
POS SDK For IOS



User's Manual_English

Shandong New Beiyang Information Technology Co., Ltd.

Contents

Contents	I
About this Manual	II
Aim of the Manual	II
Manual Contents	II
1. Overview	1
1.1 Function.....	1
1.2 Operating Environment.....	1
1.3 Contents in the Package	2
1.4 Version Record.....	2
2. Sample Program	3
2.1 Start up the Sample Program	3
2.2 How to Use the Sample Program	4
3. Programming Guide	19
3.1 Using WIFI to Connect to Printer's Port.....	19
3.2 Using Library	19
4. API Reference.....	27
4.1 WIFI Reference.....	27
4.2 POS SDK API Reference	33
5. Appendix	79
Appendix A. How to use the WIFIPortToFile class.....	79
Appendix B. List of Error Code	80
Appendix C. Barcode	82
Appendix D. Code 128.....	86
Appendix E. Programming Flow	91

About this Manual

Aim of the Manual

This manual aims to introduce to customers how to use the sample program and APIs of POS SDK For IOS.

Manual Contents

This manual is made up of the following sections:

- Chapter 1 [Overview](#)
- Chapter 2 [Sample Program](#)
- Chapter 3 [Programming Guide](#)
- Chapter 4 [API Reference](#)
- Chapter 5 [Appendix](#)

1. Overview

This chapter describes searching WIFI, connecting to printer, closing port, APIs function, operating environment, files in the package and version record.

1.1 Function

• WIFI

The functions described about WIFI are searching WIFI, opening port, sending and writing datas, setting timeout, recording communication data, closing port.

• API

- Setting system function(Set communication module instance of WIFIPort, Initialize printer, Select print mode, Select paper type, Set the horizontal and vertical motion units, Query Status, Feed line, Cut paper, Download file, Open cashdrawer).

- Text printing(Select an international character set and Code page, Set line height, Set character spacing, Set alignment mode, Select font type, White/Black reverse, Bold, Underline, Roration, Font Magnify, Bi-colour print, User defined character printing, Text raster printing).

- Image printing (8/24-dot single/double-density, Download image to RAM and print, Download image to Flash and print, Print raster image).

- Barcode 1D printing (UPC-A, UPC-E, EAN-8, EAN-13, Code39, Code93, ITF, Codabar, Code128).

- Barcode 2D printing(PDF417, QR, Maxicode, GS1 DataBar and GS1 composite barcode).

- Setting standard mode parameter (Set left margin , print area width , horizontal Starting Position).

- Setting page mode parameter (Set print area, Print direction, horizontal/vertical Starting Position, printing of page mode, clear buffer).

1.2 Operating Environment

• IOS Version

IOS Ver.4.2-4.35

IOS Ver.5.0-5.1

IOS Ver.6.1.3

- **IOS Device**

iPhone

iPad

iPod touch

- **Printer**

POS series printers of SNBC

- **Interface**

WIFI

- **Development Environment**

Xcode4.2 to Xcode4.6

1.3 Contents in the Package

Files	Description
POSSDKForIOSlib	POSSDKForIOS.a , PortIO.h, ErrorCode.h, POSCommand.h, POSSDK.h, WIFIPort.h, WIFIPortToFile.h
POSSDKForIOSDemo	Project of sample program
User's Manual	POS SDK For IOS User's Manual_Chinese.pdf POS SDK For IOS User's Manual_English.pdf

1.4 Version Record

Version	Date	Description
V1.00	30/10/2013	Initial draft

2. Sample Program

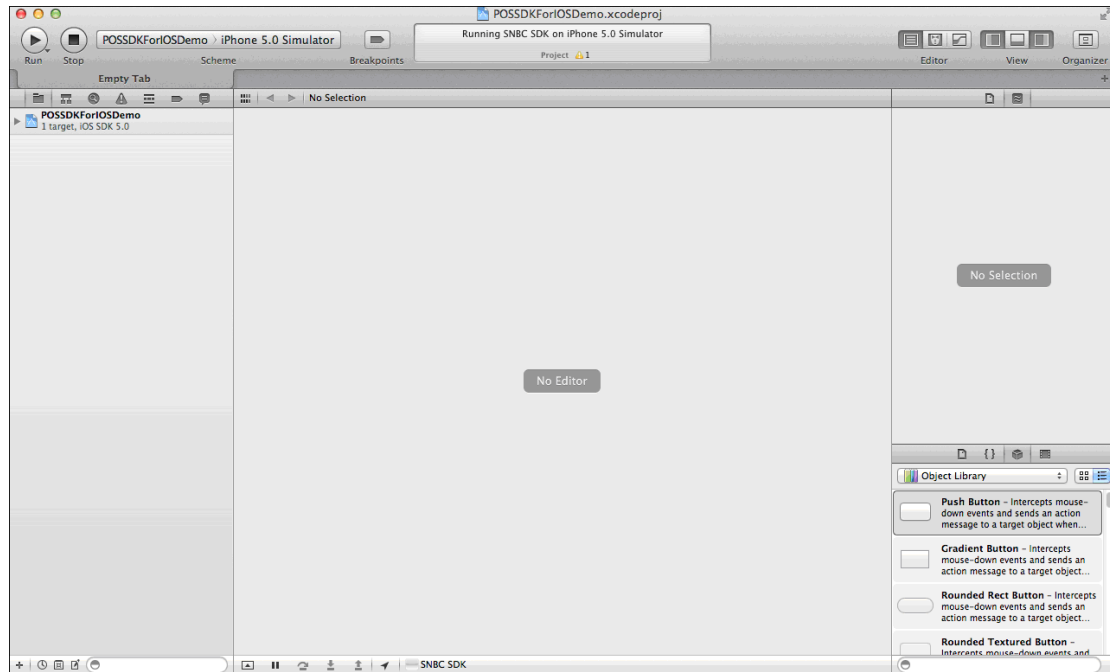
This chapter aims to describe how to use the sample program.

The sample program has the following functionality:

- Searching for printers.
- Input IP Address.
- Open port.
- Close port.
- Set standard mode parameters.
- Print page mode (text/barcode/Image).
- Text printing.
- Text raster printing.
- Custom character printing.
- Barcode printing.
- Barcode PDF417 printing.
- Barcode QR printing.
- Barcode Maxicode printing.
- Barcode GS1 Databar printing.
- Image printing(8/24-dot single/double-density).
- Image downloading for RAM/Flash printing.
- Raster image printing.
- Download file and printing.
- Printer status acquisition.
- Feed line.
- Cut paper.

2.1 Start up the Sample Program

- 1) Extract the sample program zip file to a directory of your choosing.
- 2) Go to the directory you extracted the files to (POSSDKForIOSDemo) and double click on "POSSDKForIOSDemo.xcodeproj".



- 3) Xcode will start up. Select your target device as the "Scheme."
- 4) Click the [Run] button on the upper left.
- 5) The sample program will be installed to the target IOS device, and then the program will start up.

2.2 How to Use the Sample Program

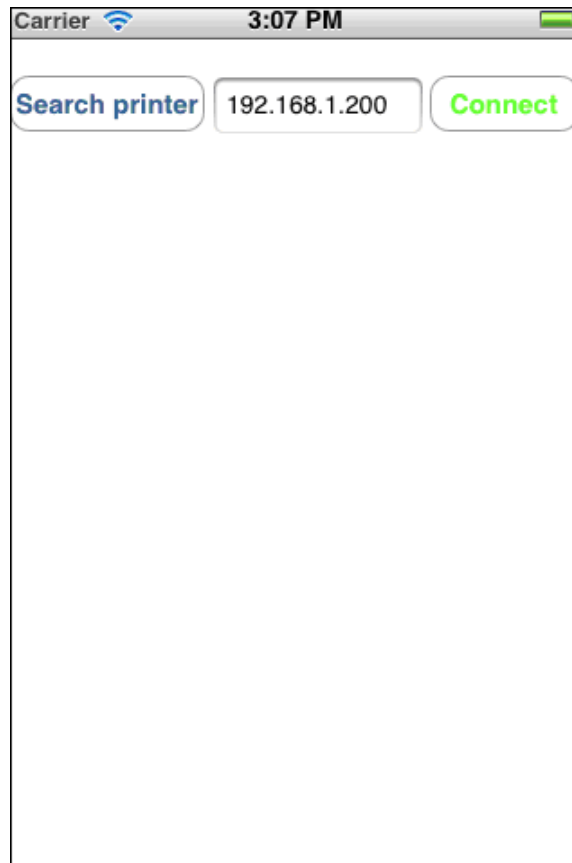
● Create WIFI connecting

To connect to the printer, the following preliminary work must be carried out.

- 1) Make sure the WIFI interface board is available and it has been connected to printer correctly.
- 2) Set IOS device and printer have the same SSID of WIFI network.
- 3) Set the IP addresses of IOS device and printer in the same network segment.

● How to connect printer using this sample program

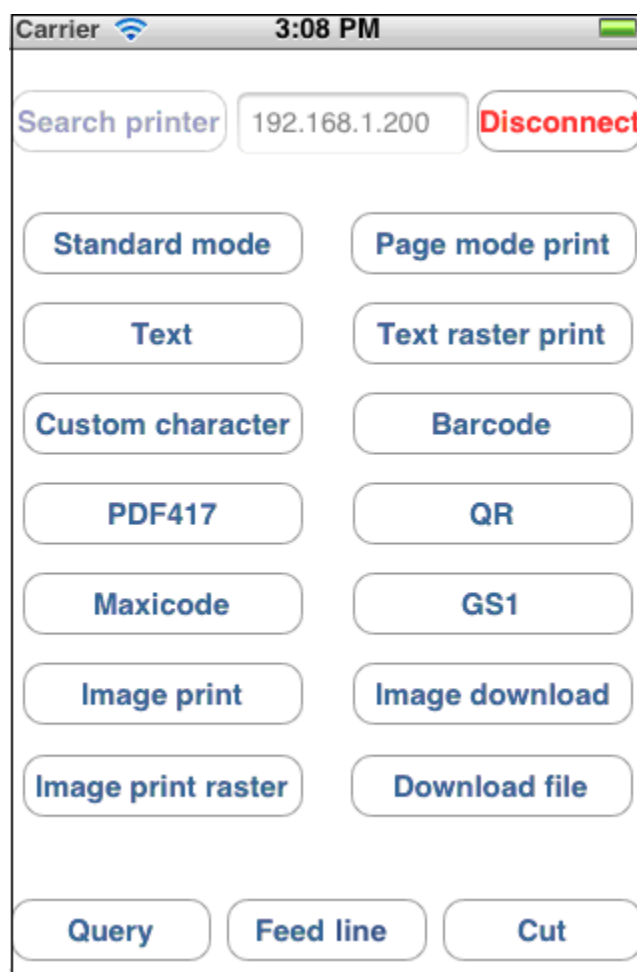
Searching WIFI and connecting to printer's port using the following screen:



Follow the procedures below to use the sample program to search WIFI and connect to port:

- 1) Start up the sample program. For details, see also: [Start up the Sample Program](#).
- 2) Search for printers. Tap [Search printer] on the main screen, the available printers can be listed. From the list of IP addresses, you can select IP address of the printer which you want to use.
- 3) After you select printer, the IP addresses will be added to the Label controller. Of course, if you don't want to carry out procedure 2), you can input the IP address into the Label controller directly. The defaulted IP is 192.168.1.200.
- 4) After the IP address selected, tap [Connect] on the main screen to open the printer's port.
- 5) If connect successfully, the button [Search printer] and the label controller "IP Address" are dark and disable to be used. At the same time, the button [Connect] turn to [Disconnect], and you can close the opened port at all times.

On the main screen, there are the following function buttons:



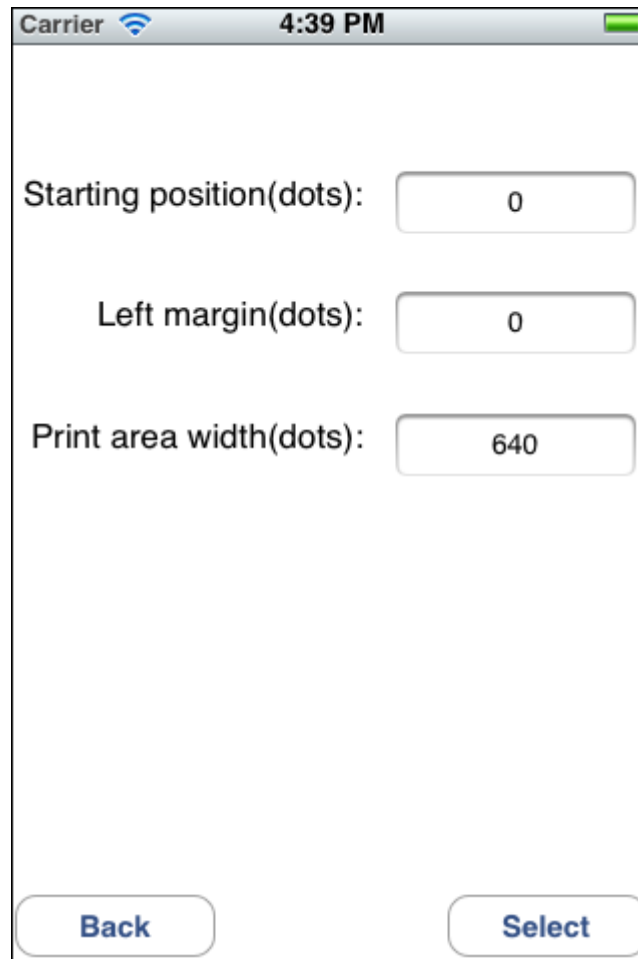
Execute the following processes:

Process	Description
Standard Mode Set	Tap [Standard mode] on the main screen. For details, refer to Setting standard mode .
Page Mode Printing	Tap [Page mode print] on the main screen.
Text Printing	Tap [Text] on the main screen. For details, refer to Text printing .
Text raster printing	Tap [Text raster print] on the main screen. For details, refer to Text raster printing .
Custom character printing	Tap [Custom character] on the main screen.
Barcode printing	Tap [Barcode] on the main screen. For details, refer to Barcode printing .
Barcode PDF417 printing	Tap [PDF417] on the main screen. For details, refer to Barcode PDF417 printing .
Barcode QR printing	Tap [QR] on the main screen. For details, refer to Barcode QR printing .

Barcode Maxicode printing	Tap [Maxicode] on the main screen.
Barcode GS1 Databar printing	Tap [GS1] on the main screen. For details, refer to Barcode GS1 Databar printing .
Image printing	Tap [Image print] on the main screen. For details, refer to Image printing .
Image download for RAM/Flash printing	Tap [Image download] on the main screen. For details, refer to Image download for RAM/Flash printing .
Raster image printing	Tap [Image print raster] on the main screen. For details, refer to Raster image printing .
Download file and printing	Tap [Download file] on the main screen.
Printer status acquisition	Tap [Query] on the main screen.
Feed line	Tap [Feed line] on the main screen.
Cut paper	Tap [Cut] on the main screen.

● Setting standard mode

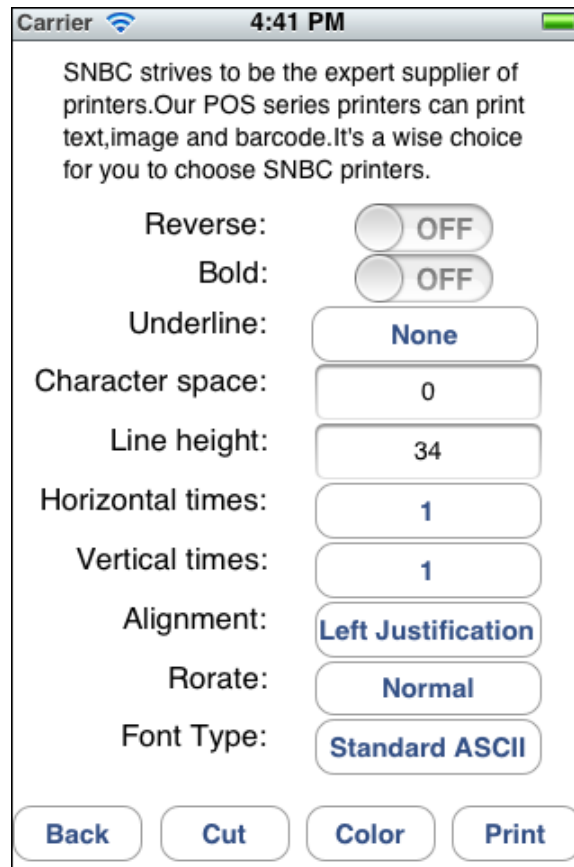
1) The following screen to specify starting position, left margin and print area width for standard mode:



- 2) Tap [Select] on the Standard mode screen, can realize to set standard mode.
- 3) Tap [Back] on the Standard mode to back to main screen from Standard mode screen.

- **Text printing**

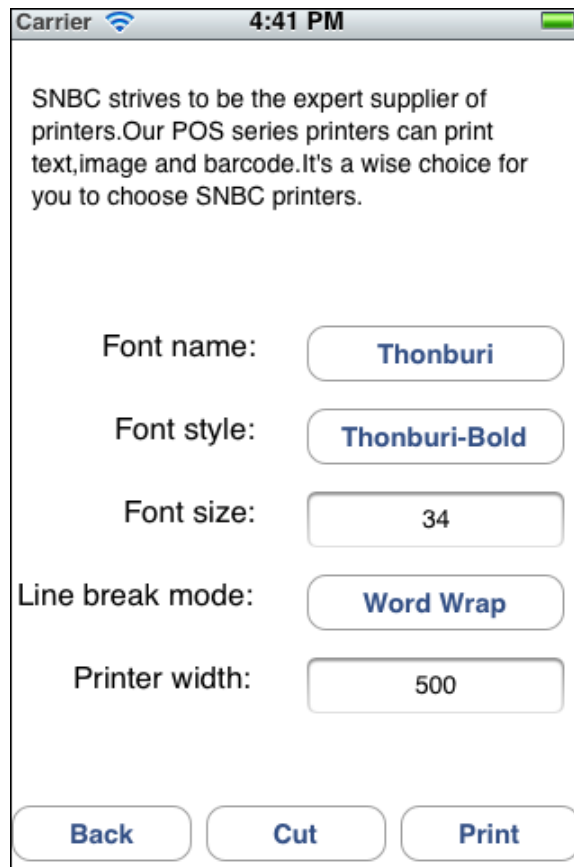
- 1) Enter a string to print for [textContent].
- 2) Specifies the character properties for the string printing. The following properties on Text screen can be specified:



- 3) Tap [Print] on Text screen to print string.
- 3) Tap [Color] on Text screen to print string using another color other than black.
- 4) Tap [Cut] on Text screen to cut paper.
- 5) Tap [Back] on Text screen to back to main screen from Text screen.

• Text raster printing

- 1) Enter a string to print for [TextRasterPrintContent].
- 2) Specifies the character properties for the string printing. The following properties on Text raster print screen can be specified:



Carrier 4:41 PM

SNBC strives to be the expert supplier of printers. Our POS series printers can print text, image and barcode. It's a wise choice for you to choose SNBC printers.

Font name: Thonburi

Font style: Thonburi-Bold

Font size: 34

Line break mode: Word Wrap

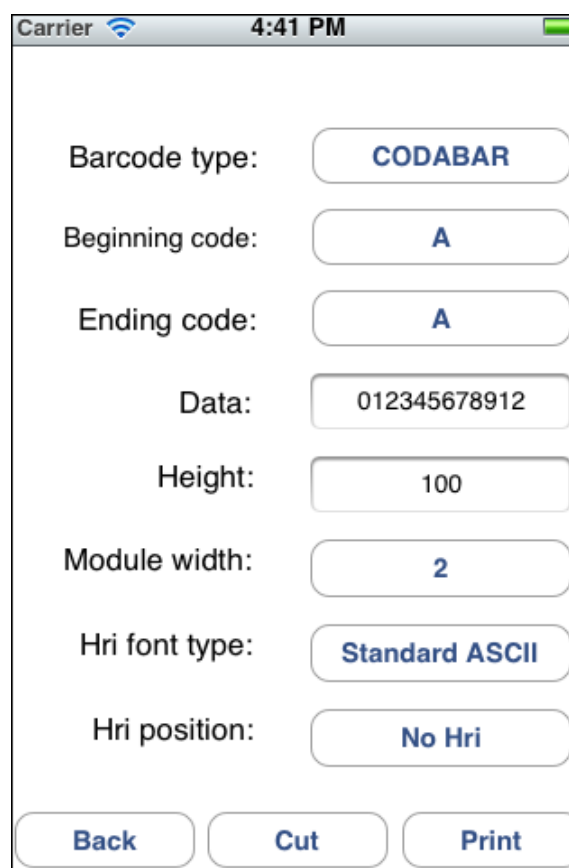
Printer width: 500

Back Cut Print

- 3) Tap [Print] on Text raster print screen to print raster text.
- 4) Tap [Cut] on Text raster print screen to cut paper.
- 5) Tap [Back] on Text raster print screen to back to main screen from Text raster print screen.

● Barcode printing

- 1) Tap [Barcode type] to choose barcode type.
- 2) Enter a barcode data string to print for [BarcodeData].
- 3) Specifies the barcode properties for the barcode to print. The following properties on Barcode screen can be specified:

A screenshot of a mobile application's barcode configuration screen. The screen has a status bar at the top showing 'Carrier', a signal strength icon, the time '4:41 PM', and a battery level icon. The main area contains several labeled input fields, each with a rounded rectangular button containing the selected value. The labels and values are: 'Barcode type:' with 'CODABAR', 'Beginning code:' with 'A', 'Ending code:' with 'A', 'Data:' with '012345678912', 'Height:' with '100', 'Module width:' with '2', 'Hri font type:' with 'Standard ASCII', and 'Hri position:' with 'No Hri'. At the bottom, there are three buttons labeled 'Back', 'Cut', and 'Print'.

- 4) Tap [Print] on Barcode screen to print 1D barcode.
- 5) Tap [Cut] on Barcode screen to cut paper.
- 6) Tap [Back] on Barcode screen to back to main screen from Barcode screen.

Remarks

- a) For barcode Codabar, it has two special properties to select: beginning code and ending code.
- b) For barcode Code128, it has one special propertie to select: character set.

• Barcode PDF417 printing

- 1) Enter a barcode data string to print for [PDF417Data].
- 2) Specifies the barcode PDF417 properties for the PDF417 to print. The following properties on PDF417 screen can be specified:

Carrier 4:42 PM

Data:

Appearance to height:

Appearance to width:

Rows:

Columns:

XSize:

Line height:

Correction:

- 3) Tap [Print] on PDF417 screen to print barcode PDF417.
- 4) Tap [Cut] on PDF417 screen to cut paper.
- 5) Tap [Back] on PDF417 screen to back to main screen from PDF417 screen.

- **Barcode QR printing**

- 1) Enter a barcode data string to print for [QRData].
- 2) Specifies the barcode QR properties for the QR to print. The following properties on QR screen can be specified:

The screenshot shows a mobile application interface with a status bar at the top displaying 'Carrier', a signal strength icon, '4:43 PM', and a battery icon. The main content area contains the following elements:

- Data:** A text input field containing the string 'QA,AABCEFGHIJKLMN123456...'.
- Basic element width:** A numeric input field with the value '5'.
- Symbol type:** A dropdown menu with 'Enhanced Type' selected.
- Language mode:** A dropdown menu with 'Chinese' selected.

At the bottom of the screen, there are three buttons: 'Back', 'Cut', and 'Print'.

- 3) Tap [Print] on QR screen to print barcode QR.
- 4) Tap [Cut] on QR screen to cut paper.
- 5) Tap [Back] on QR screen to back to main screen from QR screen.

- **Barcode GS1 Databar printing**

- 1) Enter a barcode data string to print for [GS1DataBarData].
- 2) Specifies the barcode GS1 properties for the GS1 to print. The following properties on GS1 screen can be specified:

Carrier 4:43 PM

Data: 0123456789012

Barcode type: Omnidirectional

Barcode height: 100

Basic element width: 3

Basic element height: 4

HRI: DataBar And 2D

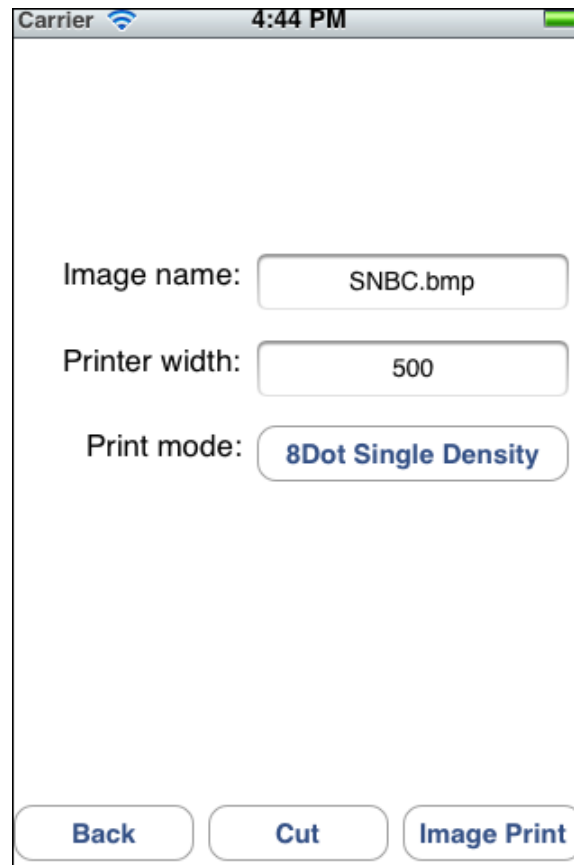
AI: 1

Back Cut Print

- 3) Tap [Print] on GS1 screen to print barcode GS1.
- 4) Tap [Cut] on GS1 screen to cut paper.
- 5) Tap [Back] on GS1 screen to back to main screen from GS1 screen.

- **Image printing**

- 1) Enter image name to print for [ImageName].
- 2) Specifies printer width and print mode for image printing:



- 4) Tap [Image Print] on Image print screen to print image.
- 5) Tap [Cut] on Image print screen to cut paper.
- 6) Tap [Back] on Image print screen to back to main screen from Image print screen.

• Image download for RAM/Flash printing

- 1) Tap UISegmentedControl controller [RAMorFlashSelectButton] to choose printing RAM or Flash image.
- 2) Enter image name to print for Label controller. Every time you can download a single image for RAM. While you can download a single image for Flash and separated by comma for every two images.
- 3) Specifies RAM/Flash image properties for RAM/Flash image to print. The following properties can be specified:

Carrier 4:45 PM

RAM Flash

Image name: SNBC.bmp

Printer width: 500

Image ID: 1

Download

Image ID: 1

Print mode: Normal Size

Print

Back Cut

Carrier 4:45 PM

RAM Flash

Image name: SNBC.bmp,Jpg.jpg,face.P...

Printer width: 500

Download

Image ID: 1

Print mode: Normal Size

Print

Back Cut

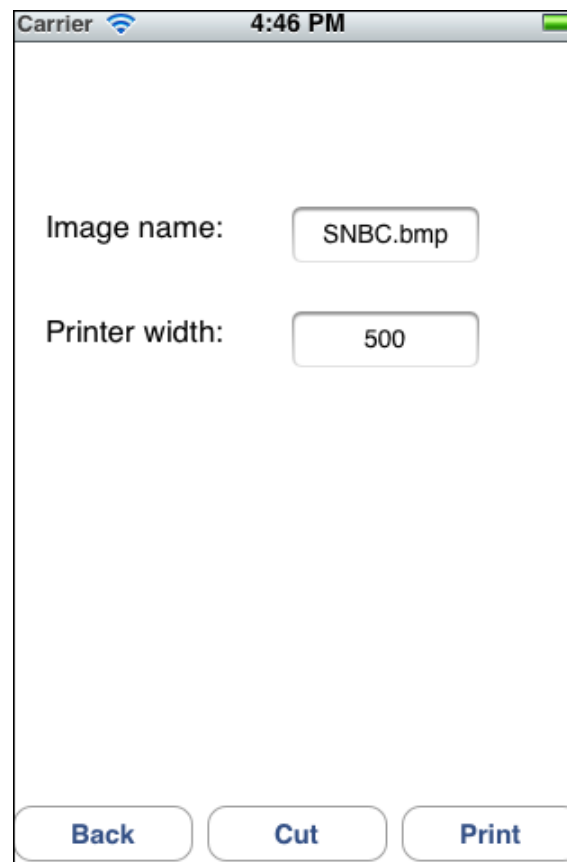
4) Tap [Download] on Image download screen to download images.

- 5) Tap [Print] on Image download screen to print image.
- 6) Tap [Cut] on Image download screen to cut paper.
- 7) Tap [Back] on Image download screen to back to main screen from Image download screen.

• **Raster image printing**

1) Enter image name to print for [RasterImageName]. Every time you can input a single image.

2) Specifies Printer width for raster image printing:



The screenshot shows a mobile application interface for raster image printing. At the top, the status bar displays 'Carrier', a Wi-Fi signal icon, the time '4:46 PM', and a battery level icon. The main content area has two labels with corresponding input fields: 'Image name:' followed by a text box containing 'SNBC.bmp', and 'Printer width:' followed by a text box containing '500'. At the bottom of the screen, there are three rounded rectangular buttons labeled 'Back', 'Cut', and 'Print' in blue text.

- 3) Tap [Print] on the Image print raster screen to print raster image.
- 4) Tap [Cut] on the Image print raster screen to cut paper.
- 5) Tap [Back] on the Image print raster screen to back to main screen from Image print raster screen.

3. Programming Guide

This chapter describes how to write programs in the application development using POS SDK For IOS.

3.1 Using WIFI to Connect to Printer's Port

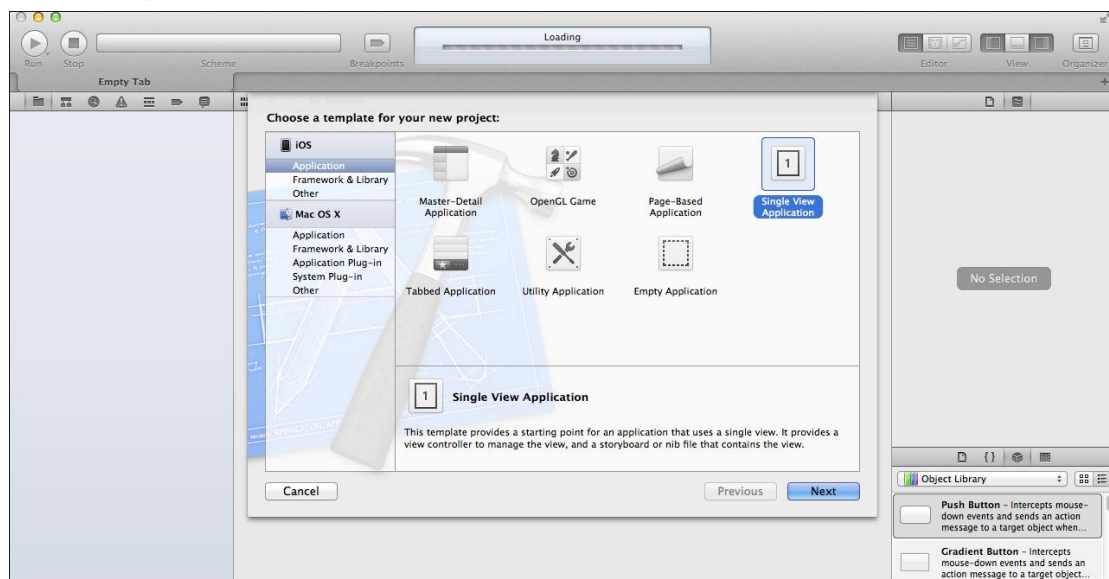
Make sure the printer's WIFI can connect to IOS device, for details see also: [Create WIFI connecting.](#)

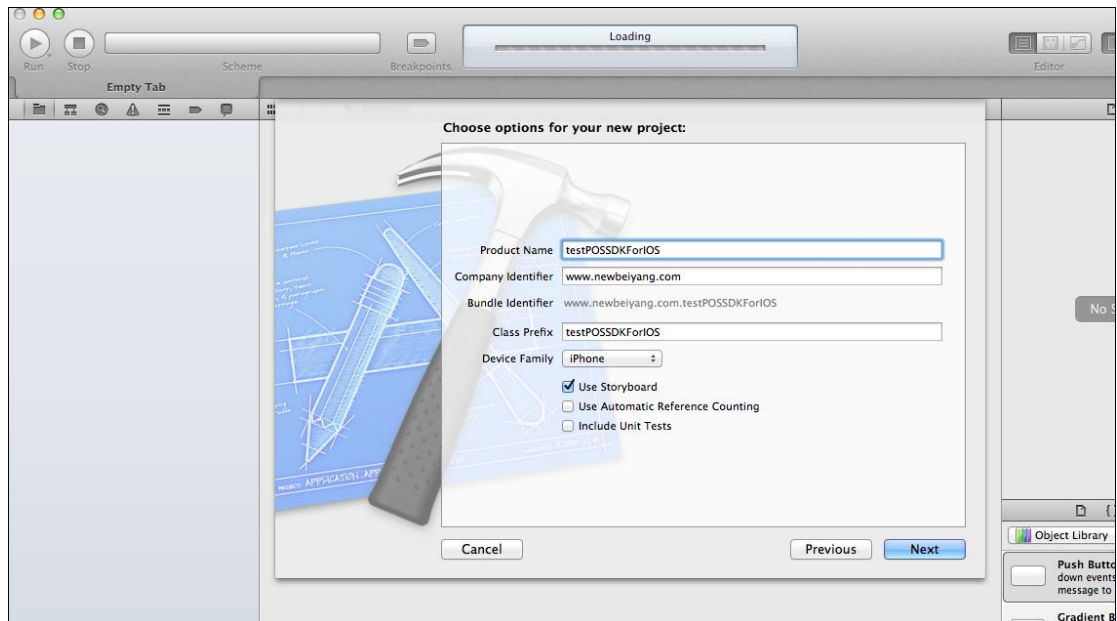
3.2 Using Library

• How to add library files

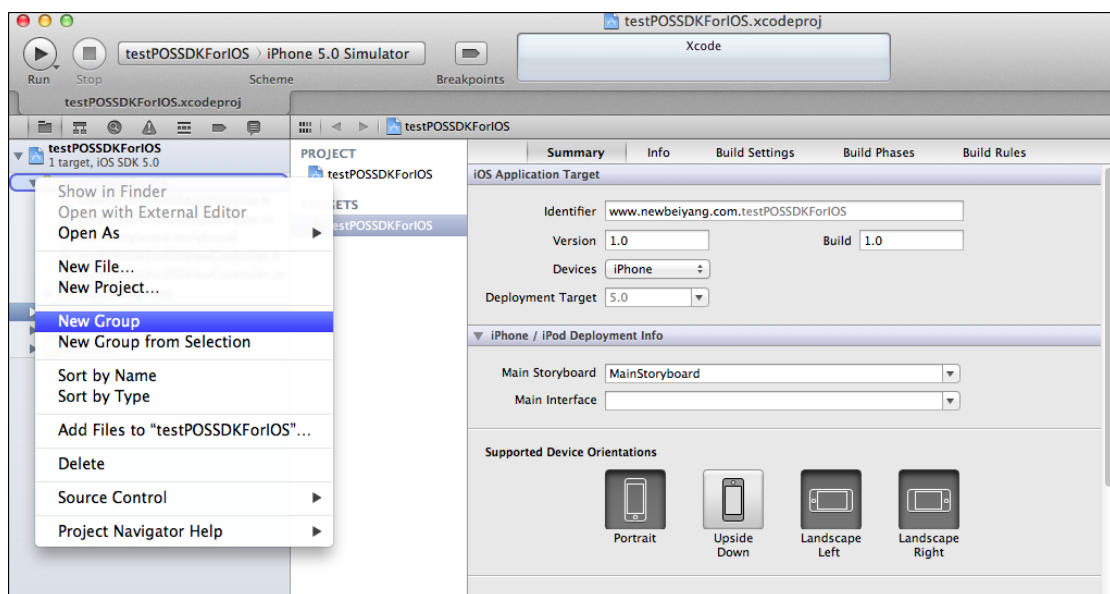
Execute the following steps to add library files to a new project:

Step1: Create a new IOS project and named it. Take testPOSSDKForIOS as an example.

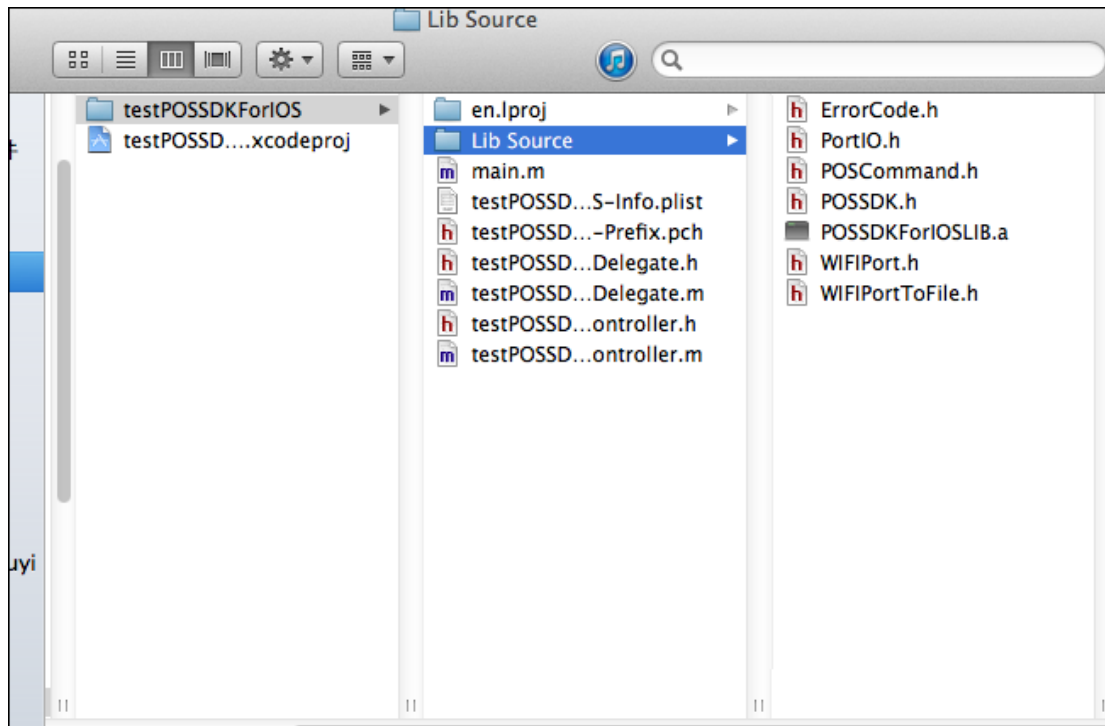




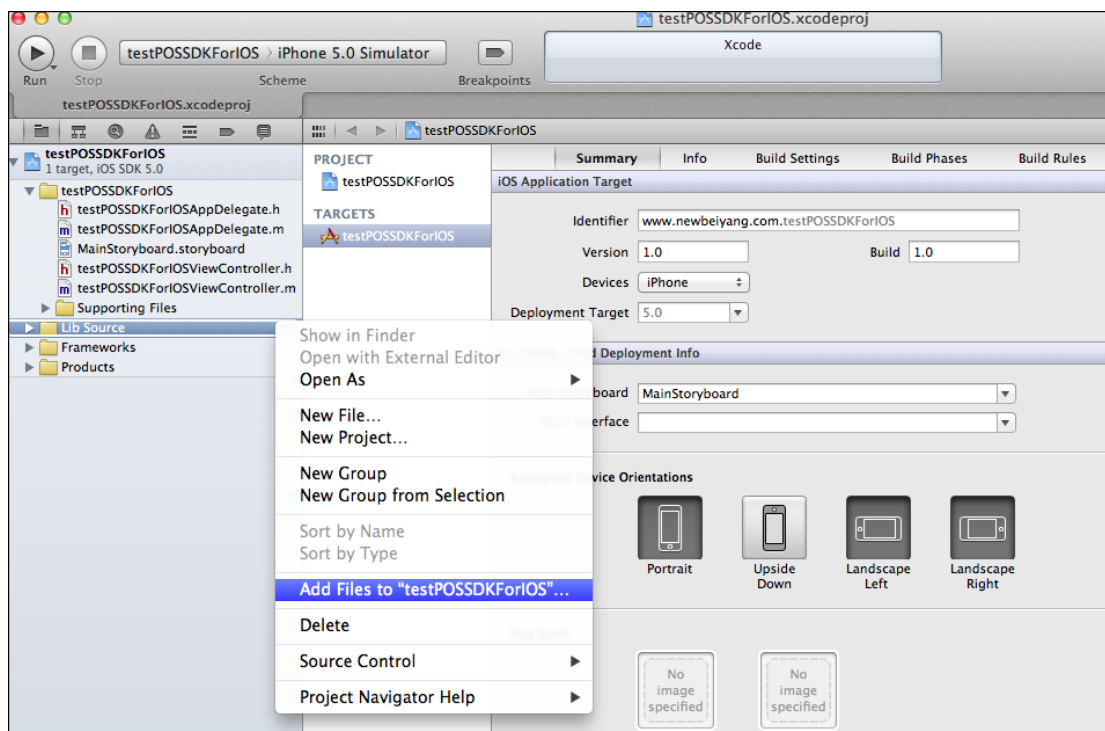
Step2: Create a new group in the project, and change it name. Take Lib Source as an Example.

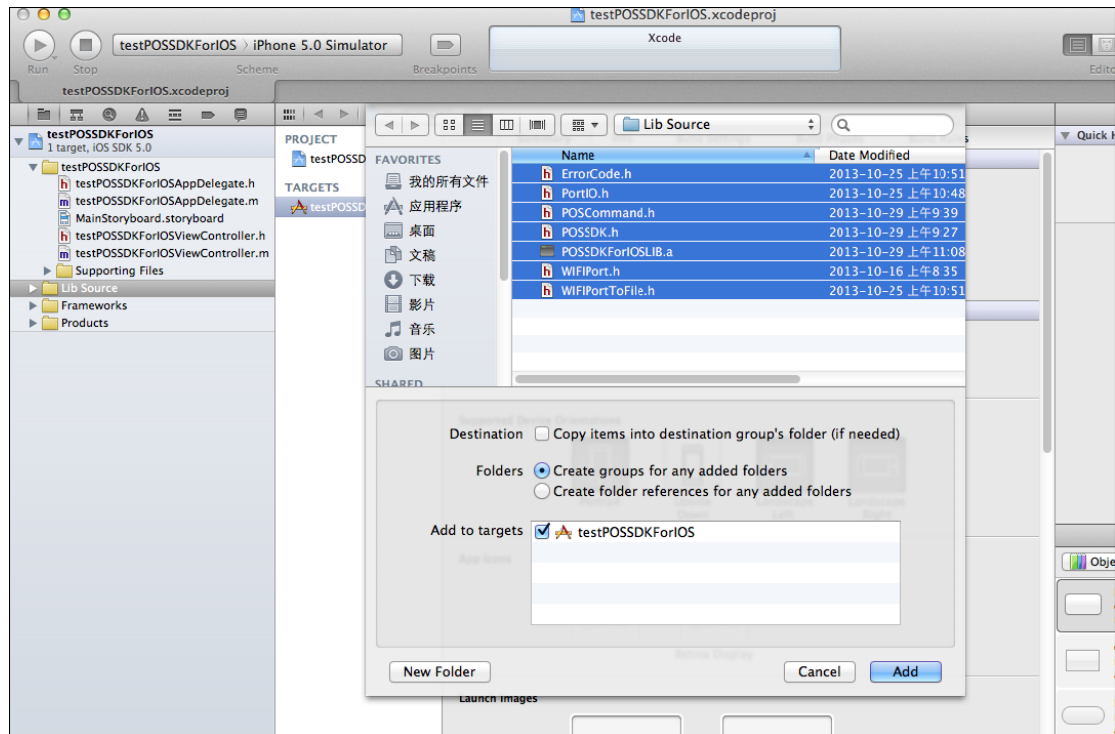


Step3: Create a folder named Lib Source in the file of testPOSSDKForIOS, and then copy POSSDKForIOSLIB.a, ErrorCode.h, PortIO.h, WiFiPort.h, WiFiPortToField.h, POSCommand.h and POSSDK.h to Lib Source.



Step4: As the following two processes to add .a and .h files to Lib Source, and then tap [Add].





After executing the above operations, the .a and .h files were added to the new project.

● How to Use Library

1) To use the library classes and methods, you need to import head files to testPOSSDKForiOSViewController.h:

```
#import "PortIO.h"
#import "WIFIPort.h"
#import "POSSDK.h"
```

2) Declare methods of these classes in testPOSSDKForiOSViewController.m:

```
@interface testPOSSDKForiOSViewController()
@property (nonatomic, retain) POSSDK *pos_sdk;
@property (nonatomic, retain) WIFIPort *printer_port;
@end
```

3) Define methods of these classes in testPOSSDKForiOSViewController.m:

```
@implementation testPOSSDKForiOSViewController
@synthesize pos_sdk = _pos_sdk;
@synthesize printer_port = _printer_port;
- (POSSDK*)pos_sdk
{
    if(_pos_sdk == nil)
    {
        _pos_sdk = [[POSSDK alloc] init];
    }
    return _pos_sdk;
}

- (WIFIPort*)printer_port
{

```

```

if(_printer_port == nil)
{
    _printer_port = [[WIFIPort alloc] init];
}
return _printer_port;
}

```

• Example of methods in sample program

Search printer

```

NSMutableArray *port_info_set = nil;
port_info_set = [self.printer_port searchPort];

```

Connect to printer's port

```

error_code = [self.printer_port openPort: @ "192.168.1.200" TimeOut: TIME_OUT_CONNECT];

```

Close port

```

error_code = [self.printer_port closePort];

```

Set parameters for standard mode

```

error_code = [self.pos_sdk systemSelectPrintMode:PrintModeStandard];
error_code = [self.pos_sdk standardModeSetStartingPosition:0];
error_code = [self.pos_sdk standardModeSetLeftMarginAndPrintAreaWidth:0 Width:640];

```

Set parameters for page mode

```

error_code = [self.pos_sdk systemSelectPrintMode:PrintModePage];
error_code = [self.pos_sdk pageModeSetPrintArea:50 Y:0 AreaWidth:580 AreaHeight:600
PrintDirection:LeftToRight];
error_code = [self.pos_sdk pageModeSetStartingPosition:0 Y:100];
error_code = [self.pos_sdk pageModePrint];
error_code = [self.pos_sdk pageModeClearBuffer];

```

Text printing

```

NSString * text_content = nil;
text_content = @"SNBC strives to be the expert supplier of printers.Our POS series printers can
print text,image and barcode.It's a wise choice for you to choose SNBC printers.";
error_code = [self.pos_sdk textPrint:text_content];

```

Text raster printing

```

error_code = [self.pos_sdk textStandardModeRasterPrint: text_content FontName:@
"Thonburi-Bold" FontSize:34 LineBreakMode: UILineBreakModeWordWrap PrinterWidth:500];

```

Print user defined character

```
NSInteger BytesOfHeight, DotsOfWidth, StartingCode = 'A', EndingCode = 'B';
NSMutableData *CharacterData = nil;
NSInteger index = 0;
UIImage *image = nil;
NSString *image_name[2] = {"1.bmp", "2.bmp"};
NSString *full_path = nil;

CharacterData = [[NSMutableData alloc] initWithCapacity:10240];
for(index = StartingCode; index <= EndingCode; index++)
{
    full_path = [[[NSBundle mainBundle] bundlePath]
stringByAppendingPathComponent:image_name[index - StartingCode]];
    image = [[UIImage alloc] initWithContentsOfFile:full_path];

    ImageDataRef image_data = {0};

    [self.pos_sdk imageFormatConvertToUserDefinedData:image ditheringSupported: FLSE
image_data:&image_data];

    BytesOfHeight = image_data.image_height >> 3;
    DotsOfWidth = image_data.image_width;
    [CharacterData appendBytes:image_data.data_buf length:image_data.image_data_len];

    if(image_data.data_buf != nil) {free(image_data.data_buf);}
    if(image != nil) {[image release];}
}

error_code = [self.pos_sdk textUserDefinedCharacterDefine:BytesOfHeight
DotsOfWidth:DotsOfWidth StartingCode:StartingCode EndingCode:EndingCode
CharacterData:CharacterData];

error_code = [self.pos_sdk textUserDefinedCharacterEnable:1];
error_code = [self.pos_sdk textPrint:@"123AACBB123"];
```

Print barcode

```
NSData *data = nil;
/// Get (NSData *)data for barcode 1D///
error_code = [self.pos_sdk barcodePrint1Dimension: data BarcodeType:BarcodeUPC_A
ModuleWidth:2 BarcodeHeight:100 HriFontType:FontStyleStandardASCII
HriPosition:HRIBelow];
```

Print PDF417

```
NSData *PDF417_Data = nil;
/// Get (NSData *)PDF417_Data for PDF417///
error_code = [self.pos_sdk barcodePrintPDF417: PDF417_Data AppearanceToHeight:1
AppearanceToWidth:1 RowNumber:3 ColumnNumber:1 XSize:4 LineHeight:l5
CorrectionGrade:0];
```

Print QR

```
NSData *QR_Data = nil;
// Get (NSData *)QR_Data for QR///
error_code = [self.pos_sdk barcodePrintQR:QR_Data BasicElementWidth:5
SymbolType:EnhancedType LanguageMode:LanguageChinese];
```

Print Maxicode

```

NSData *Maxicode_Data = nil;
// Get (NSData *)Maxicode _Data for Maxicode ///
error_code = [self.pos_sdk barcodePrintMaxicode: Maxicode_Data];

```

Print GS1 DataBar and GS1 composite barcode

```

NSData *GS1_Data = nil;
/// Get (NSData *) barcode data///
error_code = [self.pos_sdk barcodePrintGS1DataBar: GS1_Data
BarcodeType:GS1DataBar_Omnidirectional BasicElementWidth:3 BarcodeHeight:100
BasicElementHeight:4 SeparatorHeight:1 SegmentNumber:2 HRI:DataBarAnd2DHri UseAI:1];

```

Image print

```

UIImage *image = nil;
///Get (UIImage*) image according to image name///
error_code = [self.pos_sdk imageStandardModePrint: SingleDensity_8 Image:image
StartHorPos:0 PrinterWidth:500];

```

RAM/Flash image download and print**RAM**

```

error_code = [self.pos_sdk imageDownloadToPrinterRAM:0 Image: image PrinterWidth:500];
error_code = [self.pos_sdk imageRAMPrint:0 Mode: NormalMode];

```

Flash

```

NSMutableArray *image_set = nil;
NSString *flash_image_name = nil;
NSString *full_path_Flash = nil;
UIImage *image_Flash = nil;
flash_image_name = @"SNBC.bmp,Jpg.jpg,face.PNG";
list = [flash_image_name componentsSeparatedByString:@","];
image_set = [[NSMutableArray alloc] initWithCapacity:[list count]];
for(index = 0; index < [list count]; index++)
{
    full_path_Flash = [[[NSBundle mainBundle] bundlePath]
stringByAppendingPathComponent:[list objectAtIndex:index]];
    image_Flash = [[UIImage alloc] initWithContentsOfFile:full_path_Flash];
    [image_set addObject:image_Flash];
    [image_Flash release];
}
error_code = [self.pos_sdk imageDownloadToPrinterFlash:image_set PrinterWidth:500];
error_code = [self.pos_sdk imageFlashPrint:1 Mode: Double_width];
error_code = [self.pos_sdk imageFlashPrint:2 Mode: Double_height];
error_code = [self.pos_sdk imageFlashPrint:3 Mode: Quadruple];

```

Raster image print

```

error_code = [self.pos_sdk imageStandardModeRasterPrint:image PrinterWidth:500];

```

Download file

```

NSString* full_path = nil;
full_path = [[[NSBundle mainBundle] bundlePath]
stringByAppendingPathComponent:@"sample1.dat"];
[self.pos_sdk systemDownloadFile:full_path];

```

Query Status

```
#define QueryStatusSize 4
UInt8    StatusBuffer[QueryStatusSize] = {0};
error_code = [self.pos_sdk systemQueryStatus:StatusBuffer ReadSize:QueryStatusSize];
```

Feed line

```
error_code = [self.pos_sdk systemFeedLine:1];
```

Cut paper

```
error_code = [self.pos_sdk systemCutPaper:CutPartAfterFeed FeedDistance:80];
```

4. API Reference

This chapter describes the APIs provided in the POS SDK for IOS.

4.1 WIFI Reference

API	Description
searchPort	Search printer.
openPort	Connect to printer's port.
closePort	Close port.
writePort	Send data to port.
writePort: PortID	Send data to the specified port.
readPort	Receive data from port.
readPort: PortID	Receive data from the specified port.
recordCommunicationDataEnable	Record communication data.

• searchPort

Search printers and get the messages of printers which were searched successfully.

Method

- (NSMutableArray*)searchPort

Return

Return Value	Description
nil	No printer was searched.
The message list of printers which were searched successfully	Search successfully.

The printer message list, as follows:

```
@interface PortInfoWIFI : NSObject
{
    SInt32      NetPortType;
    NSString    *ModuleName;
    NSString    *SSID;
    NSString    *IPAddr;
    NSString    *Gateway;
    NSString    *MACAddr;
```

```
NSString *ExpendInfo;  
}
```

Example

See also: [Search printer](#).

• openPort

Create WIFI connection to printer's port.

Method

- (SInt32)**openPort**:(NSString*)DeviceName TimeOut:(SInt32)TimeOut

Parameter

- **DeviceName** IP address of printer, Such as 192.168.1.200.
- **TimeOut** The timeout of creating connection (ms), and the legal value is non-negative integer. We define several common TimeOut for customers, as follows:

Set Value	Description
TIME_OUT_DEFAULT	TimeOut = 0ms.
TIME_OUT_CONNECT	TimeOut = 3000ms.
TIME_OUT_SMALL_DATA	TimeOut = 1000ms.
TIME_OUT_LARGE_DATA	TimeOut = 4000ms.

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_CREATE_CONNECTION	Failed to create socket. There are not source which can be used in TimeOut. Failed to get socket.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.

Example

See also: [Connect to printer's port](#).

• closePort

Close port.

Method

- (SInt32)**closePort**

Return

Return Value	Description
SUCCESS	Processing was successful.

Example

See also: [Close port](#).

- **writePort**

Send data to a port. When the data is More than 4096 bytes, packet disassemble packet is 4096 bytes.

Method

- (SInt32)**writePort**:(const UInt8*)WriteBuffer OffsetSize:(SInt32)OffsetSize WriteSize:(SInt32)WriteSize WriteTimeOut:(SInt32)WriteTimeOut

Parameter

- **WriteBuffer** The sending data buffer. It stores data to be sent.
- **OffsetSize** Specify the offset value from the top of WriteBuffer.
(a non-negative integer)
- **WriteSize** The number of bytes which would be sent.
- **WriteTimeOut** The timeout of writing data (ms). (a non-negative integer)

Return

Return Value	Description
The bytes of sending data	Processing was successful.
ERR_INVALID_CONNECTION	Failed to connect to printer's port.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.
ERR_COMMUNICATE	Failed to set the TimeOut parameter.

Example

```
data[2] = {0x1b,0x40};
[self.printer_port writePort:data offfSize:0 WriteSize:2 WriteTimeOut:
TIME_OUT_SMALL_DATA];
```

- **writePort: PortID**

Send data to the specified port. When the data is More than 4096 bytes, packet

disassemble packet is 4096 bytes.

Method

- (SInt32)**writePort**:(const UInt8*)WriteBuffer OffsetSize:(SInt32)OffsetSize
WriteSize:(SInt32)WriteSize WriteTimeOut:(SInt32)WriteTimeOut
PortID:(SInt32)PortID

Parameter

- **WriteBuffer** The sending data buffer. It stores data to be sent.
- **OffsetSize** Specify the offset value from the top of WriteBuffer. (a non-negative integer)
- **WriteSize** The number of bytes which would be sent.
- **WriteTimeOut** The timeout of writing data (ms). (a non-negative integer)
- **PortID** The ID of port.

PortIDSet Value	Description
NET_PORT_COMMAND	NET_PORT_ID_COMMAND = 9100
NET_PORT_QUERY	NET_PORT_ID_QUERY = 4000
Other values	An invalid parameter was passed.

Return

Return Value	Description
The bytes of sending data	Processing was successful.
ERR_INVALID_CONNECTION	Failed to connect to printer's port.
ERR_COMMUNICATE	Failed to set the TimeOut parameter.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.

Example

```
[self.printer_port writePort:data offSize:0 WriteSize:2 WriteTimeOut:  
TIME_OUT_SMALL_DATA PortID: NET_PORT_COMMAND];
```

• readPort

Receive data from a port.

Method

- (SInt32)**readPort**:(UInt8*)ReadBuffer OffsetSize:(SInt32)OffsetSize
ReadSize:(SInt32)ReadSize ReadTimeOut:(SInt32)ReadTimeOut

Parameter

- **ReadBuffer** The receiving data buffer for storing received data.
- **OffsetSize** Specify the offset value from the top of ReadBuffer. (a non-negative integer)
- **ReadSize** The number of bytes which would be received. (a non-negative integer)
- **ReadTimeout** The timeout of receiving data (ms).

Return

Return Value	Description
The bytes of reading data	Processing was successful.
ERR_INVALID_CONNECTION	Failed to connect to printer.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.
ERR_ALLOC_MEMORY	Failed to alloc memory.
ERR_COMMUNICATE	Failed to set Timeout. Actually reading size is 0.

Example

```
SInt32 read_size;
UInt8 pointBuffer[512];
[self.printer_port readPort: pointBuffer offSize:0 ReadSize: read_size WriteTimeout:
TIME_OUT_SMALL_DATA];
```

- **readPort:PortID**

Receive data from the specified port.

Method

```
- (SInt32)readPort:(UInt8*)ReadBuffer OffsetSize:(SInt32)OffsetSize
ReadSize:(SInt32)ReadSize ReadTimeout:(SInt32)ReadTimeout
PortID:(SInt32)PortID
```

Parameter

- **ReadBuffer** The receiving data buffer for storing received data.
- **OffsetSize** Specify the offset value from the top of ReadBuffer. (a non-negative integer)
- **ReadSize** The number of bytes which would be read.
- **ReadTimeout** The timeout of receiving data (ms). (a non-negative integer)

-
- **PortID** The ID of port.

PortIDSet Value	Description
NET_PORT_COMMAND	NET_PORT_ID_COMMAND = 9100.
NET_PORT_QUERY	NET_PORT_ID_QUERY = 4000.
Other values	An invalid parameter was passed.

Return

Return Value	Description
The bytes of reading data	Processing was successful.
ERR_INVALID_CONNECTION	Failed to connect to printer.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.
ERR_ALLOC_MEMORY	Failed to alloc memory.
ERR_COMMUNICATE	Failed to set TimeOut. Actually reading size is 0.

Example

```
[self.printer_port readPort: pointBuffer offSize:0 ReadSize: read_size WriteTimeOut:
TIME_OUT_SMALL_DATA PortID: NET_PORT_QUERY];
```

• recordCommunicationDataEnable

Record communication data.

Method

- (SInt32)**recordCommunicationDataEnable**:(NSString*)recordFileName

Parameter

- **recordFileName** The file name recorded.

Return

Return Value	Description
SUCCESS	Processing was successful.
Other values	Processing was failed.

Example

```
[self.printer_port recordCommunicationDataEnable: @"DataFile.dat"];
```

4.2 POS SDK API Reference

Prefix	API	Description
Prefix as system (methods about system)	systemSetPortIO	Set communication module instance for printer.
	systemSetEncoding	Set encoding for printing text.
	systemDownloadFile	Download File.
	systemReset	Initialize printer, clear data in print buffer and set print mode to the default mode when powered on.
	systemSelectPrintMode	Select the print mode.
	systemSelectPaperType	Select the paper type.
	systemSetMotionUnit	Set the horizontal and vertical motion units.
	systemQueryStatus	Query the printer's status.
	systemFeedLine	Print and feed line.
Prefix as cashdrawer	systemCutPaper	Select cut paper mode and cut paper.
	cashdrawerOpen	Output the cash drawer control pulse to specified connector pin.
Prefix as text (methods about text)	textSelectCharSetAndCodePage	Select an international character set and Code page.
	textSetLineHeight	Set line height.
	textSetCharacterSpace	Set character spacing.
	textStandardModeAlignment	Align all the data in one line to the specified position (Standard mode).
	textStandardModeUpsideDown	Turn on/off upside-down printing mode.
	textPrint: String	Print text (NSString*).
	textPrint: Buffer	Print text (UInt8*).
	textSelectFontMagnifyTimes	Select character size.
	textStandardModeRotate	Rotate integer times 90 degree.
	textSelectFont	Select character font and font style.
	textSetColorPrint	Enter bi-colour print mode and set print color.
	textQuitColorPrint	Quit bi-colour print mode.

	<u>textUserDefinedCharacterEnable</u>	User-defined character is enable/disable.
	<u>textUserDefinedCharacterDefine</u>	Define user-defined character.
	<u>textUserDefinedCharacterCancel</u>	Text cancel font user-defined of char code.
	<u>textUserDefinedChineseCharacter Define</u>	Define user-defined font for Chinese Character.
	<u>textStandardModeRasterPrint</u>	Text raster print in standard mode.
Prefix as image (methods about image)	<u>imageStandardModePrint</u>	Image print in standard mode.
	<u>imageDownloadToPrinterRAM</u>	Download images to RAM.
	<u>imageRAMPrint</u>	Print image which downloaded to RAM.
	<u>imageDownloadToPrinterFlash</u>	Download images to Flash.
	<u>imageFlashPrint</u>	Print image which downloaded to Flash.
	<u>imageStandardModeRasterPrint</u>	Print raster image in standard mode.
	<u>imageFormatConvertToUserDefinedData</u>	Convert image format.
Prefix as barcode (methods about barcode)	<u>barcodeGetIDByName</u>	Get barcode ID by comparing name of barcode.
	<u>barcodeGetNameByID</u>	Get name by barcode ID.
	<u>barcodePrint1Dimension</u>	Print 1Dimension barcode.
	<u>barcodePrintQR</u>	Print barcode QR.
	<u>barcodePrintPDF417</u>	Set barcodePDF417 size and print PDF417.
	<u>barcodePrintMaxicode</u>	Print Maxicode.
	<u>barcodePrintGS1DataBar</u>	Print GS1 DataBar and GS1 composite barcode.
Prefix as standardMode (methods about standard mode)	<u>standardModeSetLeftMarginAnd PrintAreaWidth</u>	Set print area position and width in standard mode.
	<u>standardModeSetHorizontalStartingPosition</u>	Set horizontal starting position in standard mode.
Prefix as	<u>pageModeSetStartingPosition</u>	Set horizontal and vertical starting position in page mode.

pageMode (methods about page mode)	pageModeSetPrintArea	Set print area in page mode.
	pageModePrint	Print data in page mode.
	pageModeClearBuffer	Delete all the print data in current area.

• **systemSetPortIO**

Set communication module instance for printer.

Method

- (SInt32)**systemSetPortIO**:(PortIO *)Port_IO

Parameter

- **Port_IO** Instance has been connected to printer.

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_INVALID_ARGUMENT	Port_IO is not the object of PortIO or the object is invalid

Example

```
[self.pos_sdk systemSetPortIO:self.printer_port];
```

• **systemSetEncoding**

Set encoding for printing text.

Method

- (SInt32)**systemSetEncoding**:(NSStringEncoding)encoding

Parameter

- **encoding** The text encoding provided by IOS.

Return

Return Value	Description
SUCCESS	Processing was successful.

Example

```
[self.pos_sdk
```

```
systemSetEncoding:CFStringConvertEncodingToNSStringEncoding(kCFStringEncodingGB_18030_2000)];//Set encoding for chinese character
```

- **systemDownloadFile**

Download file.

Method

- (SInt32)**systemDownloadFile**:(NSString*)FileName

DownloadTimeOut:(SInt32)DownloadTimeOut

Parameter

- **FileName** The file name of the file which would be download.
- **DownloadTimeOut** Time out of download file(s). You can set 1-8. If DownloadTimeOut > 8, we define DownloadTimeOut as 8.

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_COMMUNICATE	Processing was failed.
ERR_INVALID_ARGUMENT	The file name is wrong or data is null.

Example

```
NSString * full_path = [[[NSBundle mainBundle] bundlePath]
stringByAppendingPathComponent:@"sample.dat"];
[self.pos_sdk systemDownloadFile:full_path DownloadTimeOut:3];
```

- **systemReset**

Initialize printer, clear data in print buffer and set print mode to the default mode when powered on.

Method

- (SInt32)**systemReset**

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_SYSTEM_RESET	Printer resetting failed.

Example

```
[self.pos_sdk systemReset];
```

- **systemSelectPrintMode**

Select the print mode.

Method

- (SInt32)**systemSelectPrintMode**:(SInt32)PrintMode

Parameter

- **PrintMode** Print mode

PrintMode Set Value	Description
PrintModeStandard	Select standard mode.
PrintModePage	Select page mode
Other values	Invalid parameter

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_SYSTEM_SELECT_PRINT_MODE	The print mode selecting failed.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.

Example

```
[self.pos_sdk systemSelectPrintMode: PrintModeStandard];// Standard mode
[self.pos_sdk systemSelectPrintMode:PrintModePage];// Page mode
```

- **systemSelectPaperType**

Select the paper type.

Method

- (SInt32)**systemSelectPaperType**:(SInt32)PaperType

Parameter

- **PaperType** Paper type

PaperType Set Value	Description
PaperTypeContinuous	Continuous paper

PaperTypeMarked	Marked paper
Other values	Invalid parameter

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_SYSTEM_SELECT_PAPER_TYPE	The paper type selecting failed.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.

Example

```
[self.pos_sdk systemSelectPaperType: PaperTypeCoutinuous];// Coutinuous paper
[self.pos_sdk systemSelectPaperType: PaperTypeMarked];// Marked paper
```

● systemSetMotionUnit

Set the horizontal and vertical motion units.

Method

- (SInt32)**systemSetMotionUnit**:(SInt32)HorizontalUnit
VerticalUnit:(SInt32)VerticalUnit

Parameter

- **HorizontalUnit** Horizontal unit

HorizontalUnit Set Value	Description
0-255	Legal value
Other values	Invalid parameter

- **VerticalUnit** Vertical unit

HorizontalUnit Set Value	Description
0-255	Legal value
Other values	Invalid parameter

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_SYSTEM_SET_MOTION_UNIT	The motion uint selecting failed.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.

Example

```
[self.pos_sdk systemSetMotionUnit:203 VerticalUnit:203];
```

• **systemQueryStatus**

Query the printer's status.

Method

- (SInt32)**systemQueryStatus**:(UInt8*)QueryStatusBuffer
ReadSize:(SInt32)ReadSize

Parameter

- **QueryStatusBuffer** The buffer for storing the printer's status.
- **ReadSize** The number of bytes which would be read.

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_SYSTEM_QUERY_STATUS	The printer's status querying failed.
ERR_COMMUNICATE	The bytes returning are different from bytes of datas which will be read

Example

```
#define QueryStatusSize 4
UInt8      StatusBuffer[QueryStatusSize] = {0};
[self.pos_sdk systemQueryStatus:StatusBuffer ReadSize:QueryStatusSize];
```

• **systemFeedLine**

Print and feed line.

Method

- (SInt32)**systemFeedLine**:(SInt32)LineNum

Parameter

- **LineNum** The number of feed line

LineNum	Set Value	Description
1-255		Legal value
Other values		Invalid parameter

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_SYSTEM_FEED_LINE	Feed line failed.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.

Example

```
error_code = [self.pos_sdk systemFeedLine:5];
```

• systemCutPaper

Select cut paper mode and cut paper.

Method

- (SInt32)**systemCutPaper**:(SInt32)CutMode

FeedDistance:(SInt32)FeedDistance

Parameter

- **CutMode** Feed paper distance

LineNum	Set Value	Description
CutFullImmdediately		Full cut
CutPartImmdediately		Part cut
CutPartAfterFeed		Feed line and part cut
Other values		Invalid parameter

- **FeedDistance** The distance of feed line

FeedDistance	Set Value	Description
0-255		Legal value
Other values		Invalid parameter

Remarks

a) If the parameter CutMode is specified as CutFullImmdediately or CutPartImmdediately, the parameter FeedDistance will be ignored.

b) If the parameter CutMode is specified as CutPartAfterFeed, the printer will feed FeedDistance distance and cut.

c) The parameter FeedDistance will be ignored at the Mark Paper mode. The printer will find the mark and cut.

Return

Return Value	Description
--------------	-------------

SUCCESS	Processing was successful.
ERR_SYSTEM_CUT_PAPER	The paper cutting failed.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.

Example

```
error_code = [self.pos_sdk systemCutPaper:CutPartAfterFeed FeedDistance:80];
```

- **cashdrawerOpen**

Output the cash drawer control pulse to specified connector pin.

Method

- (SInt32)**cashdrawerOpen**:(SInt32)CashdrawerID

PulseOnTimes:(SInt32)PulseOnTimes PulseOffTimes:(SInt32)PulseOffTimes

Parameter

- **CashdrawerID** Connector pin of cashdrawer

CashdrawerID	Set Value	Description
0		Drawer kick-out connector pin 2.
1		Drawer kick-out connector pin 5.
Other values		Invalid parameter.

- **PulseOnTimes** The pulse ON time

PulseOnTimes	Set Value	Description
0-255		Legal value.
Other values		Invalid parameter.

- **PulseOffTimes** The pulse Off time

PulseOffTimes	Set Value	Description
0-255		Legal value.
Other values		Invalid parameter.

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_CASH_DRAWER_OPEN	The cashdrawer opening failed.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.

Example

```
[self.pos_sdk cashdrawerOpen:0 PulseOnTimes:100 PulseOffTimes:100];
```

- **textSelectCharSetAndCodePage**

Select an international character set and code page.

Method

- (SInt32)**textSelectCharSetAndCodePage**:(SInt32)CharSet
codePage:(SInt32)CodePage

Parameter

- **CharSet** International character set

CharSet	Set Value	Description
CharacterSetUSA		U.S.A
CharacterSetFrance		France
CharacterSetGermany		Germany
CharacterSetUK		U.K
CharacterSetDenmark_I		Denmark I
CharacterSetSweden		Sweden
CharacterSetItaly		Italy
CharacterSetSpain_I		Spain I
CharacterSetJapan		Japan
CharacterSetNorway		Norway
CharacterSetDenmark_II		Denmark II
CharacterSetSpain_II		Spain II
CharacterSetLatin_America		Latin America
CharacterSetKorea		Korea
Other values		Invalid parameter

- **CodePage** Code page

CodePage	Set Value	Description
CodePagePC437		PC437
CodePageKatakana		Katakana
CodePagePC850		PC850
CodePagePC860		PC860
CodePagePC863		PC863
CodePagePC865		PC865

CodePage851_Greek	851[Greek]
CodePagePC857	PC857
CodePage737_Greek	737[Greek]
CodePage928_Greek	928[Greek]
CodePageWPC125	WPC1252
CodePagePC866	PC866
CodePagePC852	PC852
CodePagePC858	PC858
CodePageThaiTis42_Thai3	Thai Tis42(Thai3)
CodePageThaiTis11_Thai5	Thai Tis11(Thai5)
CodePageThaiTis_Thai2	Thai Tis(Thai2)
CodePageThaiKu_Thai1	Thai Ku(Thai1)
CodePageThaiTis14_Thai4	Thai Tis14(Thai4)
CodePageThaiTis18_Thai6	Thai Tis18(Thai6)
CodePageHebrew1	Hebrew1
CodePageHebrew2	Hebrew2
CodePageHebrew3	Hebrew3
CodePage775_Baltic	775[Baltic]
CodePage855_Cyrillic	855[Cyrillic]
CodePage862_hebrew	862[hebrew]
CodePage864_Arabic	864[Arabic]
CodePage869_Greek	869[Greek]
CodePageFras	Fras
CodePage772_Lithuanian	772[Lithuanian]
CodePage1250_Latin_2	1250[Latin-2]
CodePage1251_Cyrillic	1251[Cyrillic]
CodePage1253_Greek	1253[Greek]
CodePage1254_Turkish	1254[Turkish]
CodePage1255_Hebrew	1255[Hebrew]
CodePage1256_Arabic	1256[Arabic]
CodePage1257_Baltic	1257[Baltic]
CodePage771	771
CodePage774_Lithuanian	774[Lithuanian]
CodePage3840_IBM_Russian	3840 (IBM-Russian)

CodePage3841_Gost	3841 (Gost)
CodePage3843_Polish	3843 (Polish)
CodePage3844_CS2	3844 (CS2)
CodePage3845_Hungarian	3845 (Hungarian)
CodePage3846_Turkish	3846 (Turkish)
CodePage3847_Brazil_ABNT	3847 (Brazil-ABNT)
CodePage3848_Brazil_ABICOMP	3848 (Brazil-ABICOMP)
CodePage1001	1001
CodePage2001	2001
CodePage3001_Estonian_1	3001 (Estonian-1)
CodePage3002_Estonian_2	3002 (Estonian-2)
CodePage3011_Latvian_1	3011 (Latvian-1)
CodePage3012_Latvian_2	3012 (Latvian-2)
CodePage3021_Bulgarian	3021 (Bulgarian)
CodePage3041_Maitese	3041 (Maltese)
CodePage8859	8859
CodePagePersia	Persia

Remarks

- a) Several kinds of printers may not support all code page types.

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_TEXT_SELECT_CHAR_SET	The char set selecting failed.
ERR_TEXT_SELECT_CODE_PAGE	The code page selecting failed.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.

Example

```

UInt8 textdata[MAX_COMMAND_LENGTH] = {0x80};
[self.pos_sdk textSelectCharSetAndCodePage: CharacterSetUSA codePage: CodePagePC437];
[self.pos_sdk textPrint:@ "$@{}[]~"];
[self.pos_sdk textPrint: textdata Length:1];
error_code = [self.pos_sdk systemFeedLine:1];

```

● textSetLineHeight

Set line height.

Method

- (SInt32)**textSetLineHeight**:(SInt32)Height

Parameter

- **Height** Line height

Height Set Value	Description
0-255	Legal value.
Other values	Invalid parameter.

Remarks

a) If the parameter Height is 0, the printer's the height of one row will be set to default.

b) If the parameter Height is less than the height of the character, the printer will set its the height of one row to the height of the character.

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_TEXT_SET_LINE_HEIGHT	The line height setting failed.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.

Example

```
[self.pos_sdk textSetLineHeight:34];
```

- **textSetCharacterSpace**

Set character spacing.

Method

- (SInt32)**textSetCharacterSpace**:(SInt32)LeftSpace

RightSpace:(SInt32)RightSpace Mode:(SInt32)Mode

Parameter

- **LeftSpace** The character spacing for the right side of the character to inch
(When mode = ChineseCharacterMode, LeftSpace must be legal.)

LeftSpaceSet Value	Description
0-255	Legal value.

Other values	Invalid parameter.
--------------	--------------------

- **RightSpace** The character spacing for the right side of the character to inch.

RightSpaceSet Value	Description
0-255	Legal value.
Other values	Invalid parameter.

- **Mode** Character mode

ModeSet Value	Description
ChineseCharacterMode	Chinese character mode.
EnglishCharacterMode	English character mode.
Other values	Invalid parameter.

Remarks

a) If the parameter Mode is specified as ChineseCharacterMode, both the parameter LeftSpace and RightSpace must be legal. If the parameter Mode is specified as EnglishCharacterMode, it's OK as long as the parameter RightSpace legal and LeftSpace will be ignored.

b) Only ChineseCharacterMode or EnglishCharacterMode can individually be changed by calling this method for one time. If you want to change both ChineseCharacterMode and EnglishCharacterMode, you should call this method for two times.

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_TEXT_SET_CHARACTER_SPACE	The character space setting failed.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.

Example

```
[self.pos_sdk textSetCharacterSpace:10 RightSpace:50 Mode:EnglishCharacterMode];
```

- **textStandardModeAlignment**

Align all the data in one line to the specified position in standard mode.

Method

- (SInt32)textStandardModeAlignment:(SInt32)Alignment

Parameter

- **Alignment** Alignment mode of text

Alignment	Set Value	Description
TextAlignmentLeft		Left justification.
TextAlignmentCenter		Centering.
TextAlignmentRight		Right justification.
Other values		Invalid parameter.

Remarks

a) This method is enabled only when processed at the beginning of the line in the standard mode.

b) This API setting also applies to barcodes/2D-Code.

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_TEXT_STANDARD_MODE_ALIGNMENT	The alignment mode selecting failed.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.

Example

```
[self.pos_sdk textStandardModeAlignment: TextAlignmentLeft];
```

- **textStandardModeUpsideDown**

Turn on/off upside-down printing mode.

Method

- (SInt32)**textStandardModeUpsideDown**:(SInt32)UpsideDown

Parameter

- **UpsideDown** Upside-down printing mode is turned off or on.

UpsideDown	Set Value	Description
FontStyleUpsideDown		Upside-down.
Other values		Normal.

Remarks

a) Calling textStandardModeUpsideDown: FontStyleUpsideDown has the same print result as calling textStandardModeRotate: RotatePrint180.

b) When you calling textStandardModeUpsideDown: FontStyleUpsideDown first and then calling textStandardModeRotate: RotatePrint180,the result is as same as solely calling textStandardModeRotate: RotatePrint180. Reverse the call sequence, the result is unpredictable.

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_TEXT_STANDARD_MODE_UPSIDEDOWN	Upside-down failed.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.

Example

```
[self.pos_sdk textStandardModeUpsideDown: FontStyleUpsideDown];
```

- **textPrint: String**

Print text (NSString*).

Method

- (NSInteger)**textPrint:(NSString*)String**

Parameter

- **String** The content will be printed.

Remarks

a) If you want to call this function to print other language except for English, such as Chinese, firstly you should call the systemSetEncoding method to set encoding for characters.

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_TEXT_PRINT	The text printing failed.
ERR_INVALID_ARGUMENT	The string is null.

Example

```
[self.pos_sdk textPrint:@ "123456SNBC"];
```

- **textPrint: Buffer**

Print text (UInt8*).

Method

- (SInt32)**textPrint**:(UInt8*)Buffer Length:(SInt32)Length

Parameter

- **Buffer** The unsigned char data will be printed
- **Length** The length of data

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_TEXT_PRINT	The text printing failed.
ERR_INVALID_ARGUMENT	Buffer is null or Length<=0.

Example

```
UInt8 textdata[MAX_COMMAND_LENGTH] = {0x80};
[self.pos_sdk textPrint: textdata Length:1];
```

- **textSelectFontMagnifyTimes**

Select character size.

Method

- (SInt32)**textSelectFontMagnifyTimes**:(SInt32)HorizontalTimes

VerticalTimes:(SInt32)VerticalTimes

Parameter

- **HorizontalTimes** Horizontal times

HorizontalTimes Set Value	Description
1-6	The legal value of HorizontalTimes.
Other values	Invalid Parameter.

- **VerticalTimes** Vertical times

VerticalTimes Set Value	Description
1-6	The legal value of VerticalTimes.
Other values	Invalid Parameter.

Remarks

a) In standard mode, the vertical direction is the paper feed direction, and the horizontal direction is perpendicular to the paper feed direction. However, when character orientation changes in 90 ° clockwise-rotation, the relationship between vertical and horizontal directions is reversed.

b) In page mode, vertical and horizontal directions are based on the direction of print area.

c) When characters in one line are enlarged to different sizes, all the characters are aligned at the baseline.

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_TEXT_SELECT_MAGNIFY_TIMES	Magnify times selecting was failed.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.

Example

```
[self.pos_sdk textSelectFontMagnifyTimes:2 VerticalTimes:3];
```

- **textStandardModeRotate**

Rotate integer times 90 degree.

Method

- (SInt32)**textStandardModeRotate**:(SInt32)Rotate

Parameter

- **Rotate** The degree of rotation

Rotate Set Value	Description
RotatePrintNormal	Nomal.
RotatePrintR90	Turn 90 ° clockwise rotation.
RotatePrint180	180 ° rotation.
RotatePrintL90	Turn 90 ° anticlockwise rotation.
Other values	Invalid parameter.

Return

Return Value	Description
SUCCESS	Processing was successful.

ERR_TEXT_STANDARD_MODE_ROTATE	Roration setting failed.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.

Example

```
[self.pos_sdk textStandardModeRotate: RotatePrintR90];// Turn 90 °clockwise rotation
```

- **textSelectFont**

Select character font and font style.

Method

- (SInt32)**textSelectFont**:(SInt32)FontType FontStyle:(SInt32)FontStyle

Parameter

- **FontType** Font type

FontType	Set Value	Description
FontTypeStandardASCII		Standard ASCII
FontTypeCompressedASCII		Compressed ASCII
FontTypeUserDefined		User defined character
FontTypeChinese		Chinese character
Other values		Invalid parameter

- **FontStyle** Font style

FontStyle	Set Value	Description
FontStyleReverse		Reverse
FontStyleBold		Bold
FontStyleUpsideDown		UpsideDown
FontStyleUnderlineOneDotThick		One dot thick underline
FontStyleUnderlineTwoDotThick		Two dot thick underline
Other values		Invalid parameter

Remarks

- The printer cannot underline when white/black reverse mode is enable.
- The printer cannot underline characters which were clockwise rotated 90 or 270 degree.
- The print result may be the same for several kinds of printers printing Chinese characters when the parameter FontStyle is specified
FontStyleUnderlineOneDotThick and FontStyleUnderlineTwoDotThick.

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_TEXT_SELECT_FONT_TYPE	The font type selecting failed.
ERR_TEXT_SET_FONT_STYLE_REVERSE	Reverse failed.
ERR_TEXT_SET_FONT_STYLE_BOLD	Bold failed.
ERR_TEXT_SET_FONT_STYLE_UNDERLINE	Underline failed.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.

Example

```
[self.pos_sdk textSelectFont: FontTypeStandardASCII FontStyle: FontStyleBold];
```

- **textSetColorPrint**

Enter bi-colour print mode and set print color.

Method

- (SInt32)**textSetColorPrint**

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_TEXT_ENTER_QUIT_COLOR_PRINT	Failed to enter bi-color printing.
ERR_TEXT_SET_COLOR_PRINT	Color setting failed.

Example

```
[self.pos_sdk textSetColorPrint];
```

- **textQuitColorPrint**

Quit bi-color print mode.

Method

- (SInt32)**textQuitColorPrint**

Return

Return Value	Description
--------------	-------------

SUCCESS	Processing was successful.
ERR_TEXT_ENTER_QUIT_COLOR_PRINT	Failed to quitting color print.

Example

```
[self.pos_sdk textQuitColorPrint];
```

- **textUserDefinedCharacterEnable**

User-defined character is enable/disable.

Method

- (SInt32)**textUserDefinedCharacterEnable**:(SInt32)Enable

Parameter

- **Enable** Font user-defined enable or not

Enable Set Value	Description
FontUserDefinedDisable	Font user-defined disable.
FontUserDefinedEnable	Font user-defined Enable.
Other values	Invalid parameter.

Remarks

- a) All user-defined characters can be turn on/off.

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_TEXT_FONT_USER_DEFINED_ENABLE	User-defined enable failed.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.

Example

See also: [Print user defined character](#).

- **textUserDefinedCharacterDefine**

Define user-defined character.

Method

- (SInt32)**textUserDefinedCharacterDefine**:(SInt32)BytesOfHeight

DotsOfWidth:(SInt32)DotsOfWidth StartingCode:(SInt32)StartingCode
EndingCode:(SInt32)EndingCode CharacterData:(NSData*)CharacterData

Parameter

- **BytesOfHeight** Bytes of height
- **DotsOfWidth** Dots of width
- **StartingCode** Starting char code
- **EndingCode** Ending char code
- **CharacterData** The data of user defined character downloaded

Legal value of every parameter:

Parameter	Legal Value
BytesOfHeight	3
DotsOfWidth	9 or 12
StartingCode	32-127
EndingCode	32-127

Remarks

- a) The parameter BytesOfHeight must be 3.
- b) The image of user defined characters must be 9*17 or 12*24.
- c) The font style of white/black reverse,underline,character space,line height,alignment mode and rotation can affect printing user defined characters.

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_TEXT_FONT_USER_DEFINED	User-defined character defining failed.
ERR_INVALID_ARGUMENT	Invalid parameter or Data does not match the number.

Example

See also: [Print user defined character](#).

• textUserDefinedCharacterCancel

Text cancel font user-defined of CharCode.

Method

- (SInt32)textUserDefinedCharacterCancel:(SInt32)CharCode

Parameter

- **CharCode** The char code of cancel character

CharCode	Set Value	Description
32-127		Legal value.
Other values		Invalid parameter.

Remarks

- a) This method cancels the user defined character which specified by CharCode.

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_TEXT_FONT_USER_DEFINED_CANCEL	User-defined character canceling failed.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.

Example

See also: [Print user defined character.](#)

- **textUserDefinedChineseCharacterDefine**

Define user-defined font for Chinese character.

Method

- (SInt32)**textUserDefinedChineseCharacterDefine**:(SInt32)BytesOfHeight
DotsOfWidth:(SInt32)DotsOfWidth Code:(SInt32)Code
CharacterData:(NSData*)CharacterData

Parameter

- **BytesOfHeight** Bytes of height
- **DotsOfWidth** Dots of width
- **Code** The seconde code of Chinese character
- **CharacterData** The data of user defined Chinese character downloaded

Legal value of every parameter:

Parameter	Legal Value
BytesOfHeight	3
DotsOfWidth	24

Code	161-254
------	---------

Remarks

- a) The parameter BytesOfHeight must be 3.
- b) The image of user defined Chinese characters must be 24*24.

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_TEXT_FONT_USER_DEFINED	User-defined Chinese character defining failed.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.

Example

```

SInt32 BytesOfHeight, DotsOfWidth, SecondCode = 162;
NSMutableData *CharacterData = nil;
UIImage *image = nil;
NSString *image_name = @"ChineseCharacter.PNG";

///Get (UIImage*) image///
ImageDataRef image_data = {0};
[self.pos_sdk imageFormatConvertToUserDefinedData:image ditheringSupported:FALSE
image_data:&image_data];

///Get BytesOfHeight, DotsOfWidth and CharacterData///
error_code = [self.pos_sdk textUserDefinedChineseCharacterDefine:BytesOfHeight
DotsOfWidth:DotsOfWidth SecondCode:SecondCode CharacterData:CharacterData];

UInt8 buffer[2] = {0};
buffer[0] = 0xfe;
buffer[1] = SecondCode;
error_code = [self.pos_sdk textPrint:buffer length:2];
error_code = [self.pos_sdk systemFeedLine:1];

```

● textStandardModeRasterPrint

Text raster print in standard mode.

Method

- (SInt32)textStandardModeRasterPrint:(NSString*)TextToPrint
FontName:(NSString*)FontName FontSize:(CGFloat)FontSize

LineBreakMode:(UINavigationController)LineBreakMode

PrinterWidth:(NSInteger)PrinterWidth

Parameter

- **TextToPrint** The character which would be printed.
- **FontName** The font style would be chose.
- **FontSize** The font size would be specified. When PrinterWidth is not 0, the FontSize can not be larger than PrinterWidth.
- **LineBreakMode** Line break mode

LineBreakMode	Set Value	Description
UILineBreakModeWordWrap		Word wrap as line brak mode.
UILineBreakModeCharacterWarp		Character wrap as line brak mode.
UILineBreakModeClip		Clip as line brak mode.
UILineBreakModeHeadTruncation		Head Truncation wrap as line brak mode.
UILineBreakModeTailTruncation		Tail Truncation as line brak mode.
UILineBreakModeMiddleTruncation		Middle Truncation as line brak mode.
Other values		Invalid parameter.

- **PrinterWidth** Printer width

PrinterWidth	Set Value	Description
0		The image can not be zoom.
64-2040		The legal value of printer width,and the image be normally zoom according to printer width.
Other values		Invalid parameter.

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_IMAGE_STANDARD_MODE_RASTER_PRINT	The image raster printing failed.
ERR_INVALID_DATA	The data converted into raster image data was null.
ERR_INVALID_ARGUMENT	Invalid parameter was passed or FontSize >PrinterWidth.

Example

See also: [Text raster printing](#).

- **imageStandardModePrint**

Image print (standard mode).

Method

- (SInt32)**imageStandardModePrint**:(SInt32)Mode Image:(UIImage*)Image
StartHorPos:(SInt32)StartHorPos PrinterWidth:(SInt32)PrinterWidth

Parameter

- **Mode** Image print mode
- **Image** The image would be print.
- **StartHorPos** This parameter can not be used, and it fixed as 0.

Mode	Set Value	Description
SingleDensity_8		8-dot single-density
DoubleDensity_8		8-dot double-density
SingleDensity_24		24-dot single-density
DoubleDensity_24		24-dot double-density
Other values		Invalid parameter

- **PrinterWidth** Printer width

PrinterWidth	Set Value	Description
0		The image can not be zoom.
64-65535		The legal value of printer width, and the image be normally zoom according to printer width.
Other values		Invalid parameter.

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_IMAGE_DOWNLOAD_AND_PRINT	Image downloading and printing failed.
ERR_ALLOC_MEMORY	The memory allocating to store image data failed.
ERR_INVALID_ARGUMENT	Invalid parameter was passed or the image data was null.

Example

See also: [image print](#).

- **imageDownloadToPrinterRAM**

Download images to RAM.

Method

- (SInt32)**imageDownloadToPrinterRAM**:(SInt32)ImageID
Image:(UIImage*)Image PrinterWidth:(SInt32)PrinterWidth

Parameter

- **ImageID** The ID of download

ImageID	Set Value	Description
0-7		Legal value.
Other values		Invalid parameter.

- **Image** The image would be download.
- **PrinterWidth** Printer width

PrinterWidth	Set Value	Description
0		The image can not zoom.
64-65535		The legal value of printer width. The image be normally zoom according to printer width.
Other values		Invalid parameter.

Remarks

- You can only download a single image and specify the ImageID.
- The height of the downloaded image is not larger than 2040.
- Downloaded bitmaps will be cleared when turning power off.

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_IMAGE_DOWNLOAD_RAM	RAM images downloading failed.
ERR_INVALID_ARGUMENT	1. Invalid parameter was passed. 2. The image data was null. 3. The height of downloaded image is larger than 2040.
ERR_ALLOC_MEMORY	The memory allocating to store image data failed.
ERR_INVALID_DATA	The image data converted was null.

Example

See also: [RAM/Flash image download and print.](#)

• **imageRAMPrint**

Print images which have been downloaded to RAM.

Method

- (SInt32)**imageRAMPrint**:(SInt32)ImageID Mode:(SInt32)Mode

Parameter

- **ImageID** The ID of image which has been downloaded to RAM.

ImageID	Set Value	Description
0-7		Legal value.
Other values		Invalid parameter.

- **Mode** Print mode

Mode	Set Value	Description
NormalMode		Normal size.
Double_width		Double width.
Double_height		Double height.
Quadruple		Quadruple (both width and height was double).
Other values		Invalid parameter.

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_IMAGE_RAM_PRINT	RAM image printing failed.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.

Example

See also: [RAM/Flash image download and print.](#)

• **imageDownloadToPrinterFlash**

Download images to Flash.

Method

- (SInt32)**imageDownloadToPrinterFlash**:(NSMutableArray*)ImageArray

PrinterWidth:(SInt32)PrinterWidth

Parameter

- **ImageArray** The array of original images which would be downloaded to Flash.
- **PrinterWidth** Printer width

PrinterWidth	Set Value	Description
0		The image can not be zoom.
64-65535		The legal value of printer width, and the image be normally zoom according to printer width.
Other values		Invalid parameter.

Remarks

a) Every downloading will clear the images which were downloaded into Flash for the last time.

b) After processing this method, the printer need time to complete downloading images for this time for the reason that Flash need a period of time to clear the images which were downloaded for the last time . It may take 30 seconds for the updating. Please do not turn power off within this period.

c) For the situation that downloading multi-images was failed, there may be several images among them were downloaded successfully. Print flash image as the standard to images which were downloaded to Flash.

d) The height of the downloaded image is not larger than 2040.

e) Downloaded bitmaps will not be cleared when turning power off.

f) Multi-bitmaps must be compared by "," such as
 "SNBC.bmp,Jpg.jpg,face.PNG".

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_IMAGE_DOWNLOAD_FLASH	Flash images downloading failed.
ERR_INVALID_ARGUMENT	1. The ImageArray was null. 2. The height of downloaded image is larger than 2040. 3. The image data was null.
ERR_ALLOC_MEMORY	The memory allocating to store image data failed.

ERR_INVALID_DATA	The image data converted was null.
------------------	------------------------------------

Example

See also: [RAM/Flash image download and print.](#)

• imageFlashPrint

Print images which have been downloaded to Flash.

Method

- (SInt32)imageFlashPrint:(SInt32)ImageID Mode:(SInt32)Mode

Parameter

- **ImageID** The ID of image which has been downloaded to Flash.

ImageID	Set Value	Description
1-255		Legal value.
Other values		Invalid parameter.

- **Mode** Print mode

Mode	Set Value	Description
NormalMode		Normal size.
Double_width		Double width.
Double_height		Double height.
Quadruple		Quadruple(both width and height was double).
Other values		Invalid parameter.

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_IMAGE_FLASH_PRINT	Flash image printing failed.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.

Example

See also: [RAM/Flash image download and print.](#)

• imageStandardModeRasterPrint

Print raster image in standard mode.

Method

- (SInt32)**imageStandardModeRasterPrint**:(UIImage*)Image

PrinterWidth:(SInt32)PrinterWidth

Parameter

- **Image** The image would be print.
- **PrinterWidth** Printer width

PrinterWidth	Set Value	Description
0		The image can not zoom.
64-2040		The legal value of printer width,and the image be normally zoom according to printer width.
Other values		Invalid parameter.

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_IMAGE_STANDARD_MODE_RASTER_PRINT	The image raster printing failed.
ERR_INVALID_ARGUMENT	The image was null.
ERR_INVALID_DATA	The image data converted was null.

Example

See also: [Raster image print](#).

- **imageFormatConvertToUserDefinedData**

Convert image data format.

Method

- (SInt32)**imageFormatConvertToUserDefinedData**:(UIImage*)Image

ditheringSupported:(BOOL)ditheringSupported

image_data:(ImageDataRef*)image_data

Parameter

- **Image** The image would be converted.
- **ditheringSupported** Dthering is supported or not.
- **image_data** The converted image data which was suitable for user defined character.

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_ALLOC_MEMORY	Allocing space was failed.

Example

```
ImageDataRef image_data = {0};
```

```
[self.pos_sdk imageFormatConvertToUserDefinedData:image ditheringSupported: FLSE  
image_data:&image_data];
```

- **barcodeGetIDByName**

Get barcode ID by comparing name of barcode.

Method

- (SInt32)**barcodeGetIDByName**:(NSString*)BarcodeName

Parameter

- **BarcodeName** Name of barcode

BarcodeName	Set Value	Description
BarcodeUPC-A		The ID of BarcodeUPC_A
BarcodeUPC-E		The ID of BarcodeUPC_E
BarcodeJAN13orEAN13		The ID of BarcodeJAN13orEAN13
BarcodeJAN8orEAN8		The ID of BarcodeJAN8orEAN8
BarcodeCODE39		The ID of BarcodeCODE39
BarcodeITF		The ID of BarcodeITF
BarcodeCODABAR		The ID of BarcodeCODABAR
BarcodeCODE93		The ID of BarcodeCODE93
BarcodeCODE128		The ID of BarcodeCODE128
BarcodePDF417		The ID of BarcodePDF417
BarcodeQR		The ID of BarcodeQR
BarcodeMaxicode		The ID of BarcodeMaxicode
BarcodeGS1		The ID of BarcodeGS1
Other values		Invalid parameter

Return

Return Value	Description
--------------	-------------

The ID of the specified barcode	Processing was successful.
-1	Processing was failed.

Example

```
NSString *BarcodeName    = nil;
barcode_id  = [self.pos_sdk barcodeGetIDByName:BarcodeName];
```

- **barcodeGetNameByID**

Get name by barcode ID.

Method

- (NSString*)**barcodeGetNameByID**:(SInt32)BarcodeID

Parameter

- **BarcodeID** The barcode ID

BarcodeID	Set Value	Description
0-8		Legal value.
Other values		Invalid parameter.

Return

Return Value	Description
Barcode name of the specified barcode ID	Processing was successful.
nil	Processing was failed.

Example

```
[self.pos_sdk systemSetPortIO:self.printer_port];
```

- **barcodePrint1Dimension**

Print 1Dimension barcode.

Method

- (SInt32)**barcodePrint1Dimension**:(NSData*)Data BarcodeType:(SInt32)Type
ModuleWidth:(SInt32)ModuleWidth BarcodeHeight:(SInt32)Height
HriFontType:(SInt32)HriFontType HriPosition:(SInt32)HriPosition

Parameter

- **Data** Barcode data
- **BarcodeType** Barcode type

BarcodeType	Set Value	Description	Data length Set Value
BarcodeUPC_A		UPC-A	11 -12
BarcodeUPC_E		UPC-E	11-12
BarcodeJAN13orEAN13		EAN13	12-13
BarcodeJAN8orEAN8		EAN-8	7-8
BarcodeCODE39		Code39	1-255
BarcodeITF		Interleaved 2 of 5	1-255
BarcodeCODABAR		CodaBar	1-255
BarcodeCODE93		Code93	1-255
BarcodeCODE128		Code128	2-255
Other values		Invalid parameter	

- **ModuleWidth** Barcode module width

ModuleWidth	Set Value	Description
2-6		Legal value.
Other values		Invalid parameter.

- **BarcodeHeight** Barcode height

BarcodeHeight	Set Value	Description
1-255		Legal value.
Other values		Invalid parameter.

- **HriFontType** Hri font type

HriFontType	Set Value	Description
FontTypeStandardASCII		Standard ASCII
FontTypeCompressedASCII		Compressed ASCII
Other values		Invalid parameter

- **HriPosition** The position of Hri font

HriPosition	Set Value	Description
HRINone		HRI can not be printed.
HRIAbove		Above the barcode.
HRIBelow		Below the barcode.
HRIAboveAndBelow		Both above and below the barcode.
Other values		Invalid parameter.

Remarks

- a) See also: [Appendix C. Barcode](#) and [Appendix D. Code 128](#)

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_BARCODE_PRINT_1D	Barcode type selecting failed.
ERR_BARCODE_SELECT_MODULE_WIDTH	Module width selecting failed.
ERR_BARCODE_SELECT_BARCODE_HEIGHT	Barcode height selecting failed.
ERR_BARCODE_SELECT_HRI_FONT_TYPE	Hri font type selecting failed.
ERR_BARCODE_SELECT_HRI_FONT_POSITION	Hri position selecting failed.
ERR_BARCODE_1D_SEND_DATA	Data of barcode 1D sending failed.
ERR_INVALID_ARGUMENT	Invalid parameter.

Example

```

NSData *data_1D = nil;
///Get (NSData*)data_1D for barcode 1D///
[self.pos_sdk barcodePrint1Dimension:data_1D BarcodeType: BarcodeUPC_A ModuleWidth:3
BarcodeHeight:100 HriFontType: FontTypeStandardASCII HriPosition: HRIBelow];

```

- **barcodePrintQR**

Set parameter and print barcode QR.

Method

- (SInt32)**barcodePrintQR**:(NSData*)Data

BasicElementWidth:(SInt32)BasicElementWidth SymbolType:(SInt32)SymbolType

LanguageMode:(SInt32)LanguageMode

Parameter

- **Data** BarcodeQR data
- **BasicElementWidth** Basic element width

BasicElementWidth	Set Value	Description
1-255		Legal value.
Other values		Invalid parameter.

-
- **LanguageMode** Language mode

LanguageMode	Set Value	Description
LanguageChinese		Chinese
LanguageJapanese		Japanese
Other values		Invalid parameter

- **SymbolType** Symbole type

SymbolType	Set Value	Description
OriginalType		Original type.
EnhancedType		Enhance type (This type is suggested to be used).
Other values		Invalid parameter.

Remarks

a) When the barcode outside the print area by setting barcode data and basic element width, the printer can not barcode.

b) Recommended to use EnhancedType.

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_BARCODE_QR_SET_PARAM	The parameters of QR setting failed.
ERR_BARCODE_PRINT_2D	Barcode 2D type selecting to print failed.
ERR_BARCODE_QR_SEND_DATA	Barcode data of QR sending failed.
ERR_INVALID_ARGUMENT	Invalid parameter.

Example

See also: [Print QR](#).

- **barcodePrintPDF417**

Set barcodePDF417 size and print PDF417.

Method

- (SInt32)**barcodePrintPDF417**:(NSData*)Data
 AppearanceToHeight:(SInt32)AppearanceToHeight
 AppearanceToWidth:(SInt32)AppearanceToWidth RowNumber:(SInt32)RowNumber
 ColumnNumber:(SInt32)ColumnNumber XSize:(SInt32)XSize
 LineHeight:(SInt32)LineHeight CorrectionGrade:(SInt32)CorrectionGrade

Parameter

- **Data** Barcode PDF417 data
- **AppearanceToHeight** Appearance to height
- **AppearanceToWidth** Appearance to width
- **RowNumber** The number of rows
- **ColumnNumber** The number of columns
- **XSize** XSize
- **LineHeight** Line height
- **CorrectionGrade** Correction grade

The legal values of every parameter, as follow:

Parameter	Legal value
AppearanceToHeight	1-10
AppearanceToWidth	1-100
RowNumber	3-90
ColumnNumber	1-30
XSize	1-7
LineHeight	2-25
CorrectionGrade	0-8

Remarks

- a) When the barcode data outside range, the barcode can not be printed.
- b) When the barcode size outside the print area, the printer can not barcode.

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_BARCODE_PDF417_SET_SIZE	PDF417 size setting failed.
ERR_BARCODE_PDF417_SELECT_CORRECTION_GRADE	PDF417 correction grade selecting failed.
ERR_BARCODE_PRINT_2D	Barcode 2D type selecting to print failed.
ERR_BARCODE_PDF417_SEND_DATA	Barcode data of PDF417 sending failed.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.

Example

See also: [Print PDF417](#).

- **barcodePrintMaxicode**

Print barcode Maxicode.

Method

- (SInt32)**barcodePrintMaxicode**:(NSData*)Data

Parameter

- **Data** Maxicode data

Remarks

a) When the barcode data outside range, the barcode can not be printed.

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_BARCODE_PRINT_2D	Barcode 2D type selecting to print failed.
ERR_BARCODE_MAXICODE_SEND_DATA	Barcode data of Maxicode sending failed.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.

Example

See also: [Print Maxicode](#).

- **barcodePrintGS1DataBar**

Set GS1 DataBar and GS1 composite barcode parameter and print barcode.

Method

- (SInt32)**barcodePrintGS1DataBar**:(NSData*)Data

BarcodeType:(SInt32)BarcodeType BasicElementWidth:(SInt32)BasicElementWidth

BarcodeHeight:(SInt32)BarcodeHeight

BasicElementHeight:(SInt32)BasicElementHeight

SeparatorHeight:(SInt32)SeparatorHeight SegmentNumber:(SInt32)SegmentNumber

HRI:(SInt32)HRI UseAI:(SInt32)AI

Parameter

- **Data** Barcode data
- **BarcodeType** Barcode type

BarcodeType	Set Value	Description
GS1DataBarOmnidirectional		GS1DataBar Omnidirectional
GS1DataBarTruncated		GS1DataBar Truncated
GS1DataBarStacked		GS1 DataBar Stacked
GS1DataBarStackedOmnidirectional		GS1 DataBar Stacked Omnidirectiona
GS1DataBarLimited		GS1 DataBar Limited
GS1DataBarExpanded		GS1 DataBar Expanded
GS1DataBarExpandedStacked		GS1 DataBar ExpandedStacked
Other values		Invalid parameter

- **BasicElementWidth** Basic element width

BasicElementWidth	Set Value	Description
1-6		Legal value.
Other values		Invalid parameter.

- **BarcodeHeight** The height of the DataBar, Stacked, stacked omnidirectional, expanded stacked barcode indicate the height of each line of barcode.

BarcodeHeight	Set Value	Description
2-250		Legal value.
Other values		Invalid parameter.

- **BasicElementHeight** The basic element height of the 2D barcode in the composite barcode

BasicElementHeight	Set Value	Description
1-10		Legal value.
Other values		Invalid parameter.

- **SeparatorHeight** The height of the separator. This parameter should be set in DataBar composite barcode or separate stacked, stacked omnidirectional, expanded stacked barcodes.

SeparatorHeight	Set Value	Description
1-10		Legal value.
Other values		Invalid parameter.

- **SegmentNumber** The number of segments of each line of barcode. Only in expanded stacked barcode should this parameter be set.

SegmentNumber	Set Value	Description
2-20		The legal value of separate expanded stacked barcodes.
4-20		The legal value of composite expanded stacked barcodes.
Other values		Invalid parameter.

- **HRI** The content of the note character

HRI	Set Value	Description
DataBarAnd2DHri		DataBar and 2D in composite barcode. DataBar only in separate barcode.
Only DataBarHri		Print DataBar in composite or separate barcode.
Only2DHri		Print 2D in composite barcode, no print in separate barcode.
NoHri		No note character.
Other values		Invalid parameter.

- **AI** whether to use AI (use identifier): 0 indicates to not use AI; 1 indicates to use AI.

Remarks

- Several kinds of printers may not support all GS1 barcode type.
- When the barcode data outside range, the barcode can not be printed.
- When the barcode size outside the print area, the printer can not barcode.

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_BARCODE_GS1DATABAR_SET_PARAM	Parameters of GS1 setting failed.
ERR_BARCODE_PRINT_2D	Barcode 2D type selecting to print failed.
ERR_BARCODE_GS1DATABAR_SEND_DATA	Barcode data of GS1 sending failed.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.

Example

See also: [Print GS1 DataBar and GS1 composite barcode.](#)

- **standardModeSetLeftMarginAndPrintAreaWidth**

Set left margin and width in standard mode.

Method

-(SInt32)standardModeSetLeftMarginAndPrintAreaWidth:(SInt32)LeftMargin Width:(SInt32)Width

Parameter

- **LeftMargin** Left margin

LeftMargin	Set Value	Description
0-65535		Legal value.
Other values		Invalid parameter.

- **Width** Print area width

Width	Set Value	Description
0-65535		Legal value.
Other values		Invalid parameter.

Remarks

a) This method can not affect printing user defined characters, the method of imageStandardModeRasterPrint and textStandardModeRasterPrint.

b) For page mode, this method is invalid.

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_STANDARD_MODE_SET_PRINT_AREA_WIDTH	Print area width for standard mode setting failed.
ERR_STANDARD_MODE_SET_LEFT_MARGIN	Left margin for standard mode setting failed.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.

Example

```
[self.pos_sdk standardModeSetLeftMarginAndPrintAreaWidth:100 Width:500];
```

- **standardModeSetHorizontalStartingPosition**

Horizontal Starting Position in standard mode.

Method

- (SInt32)**standardModeSetStartingPosition**:(SInt32)X

Parameter

- **X** Horizontal starting position for Standard mode

Distance Set Value	Description
0-65535	Legal value.
Other values	Invalid parameter.

Remarks

a) This method can not affect user defined characters printing, image raster printing and text raster printing.

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_STANDARD_MODE_SET_HORIZONTAL_STARTING_POSITION	Horizontal starting position for standard mode setting failed.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.

Example

```
[self.pos_sdk standardModeSetStartingPosition:100];
```

- **pageModeSetStartingPosition**

Set horizontal and vertical starting position in page mode.

Method

- (SInt32)**pageModeSetStartingPosition**:(SInt32)X Y:(SInt32)Y

Parameter

- **X** Horizontal starting position

X Set Value	Description
0-65535	Legal value.
Other values	Invalid parameter.

- **Y** Vertical starting position

Y Set Value	Description
0-65535	Legal value.

Other values	Invalid parameter.
--------------	--------------------

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_STANDARD_MODE_SET_HORIZONTAL_STARTING_POSITION	Horizontal starting position for standard mode setting failed.
ERR_PAGE_MODE_SET_VERTICAL_STARTING_POSITION	Vertical starting position for page mode setting failed.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.

Example

```
[self.pos_sdk systemSetPortIO:self.printer_port];
```

- **pageModeSetPrintArea**

Set print area in page mode.

Method

- (SInt32)**pageModeSetPrintArea**:(SInt32)X Y:(SInt32)Y
 AreaWidth:(SInt32)AreaWidth AreaHeight:(SInt32)AreaHeight
 PrintDirection:(SInt32)Direction

Parameter

- **X** Horizontal starting position
- **Y** Vertical starting position
- **AreaWidth** Area width
- **AreaHeight** Area height

X/Y/AreaWidth/AreaHeight Set Value	Description
0-65535	Legal value.
Other values	Invalid parameter.

- **Direction** Print direction

Direction Set Value	Description
LeftToRight	Left to right.
BottomToTop	Bottom to top.
RightToLeft	Right to left.
TopToBottom	Top to bottom.

Other values	Invalid parameter.
--------------	--------------------

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_PAGE_MODE_SET_PRINT_AREA	Print area for page mode setting failed.
ERR_PAGE_MODE_SET_PRINT_DIRECTION	Print direction for page mode setting failed.
ERR_INVALID_ARGUMENT	An invalid parameter was passed.

Example

```
[self.pos_sdk pageModeSetPrintArea:100 Y:0 AreaWidth:400 AreaHeight:1000
PrintDirection:LeftToRight];
```

- **pageModePrint**

Print data in page mode.

Method

- (SInt32)**pageModePrint**

Return

Return Value	Description
SUCCESS	Processing was successful.
ERR_PAGE_MODE_PRINT	page mode printing failed.

Example

```
[self.pos_sdk pageModePrint];
```

- **pageModeClearBuffer**

In page mode, delete all the print data in current area.

Method

- (SInt32)**pageModeClearBuffer**

Return

Return Value	Description
--------------	-------------

SUCCESS	Processing was successful.
ERR_PAGE_MODE_CLEAR_BUFFER	Page mode clearing buffer failed.

Example

```
[self.pos_sdk pageModeClearBuffer];
```

5. Appendix

Appendix A. How to use the WIFIPortToFile class

WIFIPort can realize the function that saving data to the file while sending data to the printer successfully. That is to say, the successful opening of port is the prerequisite of saving data to the file. Sometimes the customer may not need to send data to the printer successfully and only needs to save data to the file. For this condition, we provide the WIFIPortToFile.

- **Add WIFIPortToFile.h to Project**

How to add WIFIPortToFile.h to project, please refer to [How to add library files](#).

- **Using WIFIPortToFile.h**

1) Import WIFIPortToFile.h:

```
#import "WIFIPortToFile.h"
```

2) Declare the class:

```
@property (nonatomic, retain) WIFIPortToFile *printer_port_file;
```

3) Define the class:

```
@synthesize printer_port_file = _printer_port_file;
- (WIFIPortToFile *) printer_port_file
{
    if(_printer_port_file == nil)
    {
        _printer_port_file = [[WIFIPortToFile alloc] init];
    }
    return _printer_port_file;
}
```

- **Methods in the WIFIPortToFile class**

All methods in the WIFIPort class are retained in the WIFIPortToFile class. Among them, when calling searchPort, it returns nil directly. When calling openPort, closePort or readPort, SUCCESS will be returned directly. The WIFIPortToFile class sends data by the method writePort and records communication data by the method recordCommunicationDataEnable. The sample program of recording communication data, as follow:

```
[self. printer_port_file recordCommunicationDataEnable: @ "DataFile.dat"];
```

The sample program of sending data, as follow:

```
data[2] = {0x1b,0x40};
[self. printer_port_file writePort:data offSize:0 WriteSize:2 WriteTimeOut:
TIME_OUT_SMALL_DATA];
```

Appendix B. List of Error Code

Error Code	Description
SUCCESS	Processing was successful.
ERR_INVALID_ARGUMENT	Invalid parameter was passed.
ERR_INVALID_DATA	The data converted into raster image data was null
ERR_INVALID_CONNECTION	Failed to connect to printer
ERR_CREATE_CONNECTION	Failed to create socket. There are not source which can be used in TimeOut. Failed to get socket.
ERR_COMMUNICATE	1. Failed to set TimeOut 2. The bytes of return are different from Bytes of datas which will be read
ERR_ALLOC_MEMORY	Allocating memory was failed
ERR_SYSTEM_RESET	Printer resetting failed.
ERR_SYSTEM_SELECT_PRINT_MODE	The print mode selecting failed.
ERR_SYSTEM_SELECT_PAPER_TYPE	The paper type selecting failed.
ERR_SYSTEM_SET_MOTION_UNIT	The motion unit selecting failed.
ERR_SYSTEM_QUERY_STATUS	The printer's status querying failed.
ERR_SYSTEM_FEED_LINE	Feed line failed.
ERR_SYSTEM_CUT_PAPER	The paper cutting failed.
ERR_CASH_DRAWER_OPEN	The cashdrawer opening failed.
ERR_TEXT_SELECT_CHAR_SET	The char set selecting failed.
ERR_TEXT_SELECT_CODE_PAGE	The code page selecting failed.
ERR_TEXT_SET_LINE_HEIGHT	The line height setting failed.
ERR_TEXT_SET_CHARACTER_SPACE	The character space setting failed.
ERR_TEXT_STANDARD_MODE_ALIGNM ENT	The alignment mode selecting failed.
ERR_TEXT_SELECT_FONT_TYPE	The font type selecting failed.
ERR_TEXT_SET_FONT_STYLE_REVERSE	Reverse failed.
ERR_TEXT_SET_FONT_STYLE_BOLD	Bold failed.
ERR_TEXT_SET_FONT_STYLE_UNDERLI NE	Underline failed.

ERR_TEXT_STANDARD_MODE_UPSIDEDOWN	Upside-down failed.
ERR_TEXT_SELECT_MAGNIFY_TIMES	Magnify times selecting was failed.
ERR_TEXT_STANDARD_MODE_ROTATE	Roration setting failed.
ERR_TEXT_ENTER_QUIT_COLOR_PRINT	Failed to enter /cancel bi-color printing.
ERR_TEXT_SET_COLOR_PRINT	Color setting failed.
ERR_TEXT_FONT_USER_DEFINED_ENABLE	User-defined enable failed.
ERR_TEXT_FONT_USER_DEFINED	User-defined character defining failed.
ERR_TEXT_FONT_USER_DEFINED_CANCEL	User-defined character canceling failed.
ERR_TEXT_PRINT	The text printing failed.
ERR_IMAGE_DOWNLOAD_AND_PRINT	Image downloading and printing failed.
ERR_IMAGE_DOWNLOAD_RAM	RAM images downloading failed.
ERR_IMAGE_RAM_PRINT	RAM image printing failed.
ERR_IMAGE_DOWNLOAD_FLASH	Flash images downloading failed.
ERR_IMAGE_FLASH_PRINT	Flash image printing failed.
ERR_IMAGE_STANDARD_MODE_RASTER_PRINT	The image raster printing failed.
ERR_STANDARD_MODE_SET_PRINTAREA_WIDTH	Print area width for standard mode setting failed
ERR_STANDARD_MODE_SET_LEFT_MARGIN	Left margin for standard mode setting failed
ERR_STANDARD_MODE_SET_HORIZONTAL_STARTING_POSITION	Horizontal starting position for standard mode setting failed.
ERR_PAGE_MODE_SET_VERTICAL_STARTING_POSITION	Vertical starting position for page mode setting failed
ERR_PAGE_MODE_SET_PRINT_AREA	Print area for page mode setting failed
ERR_PAGE_MODE_SET_PRINT_DIRECTION	Print direction for page mode setting failed
ERR_PAGE_MODE_PRINT	Page mode printing failed.

ERR_PAGE_MODE_CLEAR_BUFFER	Page mode clearing buffer failed.
ERR_BARCODE_PRINT_1D	Barcode type selecting failed.
ERR_BARCODE_PRINT_2D	Barcode 2D type selecting to print failed.
ERR_BARCODE_SELECT_MODULE_WIDTH	Module width selecting failed.
ERR_BARCODE_SELECT_BARCODE_HEIGHT	Barcode height selecting failed.
ERR_BARCODE_SELECT_HRI_FONT_TYPE	Hri font type selecting failed.
ERR_BARCODE_SELECT_HRI_FONT_POSITION	Hri position selecting failed.
ERR_BARCODE_QR_SET_PARAM	The parameters of QR setting failed.
ERR_BARCODE_PDF417_SELECT_CORRECTION_GRADE	PDF417 correction grade selecting failed
ERR_BARCODE_PDF417_SET_SIZE	PDF417 size setting failed.
ERR_BARCODE_GS1DATABAR_SET_PARAM	Parameters of GS1 setting failed.
ERR_BARCODE_1D_SEND_DATA	Data of barcode 1D sending failed.
ERR_BARCODE_QR_SEND_DATA	Barcode data of QR sending failed.
ERR_BARCODE_PDF417_SEND_DATA	Barcode data of PDF417 sending failed.
ERR_BARCODE_MAXICODE_SEND_DATA	Barcode data of Maxicode sending failed.
ERR_BARCODE_GS1DATABAR_SEND_DATA	Barcode data of GS1 sending failed.

Appendix C. Barcode

The data length and character set of barcode type,as follows:

Barcode Type	Data	ASCII	Remarks
--------------	------	-------	---------

	length		
UPC-A	11 ~ 12	48 ~ 57	
UPC-E	11 ~ 12	48 ~ 57	The first character must be 0.
JAN13 (EAN13)	12 ~ 13	48 ~ 57	
JAN 8 (EAN8)	7 ~ 8	48 ~ 57	
CODE39	1 ~ 255	45 ~ 57, 65 ~ 90, 32, 36, 37,43	
ITF	1 ~ 255	48 ~ 57	
CODABAR	1 ~ 255	48 ~ 57 65 ~ 68, 36, 43,45,46,47 58	The beginning code and ending code must be one of character A, B, C, D. The ending code can be replaced by T, E, *,N.
CODE93	1 ~ 255	0 ~ 127	
CODE128	2 ~ 255	0 ~ 127	You must specify the character set before barcode data.
PDF417	1 ~ 255	0 ~ 255	
QR CODE	4 ~ 255	0 ~ 255	
MAXICODE	1 ~ 138	48 ~ 57,65 ~ 90	
GS1	1 ~ 255	It depends on GS1 barcode type, see also: the following Table.	

The data length and character set of GS1, as follows:

Parameter	Barcode type	Character set	Data length
1	GS1DataBar Omnidirectional	Number 0-9	14bits, 13numbers+1bits of check characters
2	GS1DataBar Truncated	Number 0-9	14bits, 13numbers+1bits of check characters

3	GS1 DataBar Stacked	Number 0-9	14bits, 13numbers+1bits of check characters
4	GS1 DataBar Stacked Omnidirectional	Number 0-9	14bits, 13numbers+1bits of check characters
5	GS1 DataBar Limited	Number 0-9	14bits, 13numbers+1bits of check characters
6	GS1 DataBar Expanded	0 ~ 9,A ~ Z, a ~ z ! " % & ' () * + , - . / : ; < = > ? _ space FNC1	Max 74numbers or 41 letters
7	GS1 DataBar ExpandedStacked	0 ~ 9,A ~ Z, a ~ z ! " % & ' () * + , - . / : ; < = > ? _ space FNC1	Max 74numbers or 41 letters

[Notes]

When UPC-A, UPC-E, JAN13 (EAN13) or JAN8 (EAN8) is selected, if n is outside the specified range, this command is invalid

[Notes (standard mode)]

- If data is outside the specified range, the barcode can not be printed.
- If the horizontal size of the barcode exceeds printing area, the barcode can not be printed.
- This command feeds as much paper as is required to print the barcode, regardless of the line height specified by - (SInt32)textSetLineHeight:.
- It is enabled only when no data exists in the print buffer. When data exists in the print buffer, the command is ignored.
- After printing barcode, this command sets the print position to the beginning of the line.
- This command is not affected by - (SInt32)textSelectFont: (FontStyle as FontStyleReverse/FontStyleBold/FontStyleUnderlineOneDotThick/FontStyleUnderlineTwoDotThick, etc.), except for FontStyleUpsideDown.

[Notes in page mode]

- This command develops bar code data in the print buffer, but does not print it. After processing barcode data, this command moves the print position to the right side

dot of the barcode.

- If d is out of the specified range, this command is ignored.
- If barcode width exceeds the printing area, this command is ignored.

When CODE128 (m = 73) is used:

• See also: Appendix A for the information of the CODE 128 barcode and the character set.

• When using the CODE 128 in this printer, take the following points into account for data transmission:

Character set must be selected before the barcode data (one of CODE A, CODE B or CODE C).

Special characters are defined by combining two characters "{" and one character. The ASCII character "{" is defined by transmitting "{" twice consecutively.

Specific character set	Transmit data
SHIFT	{S
CODE A	{A
CODE B	{B
CODE C	{C
FNC1	{1
FNC2	{2
FNC3	{3
FNC4	{4
"{"	{{

Example: print "123456" using CODE B, You can input: {B123456

• If the top of the barcode data is not the code set selection character, the printer stops command processing and processes the following data as normal data.

• If combination of "{" and the following character does not apply to any special character, the printer stops command processing and processes the following data as normal data.

• If the printer receives characters that cannot be used in the special code set, the printer stops command processing and processes the following data as normal data.

• The printer does not print HRI characters that correspond to the shift characters or code set selection characters.

• HRI characters for the function characters are not printed.

• HRI characters for the control characters (<00>H to <1F>H and <7F>H) are not printed.

• The left-side and right –side spacing which varies from one barcode type to

another must be assured.

Appendix D. Code 128

1. Description of the CODE128 Bar Code

In CODE128 bar code system, it is possible to represent 128 ASCII characters, the one hundred numbers from 00 to 99 and some special characters with three code sets: A, B and C. Each code set is used for representing the following characters:

- Code set A: ASCII characters 00H to 5FH
- Code set B: ASCII characters 20H to 7FH
- Code set C: 100 numerals from 00 to 99

The following special characters are also available in CODE128:

- SHIFT characters

In code set A, the character just after SHIFT is processed as a character for code set B. In code set B, the character just after SHIFT is processed as a character for code set A. SHIFT characters cannot be used in code set C.

- Code set selection character (CODE A, CODE B, CODE C).

This character switches the following code set to code set A, B, or C.

- Function character (FNC1, FNC2, FNC3, FNC4)

The usage of function characters depends on the application software. In code set C, only FNC1 is available.

2. Code Tables

Printable characters in code set A

Character	Transmit Data		Character	Transmit Data		Character	Transmit Data	
	Hex	Decimal		Hex	Decimal		Hex	Decimal
NULL	00	0	(	28	40	P	50	80
SOH	01	1	)	29	41	Q	51	81
STX	02	2	*	2A	42	R	52	82
ETX	03	3	+	2B	43	S	53	83
EOT	04	4	,	2C	44	T	54	84
ENQ	05	5	-	2D	45	U	55	85
ACK	06	6	.	2E	46	V	56	86
BEL	07	7	/	2F	47	W	57	87
BS	08	8	0	30	48	X	58	88
HT	09	9	1	31	49	Y	59	89
LF	0A	10	2	32	50	Z	5A	90
VT	0B	11	3	33	51	[	5B	91
FF	0C	12	4	34	52	\	5C	92
CR	0D	13	5	35	53	]	5D	93
SO	0E	14	6	36	54	^	5E	94
SI	0F	15	7	37	55	_	5F	95
DLE	10	16	8	38	56	FNC1	7B,31	123,49
	11	17	9	39	57	FNC2	7B,32	123,50
	12	18	:	3A	58	FNC3	7B,33	123,51
	13	19	;	3B	59	FNC4	7B,34	123,52
	14	20	<	3C	60	SHIFT	7B,53	123,83
	15	21	=	3D	61	CODEB	7B,42	123,66
	16	22	>	3E	62	CODEC	7B,43	123,67
	17	23	?	3F	63			
	18	24	@	40	64			
	19	25	A	41	65			
	1A	26	B	42	66			
	1B	27	C	43	67			
	1C	28	D	44	68			
	1D	29	E	45	69			
	1E	30	F	46	70			
	1F	31	G	47	71			
NAK	20	32	H	48	72			
SYN	21	33	I	49	73			

ETB	22	34	J	4A	74			
CAN	23	35	K	4B	75			
EM	24	36	L	4C	76			
SUB	25	37	M	4D	77			
ESC	26	38	N	4E	78			
FS	27	39	O	4F	79			
GS								
RS								
US								
SP								
!								
"								
#								
\$								
%								
&								
'								

Printable characters in code set B

Character	Transmit Data		Character	Transmit Data		Character	Transmit Data	
	Hex	Decimal		Hex	Decimal		Hex	Decimal
SP	20	32	H	48	72	p	70	112
!	21	33	I	49	73	q	71	113
"	22	34	J	4A	74	r	72	114
#	23	35	K	4B	75	s	73	115
\$	24	36	L	4C	76	t	74	116
%	25	37	M	4D	77	u	75	117
&	26	38	N	4E	78	v	76	118
'	27	39	O	4F	79	w	77	119
(	28	40	P	50	80	x	78	120
)	29	41	Q	51	81	y	79	121
*	2A	42	R	52	82	z	7A	122
+	2B	43	S	53	83	{	7B,7B	123,12
,	2C	44	T	54	84	}	7C	3
-	2D	45	U	55	85	—	7D	124
.	2E	46	V	56	86	DEL	7E	125
/	2F	47	W	57	87	FNC1	7F	126

0	30	48	X	58	88	FNC2	7B,31	127
1	31	49	Y	59	89	FNC3	7B,32	123,49
2	32	50	Z	5A	90	FNC4	7B,33	123,50
3	33	51	[	5B	91	SHIFT	7B,34	123,51
4	34	52	\	5C	92	CODEA	7B,53	123,52
5	35	53	]	5D	93	CODEC	7B,41	123,83
6	36	54	^	5E	94		7B,43	123,65
7	37	55	_	5F	95			123,67
8	38	56	`	60	96			
9	39	57	a	61	97			
:	3A	58	b	62	98			
;	3B	59	c	63	99			
<	3C	60	d	64	100			
=	3D	61	e	65	101			
>	3E	62	f	66	102			
?	3F	63	g	67	103			
@	40	64	H	68	104			
A	41	65	i	69	105			
B	42	66	j	6A	106			
C	43	67	k	6B	107			
D	44	68	l	6C	108			
E	45	69	m	6D	109			
F	46	70	n	6E	110			
G	47	71	o	6F	111			

Printable characters in code set C

Character	Transmit Data		Character	Transmit Data		Character	Transmit Data	
	Hex	Decimal		Hex	Decimal		Hex	Decimal
00	00	0	40	28	40	80	50	80
01	01	1	41	29	41	81	51	81
02	02	2	42	2A	42	82	52	82
03	03	3	43	2B	43	83	53	83
04	04	4	44	2C	44	84	54	84
05	05	5	45	2D	45	85	55	85
06	06	6	46	2E	46	86	56	86
07	07	7	47	2F	47	87	57	87
08	08	8	48	30	48	88	58	88

09	09	9	49	31	49	89	59	89
10	0A	10	50	32	50	90	5A	90
11	0B	11	51	33	51	91	5B	91
12	0C	12	52	34	52	92	5C	92
13	0D	13	53	35	53	93	5D	93
14	0E	14	54	36	54	94	5E	94
15	0F	15	55	37	55	95	5F	95
16	10	16	56	38	56	96	60	96
17	11	17	57	39	57	97	61	97
18	12	18	58	3A	58	98	62	98
19	13	19	59	3B	59	99	63	99
20	14	20	60	3C	60	FNC1	7B,31	123,49
21	15	21	61	3D	61	CODEA	7B,41	123,65
22	16	22	62	3E	62	CODEB	7B,42	123,66
23	17	23	63	3F	63			
24	18	24	64	40	64			
25	19	25	65	41	65			
26	1A	26	66	42	66			
27	1B	27	67	43	67			
28	1C	28	68	44	68			
29	1D	29	69	45	69			
30	1E	30	70	46	70			
31	1F	31	71	47	71			
32	20	32	72	48	72			
33	21	33	73	49	73			
34	22	34	74	4A	74			
35	23	35	75	4B	75			
36	24	36	76	4C	76			
37	25	37	77	4D	77			
38	26	38	78	4E	78			
39	27	39	79	4F	79			

Appendix E. Programming Flow

