

标签打印机

LPAPI 打印接口说明

Android 版本

V2.0

DothanTech 道臻技术

2020 年 04 月 24 日

目 录

目 录	2
文档修改记录	5
1. LPAPI 接口文件概述	7
2. LPAPI 接口使用简述	7
3. 创建 LPAPI 对象	7
4. LPAPI.Callback 相关方法介绍	8
4.1 onProgressInfo 蓝牙设备状态发生变化时的回调函数	9
4.2 onStateChanged 打印机状态发生变化时的回调函数	9
4.3 onPrintProgress 打印进度过程发生变化时的回调函数	10
4.4 onPrinterDiscovery 通过 NFC 扫描到打印机时的回调函数	10
5. 链接打印机相关方法	11
5.1 isPrinterSupported 返回指定的蓝牙设备是否是支持的打印机	11
5.2 isPrinterSupported 返回指定的蓝牙设备名称是否是支持的打印机	11
5.3 getFirstPrinter 获取配对列表中指定型号名称的第一个打印机	12
5.4 getAllPrinters 获取配对列表中指定型号名称的所有打印机	12
5.5 openPrinter 异步方式打开指定名称或型号的打印机	13
5.6 openPrinter 异步方式打开指定的蓝牙设备	13
5.7 openPrinterByAddress 异步方式打开指定的打印机对象	14
5.8 openPrinterSync 同步方式打开指定名称或型号的打印机	14
5.9 openPrinterByAddressSync 同步方式打开指定的打印机对象	14
5.10 getPrinterName 获取已连接打印机名称	15
5.11 getPrinterInfo 获取已连接打印机详细信息	15
5.12 getPrinterState 获取打印机链接状态	16
5.13 isPrinterOpened 判断当前打印机是否打开连接成功	16
5.14 cancel 取消打印任务	17
5.15 closePrinter 断开当前已连接打印机	17
5.16 reopenPrinter 异步方式打开上次连接过的打印机	17
5.17 reopenPrinterSync 同步方式打开上次链接过的打印机	18
5.18 printBitmap 打印 Bitmap 图片	18
5.19 waitPrinterState 等待打印机状态	19
6. 打印任务与页面的方法说明	20
6.1 startJob 开始绘图任务	20
6.2 abortJob 取消绘图任务	21
6.3 endJob 结束绘图任务	21
6.4 commitJob 提交数据打印	21
6.5 commitJobWithParam 提交数据打印	21

6.6	startPage 开始绘制一个页面	22
6.7	endPage 结束一个页面的绘制	22
7.	打印内容参数的设置或获取	22
7.1	setDrawParam 设置绘制相关的参数值	24
7.2	setItemOrientation 设置后续绘制内容的顺时针旋转角度	24
7.3	getItemOrientation 得到当前绘制内容的顺时针旋转角度	24
7.4	setItemHorizontalAlignment 设置后续绘制动作的水平对齐方式	25
7.5	getItemHorizontalAlignment 得到当前绘制动作的水平对齐方式	25
7.6	setItemVerticalAlignment 设置后续绘制动作的垂直对齐方式	25
7.7	getItemVerticalAlignment 得到当前绘制动作的垂直对齐方式	26
7.8	setItemPenAlignment 设置线条画笔对齐方式	26
7.9	getItemPenAlignment 得到线条画笔对齐方式	26
8.	绘制文本字符串方法说明	27
8.1	文本绘制时统一使用的字体	27
8.2	drawText 绘制文本字符串	28
8.3	drawTextRegular 绘制文本字符串	28
8.4	drawRichText 绘制带字体风格设置符的文本字符串	29
8.5	drawTextWithIndent 绘制带缩进功能的文本字符串	30
8.6	drawTextWithScale 绘制带缩进功能的文本字符串	31
8.7	measureFontHeight 计算文本字符串高度	31
9.	绘制条码方法说明	32
9.1	draw1Dbarcode 绘制 Barcode 一维条码	33
9.2	draw2DQRCode 绘制 QRCode 二维码	33
9.3	draw2DPdf417 绘制 Pdf417 二维码	34
10.	矢量图的绘制方法说明	34
10.1	drawRectangle 以指定的线宽，绘制矩形框	34
10.2	fillRectangle 以指定的线宽，绘制填充的矩形框	35
10.3	drawRoundRectangle 以指定的线宽，绘制圆角矩形框	35
10.4	fillRoundRectangle 以指定的线宽，绘制填充的圆角矩形框	36
10.5	drawEllipse 以指定的线宽，绘制圆/椭圆	36
10.6	fillEllipse 以指定的线宽，绘制填充的圆/椭圆	37
10.7	drawCircle 以指定的位置为中心，绘制圆	38
10.8	fillCircle 以指定的位置为中心，绘制填充的圆	38
10.9	drawLine 绘制线（直线/斜线）	39
10.10	drawDashLine 绘制点划线	39
11.	图片绘制方法说明	40
11.1	drawImage 绘制指定文件的图片	40
11.2	drawImageWithActualSize 绘制给定图片实际大小的图片	41
11.3	drawImageWithThreshold 绘制灰度阀值的图片	41

11.4	drawBitmap 绘制位图对象	42
11.5	drawBitmapWithActualSize 绘制给定位图实际大小的位图对象	42
11.6	drawBitmapWithThreshold 绘制灰度阀值的位图对象	43
11.7	drawBitmapStream 绘制输入流对应的位图对象	43
11.8	drawBitmapStreamWithActualSize 绘制输入流对应的位图对象	44
11.9	drawBitmapStreamWithThreshold 绘制输入流对应的位图对象	44



文档修改记录

序号	版本号	修改内容	修改者	修改日期
1	V0.1	文档建立	杨玲美	2017-04-27
2	V0.2	startJob 函数中，去掉 DPI 参数	杨玲美	2017-05-15
3	V0.3	根据实际代码，进行同步更新	胡殿兴	2017-08-12
4	V1.0	根据最新接口,进行同步更新	胡殿兴	2017-10-10
5	V1.1	文档检测与优化	杨玲美	2017-11-11
6	V1.2	新增 LPAPI.Callback 回调函数相关文档	胡殿兴	2017-11-30
7	V1.3	新增 setDrawParam 函数	杨玲美	2018-03-20
8	V1.4	PrinterInfo 中增加 hardwareFlags 等属性	杨玲美	2018-10-22
9	V1.5	新增打印参数相关文档	胡殿兴	2019-03-30
10	V1.6	修改文档格式	杨玲美	2019-04-01
11	V1.7	PrintParamName 打印参数中添加水平和垂直偏移	杨玲美	2019-09-09
12	V1.8	PrintParamName 打印参数中添加 ANTI_COLOR 参数，用来表示是否反色打印	杨玲美	2019-10-11
13	V1.9	新增 isPrinterSupported 函数判断是否是支持的打印机	杨玲美	2019-11-27
14	V2.0	setDrawParam 函数中添加 ANTI_COLOR	杨玲美	2020-04-24

		和 FONT_WRAP 参数		
--	--	----------------	--	--

1. LPAPI 接口文件概述

LPAPI 接口方法说明文档，是针对标签绘图提出的接口方法说明，方便用户在二次开发中调用接口，缩短开发周期，加快开发进度。

LPAPI 接口为调用者提供易用的方法完成标签绘图的操作。本接口中提供了文字、一维条码、二维码、图片和各种图形等多种对象的绘制方法，同时还能进行绘制对象的旋转和绘图页面的旋转，调用者还可调用方法获得绘制完成的标签图片用于标签预览，使标签绘图的操作更加简便。

2. LPAPI 接口使用简述

LPAPI 接口的基本调用流程简单如下：

代码

```
// 得到 LPAPI 对象  
LPAPI api = LPAPI.Factory.createInstance();  
// 连接配对的第一个打印机  
api.openPrinter(null);  
// 开始绘图任务，传入参数(页面宽度，页面高度)  
api.startJob(40, 30, 0);  
// 开始一个页面的绘制，绘制文本字符串  
api.drawText(content, 4, 5, 40, 30, 4);  
// 结束绘图任务提交打印  
api.commitJob();
```

3. 创建 LPAPI 对象

LPAPI 接口中有一个 Factory 对象工厂类 (public static class Factory) 用于创建 LPAPI 对象，创建对象的方法如下：

功能简介	创建 LPAPI 对象
方法原型	public static LPAPI createInstance(Callback callback)

参数名称	callback	打印过程中的相关回调函数，详情参考： LPAPI.Callback ;
返回值	LPAPI 对象。	
使用说明	本方法每次调用会创建一个新的 LPAPI 对象。	

4. LPAPI.Callback 相关方法介绍

[ProgressInfo](#) 状态提示信息：

ProgressInfo	数值	描述
ProgressInfo.AdapterEnabling	0	正在打开蓝牙适配器； addiInfo 为 Boolean 型数据，表示是否与本对象相关；
ProgressInfo.AdapterEnabled	1	蓝牙适配器已经打开； addiInfo 为 Boolean 型数据，表示是否与本对象相关；
ProgressInfo.AdapterDisabled	2	蓝牙适配器未打开； addiInfo 为 Boolean 型数据，表示是否与本对象相关；
ProgressInfo.DeviceBonding	3	正在进行打印机配对操作； addiInfo 为 PrinterAddress 型数据，表示目标打印机
ProgressInfo.DeviceBonded	4	打印机已经配对成功； addiInfo 为 PrinterAddress 型数据，表示目标打印机
ProgressInfo.DeviceUnbonded	5	打印机已经绑定已取消； addiInfo 为 PrinterAddress 型数据，表示目标打印机；

[PrinterState](#) 打印机链接状态：

PrinterState	数值	描述
PrinterState.Connecting	0	正在连接打印机；
PrinterState.Connected	1	打印机已连接；
PrinterState.Connected2	2	打印机已连接的情况下，再次连接该打印机以测试连接是否有效？如果有效的话，会发出该状态消息；
PrinterState.Printing	3	正在打印...;
PrinterState.Working	4	除打印之外的其他工作...;
PrinterState.Disconnected	5	打印机已断开；

[PrintProgress](#) 当前打印进度过程状态：

PrintProgress	数值	描述
PrintProgress.Connected	0	已经和打印机建立连接， addiInfo 为 PrinterAddress 型数据，

		表示目标打印机；
PrintProgress.StartCopy	1	打印机已经开始打印 , addInfo 为 Integer 型数据 , 表示开始打印的页面拷贝序号 , 从 1 开始 ;
PrintProgress.DataEnded	2	打印机数据已经传输结束 , addInfo 位 Integer 型数据 , 表示开始打印的页面拷贝序号 , 从 1 开始 ;
PrintProgress.Success	3	打印机打印成功结束 , addInfo 为 Integer 型数据 , 表示成功打印的页面数 ;
PrintProgress. Failed	4	打印机打印失败 , addInfo 为 PrintFailReason 型数据,表示打印失败原因;

4.1 onProgressInfo 蓝牙设备状态发生变化时的回调函数

功能简介	蓝牙设备等状态发生变化时，会调用该回调函数。	
方法原型	<code>public void onProgressInfo(ProgressInfo info, Object addInfo);</code>	
参数名称	info	蓝牙设备或打印机配对状态；
参数名称	addInfo	蓝牙设备等状态发生变化时的附加信息，具体参考： ProgressInfo ;
返回值	无	
使用说明	无	
使用注意		

4.2 onStateChanged 打印机状态发生变化时的回调函数

功能简介	打印机状态发生变化时，会调用该回调函数。	
方法原型	<code>public void onStateChanged(PrinterAddress address, PrinterState state);</code>	
参数名称	address	目标打印机对象；
参数名称	state	当前打印机状态，具体参考： PrinterState ;

返回值	<u>无</u>
使用说明	无
使用注意	

4.3 onPrintProgress 打印进度过程发生变化时的回调函数

功能简介	当打印进度过程发生变化时，会调用该回调函数。	
方法原型	public void onPrintProgress(PrinterAddress address, Object printData, PrinterProgress progress, Object addInfo);	
参数名称	address	目标打印机的蓝牙地址对象；
参数名称	printData	打印的数据对象；位图 Bitmap 对象或 IAtESCPOS 对象；
参数名称	progress	当前打印进度过程；
参数名称	addInfo	打印进度过程的附加信息，具体参考： PrintProgress ;
返回值	<u>无</u>	
使用说明	无	
使用注意		

4.4 onPrinterDiscovery 通过 NFC 扫描到打印机时的回调函数

功能简介	当手机通过 NFC 扫描到打印机时，会调用该回调函数。	
方法原型	public void onPrinterDiscovery(PrinterAddress address, PrinterInfo info);	
参数名称	address	目标打印机对象；

参数名称	info	NFC 中的打印机信息；
返回值	无	
使用说明	无	
使用注意		

5. 链接打印机相关方法

5.1 isPrinterSupported 返回指定的蓝牙设备是否是支持的打印机

功能简介	返回指定的蓝牙设备是否是支持的打印机。	
方法原型	public static boolean isPrinterSupported(BluetoothDevice printer, String modelName);	
参数名称	printer	蓝牙设备；
参数名称	modelName	打印机型号名称。可以是多个打印机型号名称列表，用英文分号分隔；型号名称以英文减号开始，表示将该型号排除在外；null 或空字符串表示打印机型号没有限制。
返回值	<u>true</u> : 表示支持的打印机； <u>false</u> : 表示不支持的打印机；	
使用说明	无	
使用注意		

5.2 isPrinterSupported 返回指定的蓝牙设备名称是否是支持的打印机

功能简介	返回指定的蓝牙设备名称是否是支持的打印机。	
方法原型	public static boolean isPrinterSupported(String printer, String modelName);	
参数名称	printer	蓝牙设备名称；

参数名称	modelName	打印机型号名称。可以是多个打印机型号名称列表，用英文分号分隔；型号名称以英文减号开始，表示将该型号排除在外；null 或空字符串表示打印机型号没有限制。
返回值	<u>true : 表示支持的打印机；</u> <u>false : 表示不支持的打印机；</u>	
使用说明	无	
使用注意		

5.3 getFirstPrinter 获取配对列表中指定型号名称的第一个打印机

功能简介	获取配对列表中指定型号名称的第一个打印机对象。	
方法原型	public static PrinterAddress getFirstPrinter(String modelName);	
参数名称	modelName	打印机型号名称。可以是多个打印机型号名称列表，用英文分号分隔；型号名称以英文减号开始，表示将该型号排除在外；null 或空字符串表示打印机型号没有限制。
返回值	<u>如果配对列表中没有对应型号的打印机，则返回 null，否则返回第一台打印机对象。</u>	
使用说明		
使用注意	使用前要确认蓝牙已打开，并且对应打印机已进行配对且打开。	

5.4 getAllPrinters 获取配对列表中指定型号名称的所有打印机

功能简介	获取配对列表中指定型号名称的所有打印机对象。	
方法原型	public static List<PrinterAddress> getAllPrinters(String modelName);	
参数名称	modelName	打印机型号名称。可以是多个打印机型号名称列表，用英文分号分隔；型号名称以英文减号开始，表示将该型号排除在外；null 或空字符串表示打印机型号没有限制。
返回值	<u>返回指定型号的所有打印机。</u>	
使用说明	多个打印机型号需要用户自己根据英文分号 “;” 进行字符串分隔操作	

使用注意

5.5 openPrinter 异步方式打开指定名称或型号的打印机

功能简介	异步方式打开指定名称或型号的打印机。	
方法原型	public boolean openPrinter(String printerName);	
参数名称	printerName	参数可为下列类型的数据： 1、 打印机屏幕上显示的名称：用于打开指定名称的打印机； 2、 打印机型号（名称前缀）：用于打开指定型号的打印机；也可同时指定多个型号，多个型号之间以英文逗号 “,” 进行分隔； 3、 空字符串：打开配对列表中的第一台打印机；
返回值	成功与否	
使用说明	链接打印机。	
使用注意	在开始打印任务前，须先调用该方法，或者同系列的方法。	

5.6 openPrinter 异步方式打开指定的蓝牙设备

功能简介	异步方式打开指定的蓝牙设备。	
方法原型	public boolean openPrinter(BluetoothDevice device);	
参数名称	device	接口所支持的蓝牙设备，非厂家所提供的蓝牙设备为无效设备；
返回值	成功与否；true 表示目标打印机是接口所支持的打印机，反之为 false；	
使用说明	链接打印机。	
使用注意	在开始打印任务前，须先调用该方法，或者同系列的方法。	

5.7 openPrinterByAddress 异步方式打开指定的打印机对象

功能简介	异步方式打开指定对象的打印机。	
方法原型	public boolean openPrinterByAddress(PrinterAddress address);	
参数名称	address	目标打印机对象。
返回值	成功与否	
使用说明	如果指定的打印机未找到，则返回失败。	
使用注意		

5.8 openPrinterSync 同步方式打开指定名称或型号的打印机

功能简介	同步方式打开指定名称或型号的打印机。	
方法原型	public boolean openPrinterSync(String printerName);	
参数名称	printerName	参数可为下列类型的数据： 1、 打印机屏幕上显示的名称：用于打开指定名称的打印机； 2、 打印机型号（名称前缀）：用于打开指定型号的打印机；也可同时指定多个型号，多个型号之间以英文逗号 “,” 进行分隔； 3、 null 或者 “” ，打开配对列表中的第一台打印机；
返回值	成功与否	
使用说明		
使用注意	确保打印机已经打开，并进行了配对操作。	

5.9 openPrinterByAddressSync 同步方式打开指定的打印机对象

功能简介	同步方式打开指定对象的打印机。
-------------	------------------------

方法原型	public boolean openPrinterByAddressSync(PrinterAddress address);	
参数名称	address	打印机实例对象，可以通过 <code>getFirstPriner()</code> 或者 <code>getAllPrinter()</code> 函数来获取；
返回值	<u>成功与否</u>	
使用说明	如果指定的打印机未找到，则返回失败。	
使用注意		

5.10 getPrinterName 获取已连接打印机名称

功能简介	获取已连接打印机名称。
方法原型	public String getPrinterName();
参数名称	无
返回值	<u>如果已连接打印机，则返回对应打印机名称，否则返回空字符串。</u>
使用说明	
使用注意	

5.11 getPrinterInfo 获取已连接打印机详细信息

功能简介	获取已连接打印机的详细信息。
方法原型	public PrinterInfo getPrinterInfo();
参数名称	无
返回值	<u>当前打印机相关信息实例。</u>
使用说明	PrinterInfo 中常用属性有： deviceType: DEVICE_TYPE_xxx , 热敏打印机/热转印打印机等等 deviceName: 设备名称，包括型号和序列号

	deviceVersion: 硬件版本号 softwareVersion: 软件版本号 deviceAddress: MAC 地址 deviceAddrType: BLE/SPP/WiFi 等 deviceDPI: 打印精度 , DPI deviceWidth: 打印宽度 , mm hardwareFlags : 硬件标志 , 使用该值匹配不同的固件升级包 softwareFlags : 软件标志
使用注意	确保打印机已成功打开 (链接)。

5.12 getPrinterState 获取打印机连接状态

功能简介	获取打印机连接状态。
方法原型	public PrinterState getPrinterState();
参数名称	无
返回值	当前打印机的连接状态，具体类型可参考枚举类型：PrinterState
使用说明	
使用注意	

5.13 isPrinterOpened 判断当前打印机是否打开连接成功

功能简介	判断当前打印机是否打开连接成功。
方法原型	public boolean isPrinterOpened();
参数名称	无
返回值	true : 表示打印机已链接成功 , 可以进行打印 ; false : 表示未链接打印机 , 在打印之前需要链接打印机 ;
使用说明	
使用注意	

5.14 cancel 取消打印任务

功能简介	取消打印任务。
方法原型	<code>public void cancel();</code>
参数名称	无
返回值	无
使用说明	取消已经提交（执行过 <code>commitJob</code> 函数），但尚未开始的打印任务；
使用注意	

5.15 closePrinter 断开当前已连接打印机

功能简介	断开已连接打印机。
方法原型	<code>public void closePrinter();</code>
参数名称	无
返回值	无
使用说明	如果需要多个用户公用同一台打印机，在打印完成之后，可以通过该函数来断开打印机，否则其他用户无法连接到该打印机；
使用注意	在打印未完成之前尽量避免调用该函数，否则有可能会导致打印任务终止；

5.16 reopenPrinter 异步方式打开上次连接过的打印机

功能简介	异步方式打开上次连接过的打印机。
方法原型	<code>public boolean reopenPrinter();</code>

参数名称	无
返回值	<u>成功与否</u>
使用说明	
使用注意	返回成功仅仅表示操作被提交，并不代表着连接成功了，具体的连接结果会通过回调函数给出通知。

5.17 reopenPrinterSync 同步方式打开上次链接过的打印机

功能简介	同步方式打开上次连接过的打印机。
方法原型	public boolean reopenPrinterSync();
参数名称	无
返回值	<u>成功与否</u>
使用说明	
使用注意	

5.18 printBitmap 打印 Bitmap 图片

功能简介	打印 Bitmap 图片。	
方法原型	public boolean printBitmap(Bitmap bitmap, Bundle printParam);	
参数名称	bitmap	需要打印的图片；
	printParam	打印参数，如果为 null，则采用默认打印参数进行打印； 参数类型可参考： PrintParamName ;
返回值	<u>成功与否</u>	
使用说明		

使用注意	1、 bitmap 对象背景色不能为透明色，建议设置为白色背景色； 2、 返回成功仅仅表示操作被提交，并不代表着打印成功了，具体的打印结果会通过回调函数给出通知。
------	--

5.19 waitPrinterState 等待打印机状态

功能简介	等待打印机状态。	
方法原型	public boolean waitPrinterState(printerState state, int millis);	
参数名称	state	期望的打印状态；
	millis	期望的时间，单位为毫秒；小于 0 表示一直等待下去；
返回值	<u>true</u> : 指定时间后，打印机连接状态为给定的连接状态； <u>false</u> : 指定时间后，打印机连接状态跟给定的连接状态不一致；	
使用说明	该方法通常用于通过异步方式链接打印机之后，判断打印机是否链接成功，由于链接打印机是一种耗时的操作，在发起链接打印机请求后不一定能够立即连接成功，此时就需要通过该函数来判断打印机的链接状态；	
使用注意		

[PrintParamName](#) 打印参数名称：

PrintParamName	值	描述
PRINT_DARKNESS	"PRINT_DENSITY"	用于设置打印浓度，值参考 PrintParamValue ;
PRINT_SPEED	"PRINT_SPEED"	用于设置打印速度，值参考 PrintParamValue ;
PRINT_DIRECTION	"PRINT_DIRECTION"	用于设置打印方向，值为 0 , 90 , 180 , 270 ;
PRINT_COPIES	"PRINT_COPIES"	用于设置打印份数，值为任意有效整数；
GAP_TYPE	"GAP_TYPE"	用于设置纸张类型，值参考 PrintParamValue ;
GAP_LENGTH_01MM	"GAP_LENGTH_01MM"	用于设置纸张间隔长度，单位是 0.01 毫米
HORIZONTAL_OFFSET_01MM	"HORIZONTAL_OFFSET_01MM"	用于设置水平方向偏移，单位 mm ;
HORIZONTAL_OFFSET_PX	"HORIZONTAL_OFFSET_PX"	用于设置水平方向偏移，单位像素；
VERTICAL_OFFSET_01MM	"VERTICAL_OFFSET_01MM"	用于设置垂直方向偏移，单位 mm ;
VERTICAL_OFFSET_PX	"VERTICAL_OFFSET_PX"	用于设置垂直方向偏移，单位像素；
ANTI_COLOR	"ANTI_COLOR"	用于表示是否是“反色打印”；

[PrintParamValue](#) 打印参数参考值：

PrintParamValue	值	描述
MIN_PRINT_DARKNESS	0	打印浓度参考值，范围为(0-14)： 0 表示最小打印浓度， 14 表示最大打印浓度， 5 表示默认打印浓度；
DEFAULT_PRINT_DARKNESS	5	
MAX_PRINT_DARKNESS	14	
MIN_PRINT_SPEED	0	打印速度参考值，范围为(0-4)： 0 表示最小打印速度， 2 表示默认打印速度， 4 表示最大打印速度；
DEFAULT_PRINT_SPEED	2	
MAX_PRINT_SPEED	4	
GAP_NONE	0	纸张类型：默认纸张类型，随打印机设置
GAP_HOLE	1	纸张类型：使用带定位孔的标签纸进行定位
GAP_GAP	2	纸张类型：使用间隙纸进行打印
GAP_BLACK	3	纸张类型：使用黑标纸进行打印；

6. 打印任务与页面的方法说明

6.1 startJob 开始绘图任务

功能简介	开始绘图任务。	
方法原型	<code>public boolean startJob(double width, double height, int orientation);</code>	
参数名称	width	标签宽度 (基于绘制视图，不考虑标签旋转。单位毫米)。
	height	标签高度 (基于绘制视图，不考虑标签旋转。单位毫米)。
	orientation	绘图任务的标签页面顺时针旋转角度，0/90/180/270，默认值 0。
返回值	<u>成功与否</u>	
使用说明	开始一个绘图任务时需调用此方法。	
使用注意	开始绘图任务时，之前的任务/数据将会被全部丢弃。	

6.2 abortJob 取消绘图任务

功能简介	取消绘图任务。
方法原型	<code>public void abortJob();</code>
参数名称	无
返回值	无
使用注意	当前的任务/数据将会被全部丢弃，但是参数设置将会被保留。

6.3 endJob 结束绘图任务

功能简介	结束绘图任务。
方法原型	<code>public void endJob();</code>
参数名称	无
返回值	无

6.4 commitJob 提交数据打印

功能简介	提交数据打印。
方法原型	<code>public boolean commitJob();</code>
参数名称	无
返回值	成功与否

6.5 commitJobWithParam 提交数据打印

功能简介	提交数据打印。
------	---------

方法原型	public boolean commitJobWithParam(Bundle printParam);	
参数名称	printParam	设置 Bundle 格式的打印参数，参数类型可参考： PrintParamName ;
返回值	成功与否	

6.6 startPage 开始绘制一个页面

功能简介	开始绘制一个页面。
方法原型	public boolean startPage();
参数名称	无
返回值	成功与否

6.7 endPage 结束一个页面的绘制

功能简介	结束一个页面的绘制。
方法原型	public void endPage();
参数名称	无
返回值	无

7. 打印内容参数的设置或获取

[ItemAlignment](#) 绘制条码的几种风格：

ItemAlignment	数值	描述
ItemAlignment.LEFT	0	水平靠左
ItemAlignment.CENTER	1	水平居中/垂直居中
ItemAlignment.RIGHT	2	水平靠右
ItemAlignment.TOP	0	垂直靠上
ItemAlignment.MIDDLE	1	垂直居中/水平居中

ItemAlignment.BOTTOM	2	垂直靠下
ItemAlignment.SAMEASITEM	3	对象子元素的对齐方式同对象的对齐方式，当前在一维码文本的水平对齐方式中使用

[PenAlignment](#) 绘制的字体有如下几种风格：

PenAlignment	数值	描述
PenAlignment. CENTER	0	绘制的线以指定的位置为中央
PenAlignment. INSET	1	绘制的线在指定的位置内侧

[DrawParamName](#) 绘制相关参数名称：

DrawParamName	描述
DrawParamName.FONT_NAME	字体文件名称 ,存放在工程 assets 目录中的字体文件名称 ; FONT_NAME 值为 String 类型。如 : HeiTi.ttf。注意是字体文件名称 ,而不一定是字体名称。参数值没有指定后缀的情况下 ,会自动加上 .ttf 后缀 ;
DrawParamName.QRCODE_VERSION	QRCode 的编码版本号 (1~40) ; QRCODE_VERSION 值为 Integer 类型。具体含义详见 QRCode 编码规范。未指定版本号时表示根据编码内容自动采用适合的编码版本号 ;
DrawParamName.ERROR_CORRECTION	QRCode 的纠错级别 (0~3) ; ERROR_CORRECTION 值为 Integer 类型。默认为 0。对于标签打印 ,建议采用纠错级别 0 ,以便打印黑点尽量大 ,便于二维码更容易扫描识别 ;
DrawParamName.CHARACTER_SET	QRCode 的字符串编码类型 ; CHARACTER_SET 值为 String 类型。默认为 UTF-8 ;
DrawParamName.MARGIN	条码的留白 ; MARGIN 值为 Integer 类型。QRCode 编码规范中定义的留白值为 0/2/4。默认是 0 ,也即不留白 ;
DrawParamName.ANTI_COLOR	文本是否反色 ; ANTI_COLOR 值为 Boolean 类型。默认是 false ,即文字不反色 ;
DrawParamName.FONT_WRAP	文本是否换行 ; FONT_WRAP 值为 Boolean 类型。默认是 true ,即文本会自动换行 ;

7.1 setDrawParam 设置绘制相关的参数值

功能简介	设置绘制相关的参数值。	
方法原型	<code>public void setDrawParam(String name, Object value);</code>	
参数名称	name	参数名称，在 DrawParamName 接口的静态字符串变量中定义。具体参考： DrawParamName
	value	参数值，具体含义参考不同的参数名称。具体参考： DrawParamName
返回值	无	

7.2 setItemOrientation 设置后续绘制内容的顺时针旋转角度

功能简介	设置后续绘制内容的顺时针旋转角度。	
方法原型	<code>public void setItemOrientation(int orientation);</code>	
参数名称	orientation	后续绘制内容的顺时针旋转角度 (0/90/180/270)，默认值为 0。
返回值	无	

7.3 getItemOrientation 得到当前绘制内容的顺时针旋转角度

功能简介	得到当前绘制内容的顺时针旋转角度。	
方法原型	<code>public int getItemOrientation();</code>	
参数名称	无	
返回值	当前绘制内容的顺时针旋转角度 (0/90/180/270)	

7.4 setItemHorizontalAlignment 设置后续绘制动作的水平对齐方式

功能简介	设置后续绘制动作的水平对齐方式。	
方法原型	<code>public void setItemHorizontalAlignment(int alignment);</code>	
参数名称	alignment	后续绘制动作的水平对齐方式 (ItemAlignment) , 默认值为 <code>ItemAlignment.LEFT</code> 。
返回值	无	
使用说明	水平方向是基于后续绘制的内容的水平绘制方向，跟随绘制内容的顺时针旋转角度旋转。	

7.5 getItemHorizontalAlignment 得到当前绘制动作的水平对齐方式

功能简介	得到当前绘制动作的水平对齐方式。	
方法原型	<code>public int getItemHorizontalAlignment();</code>	
参数名称	无	
返回值	当前绘制动作的水平对齐方式 (ItemAlignment)	
使用说明	水平方向是基于后续绘制的内容的水平绘制方向，跟随绘制内容的顺时针旋转角度旋转。	

7.6 setItemVerticalAlignment 设置后续绘制动作的垂直对齐方式

功能简介	设置后续绘制动作的垂直对齐方式。	
方法原型	<code>public void setItemVerticalAlignment(int alignment);</code>	
参数名称	alignment	后续绘制动作的垂直对齐方式 (ItemAlignment) , 默认值为 <code>ItemAlignment.TOP</code> 。
返回值	无	

使用说明

垂直方向是基于后续绘制的内容的垂直绘制方向，跟随绘制内容的顺时针旋转角度旋转。

7.7 getItemVerticalAlignment 得到当前绘制动作的垂直对齐方式

功能简介	得到当前绘制动作的垂直对齐方式。
方法原型	<code>public int getItemVerticalAlignment();</code>
参数名称	无
返回值	当前绘制动作的垂直对齐方式 (ItemAlignment)
使用说明	垂直方向是基于后续绘制的内容的垂直绘制方向，跟随绘制内容的顺时针旋转角度旋转。

7.8 setItemPenAlignment 设置线条画笔对齐方式

功能简介	设置线条画笔对齐方式。	
方法原型	<code>public void setItemPenAlignment(int penAlignment);</code>	
参数名称	penAlignment	线条画笔对齐方式 (PenAlignment)，数值为以下两者之一： PenAlignment.CENTER ：绘制的线以指定的位置为中央； PenAlignment.INSET ：绘制的线在指定的位置内侧。
返回值	无	

7.9 getItemPenAlignment 得到线条画笔对齐方式

功能简介	得到线条画笔对齐方式。
方法原型	<code>public int getItemPenAlignment();</code>
参数名称	无
返回值	线条画笔对齐方式 (PenAlignment)，数值为以下两者之一： PenAlignment.CENTER ：绘制的线以指定的位置为中央； PenAlignment.INSET ：绘制的线在指定的位置内侧。

8. 绘制文本字符串方法说明

FontStyle 绘制的字体有如下几种风格：

FontStyle	数值	描述
FontStyle.REGULAR	0	正常
FontStyle.BOLD	1	粗体
FontStyle.ITALIC	2	斜体
FontStyle.BOLDITALIC	3	粗斜体
FontStyle.UNDERLINE	4	下划线
FontStyle.STRIKEOUT	8	删除线

8.1 文本绘制时统一使用的字体

安卓系统中不同品牌型号的手机的默认字体，有很大的差异，因此对于打印出的标签的一致性要求比较高的系统，有必要对绘制文本时使用的字体进行统一。统一文本的字体可以通过下面两种方法来进行：

- 对于整个标签中使用一种字体的情况，可以直接在安卓 APP 工程的 assets 目录下，放入要被使用的这个字体文件，将文件名称修改为 FONT.ttf。绘制文本的接口函数就会使用该字体，进行文本的绘制。在找不到该字体文件的情况下，文本绘制函数才会使用系统默认的字体；
- 对于整个标签中使用多种字体的情况，可以在安卓 APP 工程的 assets 目录下，放入需要使用的多个字体文件（对于中文名称的字体文件，建议将文件名称改成英文，因为有的安卓系统对 assets 目录下的中文名称文件，支持的不太好），然后在绘制文本之前，通过调用 [setDrawParam\(IAtBitmap.DrawParamName.FONT_NAME, "xxx"\)](#) 来指定字体文件，进行文本的绘制。

8.2 drawText 绘制文本字符串

功能简介	绘制文本字符串。	
方法原型	public boolean drawText(String text, double x, double y, double width, double height, double fontHeight);	
参数名称	text	需要绘制的文本字符串。
	x	绘制的文本框左上角水平位置 (单位毫米)。
	y	绘制的文本框左上角垂直位置 (单位毫米)。
	width	绘制的文本框水平宽度 (单位毫米)。如果 width 为 0，则会根据绘制文本的显示宽度，根据当前对齐方式进行以 x 为基准点的左中右对齐。 默认值为 0。
	height	绘制的文本框垂直高度 (单位毫米)。如果 height 为 0，则会根据绘制文本的显示高度，根据当前对齐方式进行以 y 为基准点的上中下对齐。 默认值为 0。
	fontHeight	文字大小 (单位毫米)。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行文本字符串的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。 fontHeight 是以毫米为单位。 字号和毫米的转换公式为： 字号 = 毫米 * 72 / 25.4 比方说 9 号字为 3.175 毫米，12 号字为 4.233 毫米 参考 setDrawParam 函数进行文本字体相关的参数设置。 	

8.3 drawTextRegular 绘制文本字符串

功能简介	绘制文本字符串。	
方法原型	public boolean drawTextRegular(String text, double x, double y, double width, double height, double fontHeight, int fontStyle, float lineSpace);	
参数名称	text	需要绘制的文本字符串。
	x	绘制的文本框左上角水平位置 (单位毫米)。
	y	绘制的文本框左上角垂直位置 (单位毫米)。

	width	绘制的文本框水平宽度 (单位毫米)。如果 width 为 0，则会根据绘制文本的显示宽度，根据当前对齐方式进行以 x 为基准点的左中右对齐。 默认值为 0。
	height	绘制的文本框垂直高度 (单位毫米)。如果 height 为 0，则会根据绘制文本的显示高度，根据当前对齐方式进行以 y 为基准点的上中下对齐。 默认值为 0。
	fontHeight	文字大小 (单位毫米)。
	fontStyle	字体风格，正常、粗体、斜体、粗斜体、下划线、删除线 (FontStyle)。默认值为正常字体。
	lineSpace	行间距，默认设置为 0 即可，1 表示当前字体的一倍行间距。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行文本字符串的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。 fontHeight 是以毫米为单位。 字号和毫米的转换公式为： $字号 = 毫米 * 72 / 25.4$ 比方说 9 号字为 3.175 毫米，12 号字为 4.233 毫米 本方法会根据字符串数据中包含的字体风格设置符对绘制的文本进行字体风格设置。 参考 setDrawParam 函数进行文本字体相关的参数设置。 	

8.4 drawRichText 绘制带字体风格设置符的文本字符串

功能简介	绘制带字体风格设置符的文本字符串。	
方法原型	public boolean drawRichText(String text, double x, double y, double width, double height, double fontHeight, int fontStyle);	
参数名称	text	需要绘制带字体风格设置符的文本字符串。
	x	绘制的文本框左上角水平位置 (单位毫米)。
	y	绘制的文本框左上角垂直位置 (单位毫米)。
	width	绘制的文本框水平宽度 (单位毫米)。如果 width 为 0，则会根据绘制文本的显示宽度，根据当前对齐方式进行以 x 为基准点的左中右对齐。
	height	绘制的文本框垂直高度 (单位毫米)。如果 height 为 0，则会根据绘制文本的显示高度，根据当前对齐方式进行以 y 为基准点的上中下对齐。
	fontHeight	文字大小 (单位毫米)。
	fontStyle	初始状态时的字体风格，正常、粗体、斜体、粗斜体、下划线、删除线 (FontStyle)。

返回值	成功与否
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行文本字符串的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。 fontHeight 是以毫米为单位。 字号和毫米的转换公式为： 字号 = 毫米 * 72 / 25.4 比方说 9 号字为 3.175 毫米，12 号字为 4.233 毫米 本方法会根据字符串数据中包含的字体风格设置符对绘制的文本进行字体风格设置。 参考 setDrawParam 函数进行文本字体相关的参数设置。

8.5 drawTextWithIndent 绘制带缩进功能的文本字符串

功能简介	绘制带缩进功能的文本字符串。	
方法原型	public boolean drawTextWithIndent(String text, double x, double y, double width, double height, double fontHeight, int fontStyle, float lineSpace);	
参数名称	text	需要绘制带字体风格设置符的文本字符串。
	x	绘制的文本框左上角水平位置（单位毫米）。
	y	绘制的文本框左上角垂直位置（单位毫米）。
	width	绘制的文本框水平宽度（单位毫米）。如果 width 为 0，则会根据绘制文本的显示宽度，根据当前对齐方式进行以 x 为基准点的左中右对齐。
	height	绘制的文本框垂直高度（单位毫米）。如果 height 为 0，则会根据绘制文本的显示高度，根据当前对齐方式进行以 y 为基准点的上中下对齐。
	fontHeight	文字大小（单位毫米）。
	fontStyle	初始状态时的字体风格，正常、粗体、斜体、粗斜体、下划线、删除线（ FontStyle ）。
	lineSpace	行间距，默认设置为 0 即可，1 表示当前字体的一倍行间距。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行文本字符串的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。 fontHeight 是以毫米为单位。 字号和毫米的转换公式为： 字号 = 毫米 * 72 / 25.4 比方说 9 号字为 3.175 毫米，12 号字为 4.233 毫米 本方法会根据字符串数据中包含的字体风格设置符对绘制的文本进行字体风格设置。 	

- 参考 [setDrawParam](#) 函数进行文本字体相关的参数设置。

8.6 drawTextWithScale 绘制带缩进功能的文本字符串

功能简介	绘制带缩进功能的文本字符串。	
方法原型	<code>public boolean drawTextWithScale(String text, double x, double y, double width, double height, double fontHeight, int fontStyle, float lineSpace, boolean autoIndent);</code>	
参数名称	text	需要绘制带字体风格设置符的文本字符串。
	x	绘制的文本框左上角水平位置 (单位毫米) 。
	y	绘制的文本框左上角垂直位置 (单位毫米) 。
	width	绘制的文本框水平宽度 (单位毫米) 。如果 width 为 0 , 则会根据绘制文本的显示宽度 , 根据当前对齐方式进行以 x 为基准点的左中右对齐。
	height	绘制的文本框垂直高度 (单位毫米) 。如果 height 为 0 , 则会根据绘制文本的显示高度 , 根据当前对齐方式进行以 y 为基准点的上中下对齐。
	fontHeight	文字大小 (单位毫米) 。
	fontStyle	初始状态时的字体风格 , 正常、粗体、斜体、粗斜体、下划线、删除线(FontStyle)。
	lineSpace	行间距 , 默认设置为 0 即可 , 1 表示当前字体的一倍行间距。
	autoIndent	true : 当检测到字符串中有冒号 (“ : ”) 的时候 , 将冒号前后部分分别进行对齐操作。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> ● 如果之前没有调用 startPage 而直接调用本方法 , 则本方法会自动调用 startPage 开始绘制一个页面 , 然后进行文本字符串的绘制。 ● 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。 ● fontHeight 是以毫米为单位。 字号和毫米的转换公式为 : $字号 = 毫米 * 72 / 25.4$ 比方说 9 号字为 3.175 毫米 , 12 号字为 4.233 毫米 ● 本方法会根据字符串数据中包含的字体风格设置符对绘制的文本进行字体风格设置。 ● 参考 setDrawParam 函数进行文本字体相关的参数设置。 	

8.7 measureFontHeight 计算文本字符串高度

功能简介	计算文本字符串高度。
------	------------

方法原型	<code>public double measureFontHeight(String text, double x, double y, double width, double height, double fontHeight, float lineSpace);</code>	
参数名称	text	需要绘制带字体风格设置符的文本字符串。
	x	绘制的文本框左上角水平位置 (单位毫米)。
	y	绘制的文本框左上角垂直位置 (单位毫米)。
	width	绘制的文本框水平宽度 (单位毫米)。如果 width 为 0，则会根据绘制文本的显示宽度，根据当前对齐方式进行以 x 为基准点的左中右对齐。
	height	绘制的文本框垂直高度 (单位毫米)。如果 height 为 0，则会根据绘制文本的显示高度，根据当前对齐方式进行以 y 为基准点的上中下对齐。
	fontHeight	文字大小 (单位毫米)。
	lineSpace	行间距，默认设置为 0 即可，1 表示当前字体的一倍行间距。
返回值	文本字符串的高度 (单位毫米)。	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行文本字符串的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。 fontHeight 是以毫米为单位。 字号和毫米的转换公式为： $字号 = 毫米 * 72 / 25.4$ 比方说 9 号字为 3.175 毫米，12 号字为 4.233 毫米 	

9. 绘制条码方法说明

[BarcodeType](#) 绘制条码的几种风格：

BarcodeType	数值	描述
BarcodeType.UPC_A	20	
BarcodeType.UPC_E	21	
BarcodeType.EAN13	22	
BarcodeType.EAN8	23	
BarcodeType.CODE39	24	
BarcodeType.ITF25	25	
BarcodeType.CODABAR	26	
BarcodeType.CODE93	27	
BarcodeType.CODE128	28	
BarcodeType.ISBN	29	
BarcodeType.ECODE39	30	

9.1 draw1Dbarcode 绘制 Barcode 一维条码

功能简介	绘制一维条码。	
方法原型	public boolean draw1DBarcode(String text, int type, double x, double y, double width, double height, double textHeight);	
参数名称	text	需要绘制的一维条码的内容。
	type	一维条码的编码类型 (BarcodeType) , 默认值为 AUTO。
	x	绘制的一维条码的左上角水平位置 (单位毫米)。
	y	绘制的一维条码的左上角垂直位置 (单位毫米)。
	width	一维条码的整体显示宽度。
	height	一维条码的显示高度 (包括供人识读文本)。
	textHeight	供人识读文本的高度 (单位毫米) , 建议为 3 毫米。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行一维条码的绘制。 绘制的左上角水平垂直位置、宽度、高度是基于当前绘制页面的水平垂直方向。 	

9.2 draw2DQRCode 绘制 QRCode 二维码

功能简介	绘制 QRCode 二维码。	
方法原型	public boolean draw2DQRCode(String text, double x, double y, double width);	
参数名称	text	需要绘制的 QRCode 二维码的内容。
	x	绘制的 QRCode 二维码的左上角水平位置 (单位毫米)。
	y	绘制的 QRCode 二维码的左上角垂直位置 (单位毫米)。
	width	绘制的 QRCode 二维码的水平宽度 (单位毫米)。

返回值	成功与否
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行 QRCode 二维码的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。 参考 setDrawParam 函数进行 QRCode 二维码绘制的更多参数设置。

9.3 draw2DPdf417 绘制 Pdf417 二维码

功能简介	绘制 Pdf417 二维码。	
方法原型	public boolean draw2DPdf417(String text, double x, double y, double width, double height);	
参数名称	text	需要绘制的 Pdf417 二维码的内容。
	x	绘制的 Pdf417 二维码的左上角水平位置（单位毫米）。
	y	绘制的 Pdf417 二维码的左上角垂直位置（单位毫米）。
	width	绘制的 Pdf417 二维码的水平宽度（单位毫米）。
	height	绘制的 Pdf417 二维码的垂直高度（单位毫米）。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行 Pdf417 二维码的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。 	

10. 矢量图的绘制方法说明

10.1 drawRectangle 以指定的线宽，绘制矩形框

功能简介	以指定的线宽，绘制矩形框。	
方法原型	public boolean drawRectangle(double x, double y, double width, double height, double lineWidth);	
参数名称	x	绘制的矩形框的左上角水平位置（单位 mm）。
	y	绘制的矩形框的左上角垂直位置（单位 mm）。
	width	绘制的矩形框的水平宽度（单位 mm）。

	height	绘制的矩形框的垂直高度 (单位 mm)。
	lineWidth	矩形框的线宽 (单位 mm)。矩形框的线宽是向矩形框内部延伸的。
返回值	成功与否	
使用说明		<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行矩形框的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。

10.2 fillRectangle 以指定的线宽，绘制填充的矩形框

功能简介	以指定的线宽，绘制填充的矩形框。	
方法原型	public boolean fillRectangle(double x, double y, double width, double height);	
参数名称	x	绘制的填充矩形框的左上角水平位置 (单位 mm)。
	y	绘制的填充矩形框的左上角垂直位置 (单位 mm)。
	width	绘制的填充矩形框的水平宽度 (单位 mm)。
	height	绘制的填充矩形框的垂直高度 (单位 mm)。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行矩形框的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。 	

10.3 drawRoundRectangle 以指定的线宽，绘制圆角矩形框

功能简介	以指定的线宽，绘制圆角矩形框。	
方法原型	public boolean drawRoundRectangle(double x, double y, double width, double height, double cornerWidth, double cornerHeight, double lineWidth);	
参数名称	x	绘制的圆角矩形框的左上角水平位置 (单位 mm)。
	y	绘制的圆角矩形框的左上角垂直位置 (单位 mm)。
	width	绘制的圆角矩形框的水平宽度 (单位 mm)。

	height	绘制的圆角矩形框的垂直高度 (单位 mm)。
	cornerWidth	圆角宽度 (单位 mm)。
	cornerHeight	圆角高度 (单位 mm) , 默认值为同 cornerWidth。
	lineWidth	圆角矩形框的线宽 (单位 mm)。圆角矩形框的线宽是向圆角矩形框内部延伸的。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行圆角矩形框的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。 	

10.4 fillRoundRectangle 以指定的线宽，绘制填充的圆角矩形框

功能简介	以指定的线宽，绘制填充的圆角矩形框。	
方法原型	public boolean fillRoundRectangle(double x, double y, double width, double height, double cornerWidth, double cornerHeight);	
参数名称	x	绘制的填充圆角矩形框的左上角水平位置 (单位 mm)。
	y	绘制的填充圆角矩形框的左上角垂直位置 (单位 mm)。
	width	绘制的填充圆角矩形框的水平宽度 (单位 mm)。
	height	绘制的填充圆角矩形框的垂直高度 (单位 mm)。
	cornerWidth	圆角宽度 (单位 mm)。
	cornerHeight	圆角高度 (单位 mm) , 默认值为同 cornerWidth。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行圆角矩形框的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。 	

10.5 drawEllipse 以指定的线宽，绘制圆/椭圆

功能简介	以指定的线宽，绘制圆/椭圆。
------	----------------

方法原型	public boolean drawEllipse(double x, double y, double width, double height, double lineWidth);	
参数名称	x	绘制的椭圆的左上角水平位置 (单位 mm)。
	y	绘制的椭圆的左上角垂直位置 (单位 mm)。
	width	绘制的椭圆的水平宽度 (单位 mm)。
	height	绘制的椭圆的垂直高度 (单位 mm)，默认值为同 width，即绘制圆。
	lineWidth	椭圆的线宽 (单位 mm)。椭圆的线宽是向椭圆内部延伸的。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行椭圆/圆的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。 width 和 height 相等时就是打印圆。 	

10.6 fillEllipse 以指定的线宽，绘制填充的圆/椭圆

功能简介	以指定的线宽，绘制填充的圆/椭圆。	
方法原型	public boolean fillEllipse(double x, double y, double width, double height);	
参数名称	x	绘制的填充椭圆的左上角水平位置 (单位 mm)。
	y	绘制的填充椭圆的左上角垂直位置 (单位 mm)。
	width	绘制的填充椭圆的水平宽度 (单位 mm)。
	height	绘制的填充椭圆的垂直高度 (单位 mm)，默认值为同 width，即绘制圆。
	返回值	成功与否
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行椭圆/圆的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。 width 和 height 相等时就是打印圆。 	

10.7 drawCircle 以指定的位置为中心，绘制圆

功能简介	以指定位置为中心，绘制圆。	
方法原型	public boolean drawCircle(double x, double y, double radius, double lineWidth);	
参数名称	x	绘制的圆的 x 轴中心位置 (单位 mm) 。
	y	绘制的圆的 y 轴中心位置 (单位 mm) 。
	radius	绘制圆的半径 (单位 mm) 。
	lineWidth	圆的线宽 (单位 mm) 。圆的线宽是向圆内部延伸的。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行圆的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。 	

10.8 fillCircle 以指定的位置为中心，绘制填充的圆

功能简介	以指定位置为中心，绘制填充的圆。	
方法原型	public boolean fillCircle(double x, double y, double radius);	
参数名称	x	绘制的填充圆形 X 轴坐标中心点位置 (单位 mm) 。
	y	绘制的填充圆形 y 轴坐标中心点位置 (单位 mm) 。
	radius	绘制填充圆形的半径 (单位 mm) 。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行圆的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。 	

10.9 drawLine 绘制线 (直线/斜线)

功能简介	绘制线 (直线/斜线)。	
方法原型	public boolean drawLine(double x1, double y1, double x2, double y2, double lineWidth);	
参数名称	x1	线的起点的水平位置 (单位 mm)。
	y1	线的起点的垂直位置 (单位 mm)。
	x2	线的终点的水平位置 (单位 mm)。
	y2	线的终点的垂直位置 (单位 mm)。
	lineWidth	线宽 (单位 mm)。线宽是向线的下方延伸的。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行直线/斜线的绘制。 绘制的水平垂直位置是基于当前绘制页面的水平垂直方向。 	

10.10 drawDashLine 绘制点划线

功能简介	绘制点划线。	
方法原型	public boolean drawDashLine(double x1, double y1, double x2, double y2, double lineWidth, double[] dashLen, int dashCount); public boolean drawDashLine2(double x1, double y1, double x2, double y2, double lineWidth, double dashLen1, double dashLen2); public boolean drawDashLine4(double x1, double y1, double x2, double y2, double lineWidth, double dashLen1, double dashLen2, double dashLen3, double dashLen4);	
参数名称	x1	线的起点的水平位置 (单位 mm)。
	y1	线的起点的垂直位置 (单位 mm)。
	x2	线的终点的水平位置 (单位 mm)。
	y2	线的终点的垂直位置 (单位 mm)。
	lineWidth	线宽 (单位 mm)。线宽是向线的下方延伸的。

	dashLen1	点划线第一段的长度 (单位 mm)。
	dashLen2	点划线第二段的长度 (单位 mm)。
	dashLen3	点划线第三段的长度 (单位 mm)。
	dashLen4	点划线第四段的长度 (单位 mm)。
	dashLen	点划线线段长度的数组 (单位 mm)。
	dashCount	点划线线段长度数组的元素个数。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行点划线的绘制。 绘制的水平垂直位置是基于当前绘制页面的水平垂直方向。 	

11. 图片绘制方法说明

11.1 drawImage 绘制指定文件的图片

功能简介	绘制指定文件的图片。	
方法原型	<code>public boolean drawImage(String imageFile, double x, double y, double width, double height);</code>	
	imageFile	图片路径。
	x	打印对象在水平方向上的位置 (单位 mm)。
	y	打印对象在垂直方向上的位置 (单位 mm)。
	width	打印对象的宽度 (单位 mm)。如果 width 为 0，则采用加载的位图的宽度。默认值为 0。
	height	打印对象的高度 (单位 mm)。如果 height 为 0，则采用加载的位图的高度。默认值为 0。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行图片的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。 	

11.2 drawImageWithActualSize 绘制给定图片实际大小的图片

功能简介	绘制给定图片实际大小的图片。	
方法原型	public boolean drawImageWithActualSize(String imageFile, double x, double y);	
参数名称	imageFile	图片路径。
	x	打印对象在水平方向上的位置 (单位 mm) 。
	y	打印对象在垂直方向上的位置 (单位 mm) 。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行图片的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。 	

11.3 drawImageWithThreshold 绘制灰度阀值的图片

功能简介	绘制灰度阀值的图片。	
方法原型	public boolean drawImageWithThreshold(String imageFile, double x, double y, double width, double height, int threshold);	
参数名称	imageFile	图片路径。
	x	打印对象在水平方向上的位置 (单位 mm) 。
	y	打印对象在垂直方向上的位置 (单位 mm) 。
	width	打印对象的宽度(单位 mm)。如果 width 为 0，则采用加载的位图的宽度。默认值为 0。
	height	打印对象的高度 (单位 mm)。如果 height 为 0，则采用加载的位图的高度。默认值为 0。
	threshold	绘制位图的灰度阀值。256 表示绘制灰度图片；257 表示绘制原色图片；0 ~ 255 表示绘制黑白图片，原图颜色>灰度阀值的点会被认为是白色，而原图颜色<=灰度阀值的点会被认为是黑色。默认值为 192。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行图片的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。 	

11.4 drawBitmap 绘制位图对象

功能简介	绘制位图对象。	
方法原型	public boolean drawBitmap(Bitmap bitmap, double x, double y, double width, double height);	
参数名称	bitmap	需要绘制的位图 Bitmap 对象。
	x	绘制的位图的左上角水平位置 (单位 mm)。
	y	绘制的位图的左上角垂直位置 (单位 mm)。
	width	绘制的位图的水平宽度 (单位 mm)。如果 width 为 0，则采用加载的位图的宽度。默认值为 0。
	height	绘制的位图的垂直高度 (单位 mm)。如果 height 为 0，则采用加载的位图的高度。默认值为 0。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行位图的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。 位图绘制时会被缩放到指定的宽度和高度。 	

11.5 drawBitmapWithActualSize 绘制给定位图实际大小的位图对象

功能简介	绘制给定位图实际大小的位图对象。	
方法原型	public boolean drawBitmapWithActualSize(Bitmap bitmap, double x, double y);	
参数名称	bitmap	需要绘制的位图 Bitmap 对象。
	x	绘制的位图的左上角水平位置 (单位 mm)。
	y	绘制的位图的左上角垂直位置 (单位 mm)。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行位图的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。 	

- 位图绘制时会被缩放到指定的宽度和高度。

11.6 drawBitmapWithThreshold 绘制灰度阀值的位图对象

功能简介	绘制灰度阀值的位图对象。	
方法原型	public boolean drawBitmapWithThreshold(Bitmap bitmap, double x, double y, double width, double height, int threshold);	
参数名称	bitmap	需要绘制的位图 Bitmap 对象。
	x	绘制的位图的左上角水平位置 (单位 mm)。
	y	绘制的位图的左上角垂直位置 (单位 mm)。
	width	绘制的位图的水平宽度 (单位 mm)。如果 width 为 0，则采用加载的位图的宽度。默认值为 0。
	height	绘制的位图的垂直高度 (单位 mm)。如果 height 为 0，则采用加载的位图的高度。默认值为 0。
	threshold	绘制位图的灰度阀值。256 表示绘制灰度图片；257 表示绘制原色图片；0~255 表示绘制黑白图片，原图颜色>灰度阀值的点会被认为是白色，而原图颜色<=灰度阀值的点会被认为是黑色。默认值为 192。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法，则本方法会自动调用 startPage 开始绘制一个页面，然后进行位图的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。 位图绘制时会被缩放到指定的宽度和高度。 	

11.7 drawBitmapStream 绘制输入流对应的位图对象

功能简介	绘制输入流对应的位图对象。	
方法原型	public boolean drawBitmapStream(InputStream is, double x, double y, double width, double height);	
参数名称	is	需要绘制的位图对象对应的输入流对象。
	x	绘制的位图的左上角水平位置 (单位 mm)。
	y	绘制的位图的左上角垂直位置 (单位 mm)。

	width	绘制的位图的水平宽度 (单位 mm)。如果 width 为 0 , 则采用加载的位图的宽度。默认值为 0。
	height	绘制的位图的垂直高度 (单位 mm)。如果 height 为 0 , 则采用加载的位图的高度。默认值为 0。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法 , 则本方法会自动调用 startPage 开始绘制一个页面 , 然后进行位图的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。 位图绘制时会被缩放到指定的宽度和高度。 	

11.8 drawBitmapStreamWithActualSize 绘制输入流对应的位图对

象

功能简介	绘制输入流对应的位图对象。	
方法原型	public boolean drawBitmapStreamWithActualSize(InputStream is, double x, double y);	
参数名称	is	需要绘制的位图对象对应的输入流对象。
	x	绘制的位图的左上角水平位置 (单位 mm)。
	y	绘制的位图的左上角垂直位置 (单位 mm)。
返回值	成功与否	
使用说明	<ul style="list-style-type: none"> 如果之前没有调用 startPage 而直接调用本方法 , 则本方法会自动调用 startPage 开始绘制一个页面 , 然后进行位图的绘制。 绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。 位图绘制时会被缩放到指定的宽度和高度。 	

11.9 drawBitmapStreamWithThreshold 绘制输入流对应的位图对

象

功能简介	绘制输入流对应的位图对象。	
方法原型	public boolean drawBitmapStreamWithThreshold(InputStream is, double x, double y, double width, double height, int threshold);	
参数名称	is	需要绘制的位图对象对应的输入流对象。

	x	绘制的位图的左上角水平位置 (单位 mm)。
	y	绘制的位图的左上角垂直位置 (单位 mm)。
	width	绘制的位图的水平宽度 (单位 mm)。如果 width 为 0 , 则采用加载的位图的宽度。默认值为 0。
	height	绘制的位图的垂直高度 (单位 mm)。如果 height 为 0 , 则采用加载的位图的高度。默认值为 0。
	threshold	绘制位图的灰度阀值。256 表示绘制灰度图片 ;257 表示绘制原色图片 ;0 ~ 255 表示绘制黑白图片 , 原图颜色 > 灰度阀值的点会被认为是白色 , 而原图颜色 <= 灰度阀值的点会被认为是黑色。默认值为 192。
返回值	成功与否	
使用说明	<ul style="list-style-type: none">如果之前没有调用 startPage 而直接调用本方法 , 则本方法会自动调用 startPage 开始绘制一个页面 , 然后进行位图的绘制。绘制的左上角水平垂直位置、水平宽度、垂直高度是基于当前绘制页面的水平垂直方向。位图绘制时会被缩放到指定的宽度和高度。	