryDefaults	This function can demand the printer to make itself to be factory defaults.
DemandLabelFeed	This function can demand the printer to feed out a blank label.
DemandPrintConfiguration	This function can demand the printer to print out a label with printer configuration.
DemandResetPrinter	This function can demand the printer to make a reset.

DemandCalibrationTest

This function can demand the printer to run a calibration test.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall DemandCalibrationTest();
```

Visual Basic

```
Private Declare Sub DemandCalibrationTest Lib "SBSDK_API.DLL" ()
```

Delphi

```
Procedure DemandCalibrationTest ();stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool DemandCalibrationTest();
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Example (VC)

```
char acPortBuffer[40];

int nPortCount = PortEnumCount(0); // Get USB count

PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

if(PortOpen(acPortBuffer) == 0)

{

        AfxMessageBox(_T("Can't open port!"));

        return;

}


DemandCalibrationTest(); // demand the printer to run a calibration test.

PortClose();
```

See Also

Demand Functions, Port Functions

DemandFactoryDefaults

This function can demand the printer to make itself to be factory defaults.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall DemandFactoryDefaults();
```

Visual Basic

```
Private Declare Sub DemandFactoryDefaults Lib "SBSDK_API.DLL" ()
```

Delphi

```
Procedure DemandFactoryDefaults ();stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool DemandFactoryDefaults();
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Example (VC)

```
char acPortBuffer[40];

int nPortCount = PortEnumCount(0); // Get USB count

PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

if(PortOpen(acPortBuffer) == 0)

{

        AfxMessageBox(_T("Can't open port!"));

        return;

}


DemandFactoryDefaults(); // demand the printer to make itself to be factory defaults.

PortClose();
```

See Also

Demand Functions, Port Functions

DemandLabelFeed

This function can demand the printer to feed out a blank label.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall DemandLabelFeed();
```

Visual Basic

```
Private Declare Sub DemandLabelFeed Lib "SBSDK_API.DLL" ()
```

Delphi

```
Procedure DemandLabelFeed ();stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool DemandLabelFeed();
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Example (VC)

```
char acPortBuffer[40];

int nPortCount = PortEnumCount(0); // Get USB count

PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

if(PortOpen(acPortBuffer) == 0)

{

        AfxMessageBox(_T("Can't open port!"));

        return;

}


DemandLabelFeed(); // demand the printer to feed out a blank label.

PortClose();
```

See Also

Demand Functions, Port Functions

DemandPrintConfiguration

This function can demand the printer to print out a label with printer configuration.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall DemandPrintConfiguration();
```

Visual Basic

```
Private Declare Sub DemandPrintConfiguration Lib "SBSDK_API.DLL" ()
```

Delphi

```
Procedure DemandPrintConfiguration ();stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool DemandPrintConfiguration();
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Example (VC)

```
char acPortBuffer[40];
int nPortCount = PortEnumCount(0); // Get USB count
PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port
if(PortOpen(acPortBuffer) == 0)
{
        AfxMessageBox(_T("Can't open port!"));
        return;
}

DemandPrintConfiguration(); // demand the printer to print out a label with printer
configuration.
PortClose();
```

See Also

Demand Functions, Port Functions

DemandResetPrinter

This function can demand the printer to make a reset.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall DemandResetPrinter();
```

Visual Basic

```
Private Declare Sub DemandResetPrinter Lib "SBSDK_API.DLL" ()
```

Delphi

```
Procedure DemandResetPrinter ();stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool DemandResetPrinter();
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Example (VC)

```
char acPortBuffer[40];

int nPortCount = PortEnumCount(0); // Get USB count

PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

if(PortOpen(acPortBuffer) == 0)

{

        AfxMessageBox(_T("Can't open port!"));

        return;

}


DemandResetPrinter(); // demand the printer to make a reset.

PortClose();
```

See Also

Demand Functions, Port Functions

GETTING FUNCTIONS

Name	Description
GetEmulationType	This function can get printer emulation type.
GetFirmwareVersion	This function can get printer firmware version.
GetPrintDistance	This function can get printed distance of printer.
GetPrinterStatus	This function can get current status of printer.
GetReferencePoint	This function can get X and Y axis.
GetResolution	This function can get printer resolution.
GetSerialNumber	This function can get printer serial number.
GetTPH_YOffset	This function can get Y offset of TPH.

GetEmulationType

This function can get printer emulation type.

C++

```
extern "C" __declspec(dllimport) int _stdcall GetEmulationType();
```

Visual Basic

```
Private Declare Function GetEmulationType Lib "SBSDK_API.DLL" () As Long
```

Delphi

```
function GetEmulationType ():integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern int GetEmulationType();
```

Return Value

If the function succeeds, the return value is emulation value.

Value	Description
100	PEPL

If the function fails, the return value is 0(unknown emulation), -1 (port is unopened), -2 (not support LPT1,File), -3 (Read fail).

Remarks

Example

```
char acPortBuffer[40];

int nPortCount = PortEnumCount(0); // Get USB count

PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

if(PortOpen(acPortBuffer) == 0)

{

        AfxMessageBox(_T("Can't open port!"));

        return;

}


//PortOpen("COM1"); // not for LPT port

//PortSetupSerial(115200, 8, 0, 1, 1); // 115200 bps, 8 data-bits, no parity, 1 stop bit,
hardware flowing control.

if(GetEmulationType() == 100)

        AfxMessageBox(_T("PEPL Emulation"));

else

        AfxMessageBox(_T("Can't get emulation from printer"));
```

```
    PortClose();
```

See Also

Getting Functions, PortOpen, PortSetupSerial, PortClose, Setting Functions

GetFirmwareVersion

This function can get printer firmware version.

C++

```
extern "C" __declspec(dllimport) int _stdcall GetFirmwareVersion(char* pBuffer);
```

Visual Basic

```
Private Declare Function GetFirmwareVersion Lib "SBSDK_API.DLL" (ByVal pBuffer As String) As Long
```

Delphi

```
function GetFirmwareVersion (pBuffer:pchar):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern int GetFirmwareVersion(StringBuilder pBuffer);
```

Return Value

If the function succeeds, the return value is 0.

If the function fails, the return value is -1 (port is unopened), -2 (not support LPT1 and File), -3 (Read fail).

Parameters

- *pBuffer*

  Buffer for receiving a firmware string. (Max. Length: 32 bytes)

Remarks

Example (VC)

```
char acBuffer[32];
CString strTemp;
char acPortBuffer[40];
int nPortCount = PortEnumCount(0); // Get USB count
PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port
if(PortOpen(acPortBuffer) == 0)
{
        AfxMessageBox(_T("Can't open port!"));
        return;
}

//PortOpen("COM1"); // not for LPT port
//PortSetupSerial(115200, 8, 0, 1, 1); // 115200 bps, 8 data-bits, no parity, 1 stop bit, hardware flowing control.
```

```
GetFirmwareVersion(acBuffer);

strTemp = acBuffer;

AfxMessageBox(strTemp);

PortClose();
```

See Also

Getting Functions, PortOpen, PortSetupSerial, PortClose, Setting Functions

GetPrintDistance

This function can get printed distance of printer.

C++

```
extern "C" __declspec(dllimport) int _stdcall GetPrintDistance(int nMeasurement);
```

Visual Basic

```
Private Declare Function GetPrintDistance Lib "SBSDK_API.DLL" (ByVal nMeasurement As Long) As
Long
```

Delphi

```
function GetPrintDistance (nMeasurement:integer):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern int GetPrintDistance(int nMeasurement);
```

Return Value

If the function succeeds, the return value is distance value.

If the function fails, the return value is -1 (port is unopened), -2 (not support LPT1,File), -3 (Read fail).

Parameters

- *nMeasurement*

  Measurement of printing distance.

Value	Description
0	Dots
1	Inches
2	Centimeters
3	Millimeters

Remarks

Example

```
CString strTemp;

char acPortBuffer[40];

int nPortCount = PortEnumCount(0); // Get USB count

PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

if(PortOpen(acPortBuffer) == 0)

{

        AfxMessageBox(_T("Can't open port!"));

        return;

}
```

```
//PortOpen("COM1"); // not for LPT port

//PortSetupSerial(115200, 8, 0, 1, 1); // 115200 bps, 8 data-bits, no parity, 1 stop bit,
hardware flowing control.

int nDistanceDots = GetPrintDistance(0); // Dots

int nDistanceInchs = GetPrintDistance(1); // inchs

int nDistanceCm = GetPrintDistance(2); // CentiMeters

int nDistanceMm = GetPrintDistance(3); // Millimeters

strTemp.Format(_T("%d dots, %d inchs, %d cm, %d mm, %d km"), nDistanceDots,
nDistanceInchs, nDistanceCm, nDistanceMm, nDistanceCm/100/1000);

AfxMessageBox(strTemp);

PortClose();
```

See Also

Getting Functions, PortOpen, PortSetupSerial, PortClose, Setting Functions

GetPrinterStatus

This function can get current status of printer.

C++

```
extern "C" __declspec(dllimport) int _stdcall GetPrinterStatus();
```

Visual Basic

```
Private Declare Function
```

Delphi

```
function GetPrinterStatus ():integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern int GetPrinterStatus();
```

Return Value

If the function succeeds, the return value is printer current status.

Value	Description
0	Ready
100	Feeding
101	Pausing
102	Peeling
103	On Demand
200	Label gap out
201	Label out
202	Ribbon out
203	Cover open (ready)
204	Cover open (printing)
205	TPH fail
206	Update checksum error
207	Update fail

If the function fails, the return value is -1 (port is unopened), -2 (not support LPT1,File), -3 (Read fail).

Example

```
CString strTemp;

char acPortBuffer[40];

int nPortCount = PortEnumCount(0); // Get USB count

PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

if(PortOpen(acPortBuffer) == 0)

{

        AfxMessageBox(_T("Can't open port!"));
        return;
    }

    //PortOpen("COM1"); // not for LPT port
    //PortSetupSerial(115200, 8, 0, 1, 1); // 115200 bps, 8 data-bits, no parity, 1 stop bit,
    hardware flowing control.
    int nErrorCode = GetPrinterStatus();
    if( nErrorCode != 0)
        strTemp.Format(_T("Printr Status: Error Code %d"), nErrorCode);
    else
        strTemp = _T("Printr Status: Ready");
    AfxMessageBox(strTemp);
    PortClose();
```

See Also

Getting Functions, PortOpen, PortSetupSerial, PortClose, Setting Functions

GetReferencePoint

This function can get X and Y axis.

C++

```
extern "C" __declspec(dllimport) int _stdcall GetReferencePoint(int nAxisType);
```

Visual Basic

```
Private Declare Function GetReferencePoint Lib "SBSDK_API.DLL" (ByVal nAxisType As Long) As Long
```

Delphi

```
function GetReferencePoint (nAxisType:integer):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern int GetReferencePoint (int nAxisType);
```

Return Value

If the function succeeds, the return value is X or Y axis value.

If the function fails, the return value is -1 (port is unopened).

Parameters

- *nAxis*

  Coordinate type.

Value	Description
0	X axis
1	Y axis

Remarks

Example

```
CString strTemp;

char acPortBuffer[40];

int nPortCount = PortEnumCount(0); // Get USB count

PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

if(PortOpen(acPortBuffer) == 0)

{

        AfxMessageBox(_T("Can't open port!"));

        return;

}


//PortOpen("COM1"); // not for LPT port
```

```
//PortSetupSerial(115200, 8, 0, 1, 1); // 115200 bps, 8 data-bits, no parity, 1 stop bit,
hardware flowing control.

int nRefPointDots_X = GetReferencePoint(0);

int nRefPointDots_Y = GetReferencePoint(1);

strTemp.Format(_T("X axis = %d, Y axis= %d (dots) "), nRefPointDots_X,
nRefPointDots_Y);

AfxMessageBox(strTemp);

PortClose();
```

See Also

Getting Functions, PortOpen, PortSetupSerial, PortClose, Setting Functions

GetResolution

This function can get printer resolution.

C++

```
extern "C" __declspec(dllimport) int _stdcall GetResolution();
```

Visual Basic

```
Private Declare Function GetResolution Lib "SBSDK_API.DLL" () As Long
```

Delphi

```
function GetResolution ():integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern int GetResolution();
```

Return Value

If the function succeeds, the return value is resolution value.

Value	Description
200	200 dpi
300	300 dpi

If the function fails, the return value is -1 (port is unopened), -2 (not support LPT1,File), -3 (Read fail).

Remarks

Example

```
CString strTemp;

char acPortBuffer[40];

int nPortCount = PortEnumCount(0); // Get USB count

PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

if(PortOpen(acPortBuffer) == 0)

{

        AfxMessageBox(_T("Can't open port!"));

        return;

}


//PortOpen("COM1"); // not for LPT port

//PortSetupSerial(115200, 8, 0, 1, 1); // 115200 bps, 8 data-bits, no parity, 1 stop bit,
hardware flowing control.

int nResolution = GetResolution();

strTemp.Format(_T("%d dpi"), nResolution);
```

```
    AfxMessageBox(strTemp);

    PortClose();
```

See Also

Getting Functions, PortOpen, PortSetupSerial, PortClose, Setting Functions

GetSerialNumber

This function can get printer serial number.

C++

```
extern "C" __declspec(dllimport) int _stdcall GetSerialNumber(char* pBuffer);
```

Visual Basic

```
Private Declare Function GetSerialNumber Lib "SBSDK_API.DLL" (ByVal pBuffer As String) As Long
```

Delphi

```
function GetSerialNumber (pBuffer:pchar):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern int GetSerialNumber(StringBuilder pBuffer);
```

Return Value

If the function succeeds, the return value is length of serial number. (Bytes)

If the function fails, the return value is -1 (port is unopened), -2 (not support LPT1,File), -3 (Read fail).

Parameters

- *pBuffer*

  Buffer for receiving a serial number string. (Max. Length: 32 bytes)

Remarks

Example

```
char acBuffer[32];

CString strTemp;

char acPortBuffer[40];

int nPortCount = PortEnumCount(0); // Get USB count

PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

if(PortOpen(acPortBuffer) == 0)

{

    AfxMessageBox(_T("Can't open port!"));

    return;

}


//PortOpen("COM1"); // not for LPT port

//PortSetupSerial(115200, 8, 0, 1, 1); // 115200 bps, 8 data-bits, no parity, 1 stop bit,
hardware flowing control.
```

```
if(GetSerialNumber(acBuffer) > 0)

{

        strTemp = _T("Serial Number: ");

        strTemp += acBuffer;

        AfxMessageBox(strTemp);

}

else

        AfxMessageBox(_T("Can't get serial number"));

PortClose();
```

See Also

Getting Functions, PortOpen, PortSetupSerial, PortClose, Setting Functions

GetTPH_YOffset

This function can get Y offset of TPH.

C++

```
extern "C" __declspec(dllimport) int _stdcall GetTPH_YOffset();
```

Visual Basic

```
Private Declare Function GetTPH_YOffset Lib "SBSDK_API.DLL" () As Long
```

Delphi

```
function GetTPH_YOffset ():integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern int GetTPH_YOffset ();
```

Return Value

If the function succeeds, the return value is Y-offset of TPH. Range: -720 ~ 80 dots.

If the function fails, the return value is -1 (port is unopened), -2 (not support LPT1,File), -3 (Read fail).

Remarks

Example

```
CString strTemp;

char acPortBuffer[40];

int nPortCount = PortEnumCount(0); // Get USB count

PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

if(PortOpen(acPortBuffer) == 0)

{

        AfxMessageBox(_T("Can't open port!"));

        return;

}


//PortOpen("COM1"); // not for LPT port

//PortSetupSerial(115200, 8, 0, 1, 1); // 115200 bps, 8 data-bits, no parity, 1 stop bit,
hardware flowing control.

int nYOffset = GetTPH_YOffset();

strTemp.Format(_T("%d dots"), nYOffset);

AfxMessageBox(strTemp);

PortClose();
```

See Also

Getting Functions, PortOpen, PortSetupSerial, PortClose, Setting Functions

SETTING FUNCTIONS

Name	Description
SetAfterPrint	This function can enable cutter, tear-off, peel-off or normal after printing.
SetFontCharacterSet	This function can set the appropriate character set for printing.
SetMeasurement	This function can set coordinate of measurement in printing.
SetMedia_Continuous	This function can set width and length of continuous label, set stop-offset after printing.
SetMedia_LabelWithGaps	This function can set label width, label length, gap length and gap-offset of gap label.
SetMedia_LabelWithMarks	This function can set label width, label length, black-line length and black-line thickness offset of black-line label.
SetPrinter	This function can set speed, darkness and method of printing.
SetReferencePoint	This function can set the reference point for X and Y axes.
SetTPH_YOffset	This function can set the Y-offset of TPH.

SetAfterPrint

This function can enable cutter, tear-off, peel-off, tap-to-print or normal after printing.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall SetAfterPrint(int nType, int nCutPiece);
```

Visual Basic

```
Private Declare Function SetAfterPrint Lib "SBSDK_API.DLL" (ByVal nType As Long, ByVal nCutPiece
As Long) As Long
```

Delphi

```
function SetAfterPrint (nType:integer; nCutPiece:integer):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool SetAfterPrint(int nType, int nCutPiece);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Parameters

- *nType*

  Enables type of action after printing.

  You can combine options listed below by using the bitwise-OR (|) operator.

Value	Value Define in C++	Description
0	DEV_OPT_NORMAL	Normal
1	DEV_OPT_TEAR	Enable Tear-Off
2	DEV_OPT_PEEL	Enable Peel-Off
4	DEV_OPT_CUTTER	Enable Cutter
8	DEV_OPT_TAPPRINT	Enable Tap to print

- *nCutPiece*

  Sets the number of labels to print prior to cut when nType is DEV_OPT_CUTTER. The value is between 1 - 255.

Remarks

Example

```
char acPortBuffer[40];

int nPortCount = PortEnumCount(0); // Get USB count

PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

if(PortOpen(acPortBuffer) == 0)

{
        AfxMessageBox(_T("Can't open port!"));
        return;
    }
    SetAfterPrint (DEV_OPT_TEAR, 0); // Enable Tear-Off mode after printing.
    PrintLabel(0, 0, 0); // Only send data to printer without printing
    PortClose();
```

See Also

Port Functions, Setting Functions, Getting Functions

SetFontCharacterSet

This function can set the appropriate character set for printing.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall SetFontCharacterSet(int nNumberBit, int
nCharacterSet);
```

Visual Basic

```
Private Declare Function SetFontCharacterSet Lib "SBSDK_API.DLL" (ByVal nNumberBit As Long,
ByVal nCharacterSet As Long) As Long
```

Delphi

```
function SetFontCharacterSet (nNumberBit:integer; nCharacterSet:integer):integer;stdcall;external
'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool SetFontCharacterSet(int nMeasurement, int nResolution);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Parameters

- *nNumberBit*

  Number of data bits - 8 for 8 bit data or 7 for 7 bit data.

- *nCharacterSet*

  Printer Codepage/Language Support. Refer below table.

nNimberBit=8 (8 Bit Data)			**nNimberBit =7 (7 Bit Data)**	
nCharacterSet	**Code Page**	**Description**	**nCharacterSet**	**Description**
0	DOS_437	English-US	**0**	USA
1	DOS_850	Latin-1	**1**	British
2	DOS_852	Latin-2 (Cyrillic II/Slavic)	**2**	German
3	DOS_860	Portuguese	**3**	French
4	DOS_863	French Canadian	**4**	Danish
5	DOS_865	Nordic	**5**	Italian
6	DOS_857	Turkish	**6**	Spanish
7	DOS_861	Icelandic	**7**	Swedish
8	DOS_862	Hebrew	**8**	Swiss
9	DOS_855	Cyrillic		
10	DOS_866	Cyrillic CIS-1		
11	DOS_737	Greek		
12	DOS_851	Greek-1		
13	DOS_869	Greek-2		

65	Windows-1252	Latin 1		
66	Windows-1250	Latin 2		
67	Windows-1251	Cyrillic		
68	Windows-1253	Greek		
69	Windows-1254	Turkish		
70	Windows-1255	Hebrew		

Remarks

Example

```
char acPortBuffer[40];

int nPortCount = PortEnumCount(0); // Get USB count

PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

if(PortOpen(acPortBuffer) == 0)

{

        AfxMessageBox(_T("Can't open port!"));

        return;

}


SetFontCharacterSet(7, 0); // 7 Bit Data with USA character Set

PrintLabel(0, 0, 0); // Only send data to printer without printing

PortClose();
```

See Also

Port Functions, Setting Functions, Getting Functions

SetMeasurement

This function can set coordinate of measurement in printing.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall SetMeasurement(int nMeasurement, int
nResolution);
```

Visual Basic

```
Private Declare Function SetMeasurement Lib "SBSDK_API.DLL" (ByVal nMeasurement As Long,
ByVal nResolution As Long) As Long
```

Delphi

```
function SetMeasurement (nMeasurement:integer; nResolution:integer):integer;stdcall;external
'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool SetMeasurement(int nMeasurement, int nResolution);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0.

Parameters

- *nMeasurement*

  Measurement of printing coordinate or length.

Value	Description
0	Dots
1	Inches
2	Centimeters
3	Millimeters

- *nResolution*

  Printer resolution.

Value	Description
0	Refer current value
200	200 Dpi Printer
300	300 Dpi Printer

Remarks

Example (VC)

```
char acPortBuffer[40];

int nPortCount = PortEnumCount(0); // Get USB count

PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

if(PortOpen(acPortBuffer) == 0)
```

```
{
    AfxMessageBox(_T("Can't open port!"));
    return;
}

SetMeasurement(1, 200); // inches, 200 dpi printer
PrintText(2.1f, 1.1f, 0, 1, 1, 1, 0, 8, “AaBbCcDd”); // X=2.1, Y=1.1 (inches), No rotation,
Font1, Hor Expand=1, Ver Expand=1, normal image
PrintLabel(0, 0, 0); // Only send data to printer without printing
PortClose();
```

See Also

Port Functions, Setting Functions, Getting Functions

SetMedia_Continuous

This function can set width and length of continuous label, set stop-offset after printing.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall SetMedia_Continuous(float fLabelWidth, float
fLabelLen, float fStopOffset);
```

Visual Basic

```
Private Declare Function SetMedia_Continuous Lib "SBSDK_API.DLL" (ByVal fLabelWidth As Single,
ByVal fLabelLen As Single, ByVal fStopOffset As Single) As Long
```

Delphi

```
function SetMedia_Continuous (fLabelWidth:Single; fLabelLen:Single;
fStopOffset:Single):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool SetMedia_Continuous(float fLabelWidth, float fLabelLen, float fStopOffset);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Parameters

- *fLabelWidth*

  The width of the printable area of the media in library measurement. (See below “W” of picture1)

- *fLabelLen*

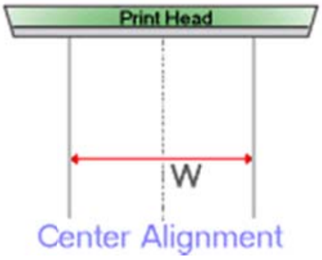
  Label length in library measurement. Range: 1 ~ 65535 (dots)

- *fStopOffset*

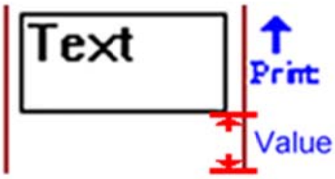
  Stop Offset in library measurement. Range: 0~ 400 (dots) about 5.08 cm. (See below “Value” of picture2)

Remarks

(Picture1)



(Picture2)



Example

```
char acPortBuffer[40];

int nPortCount = PortEnumCount(0); // Get USB count
```

```
PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

if(PortOpen(acPortBuffer) == 0)

{

        AfxMessageBox(_T("Can't open port!"));

        return;

}


SetMeasurement(0, 0); // nMeasurement=0(Dots), 1(Inches), 2(Centimeters),

3(Millimeters); nResolution=0(Currecnt Set)

SetMedia_Continuous(800.f, 600.f, 20.f); // W=4 inch, H=3Inch, StopOffset=0.1 inch;

PrintLabel(0, 0, 0); // Only send data to printer without printing

PortClose();
```

See Also

Port Functions, Setting Functions, Getting Functions

SetMedia_LabelWithGaps

This function can set label width, label length, gap length and gap-offset of gap label.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall SetMedia_LabelWithGaps(float fLabelWidth, float
fLabelLen, float fGapLen, float fGapOffset);
```

Visual Basic

```
Private Declare Function SetMedia_LabelWithGaps Lib "SBSDK_API.DLL" (ByVal nLabelWidth As
Single, ByVal nLabelLen As Single, ByVal nGapLen As Single, ByVal nGapOffset As Single) As Long
```

Delphi

```
function SetMedia_LabelWithGaps (nLabelWidth:Single; nLabelLen:Single; nGapLen:Single;
nGapOffset:Single):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool SetMedia_LabelWithGaps(float nLabelWidth, float nLabelLen, float nGapLen,
float nGapOffset);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Parameters

- *fLabelWidth*

  The width of the printable area of the media in library measurement. (See below "W" of picture)

- *fLabelLen*

  Label length in library measurement. Range: 1 ~ 65535 (dots). (See below "P1" of picture)

- *fGapLen*

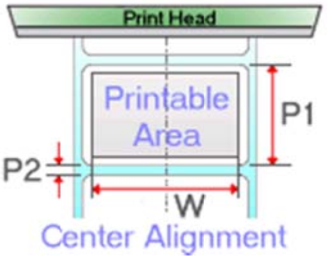
  Gap length in library measurement. Range: 16~ 240 (dots) about 3 cm for 200 dpi. (See below "P2" of picture)

- *fGapOffset*

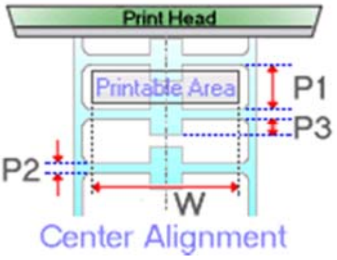
  Gap offset in library measurement. Range: -80~ 80 (dots) about 1 cm for 200 dpi. (See below "P3" of picture)

Remarks

(Picture1)



(Picture2)



Example

```
char acPortBuffer[40];
int nPortCount = PortEnumCount(0); // Get USB count
PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port
if(PortOpen(acPortBuffer) == 0)
{
        AfxMessageBox(_T("Can't open port!"));
        return;
}

SetMeasurement(0, 0); // nMeasurement=0(Dots), 1(Inches), 2(Centimeters), 3(Millimeters); nResolution=0(Currecnt Set)
SetMedia_LabelWithGaps(800.f, 600.f, 32.f, 0); // W=4 inch, H=3Inch, GapLen=32 dots, GapOffset=0;
PrintLabel(0, 0, 0); // Only send data to printer without printing
PortClose();
```

See Also

Port Functions, Setting Functions, Getting Functions

SetMedia_LabelWithMarks

This function can set label width, label length, black-line length and black-line thickness offset of black-line label.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall SetMedia_LabelWithMarks(float fLabelWidth, float
fLabelLen, float fBlackLineLen, float fThicknessOffset);
```

Visual Basic

```
Private Declare Function SetMedia_LabelWithMarks Lib "SBSDK_API.DLL" (ByVal fLabelWidth As
Single, ByVal fLabelLen As Single, ByVal fBlackLineLen As Single, ByVal fThicknessOffset As Single)
As Long
```

Delphi

```
function SetMedia_LabelWithMarks (fLabelWidth:Single; fLabelLen:Single; fBlackLineLen:Single;
fThicknessOffset:Single):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool SetMedia_LabelWithMarks(float fLabelWidth, float fLabelLen, float
fBlackLineLen, float fThicknessOffset);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Parameters

- *fLabelWidth*

  The width of the printable area of the media in library measurement. (See below “W” of picture)

- *fLabelLen*

  Label length in library measurement. Range: 1 ~ 65535 (dots). (See below “P1” of picture)

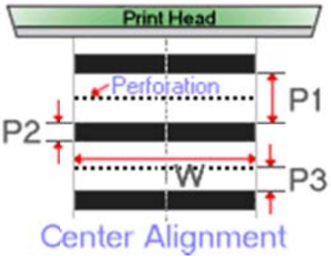
- *fBlackLineLen*

  Back-line length in library measurement. Range: 16~ 240 (dots) about 3 cm for 200 dpi. (See below “P2” of picture)

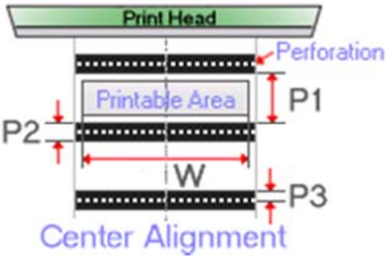
- *fThicknessOffset*

  Thickness of back-line in library measurement. Range: -80~ 80 (dots) about 1 cm for 200 dpi. (See below “P3” of picture)

Remarks



(Picture1)



(Picture2)

Example

```
char acPortBuffer[40];

int nPortCount = PortEnumCount(0); // Get USB count

PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

if(PortOpen(acPortBuffer) == 0)

{

        AfxMessageBox(_T("Can't open port!"));

        return;

}


SetMeasurement(0, 0); // nMeasurement=0(Dots), 1(Inches), 2(Centimeters),
3(Millimeters); nResolution=0(Currecnt Set)

SetMedia_LabelWithMarks(800.f, 600.f, 40.f, 0); // W=4 inch, H=3Inch, Black-line Len=40
dots, Thickness Offset=0;

PrintLabel(0, 0, 0); // Only send data to printer without printing

PortClose();
```

See Also

Port Functions, Setting Functions, Getting Functions

SetPrinter

This function can set speed, darkness and method of printing.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall SetPrinter(int nSpeed, int nDarkness, BOOL
bDirectThermal);
```

Visual Basic

```
Private Declare Function SetPrinter Lib "SBSDK_API.DLL" (ByVal nSpeed As Long, ByVal nDarkness
As Long, ByVal bDirectThermal As Long) As Long
```

Delphi

```
function SetPrinter (nSpeed:integer; nDarkness:integer;
bDirectThermal:integer):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool SetPrinter(int nSpeed, int nDarkness, int bDirectThermal);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Parameters

- *nSpeed*

  Print speed.

Model	Value	Speed
T4	1	1 ips (25 mm/s)
	2	2 ips (50 mm/s)
	3	3 ips (75 mm/s)
	4	4 ips (100 mm/s)

- *nDarkness*

  Print darkness.

Model	Value
T4	0 ~ 15

- *bDirectThermal*

  Enable Direct-Thermal printing.

nValue	Description
0	Thermal Transfer
1	Direct Thermal

Remarks

Example

```
char acPortBuffer[40];

int nPortCount = PortEnumCount(0); // Get USB count

PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

if(PortOpen(acPortBuffer) == 0)

{

        AfxMessageBox(_T("Can't open port!"));

        return;

}


SetPrinter(3, 10, 1); // nSpeed=3, nDarkness=10, bDirectThermal=1;

PrintLabel(0, 0, 0); // Only send data to printer without printing

PortClose();
```

See Also

Port Functions, Setting Functions, Getting Functions

SetReferencePoint

This function can set the reference point for X and Y axes.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall SetReferencePoint(int nHoriDot, int nVertDot);
```

Visual Basic

```
Private Declare Function SetReferencePoint Lib "SBSDK_API.DLL" (ByVal nHoriDot As Long, ByVal
nVertDot As Long) As Long
```

Delphi

```
function SetReferencePoint (nHoriDot:integer; nVertDot:integer):integer;stdcall;external
'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool SetReferencePoint(int nHoriDot, int nVertDot);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Parameters

- *nHoriDot*

  Horizontal (left) margin measured in dots.

- *nVertDot*

  Vertical (top) margin measured in dots.

Remarks

Example

```
char acPortBuffer[40];

int nPortCount = PortEnumCount(0); // Get USB count

PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

if(PortOpen(acPortBuffer) == 0)

{

        AfxMessageBox(_T("Can't open port!"));

        return;

}


SetReferencePoint(0, 100); // nVertDot=100, Move Y axes 100 dots.

PrintLabel(0, 0, 0); // Only send data to printer without printing
```

```
PortClose();
```

See Also

Port Functions, Setting Functions, Getting Functions

SetTPH_YOffset

This function can set the Y-offset of TPH.

C++

```
extern "C" __declspec(dllimport) BOOL _stdcall SetTPH_YOffset(int nOffsetDot);
```

Visual Basic

```
Private Declare Function SetTPH_YOffset Lib "SBSDK_API.DLL" (ByVal nOffsetDot As Long) As Long
```

Delphi

```
function SetTPH_YOffset (nOffsetDot:integer):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
 [DllImport("SBSDK_API.dll")]
        static extern bool SetTPH_YOffset(int nOffsetDot);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0 (port is unopened).

Parameters

- *nOffsetDot*

  Y-Offset of TPH measured in dots. Range: -720 ~ 80 dots.

Remarks

Example

```
char acPortBuffer[40];
int nPortCount = PortEnumCount(0); // Get USB count
PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port
if(PortOpen(acPortBuffer) == 0)
{
        AfxMessageBox(_T("Can't open port!"));
        return;
}

SetTPH_YOffset(-10); // Shift TPH Y to -10 dots.
PrintLabel(0, 0, 0); // Only send data to printer without printing
PortClose();
```

See Also

Port Functions, Setting Functions, Getting Functions

OTHER FUNCTIONS

Name	Description
DllAbout	This function can display an about-dialog of DLL and return a version of DLL.
SendCmd	This function can send a buffer of data to the printer or write some commands in dialog to send the printer.
SendFileA, SendFileW	This function can send a file to printer.
GetScaleData	This is a scale model can be specified so that the weight after weighing the return value, and specify from which incoming RS232.

DllAbout

This function can display an about-dialog of DLL and return a version of DLL.

C++

```
extern "C" __declspec(dllimport) double _stdcall DllAbout(BOOL bDisplay);
```

Visual Basic

```
Private Declare Function DllAbout Lib "SBSDK_API.DLL" (ByVal bDisplay As Long) As Double
```

Delphi

```
function DllAbout (bDisplay:integer):Double;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern double DllAbout(int bDisplay);
```

Return Value

If the function succeeds, the return value is version of DLL.

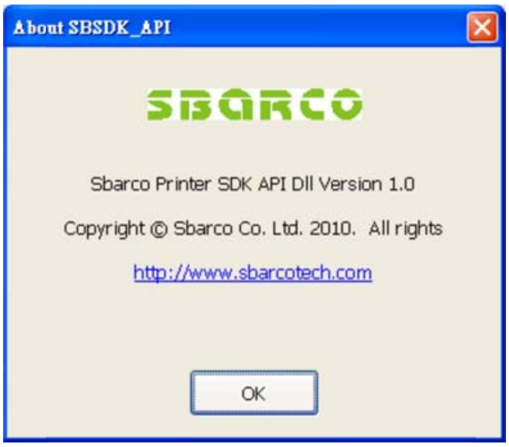
If the function fails, the return value is 0.

Parameters

- *bDisplay*

  0=only return version, 1=display dialog and return version.

Remarks



Sbarco Printer SDK API Dll Version 1.0

Copyright © Sbarco Co. Ltd. 2010. All rights

http://www.sbarcotech.com

Example

```
double dbVersion = DllAbout(TRUE); // 1=display dialog, 0=only return version

CString strVersion;

strVersion.Format(_T("DLL Version: %.1f"), dbVersion);

AfxMessageBox(strVersion);
```

See Also

Other Functions

SendCmd

This function can send a buffer of data to the printer or write some commands in dialog to send the printer.

C++

```
extern "C" __declspec(dllimport) int _stdcall SendCmd(int nDataLen, LPBYTE lpData);
```

Visual Basic

```
Private Declare Function SendCmd Lib "SBSDK_API.DLL" (ByVal nDataLen As Long, ByVal lpData As String) As Long
```

Delphi

```
function SendCmd (nDataLen:integer; lpData:string):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern int SendCmd(int nDataLen, StringBuilder lpData);
```

Return Value

If the function succeeds, the return value is length of data.

If the function fails, the return value is 0 (port is unopened).

Parameters

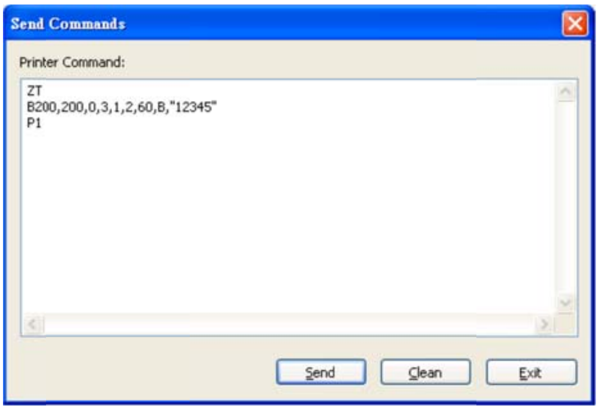
- *nDataLen*

  length of data buffer. When the value is -1, the function will appear a dialog.(see below picture)

- *lpData*

  Data buffer.

Remarks



```
ZT
B200,200,0,3,1,2,60,B,"12345"
P1
```

Example

```
char acPortBuffer[40];

int nPortCount = PortEnumCount(0); // Get USB count
```

```
PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port
if(PortOpen(acPortBuffer) == 0)
{
        AfxMessageBox(_T("Can't open port!"));
        return;
}

int nDataLen = SendCmd(-1, NULL);
CString strMessage;
strMessage.Format(_T("Data length = %d"), nDataLen);
AfxMessageBox(strMessage);
PortClose();
```

See Also

Other Functions, Port Functions

SendFileA, SendFileW

This function can send a file to printer.

C++

```
extern "C" __declspec(dllimport) int _stdcall SendFileA(char* pFileName);

extern "C" __declspec(dllimport) int _stdcall SendFileW(TCHAR* ptcFileName);
```

Visual Basic

```
Private Declare Function SendFileA Lib "SBSDK_API.DLL" (ByVal pFileName As String) As Long
```

Delphi

```
function SendFileA (pFileName:string):integer;stdcall;external 'SBSDK_API.DLL';
function SendFileW (ptcFileName: WideString):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern int SendFileA(string pFileName);

 [DllImport("SBSDK_API.dll")]
        static extern int SendFileW(string ptcFileName);
```

Return Value

If the function succeeds, the return value is length of file.

If the function fails, the return value is 0 (port is unopened).

Parameters

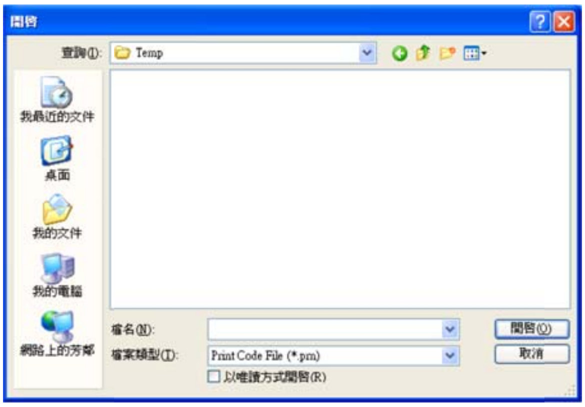
- *pFileName*

  Print out file name with path. If the value is “open”, it will appear a dialog for browsing file.

- *ptcFileName*

  Print out file name with path that supports Unicode. If the value is _T(“open”), it will appear a dialog for browsing file.

Remarks



查詢(I): Temp
檔案類型(T): Print Code File (*.prn)
☐ 以唯讀方式開啟(R)

Example

```
char acPortBuffer[40];
```

```
int nPortCount = PortEnumCount(0); // Get USB count

PortEnumGet(0, 0, acPortBuffer); // Get name of number 0 of USB port

if(PortOpen(acPortBuffer) == 0)

{

        AfxMessageBox(_T("Can't open port!"));

        return;

}


int nDataLen = SendFileA("testcmd.txt");

//int nDataLen = SendFileW(_T("testcmd.txt"));

//int nDataLen = SendFileA("open");

//int nDataLen = SendFileW(_T("open"));

CString strMessage;

strMessage.Format(_T("Data length = %d"), nDataLen);

AfxMessageBox(strMessage);

PortClose();
```

See Also

Other Functions, Port Functions

GetScaleData

This is a scale model can be specified so that the weight after weighing the return value, and specify from which incoming RS232.

這是可以指定磅秤型號，使之回傳秤重完後的重量數值，並指定從哪個 RS232 傳入。

C++

```
extern "C" __declspec(dllimport) double _stdcall GetScaleData(int nMode, int nComPortNum);
```

Visual Basic

```
Private Declare Function GetScaleData Lib "SBSDK_API.DLL" (ByVal nMode As Long, ByVal
nComPortNum As Long) As Double
```

Delphi

```
function GetScaleData (nMode:integer, nComPortNum:integer):Double;stdcall;external
'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern double GetScaleData (int nMode, int nComPortNum);
```

Return Value

If the function succeeds, the return weight value.

If the function fails, the return value is 0.

Parameters

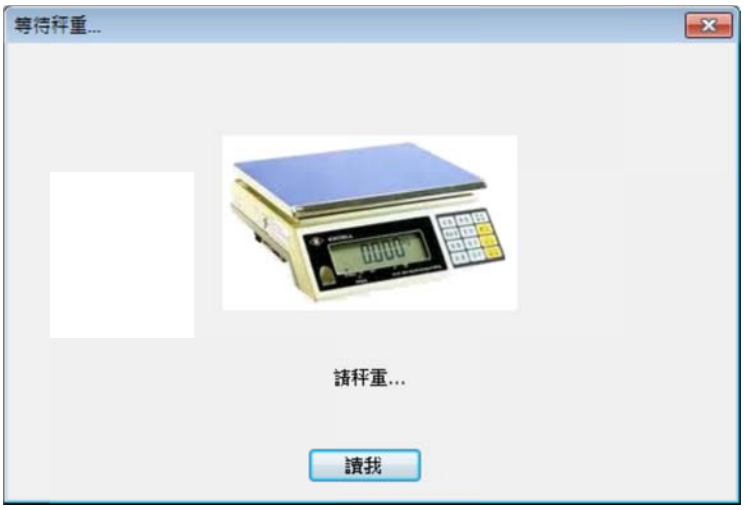
- *nMode*

  1=EXCELL Scale.

- *nComPortNum*

  Ex: 1=COM1.

Remarks



請秤重...

Example

```
CString strTemp;

double dbResult = GetScaleData(1, 1);

strTemp.Format(_T("%g"), dbResult);

AfxMessageBox(strTemp);
```

See Also

Other Functions

BARDRAWER FUNCTIONS

Name	Description
BarDrawer_DatabaseSet	This function can set database setting of bardrawer label file.
BarDrawer_Launch	This function can launch bardrawer application with existing label file.
BarDrawer_LaunchNew	This function can launch bardrawer application with new label file.

BarDrawer_DatabaseSet

This function can set database setting of bardrawer label file.

C++

```
extern "C" __declspec(dllimport) double _stdcall BarDrawer_DatabaseSet (int nType, BOOL
nSaveInfos, char* pstrFilePath, char* pstrUserName, char* pstrPassword, char* pstrTable);
```

Visual Basic

```
Private Declare Function BarDrawer_DatabaseSet Lib "SBSDK_API.DLL" (ByVal nType As Long,
ByVal nSaveInfos As Long, ByVal pstrFilePath As String, ByVal pstrUserName As String, ByVal
pstrPassword As String, ByVal pstrTable As String) As Long
```

Delphi

```
function BarDrawer_DatabaseSet (nType:integer; nSaveInfos:integer; pstrFilePath:string;
pstrUserName:string; pstrPassword:string; pstrTable:string):integer;stdcall;external
'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool BarDrawer_DatabaseSet(int nType, int nSaveInfos, string pstrFilePath, string
pstrUserName, string pstrPassword, string pstrTable);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0.

Parameters

- *nType*

  Database Type, 0= Microsoft Access, 1= Microsoft Excel.

- *nSaveInfos*

  To save user name and password into file. 0=No, 1=Save.

- *pstrFilePath*

  Database file name with full path. If it is only file name, it will load file in current path.

- *pstrUserName*

  User name of database security.

- *pstrPassword*

  Password of database security.

- *pstrTable*

  Used table name of database.

Remarks

This function supports BarDrawer application form version 1.1.5 .

Example

```
BarDrawer_DatabaseSet(1 /*Excel*/, TRUE, "Excel database.xls", "Admin", "", "Sheet1$");
```

```
BarDrawer_Launch("DLLTest.bdw");

//BarDrawer_LaunchNew("Sample1.bdw");
```

See Also

BarDrawer Functions

BarDrawer_Launch

This function can launch bardrawer application with existing label file.

C++

```
extern "C" __declspec(dllimport) double _stdcall BarDrawer_Launch(char* pFileName);
```

Visual Basic

```
Private Declare Function BarDrawer_Launch Lib "SBSDK_API.DLL" (ByVal pFileName As String) As Long
```

Delphi

```
function BarDrawer_Launch (pFileName:string):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool BarDrawer_Launch(string pFileName);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0.

Parameters

- *pFileName*

  Label file name with full path. If it is only file name, it will load file in current path.

Remarks

This function supports BarDrawer application form version 1.1.5 .

Example

```
BarDrawer_Launch("DLLTest.bdw");
```

Or

```
BarDrawer_Launch("C:\\DLLTest.bdw");
```

See Also

BarDrawer Functions

BarDrawer_LaunchNew

This function can launch bardrawer application with new label file.

C++

```
extern "C" __declspec(dllimport) double _stdcall BarDrawer_LaunchNew(char* pFileName);
```

Visual Basic

```
Private Declare Function BarDrawer_LaunchNew Lib "SBSDK_API.DLL" (ByVal pFileName As String)
As Long
```

Delphi

```
function BarDrawer_LaunchNew (pFileName:string):integer;stdcall;external 'SBSDK_API.DLL';
```

CShap

```
[DllImport("SBSDK_API.dll")]
        static extern bool BarDrawer_LaunchNew(string pFileName);
```

Return Value

If the function succeeds, the return value is 1.

If the function fails, the return value is 0.

Parameters

- *pFileName*

  Label file name with full path. If it is only file name, it will load file in current path.

Remarks

This function supports BarDrawer application form version 1.1.5 .

Example

```
BarDrawer_LaunchNew("Sample1.bdw");
```

or

```
BarDrawer_LaunchNew("C:\\Sample1.bdw");
```

See Also

BarDrawer Functions