

打印机接口说明

打印机接口说明.....	1
一 概述.....	5
二 接口.....	7
CSNIO.....	7
Write.....	7
Read.....	8
IsOpened.....	9
CSNIOCallBack.....	10
OnOpen.....	10
OnOpenFailed.....	11
OnClose.....	12
CSNBTPrinting.....	13
Open.....	13
Listen.....	14
Close.....	15
Write.....	16
Read.....	17
SkipAvailable.....	18
IsOpened.....	19
SetCallBack.....	20
CSNBLEPrinting.....	21
Open.....	21
Close.....	22
Write.....	23
Read.....	24
SkipAvailable.....	25
IsOpened.....	26
SetCallBack.....	27
CSNNETPrinting.....	28
Open.....	28
Close.....	29
Write.....	30
Read.....	31
SkipAvailable.....	32
IsOpened.....	33
SetCallBack.....	34
CSNCOMPrinting.....	35
Open.....	35

Close.....	37
Write.....	38
Read.....	39
CSNUSBPrinting.....	39
Open.....	39
Close.....	41
Write.....	42
Read.....	43
SkipAvailable.....	44
IsOpened.....	45
SetCallBack.....	46
MemoryPrinting.....	47
Write.....	47
GetWriteBuffer.....	48
CSNPOS.....	49
Set.....	49
GetIO.....	50
POS_PrintPicture.....	51
POS_S_TextOut.....	52
POS_TextOut.....	53
POS_S_SetBarcode.....	54
POS_S_SetQRcode.....	56
POS_DoubleQRCode.....	57
POS_FeedLine.....	58
POS_S_Align.....	59
POS_SetLineHeight.....	60
POS_Reset.....	61
POS_SetMotionUnit.....	62
POS_S_SetAreaWidth.....	63
POS_SetDoubleByteMode.....	64
POS_SetSingleByteMode.....	65
POS_SetDoubleByteEncoding.....	66
POS_FullCutPaper.....	67
POS_HalfCutPaper.....	68
POS_Beep.....	68
POS_KickDrawer.....	69
POS_SetPrintSpeed.....	70
POS_QueryStatus.....	71
POS_RTQueryStatus.....	72
POS_SetBaudrate.....	74
POS_SetCharSetAndCodePage.....	74
POS_Codepage.....	77
POS_SetBasic.....	78
CSNLabel.....	81

Set.....	81
GetIO.....	82
PageBegin.....	83
PageEnd.....	84
PagePrint.....	85
PageFeed.....	86
DrawPlainText.....	87
DrawLine.....	88
DrawBox.....	89
DrawRectangel.....	90
DrawBarcode.....	91
DrawQRCode.....	93
DrawPDF417.....	94
DrawBitmap.....	95
DrawBitmap.....	96
Page.....	97
Set.....	97
GetIO.....	98
PageEnter.....	99
PagePrint.....	100
PageExit.....	101
SetPrintArea.....	102
DrawText.....	104
DrawBarcode.....	106
DrawQRCode.....	108
DrawBitmap.....	109
Canvas.....	110
Set.....	110
GetIO.....	111
CanvasBegin.....	112
CanvasEnd.....	113
CanvasPrint.....	114
SetPrintDirection.....	115
DrawText.....	116
DrawTextAutoNewLine.....	118
DrawTextMultiLine.....	119
DrawLine.....	120
DrawBox.....	121
DrawRect.....	122
DrawBitmap.....	123
DrawQRCode.....	124
DrawBarcode.....	125

一 概述

1 PrinterLibs 是 Android 平台下控制打印机的 jar 库

2 PrinterLibs 有以下几个公共类

A IO 类

包括 IO、BLEPrinting、BTPrinting、NETPrinting、USBPrinting、MemoryPrinting
实现基本的 Open、Close、Write、Read 等功能

IOCallBack 提供了 Open 和 Close 的回调接口、便于获取当前的连接状况

B 打印类

包括 Pos、Label、Page、Canvas

Pos 类实现了诸如打印文本、打印条码、打印二维码、打印图片等功能

Label 类实现了标签打印功能，需要标签打印机支持

Page 类实现了页模式打印功能，需要页模式打印机支持

Canvas 类封装了安卓画布函数，使用图片的方式打印，支持所有打印机

C 使用方法

Android 代码片段：

```
CSNPos mPos = new CSNPos();
CSNBTPrinting mBt = new CSNBTPrinting();
mPos.Set(mBt);
mBt.SetCallBack(this);
```

之后启用异步调用：

```
public class TaskOpen implements Runnable
{
    CSNBTPrinting bt = null;
    String address = null;
    Context context = null;

    public TaskOpen(CSNBTPrinting bt, String address,
                    Context context)
    {
        this.bt = bt;
        this.address = address;
        this.context = context;
    }

    @Override
    public void run() {
```

```
// TODO Auto-generated method stub  
bt.Open(address, context);  
}  
}
```

连接成功之后，调用相应的函数即可打印。

二 接口

CSNIO

Write

Syntax

```
public int Write(byte[] buffer, int offset, int count)
```

Parameters

buffer

发送缓冲区

offset

从指定偏移开始发送数据

count

要发送的字节数

Return value

如果写入成功，返回成功写入的字节数、如果写入失败，返回-1

Remarks

IO 类的 Write 函数为空实现，始终返回-1

Read

Syntax

```
public int Read(byte[] buffer, int offset, int count, int timeout)
```

Parameters

buffer

接收缓冲区

offset

从指定偏移开始存放收到的数据

count

要接收的字节数

timeout

超时毫秒时间

Return value

如果读取成功，返回成功读入的字节数、如果读取失败，返回-1。

Remarks

IO 类的 Read 函数为空实现，始终返回-1

IsOpened

Syntax

```
public boolean IsOpened()
```

Parameters

Return value

如果以连接到打印机，返回 true、否则，返回 false

Remarks

IO 类的 IsOpened 函数为空实现，始终返回 false

CSNIOCallBack

处理底层连接的 4 个类:

BLEPrinting BTPrinting NETPrinting USBPrinting

Open 成功时，会调用 OnOpen

Open 失败时，会调用 OnOpenFailed

Close 或异常断开时，会调用 OnClose

OnOpen

连接成功之后，会调用 OnOpen

Syntax

void OnOpen ()

Parameters

Return value

Remarks

OnOpenFailed

连接失败，会调用 OnOpenFailed

Syntax

```
void OnOpenFailed()
```

Parameters

Return value

Remarks

OnClose

连接断开（主动断开或异常中断），会调用 OnClose

Syntax

void OnClose()

Parameters

Return value

Remarks

CSNBTPrinting

蓝牙 2.0 连接、读写封装

Open

连接指定蓝牙打印机

Syntax

```
public boolean Open(String BTAddress, Context mContext)
```

Parameters

BTAddress

蓝牙打印机地址：形如 00:11:22:33:44:55

mContext

Application Context

Return value

连接成功，返回 true、否则，返回 false。

Remarks

连接成功之后，会调用回调接口 OnOpen，连接失败会调用 OnOpenFailed

Listen

连接 2.0 蓝牙打印机（作为主模式，等待打印机主动上连）

Syntax

```
public boolean Listen(String BTAddress, int timeout, Context mContext)
```

Parameters

BTAddress

蓝牙打印机地址：形如 00:11:22:33:44:55，暂不使用

timeout

等待超时毫秒时间

mContext

Application Context

Return value

连接成功，返回 true、否则，返回 false。

Remarks

连接成功之后，会调用回调接口 OnOpen，连接失败会调用 OnOpenFailed

Close

关闭连接

Syntax

```
public void Close()
```

Parameters

Return value

Remarks

关闭连接，会调用回调接口 OnClose，重复 Close 不会多次调用回调。

Write

通过蓝牙写入数据

Syntax

```
public int Write(byte[] buffer, int offset, int count)
```

Parameters

buffer

发送缓冲区

offset

从指定偏移开始发送数据

count

要发送的字节数

Return value

如果写入成功，返回成功写入的字节数、如果写入失败，返回-1

Remarks

Read

读数据

Syntax

```
public int Read(byte[] buffer, int offset, int count, int timeout)
```

Parameters

buffer

接收缓冲区

offset

从指定偏移开始存放收到的数据

count

要接收的字节数

timeout

超时毫秒时间

Return value

如果读取成功，返回成功读入的字节数、如果读取失败，返回-1。

Remarks

SkipAvailable

忽略缓冲区中的数据

Syntax

```
public void SkipAvailable()
```

Parameters

Return value

Remarks

IsOpened

是否已连接

Syntax

```
public boolean IsOpened()
```

Parameters

Return value

返回 true，表示已经连接、返回 false，表示未连接。

Remarks

IsOpened 函数是建立在心跳的基础上，并不能实时获取连接状态。

如果打印机突然关机，IsOpened 可能需要几秒钟，才能返回正确的结果。

如果想确定打印机是否已连接，可以使用 POS 系列函数中的 RTQueryStatus。

SetCallBack

设置回调接口

Syntax

```
public void SetCallBack(IOCallBack callBack)
```

Parameters

callBack

回调接口，只有设置了该项，在连接成功或连接断开的时候，才会有回调。

Return value

Remarks

CSNBLEPrinting

蓝牙 4.0 连接、读写封装

Open

连接指定蓝牙打印机

Syntax

```
public boolean Open(String BTAddress)
```

Parameters

BTAddress

蓝牙打印机地址：形如 00:11:22:33:44:55

Return value

连接成功，返回 true、否则，返回 false。

Remarks

连接成功之后，会调用回调接口 OnOpen，连接失败会调用 OnOpenFailed

Close

关闭连接

Syntax

```
public void Close()
```

Parameters

Return value

Remarks

关闭连接，会调用回调接口 OnClose，重复 Close 不会多次调用回调。

Write

通过蓝牙写入数据

Syntax

```
public int Write(byte[] buffer, int offset, int count)
```

Parameters

buffer

发送缓冲区

offset

从指定偏移开始发送数据

count

要发送的字节数

Return value

如果写入成功，返回成功写入的字节数、如果写入失败，返回-1

Remarks

蓝牙 4.0 由于标准限制，速度会比 2.0 慢不少。

Read

读数据

Syntax

```
public int Read(byte[] buffer, int offset, int count, int timeout)
```

Parameters

buffer

接收缓冲区

offset

从指定偏移开始存放收到的数据

count

要接收的字节数

timeout

超时毫秒时间

Return value

如果读取成功，返回成功读入的字节数、如果读取失败，返回-1。

Remarks

SkipAvailable

忽略缓冲区中的数据

Syntax

```
public void SkipAvailable()
```

Parameters

Return value

Remarks

IsOpened

是否已连接

Syntax

```
public boolean IsOpened()
```

Parameters

Return value

返回 true，表示已经连接、返回 false，表示未连接。

Remarks

IsOpened 可能会有延时。

如果想确定打印机是否已连接，可以使用 POS 系列函数中的 RTQueryStatus。

SetCallBack

设置回调接口

Syntax

```
public void SetCallBack(IOCallBack callBack)
```

Parameters

callBack

回调接口，只有设置了该项，在连接成功或连接断开的时候，才会有回调。

Return value

Remarks

CSNNETPrinting

WIFI 底层连接、读写封装

Open

连接指定网络打印机

Syntax

```
public boolean Open(String IPAddress, int PortNumber)
```

Parameters

IPAddress

打印机 IP 地址：可以在打印机自检页中获取，打印机默认 IP：192.168.1.87

PortNumber

打印机端口号：固定为 9100

Return value

连接成功，返回 true、否则，返回 false。

Remarks

连接成功之后，会调用回调接口 OnOpen，连接失败会调用 OnOpenFailed

Close

关闭连接

Syntax

```
public void Close()
```

Parameters

Return value

Remarks

关闭连接，会调用回调接口 OnClose，重复 Close 不会多次调用回调。

Write

通过网口写入数据

Syntax

```
public int Write(byte[] buffer, int offset, int count)
```

Parameters

buffer

发送缓冲区

offset

从指定偏移开始发送数据

count

要发送的字节数

Return value

如果写入成功，返回成功写入的字节数、如果写入失败，返回-1

Remarks

如果无线路由器信号不好，或网络环境不佳，可能会造成卡顿。正常情况下，打印巨量数据都不会有问题。

Read

读数据

Syntax

```
public int Read(byte[] buffer, int offset, int count, int timeout)
```

Parameters

buffer

接收缓冲区

offset

从指定偏移开始存放收到的数据

count

要接收的字节数

timeout

超时毫秒时间

Return value

如果读取成功，返回成功读入的字节数、如果读取失败，返回-1。

Remarks

SkipAvailable

忽略缓冲区中的数据

Syntax

```
public void SkipAvailable()
```

Parameters

Return value

Remarks

IsOpened

是否已连接

Syntax

```
public boolean IsOpened()
```

Parameters

Return value

返回 true，表示已经连接、返回 false，表示未连接。

Remarks

IsOpened 函数是建立在心跳的基础上，并不能实时获取连接状态。

如果打印机突然关机，IsOpened 可能需要几秒钟，才能返回正确的结果。

如果想确定打印机是否已连接，可以使用 POS 系列函数中的 RTQueryStatus。

SetCallBack

设置回调接口

Syntax

```
public void SetCallBack(IOCallBack callBack)
```

Parameters

callBack

回调接口，只有设置了该项，在连接成功或连接断开的时候，才会有回调。

Return value

Remarks

CSNCOMPrinting

串口底层连接、读写封装

Open

连接指定 COM 打印机

Syntax

```
public boolean Open(String name, int baudrate, int stopBits, int dataBits,  
int parity, int flowCon, int flags)
```

Parameters

name

串口名称

使用 CSNCOMIO.enumPorts() 获得, 或者手动输入

baudrate

串口波特率

打印机的串口波特率

stopBits

串口停止位, [1, 2]

dataBits

串口数据位, [5, 6, 7, 8]

parity

串口检验位

值	说明
---	----

0	无校验
---	-----

1	奇校验
---	-----

2	偶校验
---	-----

flowCon

串口流控

值	说明
---	----

0	不开启流控
---	-------

1	硬件流控
---	------

2	软件流控
---	------

Flags

占位符，直接填 0 即可

Return value

连接成功，返回 true、否则，返回 false。

Remarks

连接成功之后，会调用回调接口 OnOpen，连接失败会调用 OnOpenFailed

Close

关闭连接

Syntax

```
public void Close()
```

Parameters

Return value

Remarks

关闭连接，会调用回调接口 OnClose，重复 Close 不会多次调用回调。

Write

通过串口写入数据

Syntax

```
public int Write(byte[] buffer, int offset, int count)
```

Parameters

buffer

发送缓冲区

offset

从指定偏移开始发送数据

count

要发送的字节数

Return value

如果写入成功，返回成功写入的字节数、如果写入失败，返回-1

Remarks

USB 发送数据速度最快最稳定，建议使用 USB 进行打印。

Read

读数据

Syntax

```
public int Read(byte[] buffer, int offset, int count, int timeout)
```

Parameters

buffer

接收缓冲区

offset

从指定偏移开始存放收到的数据

count

要接收的字节数

timeout

超时毫秒时间

Return value

如果读取成功，返回成功读入的字节数、如果读取失败，返回-1。

Remarks

CSNUSBPrinting

USB 底层连接、读写封装

Open

连接指定 USB 打印机

Syntax

```
public boolean Open(UsbManager manager, UsbDevice device)
```

Parameters

manager

UsbManager

使用 (UsbManager) getSystemService(Context.*USB_SERVICE*) 获得

device

UsbDevice

通过枚举 USB 设备获得 UsbDevice (mUsbManager.getDeviceList())

Return value

连接成功，返回 true、否则，返回 false。

Remarks

连接成功之后，会调用回调接口 OnOpen，连接失败会调用 OnOpenFailed

Close

关闭连接

Syntax

```
public void Close()
```

Parameters

Return value

Remarks

关闭连接，会调用回调接口 OnClose，重复 Close 不会多次调用回调。

Write

通过 USB 写入数据

Syntax

```
public int Write(byte[] buffer, int offset, int count)
```

Parameters

buffer

发送缓冲区

offset

从指定偏移开始发送数据

count

要发送的字节数

Return value

如果写入成功，返回成功写入的字节数、如果写入失败，返回-1

Remarks

USB 发送数据速度最快最稳定，建议使用 USB 进行打印。

Read

读数据

Syntax

```
public int Read(byte[] buffer, int offset, int count, int timeout)
```

Parameters

buffer

接收缓冲区

offset

从指定偏移开始存放收到的数据

count

要接收的字节数

timeout

超时毫秒时间

Return value

如果读取成功，返回成功读入的字节数、如果读取失败，返回-1。

Remarks

SkipAvailable

忽略缓冲区中的数据

Syntax

```
public void SkipAvailable()
```

Parameters

Return value

Remarks

IsOpened

是否已连接

Syntax

```
public boolean IsOpened()
```

Parameters

Return value

返回 true，表示已经连接、返回 false，表示未连接。

Remarks

IsOpened 函数是建立在心跳的基础上，并不能实时获取连接状态。

如果打印机突然关机，IsOpened 可能需要几秒钟，才能返回正确的结果。

如果想确定打印机是否已连接，可以使用 POS 系列函数中的 RTQueryStatus。

SetCallBack

设置回调接口

Syntax

```
public void SetCallBack(IOCallBack callBack)
```

Parameters

callBack

回调接口，只有设置了该项，在连接成功或连接断开的时候，才会有回调。

Return value

Remarks

MemoryPrinting

内存打印读写封装，将指令打印到内存中，再使用任意其他方式送到打印机

Write

写数据到内存缓冲区

Syntax

```
public int Write(byte[] buffer, int offset, int count)
```

Parameters

buffer

发送缓冲区

offset

从指定偏移开始发送数据

count

要发送的字节数

Return value

如果写入成功，返回成功写入的字节数、如果写入失败，返回-1

Remarks

GetWriteBuffer

获取内存缓冲区的数据

Syntax

```
public byte[] GetWriteBuffer()
```

Parameters

Return value

返回内存缓冲区的数据

Remarks

调用该函数，会返回内存缓冲区的数据，并清空内存缓冲区。

CSNPOS

POS 通过持有一个 IO 对象来与打印机通信
使用 Set (IO) 即可设置 POS 持有的 IO 对象
后续一系列指令，都是通过指定 IO 传达

Set

指定 IO 对象

Syntax

```
public void Set(IO io)
```

Parameters

io

需要使用的 IO 对象

Return value

Remarks

调用该函数，将一个底层读写类绑定到 Pos 这个上层逻辑处理类。

GetIO

获取当前的 IO 对象

Syntax

```
public IO GetIO()()
```

Parameters

Return value

当前 IO 对象

Remarks

POS_PrintPicture

打印图片

Syntax

```
public boolean POS_PrintPicture(Bitmap mBitmap, int nWidth, int  
nBinaryAlgorithm, int nCompressMethod)
```

Parameters

mBitmap

需要打印的位图

nWidth

需要打印的宽度

如果 nWidth 和 Bitmap 的宽度不一致，会等比例缩放到 nWidth 宽

2 寸打印机（58mm 打印机）最大宽度不超过 384 点

3 寸打印机（80mm 打印机）最大宽度不超过 576 点

nBinaryAlgorithm

二值化算法

0 使用抖动算法，对彩色图片有较好的效果。

1 使用平均阀值算法，对文本类图片有较好的效果

nCompressMethod

压缩算法

0 不使用压缩算法

1 使用压缩算法

Return value

Remarks

打印图片使用抖动算法，对彩色图片打印效果较好，打印的图片有灰阶的效果。

POS_S_TextOut

按照一定的格式打印字符串

Syntax

```
public boolean POS_S_TextOut(String pszString, int nOrgx, int  
nWidthTimes,  
int nHeightTimes, int nFontType, int nFontStyle)
```

Parameters

pszString

需要打印的字符串

nOrgx

指定 x 方向（水平）的起始点位置离左边界点数。

2 寸打印机一行 384 点，3 寸打印机一行 576 点。

nWidthTimes

指定字符宽度方向上的放大倍数。

可以为 0 到 1。

nHeightTimes

指定字符高度方向上的放大倍数。

可以为 0 到 1。

nFontType

指定字符的字体类型。

(0x00 标准 ASCII 12x24)

(0x01 压缩 ASCII 9x17)

nFontStyle

指定字符的字体风格。可以为以下列表中的一个或若干个。

(0x00 正常)

(0x08 加粗)

(0x80 1 点粗的下划线)

(0x100 2 点粗的下划线)

(0x200 倒置、只在行首有效)

(0x400 反显、黑底白字)

(0x1000 每个字符顺时针旋转 90 度)

Return value

Remarks

POS_TextOut

按照一定的格式打印字符串（可打印中日韩文编码）

Syntax

```
public boolean POS_TextOut(String pszString, int nLan, int nOrgx, int  
nWidthTimes, int nHeightTimes, int nFontType, int nFontStyle)
```

Parameters

pszString

需要打印的字符串

nLan

0-GBK 1-UTF8 3-BIG5 4-SHIFT-JIS 5-EUC-KR

nOrgx

指定 x 方向（水平）的起始点位置离左边界点数。

2 寸打印机一行 384 点，3 寸打印机一行 576 点。

nWidthTimes

指定字符宽度方向上的放大倍数。

可以为 0 到 1。

nHeightTimes

指定字符高度方向上的放大倍数。

可以为 0 到 1。

nFontType

指定字符的字体类型。

(0x00 标准 ASCII 12x24)

(0x01 压缩 ASCII 9x17)

nFontStyle

指定字符的字体风格。可以为以下列表中的一个或若干个。

(0x00 正常)

(0x08 加粗)

(0x80 1 点粗的下划线)

(0x100 2 点粗的下划线)

(0x200 倒置、只在行首有效)

(0x400 反显、黑底白字)

(0x1000 每个字符顺时针旋转 90 度)

Return value

Remarks

POS_S_SetBarcode

打印条码

Syntax

```
public boolean POS_S_SetBarcode(String strCodedata, int nOrgx, int nType,  
                                int nWidthX, int nHeight, int nHriFontType,  
                                int nHriFontPosition)
```

Parameters

strCodedata

需要打印的条码的字符串

部分条码有格式要求，请按照条码规范打印条码

nOrgx

指定 x 方向（水平）的起始点位置离左边界点数。

2 寸打印机一行 384 点，3 寸打印机一行 576 点。

nType

指定条码的类型。

可以为以下列表中所列值之一。

Value	Meaning
0x41	UPC-A
0x42	UPC-C
0x43	JAN13(EAN13)
0x44	JAN8(EAN8)
0x45	CODE39
0x46	ITF
0x47	CODEBAR
0x48	CODE93
0x49	CODE 128

nWidthX

指定条码的基本元素宽度

范围：[2,6]

nHeight

指定条码的高度点数

可以为 1 到 255

nHriFontType

指定 HRI (Human Readable Interpretation) 字符的字体类型。

可以为以下列表中所列值之一。

Value	Meaning
0x00	标准 ASCII
0x01	压缩 ASCII

nHriFontPosition

指定 HRI (Human Readable Interpretation) 字符的位置。

可以为以下列表中所列值之一。

Value	Meaning
0x00	不打印
0x01	只在条码上方打印
0x02	只在条码下方打印
0x03	条码上、下方都打印

Return value

Remarks

如果条码太宽超出打印机最大打印宽度，则条码不会被打印。

如果条码格式有误，条码也不会打印。

POS_S_SetQRcode

打印二维码

Syntax

```
public boolean POS_S_SetQRcode(String strCodedata, int nWidthX,  
int nVersion, int nErrorCorrectionLevel)
```

Parameters

strCodedata

二维码字符串

nWidthX

二维码每个模块的单元宽度, [1,16]

适当设置模块宽度, 可以使得二维码看起来更漂亮

nVersion

二维码版本大小, 该值和二维码大小相关。 [0,16]

设置为 0 自动计算二维码版本大小。

如果希望二维码大小固定不变, 请设置该值为合适的值。

nErrorCorrectionLevel

纠错等级。

[1, 4]

Return value

Remarks

POS_DoubleQRCode

在一行上打印双二维码

Syntax

```
public boolean POS_DoubleQRCode (
    String QR1Data, int QR1Position, int QR1Ecc, int QR1Version,
    String QR2Data, int QR2Position, int QR2Ecc, int QR2Version,
    int ModuleSize)
```

Parameters

QR1Data

 QR1 数据

QR1Position

 QR1 位置 (2 寸打印机 384 点宽, 3 寸打印机 576 点宽)

QR1Ecc

 QR1 效验 1-4

QR1Version

 QR1 版本 (0-40, 0 自动选择)

QR2Data

 QR2 数据

QR2Position

 QR2 位置

QR2Ecc

 QR2 效验 1-4

QR2Version

 QR2 版本 (0-40, 0 自动选择)

ModuleSize

 QR 码模块宽度

Return value

Remarks

POS_FeedLine

走纸一行

Syntax

```
public boolean POS_FeedLine()
```

Parameters

Return value

Remarks

POS_S_Align

设置对齐方式

Syntax

```
public boolean POS_S_Align(int align)
```

Parameters

Align

设置对其方式

(0 左对齐)

(1 居中对齐)

(2 右对齐)

Return value

Remarks

POS_SetLineHeight

设置行高

Syntax

```
public boolean POS_SetLineHeight(int nHeight)
```

Parameters

nHeight

行高 (0, 255]

Return value

Remarks

POS_Reset

复位打印机（软件复位）

Syntax

```
public boolean POS_Reset()
```

Parameters

Return value

Remarks

POS_SetMotionUnit

设置打印机的移动单位

Syntax

```
public boolean POS_SetMotionUnit(int nHorizontalMU, int nVerticalMU)
```

Parameters

nHorizontalMU

把水平方向上的移动单位设置为 25.4 / nHorizontalMU 毫米。

nVerticalMU

把垂直方向上的移动单位设置为 25.4 / nVerticalMU 毫米。

Return value

Remarks

POS_S_SetAreaWidth

设置标准模式下的打印区域宽度

Syntax

```
public boolean POS_S_SetAreaWidth(int nWidth)
```

Parameters

nWidth

指定打印区域的宽度

Return value

Remarks

POS_SetDoubleByteMode

启用双（多）字节编码

Syntax

```
public boolean POS_SetDoubleByteMode ()
```

Parameters

Return value

Remarks

启用以支持中日韩文

POS_SetSingleByteMode

禁用双（多）字节编码

Syntax

```
public boolean POS_SetSingleByteMode ()
```

Parameters

Return value

Remarks

启用纯英文打印

POS_SetDoubleByteEncoding

切换编码

Syntax

```
public boolean POS_SetDoubleByteEncoding(int nEncoding)
```

Parameters

nEncoding

	编码
0	GBK
1	UTF8
2	保留
3	BIG5
4	SHIFT-JIS
5	EUC-KR

Return value

Remarks

POS_FullCutPaper

全切纸

Syntax

```
public boolean POS_FullCutPaper()
```

Parameters

Return value

Remarks

只对带全切切刀的机器有效

POS_HalfCutPaper

半切纸

Syntax

```
public boolean POS_HalfCutPaper()
```

Parameters

Return value

Remarks

只对带半切切刀的机器有效

POS_Beep

蜂鸣器鸣叫

Syntax

```
public boolean POS_Beep(int nBeepCount, int nBeepMillis)
```

Parameters

nBeepCount

 鸣叫次数

nBeepMillis

 每次鸣叫的时间 = 100 * nBeemMillis ms

Return value

Remarks

POS_KickDrawer

打开钱箱

Syntax

```
public boolean POS_KickDrawer(int nDrawerIndex, int nPulseTime)
```

Parameters

nDrawerIndex

0 表示：脉冲发送到钱箱输出引脚 2
1 表示：脉冲发送到钱箱输出引脚 5

nPulseTime

脉冲时间
高电平时间：nPulseTime*2ms
低电平时间：nPulseTime*2ms

Return value

Remarks

POS_SetPrintSpeed

设置打印速度 注：如果打印速度大于发送速度，打印会有卡顿感。

Syntax

```
public boolean POS_SetPrintSpeed(int nSpeed)
```

Parameters

nSpeed

打印速度 (mm/s)

Return value

Remarks

将打印速度设置为数据发送速度，可以是打印效果达到最好。

可以通过打印一张单据，测量单据的长度和所用时间，用长度/时间，即可。

POS_QueryStatus

查询状态

打印机忙时，该命令会一直阻塞

返回的状态保存在 status 中

Syntax

```
public boolean POS_QueryStatus(byte[] status, int timeout, int MaxRetry)
```

Parameters

status

status = new byte[1] 该值目前无意义

timeout

单次查询状态的超时毫秒时间

MaxRetry

失败重试次数

Return value

返回 true，表明打印机状态 OK。否则，打印机未联机或打印机正忙。

Remarks

POS_RTQueryStatus

实时状态查询

无论打印机处于何种状态，只要打印机收到该命令就立刻回送状态
返回的状态保存在 status 中

Syntax

```
public boolean POS_RTQueryStatus(byte[] status, int type, int timeout,  
int MaxRetry)
```

Parameters

status

```
status = new byte[1]
```

type

type 可取值 [1, 4]

1: 打印机状态

位	0/1	十六进制码	十进制码	功能
0	0	00	0	固定为 0
1	1	02	2	固定为 1
2	0	00	0	一个或两个钱箱打开 (没有钱箱的机器该位固定为零)
	1	04	4	两个钱箱都关闭
3	0	00	0	联机
	1	08	8	脱机
4	1	10	16	固定为 1
5, 6		--	--	未定义
	7	00	00	纸已撕走
		80	96	纸未撕走

2: 传送脱机状态

位	0/1	十六进制码	十进制码	功能
0	0	00	0	固定为 0
1	1	02	2	固定为 1
2	0	00	0	上盖关
	1	04	4	上盖开
3	0	00	0	未按走纸键
	1	08	8	按下走纸键
4	1	10	16	固定为 1

5	0	00	0	打印机不缺纸
	1	20	32	打印机缺纸
6	0	00	00	没有出错情况
	1	40	64	有错误情况
7	0	00	0	固定为 0

3: 传送错误状态

位	0/1	十六进制码	十进制码	功能
0	0	00	0	固定为 0
1	1	02	2	固定为 1
2		--	--	未定义
3	0	00	0	切刀无错误
	1	08	8	切刀有错误
4	1	10	16	固定为 1
5	0	00	0	无不可恢复错误
	1	20	32	有不可恢复错误
6	0	00	00	打印头温度和电压正常
	1	40	64	打印头温度或电压超出范围
7	0	00	0	固定为 0

4: 传送纸传感器状态

位	0/1	十六进制码	十进制码	功能
0	0	00	0	固定为 0
1	1	02	2	固定为 1
2,	0	00	0	有纸
	1	0C	12	纸将近
4	1	10	16	固定为 1
5,	0	00	0	有纸
	1	60	96	纸尽
7	0	00	0	固定为 0

timeout

单次查询状态的超时毫秒时间

MaxRetry

失败重试次数

Return value

返回 true，表明打印机通讯正常，查询的状态保存在 status 中。

Remarks

POS_SetBaudrate

修改打印机波特率

Syntax

```
public boolean POS_SetBaudrate(int nBaudrate)
```

Parameters

nBaudrate

需要修改的波特率: 9600 19200 38400 57600 115200

Return value

返回 true, 表明打印机通讯正常, 返回 false, 代表打印机通讯异常

Remarks

注意打印波特率变化, 以免造成通讯异常, 修改后请重新打开串口

POS_SetCharSetAndCodePage

修改打印机字符集（重启或 RESET 后失效）

Syntax

```
public boolean POS_SetCharSetAndCodePage(int nCharSet, int nCodePage)
```

Parameters

nCharSet

需要修改国际字符集

<u>nCharSet</u>	字符集
0	美国
1	法国
2	德国
3	英国
4	丹麦 I
5	瑞典
6	意大利
7	西班牙 I
8	日本
9	挪威
10	丹麦 II
11	西班牙 II
12	拉丁美洲
13	韩国
14	斯洛文尼亚
15	中国

nCodePage

需要修改的代码页

nCodePage 代码页

- 0 CP437 [美国, 欧洲标准]
- 1 KataKana [片假名]
- 2 CP850 [多语言]
- 3 CP860 [葡萄牙]
- 4 CP863 [加拿大-法语]
- 5 CP865 [北欧]
- 6 WCP1251 [斯拉夫语]
- 7 CP866 斯拉夫 2
- 8 MIK[斯拉夫/保加利亚]
- 9 CP755 [东欧, 拉脱维亚 2]
- 10 [伊朗, 波斯]
- 11 保留
- 12 保留
- 13 保留
- 14 保留
- 15 CP862 [希伯来]
- 16 WCP1252 [拉丁语 1]
- 17 WCP1253 [希腊]
- 18 CP852 [拉丁语 2]

19	CP858 [多种语言拉丁语 1+欧符]
20	伊朗 II [波斯语]
21	拉脱维亚
22	CP864 [阿拉伯语]
23	ISO-8859-1 [西欧]
24	CP737 [希腊]
25	WCP1257 [波罗的海]
26	泰文
27	CP720[阿拉伯语]
28	CP855
29	CP857[土耳其语]
30	WCP1250[中欧]
31	CP775
32	WCP1254[土耳其语]
33	WCP1255[希伯来语]
34	WCP1256[阿拉伯语]
35	WCP1258[越南语]
36	ISO-8859-2[拉丁语 2]
37	ISO-8859-3[拉丁语 3]
38	ISO-8859-4[波罗的语]
39	ISO-8859-5[斯拉夫语]
40	ISO-8859-6[阿拉伯语]
41	ISO-8859-7[希腊语]
42	ISO-8859-8[希伯来语]
43	ISO-8859-9[土耳其语]
44	ISO-8859-15[拉丁语 9]
45	[泰文 2]
46	CP856
47	Cp874
252	CP932 SHIFT_JIS
253	UNICODE UCS-2
254	BIG5
255	GBK

Return value

返回 true，表明打印机通讯正常，返回 false，代表打印机通讯异常

Remarks

重启打印机或 RESET 后失效

POS_Codepage

[修改打印机代码页](#)

Syntax

```
public boolean POS_Codepage(int Value)
```

Parameters

Value

需要修改的代码页

Value	Codepage
0	CP437:USA/Standard Europe
1	Katakana
2	CP850:Latin1;Western European
3	CP860:Portuguese
4	CP863:French Canadian
5	CP865:Nordic
6	CP1251:Cyrillic
7	CP866:Russian;Cyrillic
8	CP1025:Cyrillic/Bulgarian
9	CP773:Estonian/Lithuanian/Latvian
10	Iran
15	CP862:Hebrew
16	CP1252:Latin1
17	CP1253:Greek
18	CP852:Latin2
19	CP858:Latin1+Euro
20	Iran II
21	CP1117:Latvian
22	CP864:Arabic
23	SO-8859-1:West Europe
24	CP737:Greek
25	CP1257:Baltic
26	Thai
27	CP720:Arabic
28	CP855:Cyrillic
29	CP857:Turkish
30	CP1250:Central Europe
31	CP775:Estonian/Lithuanian/Latvian
32	CP1254:Turkish
33	CP1255:Hebrew
34	CP1256:Arabic

35	CP1258:Vietnam
36	ISO-8859-2:Latin2
37	ISO-8859-3:Latin3
38	ISO-8859-4:Baltic
39	ISO-8859-5:Cyrillic
40	ISO-8859-6:Arabic
41	ISO-8859-7:Greek
42	ISO-8859-8:Hebrew
43	ISO-8859-9:Turkish
44	ISO-8859-15:Latin3
45	Thai2
46	CP856:Hebrew
47	CP874:Thai
252	CP932 SHIFT_JIS
253	UNICODE UCS-2
254	CP950 BIG5
255	CP936 GBK

Return value

返回 true, 表明打印机通讯正常, 返回 false, 代表打印机通讯异常

Remarks

重启打印机有效

POS_SetBasic

修改打印机基础设置

Syntax

```
public boolean POS_SetBasic(int nFontStyle, int nDensity, int nLine, int
nBeep, int nCut)
```

Parameters

nFontStyle

设置字体规格,各值定义如下:

值	字体
0	9*17
1	12*24
2	9*24
3	16*18

nDensity

设置浓度,各值定义如下:

值	字体

0	微淡
1	正常
2	微浓
3	高浓度

nLine

设置进纸模式,各值定义如下:

值	模式
0	0x0A
1	0x0D

nBeep

是否启用蜂鸣器,各值定义如下:

值	开关
0	关闭蜂鸣器
1	开启蜂鸣器

nCut

是否启用切刀,各值定义如下:

值	开关
0	关闭切刀功能
1	开启切刀功能

Return value

返回 true, 表明打印机通讯正常, 返回 false, 代表打印机通讯异常

Remarks

CSNLabel

Label 通过持有一个 IO 对象来与打印机通信
使用 Set (IO) 即可设置 Label 持有的 IO 对象
后续一系列指令，都是通过指定 IO 传达

Set

指定 IO 对象

Syntax

```
public void Set(IO io)
```

Parameters

io

需要使用的 IO 对象

Return value

Remarks

调用该函数，将一个底层读写类绑定到 Label 这个上层逻辑处理类。

GetIO

获取当前的 IO 对象

Syntax

```
public IO GetIO()
```

Parameters

Return value

当前 IO 对象

Remarks

PageBegin

描述：指示一个 Page 页面的开始，并设置 Page 页的大小，参考点坐标和页面旋转角度。

Syntax

```
public boolean PageBegin(int startx, int starty, int width, int height,  
    int rotate)
```

Parameters

`startx`

页面起始点 x 坐标

`starty`

页面起始点 y 坐标

`width`

页面页宽

`startx + width` 的范围为 [1, 384]。编写 SDK 的时候，该打印机一行的打印点数为 384 点。如果你不确定每行打印点数，请参考打印机规格书。一般来说有 384, 576, 832 这三种规格。

`height`

页面页高

`starty + height` 的范围 [1, 936]。编写 SDK 的时候，限制是 936，但是这个值并不确定，这和打印机的资源有关。即便如此，也不建议把页高设置过大。建议页宽和页高设置和标签纸匹配即可。

`rotate`

页面旋转。`rotate` 的取值范围 {0, 1}。为 0，页面不旋转打印，为 1，页面旋转 90 度打印。

Return value

Remarks

PageEnd

描述：指示一个 Page 页面的结束。

Syntax

```
public boolean PageEnd()
```

Parameters

Return value

Remarks

PagePrint

描述：将 Page 页上的内容打印到标签纸上。

Syntax

```
public boolean PagePrint(int num)
```

Parameters

num

打印的次数，1-255。

Return value

Remarks

PageFeed

描述：走纸到标签缝隙处。

Syntax

```
public boolean PageFeed()
```

Parameters

Return value

Remarks

DrawPlainText

描述：在 Page 页面上指定位置绘制文本。只能单行打印。

Syntax

```
public boolean DrawPlainText(int startx, int starty, int font, int style,  
                           byte[] str)
```

Parameters

`startx`

定义文本起始位置 x 坐标，取值范围：[0, Page_Width-1]

`starty`

定义文本起始位置 y 坐标，取值范围：[0, Page_Height-1]

`font`

选择字体，有效值范围为{16, 24, 32, 48, 64, 80, 96}，当前打印机只可以使用 24。

`style`

字符风格。

数据位

定义

0 加粗标志位：置 1 字体加粗，清零则字体不加粗。

1 下划线标志位：置 1 文本带下划线，清零则无下划线。

2 反白标志位：置 1 文本反白(黑底白字)，清零不反白。

3 删除线标志位：置 1 文本带删除线，清零则无删除线。

[5,4] 旋转标志位：00 旋转 0°；

01 旋转 90°；

10 旋转 180°；

11 旋转 270°；

[11,8] 字体宽度放大倍数；

[15,12] 字体高度放大倍数；

`str`

字符串数据流

Return value

Remarks

DrawLine

描述：在 Page 页指定两点间绘制一条直线段。

Syntax

```
public boolean DrawLine(int startx, int starty, int endx, int endy, int width,  
int color)
```

Parameters

`startx`

直线段起始点 x 坐标值，取值范围：[0, Page_Width-1]。

`starty`

直线段起始点 y 坐标值，取值范围：[0, Page_Height-1]。

`endx`

直线段终止点 x 坐标值，取值范围：[0, Page_Width-1]。

`endy`

直线段终止点 y 坐标值，取值范围：[0, Page_Height-1]。

`width`

直线段线宽，取值范围：[1, Page_Height-1]。

`color`

直线段颜色，取值范围：{0, 1}。

当 Color 为 1 时，线段为黑色。

当 Color 为 0 时，线段为白色。

Return value

Remarks

DrawBox

描述：在 Page 页指定位置绘制指定大小的矩形框。

Syntax

```
public boolean DrawBox(int left, int top, int right, int bottom,  
                      int borderwidth, int bordercolor)
```

Parameters

left

矩形框左上角 x 坐标值，取值范围：[0, Page_Width-1]。

top

矩形框左上角 y 坐标值。取值范围：[0, Page_Height-1]。

right

矩形框右下角 x 坐标值。取值范围：[0, Page_Width-1]。

bottom

矩形框右下角 y 坐标值。取值范围：[0, Page_Height-1]。

borderwidth

矩形框线宽。

bordercolor

矩形框线颜色，曲直范围{0, 1}。当 Color = 1 时，绘制黑色矩形宽，Color = 0 时，绘制白色矩形框。

Return value

Remarks

DrawRectangel

描述：在 Page 页指定位置绘制矩形块。

Syntax

```
public boolean DrawRectangel(int left, int top, int right, int bottom,  
    int color)
```

Parameters

left

矩形块左上角 x 坐标值，取值范围：[0, Page_Width-1]。

top

矩形块左上角 y 坐标值。取值范围：[0, Page_Height-1]。

right

矩形块右下角 x 坐标值。取值范围：[0, Page_Width-1]。

bottom

矩形块右下角 y 坐标值。取值范围：[0, Page_Height-1]。

color

矩形块颜色，取值范围：{0, 1}。当 color 为 1 时，矩形块为黑色。当 color 为 0 时，矩形块为白色。

Return value

Remarks

DrawBarcode

描述：在 Page 页指定位置绘制一维条码。

Syntax

```
public boolean DrawBarcode(int startx, int starty, int type, int height,
                           int unitwidth, int rotate, byte[] str)
```

Parameters

startx

条码左上角 x 坐标值，取值范围：[0, Page_Width-1]。

starty

条码左上角 y 坐标值，取值范围：[0, Page_Height-1]。

type

标识条码类型，取值范围：[0, 29]。各值定义如下：

type	类型	长度	条码值范围（十进制）
0	UPC-A	11	48-57
1	UPC-E	6	48-57
2	EAN13	12	48-57
3	EAN8	7	48-57
4	CODE39	1-	48-57, 65-90, 32, 36, 37, 43, 45, 46, 47
5	I25	1-	偶数 48-57
6	CODABAR	1-	48-57, 65-68, 36, 43, 45, 46, 47, 58
7	CODE93	1-255 0-127	
8	CODE128	2-255 0-127	
9	CODE11		
10	MSI		
11	"128M", //可以根据数据切换编码模式-> !096 - !105		
12	"EAN128", // 自动切换编码模式		
13	"25C", // 25C Check use mod 10-> 奇数先在前面补 0, 10 的倍数-[(奇数位的数字之和<从左至右)+(偶数位数字之和)*3]		
14	"39C", //39 碼的檢查碼必須搭配「檢查碼相對值對照表」，如表所示，將查出的相對值累加後再除以 43，得到的餘數再查出相對的編碼字元，即為檢查碼字元。		
15	"39", //Full ASCII 39 Code, 特殊字符用两个可表示的字来表示，39C 同样是包含 Full ASCII, 注意宽窄比处理		
16	"EAN13+2", // 附加码与主码间隔 7-12 单位，起始为 1011 间隔为 01, (_0*10+_1) Mod 4-> 0--AA 1--AB 2--BA 3--BB		
17	"EAN13+5", //附加码部分同上，模式((_0+_2+_4)*3+(_1+_3)*9) mod 10->"bbaaa", "babaa", "baaba", "baaab", "abbaa", "aabba", "aaabb", "ababa", "abaab", "aabab		
18	"EAN8+2", // 同 EAN13+2		
19	"EAN8+5", // 同 EAN13+5		

```
20      "POST", // 详见规格说明, 是高低条码, 不是宽窄条码
21      "UPCA+2", // 附加码见 EAN
22      "UPCA+5", // 附加码见 EAN
23      "UPCE+2", // 附加码见 EAN
24      "UPCE+5", // 附加码见 EAN
25      "CPOST", // 测试不打印。。
26      "MSIC", // 将检查码作为数据再计算一次检查码
27      "PLESSEY", // 测试不打印。。
28      "ITF14", // 25C 变种, 第一个数前补 0, 检查码计算时需扣除最后一个数,
但仍填充为最尾端
29      "EAN14"
```

height

定义条码高度。

unitwidth

定义条码码宽。取值范围: [1, 4]。

各值定义如下:

width 取值	多级条码单位宽度 (mm)	二进制条码窄线条宽度	二进制条码宽线条宽度
1	0.125	0.125	0.25
2	0.25	0.25	0.50
3	0.375	0.375	0.75
4	0.50	0.50	1.0

rotate

表示条码旋转角度。取值范围: [0, 3]。各值定义如下:

Rotate 取值	定义
0	条码不旋转绘制。
1	条码旋转 90° 绘制。
2	条码旋转 180° 绘制。
3	条码旋转 270° 绘制。

str

文本字符数据流

Return value

Remarks

DrawQRCode

描述：在 Page 页指定位置绘制 QRCode 码。

Syntax

```
public boolean DrawQRCode(int startx, int starty, int version, int ecc,  
    int unitwidth, int rotate, byte[] str)
```

Parameters

startx

QRCode 码左上角 x 坐标值，取值范围：[0, Page_Width-1]。

starty

QRCode 码左上角 y 坐标值，取值范围：[0, Page_Height-1]。

version

指定字符版本。取值范围：[0, 20]。当 version 为 0 时，打印机根据字符串长度自动计算版本号。

ecc

指定纠错等级。取值范围：[1, 4]。各值定义如下：

ECC 纠错等级

1 L: 7%，低纠错，数据多。

2 M: 15%，中纠错

3 Q: 优化纠错

4 H: 30%，最高纠错，数据少。

unitwidth

QRCode 码码块，取值范围：[1, 4]。各值定义与一维条码指令输入参数 UniWidth 相同。

rotate

QRCode 码旋转角度，取值范围：[0, 3]。各值定义与一维条码指令输入参数 Rotate 相同。

str

QRCode 文本字符数据流

Return value

Remarks

DrawPDF417

描述：在 Page 页指定位置绘制 PDF417 条码。

Syntax

```
public boolean DrawPDF417(int startx, int starty, int colnum, int lwratio,
                           int ecc, int unitwidth, int rotate, byte[] str)
```

Parameters

startx

PDF417 码左上角 x 坐标值，取值范围：[0, Page_Width-1]。

starty

PDF417 码左上角 y 坐标值，取值范围：[0, Page_Height-1]。

colnum

ColNum 为列数，表述每行容纳多少码字。一个码字为 $17 * \text{UnitWidth}$ 个点。行数由打印机自动产生，行数范围限定为 3~90。ColNum 的取值范围：[1, 30]。

lwratio

宽高比。取值范围：[3, 5]。

ecc

纠错等级，取值范围：[0..8]。

ecc 取值 纠错码数 可存资料量（字节）

0 2 1108

1 4 1106

2 8 1101

3 16 1092

4 32 1072

5 64 1024

6 128 957

7 256 804

8 512 496

unitwidth

PDF417 码码块，取值范围：[1, 3]。各值定义与一维条码指令输入参数 UniWidth 相同。

rotate

PDF417 码旋转角度，取值范围：[0, 3]。各值定义与一维条码指令输入参数 Rotate 相同。

str

PDF417 文本字符数据流。

Return value

Remarks

DrawBitmap

描述：在 Page 页指定位置绘制位图。

Syntax

```
public boolean DrawBitmap(int startx, int starty, int width, int height,  
    int style, byte[] pdata)
```

Parameters

`startx`

位图左上角 x 坐标值，取值范围：[0, Page_Width]。

`starty`

位图左上角 y 坐标值，取值范围：[0, Page_Height]。

`width`

位图的像素宽度。

`height`

位图的像素高度。

`style`

位图打印特效，各位定义如下：

位 定义

0 反白标志位，置 1 位图反白打印，清零正常打印。

[2:1] 旋转标志位：

00 旋转 0°；

01 旋转 90°；

10 旋转 180°；

11 旋转 270°

[7:3] 保留。

[11:8] 位图宽度放大倍数。

[12:15] 位图高度放大倍数。

`pdata`

位图的点阵数据。

Return value

Remarks

DrawBitmap

描述：在 Page 页指定位置绘制位图。

Syntax

```
public boolean DrawBitmap(int startx, int starty, int dstWidth, int dstHeight, int style, Bitmap mBitmap, int nBinaryAlgorithm)
```

Parameters

startx

位图左上角 x 坐标值，取值范围：[0, Page_Width]。

starty

位图左上角 y 坐标值，取值范围：[0, Page_Height]。

dstWidth

要打印的像素宽度。

dstHeight

要打印的像素高度。

style

同上

mBitmap

要打印的位图

nBinaryAlgorithm

二值化算法，彩色图片无法直接打印，需要二值化为黑白图像才可以打印。

0 使用抖动算法

1 使用平均阀值算法

Return value

Remarks

Page

Page 通过持有一个 IO 对象来与打印机通信
使用 Set (IO) 即可设置 Page 持有的 IO 对象
后续一系列指令，都是通过指定 IO 传达

Set

指定 IO 对象

Syntax

```
public void Set(IO io)
```

Parameters

io

需要使用的 IO 对象

Return value

Remarks

调用该函数，将一个底层读写类绑定到 Page 这个上层逻辑处理类。

GetIO

获取当前的 IO 对象

Syntax

```
public IO GetIO()
```

Parameters

Return value

当前 IO 对象

Remarks

PageEnter

选择页模式

Syntax

```
public boolean PageEnter()
```

Parameters

Return value

如果指令写入成功，返回 true。否则，返回 false。

Remarks

PagePrint

页模式下打印页面内容

Syntax

```
public boolean PagePrint()
```

Parameters

Return value

如果指令写入成功，返回 true。否则，返回 false。

Remarks

PageExit

退出页模式

Syntax

```
public boolean PageExit()
```

Parameters

Return value

如果指令写入成功，返回 true。否则，返回 false。

Remarks

SetPrintArea

页模式下设置打印区域

Syntax

```
public boolean SetPrintArea(int left, int top, int right, int bottom, int direction)
```

Parameters

left

打印区域左上角 x 坐标

top

打印区域左上角 y 坐标

right

打印区域右下角 x 坐标

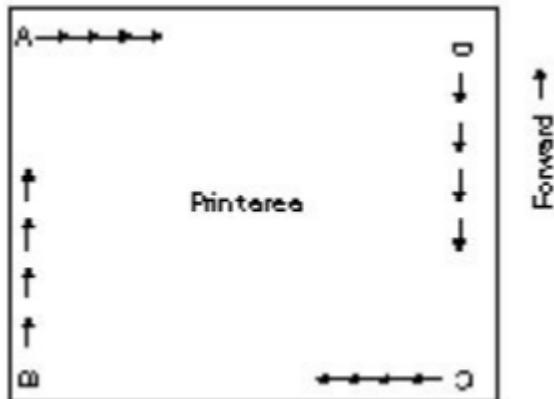
bottom

打印区域右下角 y 坐标

direction

打印区域方向

direction	打印方向	起始位置
0	自左向右	左上角(图中的 A)
1	自下向上	左下角(图中的 B)
2	自右向左	右下角(图中的 C)
3	自上向下	右上角(图中的 D)

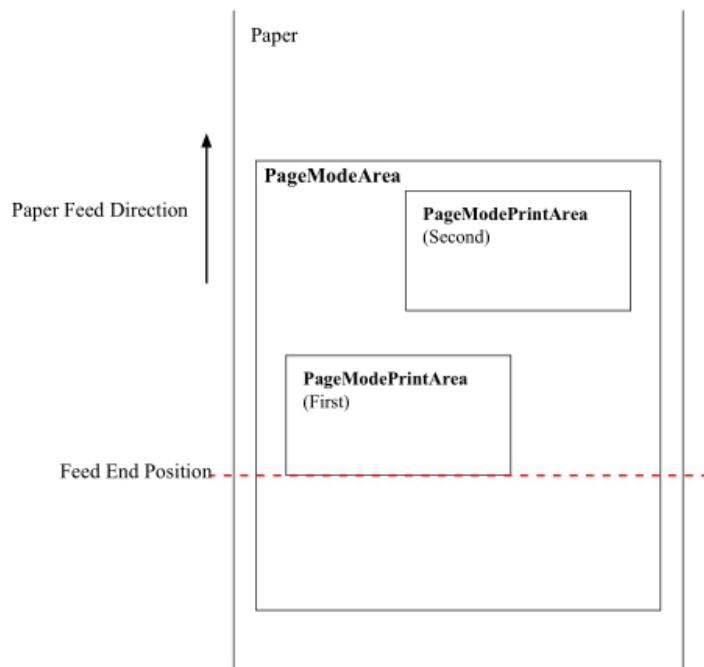


Return value

如果指令写入成功，返回 true。否则，返回 false。

Remarks

下图展示了打印区域的概念



DrawText

画文本

Syntax

```
public boolean DrawText(String pszString, int x, int y, int nWidthScale,  
int nHeightScale, int nFontType, int nFontStyle)
```

Parameters

pszString

要打印的内容。

x

指定水平方向的起始点位置离打印区域左边界点数。（横坐标）

支持左对齐，居中，右对齐

传入 x 见下表

x	含义
-1	左对齐
-2	居中对齐
-3	右对齐
大于等于 0	横坐标

y

指定垂直方向的起始点位置离打印区域上边界点数。（纵坐标）

nWidthScale

指定宽度放大倍数 [0,7]

nHeightScale

指定高度放大倍数 [0,7]

nFontType

字体类型

0 标准字体

1 压缩字体

nFontStyle

指定字体风格，可以为下表中的一个或者若干个（相加即可）

Value	Meaning
0x00	正常

0x08	加粗
0x100	下划线
0x200	倒置（只在行首有效）
0x400	反显（黑底白字）

Return value

如果指令写入成功，返回 true。否则，返回 false。

Remarks

页模式 Draw 函数并不立刻打印，只是填在页面中，直到调用了 PagePrint，才开始打印。

DrawBarcode

画条码

Syntax

```
public boolean DrawBarcode(String pszString, int x, int y, int  
nBarcodeUnitWidth, int nBarcodeHeight, int nHriFontType, int  
nHriFontPosition, int nBarcodeType)
```

Parameters

pszString

条码内容

x

指定水平方向的起始点位置离打印区域左边界点数。（横坐标）

支持左对齐，居中，右对齐

传入 x 见下表

x	含义
-1	左对齐
-2	居中对齐
-3	右对齐
大于等于 0	横坐标

y

指定垂直方向的起始点位置离打印区域上边界点数。（纵坐标）

nBarcodeUnitWidth

指定条码的基本元素宽度。

可以为以下列表中所列值（n）之一。

n	单基本模块宽度 (连续型)	双基本模块宽度 (离散型)	
		窄元素宽度	宽元素宽度
2	0. 25mm	0. 25mm	0. 625mm
3	0. 375mm	0. 375mm	1. 0mm
4	0. 5mm	0. 5mm	1. 25mm
5	0. 625mm	0. 625mm	1. 625mm
6	0. 75mm	0. 75mm	1.875mm

nBarcodeHeight

条码高度

nHriFontType

指定 HRI (Human Readable Interpretation) 字符的字体类型。

可以为以下列表中所列值之一。

Value	Meaning
0x00	标准 ASCII
0x01	压缩 ASCII

nHriFontPosition

指定 HRI (Human Readable Interpretation) 字符的位置。

可以为以下列表中所列值之一。

Value	Meaning
0x00	不打印
0x01	只在条码上方打印
0x02	只在条码下方打印
0x03	条码上、下方都打印

nBarcodeType

可以为以下列表中所列值之一。

Value	Meaning
0x41	UPC-A
0x42	UPC-C
0x43	JAN13(EAN13)
0x44	JAN8(EAN8)
0x45	CODE39
0x46	ITF
0x47	CODEBAR
0x48	CODE93
0x49	CODE 128

Return value

如果指令写入成功，返回 true。否则，返回 false。

Remarks

！！！时间比较赶，条码的对齐还没有做进去。下一个版本会加上。

DrawQRCode

画二维码

Syntax

```
public boolean DrawQRCode(String pszString, int x, int y, int nQRCodeUnitWidth, int nVersion, int nEcLevel)
```

Parameters

pszString

二维码文本

x

指定水平方向的起始点位置离打印区域左边界点数。（横坐标）

支持左对齐，居中，右对齐

传入 x 见下表

x	含义
-1	左对齐
-2	居中对齐
-3	右对齐
大于等于 0	横坐标

y

指定垂直方向的起始点位置离打印区域上边界点数。（纵坐标）

nQRCodeUnitWidth

QR 码单元宽度，范围[1,16]。

QR 码单元宽度越大，QR 码越大。

nVersion

QR 码版本。0 表示自动计算版本。

QR 码版本越大，能编码的字符就越多，QR 码也越大。

nEcLevel

QR 码纠错等级。[1,4]

Return value

如果指令写入成功，返回 true。否则，返回 false。

Remarks

DrawBitmap

画 Bitmap 位图

Syntax

```
public boolean DrawBitmap(Bitmap mBitmap, int x, int y, int dwWidth, int dwHeight, int nBinaryAlgorithm)
```

Parameters

mBitmap

位图

x

指定水平方向的起始点位置离打印区域左边界点数。（横坐标）

支持左对齐，居中，右对齐

传入 x 见下表

x	含义
-1	左对齐
-2	居中对齐
-3	右对齐
大于等于 0	横坐标

y

指定垂直方向的起始点位置离打印区域上边界点数。（纵坐标）

dwWidth

要打印的宽度

dwHeight

要打印的高度

nBinaryAlgorithm

二值化算法

0 使用抖动算法，对彩色图片有较好的效果。

1 使用平均阀值算法，对文本类图片有较好的效果

Return value

如果指令写入成功，返回 true。否则，返回 false。

Remarks

Canvas

Canvas 通过持有一个 IO 对象来与打印机通信
使用 Set(IO) 即可设置 Canvas 持有的 IO 对象
后续一系列指令，都是通过指定 IO 传达

Set

指定 IO 对象

Syntax

```
public void Set(IO io)
```

Parameters

io

需要使用的 IO 对象

Return value

Remarks

调用该函数，将一个底层读写类绑定到 Canvas 这个上层逻辑处理类。

GetIO

获取当前的 IO 对象

Syntax

```
public IO GetIO()
```

Parameters

Return value

当前 IO 对象

Remarks

CanvasBegin

初始化指定宽高的画布。后续画图都以此画布范围内展开。

Syntax

```
public void CanvasBegin(int width, int height)
```

Parameters

width

画布宽度（像素为单位，不超过打印机可打印宽度。建议 58 毫米打印机使用 384, 80 毫米打印机使用 576）

height

画布高度

Return value

Remarks

CanvasEnd

结束该画布。结束之后，只可以调用 CanvasPrint 进行打印操作。不可做其他操作。

Syntax

```
public void CanvasEnd()
```

Parameters

Return value

Remarks

CanvasPrint

打印画布内容

Syntax

```
public boolean CanvasPrint(int nBinaryAlgorithm, int nCompressMethod)
```

Parameters

nBinaryAlgorithm

二值化算法

0 使用抖动算法，对彩色图片有较好的效果。

1 使用平均阀值算法，对文本类图片有较好的效果

nCompressMethod

压缩算法

0 不使用压缩算法

1 使用压缩算法

Return value

Remarks

SetPrintDirection

设置打印区域方向

Syntax

```
public void SetPrintDirection(int direction)
```

Parameters

direction

- 0 自左向右,
- 1 自下向上,
- 2 自右向左,
- 3 自上向下

Return value

Remarks

DrawText

画文本到画布指定位置

Syntax

```
public void DrawText(String text, float x, float y, float rotation,  
Typeface typeface, float textSize, int nFontStyle)
```

Parameters

text

文本内容

x

指定水平方向的起始点位置离打印区域左边界点数。（横坐标）

支持左对齐，居中，右对齐

传入 x 见下表

x	含义
-1	左对齐
-2	居中对齐
-3	右对齐
大于等于 0	横坐标

y

指定垂直方向的起始点位置离打印区域上边界点数。（纵坐标）

支持上对齐，居中，下对齐

传入 y 见下表

y	含义
-1	上对齐
-2	居中对齐
-3	下对齐
大于等于 0	纵坐标

rotation

旋转角度（顺时针方向）

typeface

字体类型

textSize

字体大小

nFontStyle

字体风格

0x00 普通

0x08 加粗

0x80 下划线

Return value

Remarks

DrawTextAuto.NewLine

画文本到画布指定位置

Syntax

```
public void DrawTextAuto.NewLine(String text, float x, float y, float rotation, Typeface typeface, float textSize, int nFontStyle, int outerwidth, float spacingmult, float spacingadd)
```

Parameters

text

x

y

rotation

typeface

textSize

nFontStyle

同 DrawText

outerwidth

超过该宽度自动换行

spacingmult

相对字体高度行间距

spacingadd

额外添加的行间距

Return value

Remarks

DrawTextMultiLine

画文本到画布指定位置

Syntax

```
public void DrawTextMultiLine(String text, float x, float y, float  
rotation, Typeface typeface, float textSize, int nFontSize, float  
rowWidth, float lineSpacing)
```

Parameters

Return value

Remarks

和 `DrawTextAuto.NewLine` 功能一致, 请使用 `DrawTextAuto.NewLine`

DrawLine

画一条线段

Syntax

```
public void DrawLine(float startX, float startY, float stopX, float stopY)
```

Parameters

startX

线段起点横坐标

startY

线段起点纵坐标

stopX

线段终点横坐标

stopY

线段终点纵坐标

Return value

Remarks

DrawBox

画方框（方框线条粗细为 1 个像素点）

Syntax

```
public void DrawBox(float left, float top, float right, float bottom)
```

Parameters

`left`

矩形框左上角横坐标

`top`

矩形框左上角纵坐标

`right`

矩形框右下角横坐标

`bottom`

矩形框右下角纵坐标

Return value

Remarks

DrawRect

画矩形

Syntax

```
public void DrawRect(float left, float top, float right, float bottom)
```

Parameters

`left`

矩形左上角横坐标

`top`

矩形左上角纵坐标

`right`

矩形右下角横坐标

`bottom`

矩形右下角纵坐标

Return value

Remarks

DrawBitmap

画 Bitmap 位图

Syntax

```
public void DrawBitmap(Bitmap bitmap, float x, float y, float rotation)
```

Parameters

bitmap

位图

x

指定水平方向的起始点位置离打印区域左边界点数。（横坐标）

支持左对齐，居中，右对齐

传入 x 见下表

x	含义
-1	左对齐
-2	居中对齐
-3	右对齐
大于等于 0	横坐标

y

指定垂直方向的起始点位置离打印区域上边界点数。（纵坐标）

支持上对齐，居中，下对齐

传入 y 见下表

y	含义
-1	上对齐
-2	居中对齐
-3	下对齐
大于等于 0	纵坐标

rotation

旋转角度（顺时针方向）

Return value

Remarks

DrawQRCode

画二维码

Syntax

```
public void DrawQRCode(String text, float x, float y, float rotation, int unitWidth, int version, int ecc)
```

Parameters

text

二维码文本

x

指定水平方向的起始点位置离打印区域左边界点数。（横坐标）

支持左对齐，居中，右对齐

传入 x 见下表

x	含义
-1	左对齐
-2	居中对齐
-3	右对齐
大于等于 0	横坐标

y

指定垂直方向的起始点位置离打印区域上边界点数。（纵坐标）

支持上对齐，居中，下对齐

传入 y 见下表

y	含义
-1	上对齐
-2	居中对齐
-3	下对齐
大于等于 0	纵坐标

rotation

旋转角度（顺时针方向）

unitWidth

QR 码单元宽度，范围 [1, 16]。 QR 码单元宽度越大，QR 码越大。

version

QR 码版本。0 表示自动计算版本。QR 码版本越大，能编码的字符就越多，QR 码也越大。

ecc

QR 码纠错等级。[1, 4]

Return value

Remarks

DrawBarcode

画条码

Syntax

```
public void DrawBarcode(String text, float x, float y, float rotation,  
int unitWidth, int height, int barcodeType)
```

Parameters

text

条码文本

x

指定水平方向的起始点位置离打印区域左边界点数。（横坐标）

支持左对齐，居中，右对齐

传入 x 见下表

x	含义
-1	左对齐
-2	居中对齐
-3	右对齐
大于等于 0	横坐标

y

指定垂直方向的起始点位置离打印区域上边界点数。（纵坐标）

支持上对齐，居中，下对齐

传入 y 见下表

y	含义
-1	上对齐
-2	居中对齐
-3	下对齐
大于等于 0	纵坐标

rotation

旋转角度（顺时针方向）

unitWidth

条形码单元宽度，范围 [1, 16]。条形码单元宽度越大，条形码越大。

height

条码高度。（打印机 dpi 一般是 203，这时候 8 点就是 1mm）。设置为 80 点就是 1CM。

barcodeType

条码类型。当前只支持 CODE128 条码，更多条码，后续支持。

可以为以下列表中所列值之一。

Value	Meaning
0x49	CODE 128

Return value

Remarks