目录

IO....3
Write....3
Read....3
IsOpened....4
IOCallBack....4
OnClose....4
BTPrinting....5
Open....5
Listen....6
Write....7
Read....7
IsOpened....8
SetCallBack....8
BLEPrinting....9
Open....9
Close....9
Write....10
Read....10
IsOpened....11
SetCallBack....11
NETPrinting....12
Open....12
Close....12
Write....13
Read....13
IsOpened....14
SetCallBack....14
USBPrinting....15
Open....15
Close....15
Write....16
Read....16
IsOpened....17
SetCallBack....17
Pos....18
Set....18
POS_PrintPicture....18
POS_S_TextOut....19
POS_S_SetBarcode....20
POS_S_SetQRcode....22
POS_FeedLine()....22
POS_S_Align(int align)....23
POS_SetLineHeight(int nHeight)....23

POS_SetMotionUnit..... 24
POS_S_SetAreaWidth.....24
Label..... 25
Set.....25
PageBegin..... 25
PageEnd..... 26
PagePrint..... 26
DrawPlainText.....27
DrawLine.....28
DrawBox..... 29
DrawRectangel..... 30
DrawBarcode.....31
DrawQRCode..... 33
DrawPDF417.....34
DrawBitmap.....35

IO

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

IOCallBack

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

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

BTPrinting

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

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

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

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 RTQeuryStatus。

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

BLEPrinting

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

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 RTQeuryStatus。

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

NETPrinting

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

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 RTQeuryStatus。

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

USBPrinting

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

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 RTQeuryStatus。

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

Pos

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。

POS_PrintPicture

打印图片

Syntax

Pubic void POS_PrintPicture(Bitmap mBitmap,int nWidth, int nMode)

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

Return value

Remark

POS_S_TextOut

按照一定的格式打印字符串（可打印中日韩文编码）

Syntax

public void POS_S_TextOut(String pszString, String encoding, int nOrgx, int nWidthTimes, int nHeightTimes, int nFontType, int nFontStyle)

Parameters

pszString

需要打印的字符串

Encoding

GBK　　UTF8　　BIG5　　SHIFT-JIS　　EUC-KR

nOrgx

指定 X 方向（水平）的起始点位置离左边界的点数。 2 寸打印机一行 384 点，3 寸打印机一行 576 点。

nWidthTimes

指定字符的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

Syntax

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

Parameters

strCodedata

需要打印得到条码的字符串

部分条码又格式要求，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的点数。

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	JAN12(EAN13)
0x44	JAN8(EAN8)
0X45	CODE39
0X46	ITF
0X47	CODEBAR
0X48	CODE93
0X49	CODE128

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,范围:[1,255]

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

nHriFontPositon

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

Syntax

public void 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,范围:[1,16]

适当设置模块宽度，可以使得二维码更美观

nVersion

二维码模板大小，该值和二维码大小相关,范围:[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表，请设置该值为合适的值

nErrorCorrectionLevel

纠错等级,范围:[1,4]

Return value

Remarks

POS_FeedLine()

走纸一行

Syntax

public void POS_FeedLine()

Parameters

Return value

Remarks

POS_S_Align(int align)

Syntax
public void 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(int nHeight)

设置行高
Syntax
public void POS_SetLineHeight(int nHeight)
Parameters
nHeight
　　行高 (0,255]
Return value
Remarks

POS_Reset()
复位打印机（软件复位）

POS_SetMotionUnit

设置打印机的移动单位

Syntax

public void 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 void POS_S_SetAreaWidth(int nWidth)

Parameters

nWidth

指定打印区域的宽度

Return value

Remarks

Label

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。

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 void 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 的取值范围为{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 void PageEnd()

Parameters

Return value

Remarks

PagePrint

描述: 将 Page 页上的内容打印到标签纸上。

Syntax

public void PagePrint(int num)

Parameters

num

打印的次数，1-255。

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 void 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 void DrawLine(intstartx,intstarty,intendx,intendy,intwidth, 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 void 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 void 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 void 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)mod10 ->"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 void 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 void 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*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 void 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