

打印开发指南

文档修订记录

版本号	*变化 状态	简要说明	日期	变更人	批准日期	批准人
V1.0	C	初始版本	2021/8/9	LinJinXing		
V1.1	M	迭代版本	2021/9/26	LinJinXing		

*变化状态：C= 创立，A= 增加，M= 修改，D= 删除

文档审批记录

序号	审批人	角色	审批日期	签字	备注

目录

打印开发指南	1
1 概述	3
2 引入方法	3
3 打印基本流程	4
4 快速上手	4
4.1 标签打印	4
4.2 热敏打印	5
4.3 打印监听示例	5
5 接口说明	6
5.1 获取打印机实例	6
5.2 添加打印监听	6
5.3 移除打印监听	7
5.4 获取打印版本	7
5.5 打开打印	7
5.6 关闭打印	8
5.7 设置浓度	8
5.8 获取浓度	8
5.9 重置打印机	8
5.10 设置字体大小	8
5.11 获取字体大小	9
5.12 设置粗体字	9
5.13 是否粗体字	9
5.14 设置黑标检测	9

5.15 检测黑标状态	10
5.16 设置下划线	10
5.17 检测下划线	10
5.18 设置走纸距离	10
5.19 获取走纸距离	10
5.20 设置回纸距离	11
5.21 获取回纸距离	11
5.22 添加打印文字	11
5.23 添加打印文字	12
5.24 添加一维码	12
5.25 添加一维码	12
5.26 添加二维码	13
5.27 添加二维码	13
5.28 添加图片	13
5.29 添加图片	13
5.30 添加图片	14
5.31 添加图片	14
5.32 添加空白行	14
5.33 开始走纸	14
5.34 设置黑白反转	15
5.35 是否黑白反转	15
5.36 开始走纸	15
5.37 开始走纸	15
6 附录	16

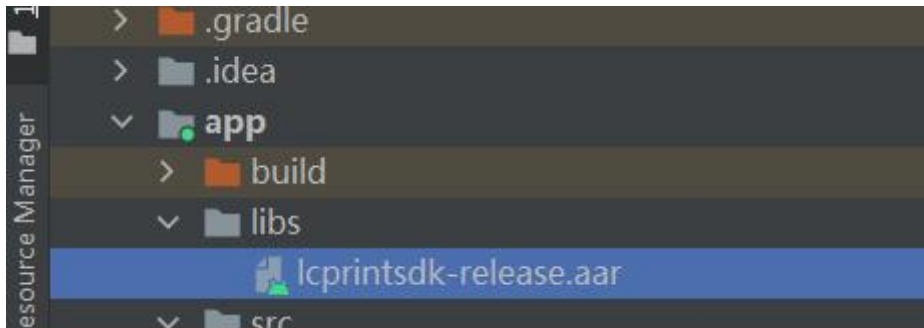
6.1 常量类	16
7 Q&A	19
3.1 Q:如果已有老的 SDK , 如何兼容升级	19
3.2 Q:新旧依赖包冲突怎么办	19
3.3 Q:已有旧的 SDK 如何迁移到新的 SDK	19

1 概述

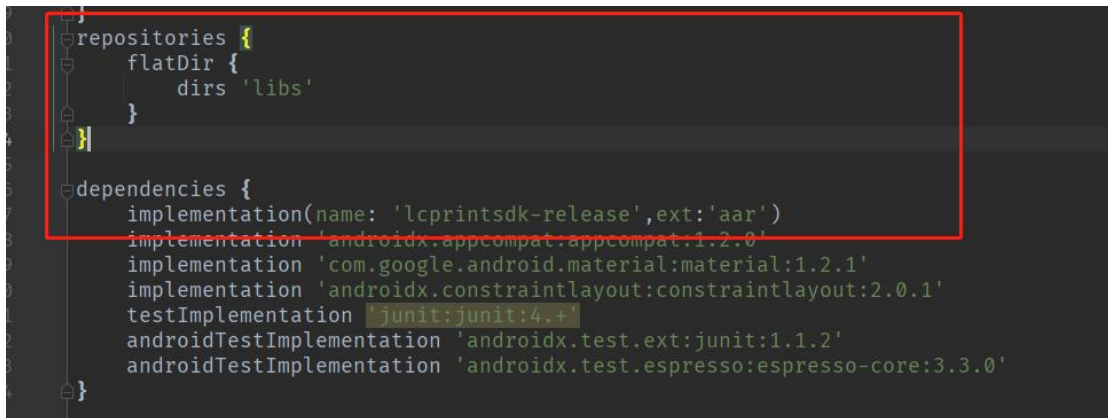
为了方便进行二次开发，我们提供了可以在 Java 平台进行运行的函数库。该库采用 Java 语言编写。

2 引入方法

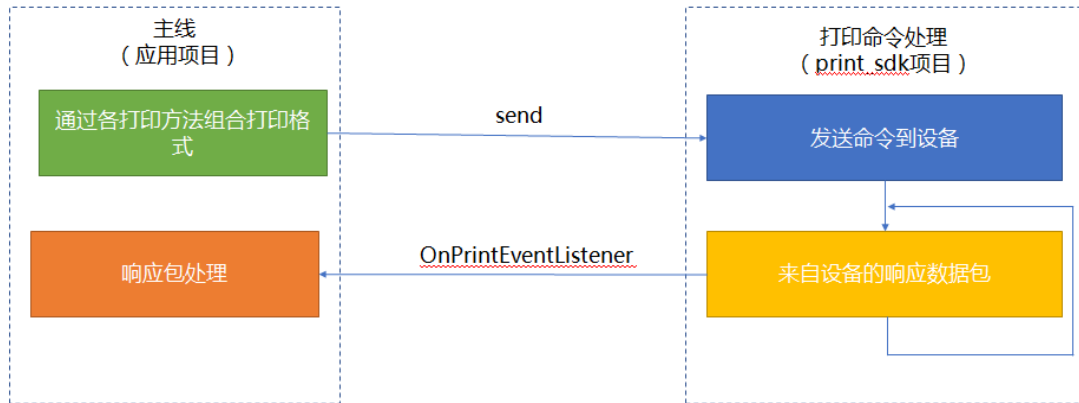
1、集成将 aar 包添加到项目中。



2、在项目里的 gradle 下添加 repositories 和 arr 包的引用（repositories 和 android 节点同级）。



3 打印基本流程



4 快速上手

4.1 标签打印

//步骤 1：获取打印示例

```
printUtil=PrintUtil.getInstance (TestLabelTemplateActivity.this);
```

//步骤 2：设置打印结果回调监听

```
printUtil.setPrintEventListener (this);//设置监听
```

//步骤 3：标签设置

```
printUtil.setUnwindPaperLen (return_distance);//设置回纸距离
```

```
printUtil.printEnableMark (true);//开启黑标
```

```
printUtil.printConcentration (25);//设置浓度
```

//步骤 4：打印文本

```
printUtil.printText ( "打印文本 1" );
```

```
printUtil.printText ( "打印文本 2" );
```

```
printUtil.printText ( "打印文本 3" );
```

//步骤 5：走纸

```
printUtil.printGoToNextMark ();//走纸
```

4.2 热敏打印

//步骤 1：获取打印示例

```
printUtil=PrintUtil.getInstance (this);//获取实例
```

//步骤 2：设置打印结果回调监听

```
printUtil.setPrintEventListener (this);//设置监听
```

//步骤 3：标签设置

```
printUtil.printEnableMark (false);//关闭黑标
```

```
printUtil.printConcentration (25);//打印浓度
```

//步骤 4：打印文本

```
printUtil.printText ("文本");//打印文本
```

```
printUtil.printLine (4);//打印行数
```

//步骤 5：走纸

```
printUtil.start ();//开始走纸
```

4.3 打印监听示例

```
public class NoLabelTemplateActivity extends BaseActivity implements  
PrintUtil.PrinterBinderListener {
```

```
protected void onCreate(Bundle savedInstanceState) {
```

```
//设置打印监听
```

```
printUtil.setPrintEventListener (this);
```

```
}
```

```
@Override
```

```
public void onPrintCallback(int state) {
```



```
if (PrintConfig.IErrorCode.ERROR_NO_ERROR == state) {  
    //打印成功  
  
    showToast (getString (R.string.toast_print_success));  
}  
  
}  
  
@Override  
public void onVersion(String s) {  
    //版本号  
  
    showToast (s);  
}  
}
```

详情参考 [demo](#)

5 接口说明

5.1 获取打印机实例

函数接口	PrintUtil getInstance(Context context)
功能说明	获取打印接口实例化对象。
参数说明	上下文
返回值	PrintUtil 对象

5.2 添加打印监听

函数接口	void setPrintEventListener(PrintUtil.PrinterBinderListener Listener)
功能说明	添加打印监听
参数说明	打印监听器 Listener , 需实现 onPrintCallback(int state) ,

	监听状态回调 state：详见 6.1 章状态说明表。 需实现 onVersion(String version) Version:版本号 例子：详见 4.3 打印监听示例。
返回值	无

5.3 移除打印监听

函数接口	void removePrintListener(PrintUtil.PrinterBinderListener aListener)
功能说明	移除打印监听
参数说明	PrintUtil.PrinterBinderListener 监听器
返回值	无

5.4 获取打印版本

函数接口	String getVersion()
功能说明	获取打印版本
参数说明	无
返回值	版本号

5.5 打开打印

函数接口	boolean open()
功能说明	打开打印
参数说明	无
返回值	true 成功：false 失败

5.6 关闭打印

函数接口	boolean close()
功能说明	关闭打印
参数说明	无

返回值	true 成功：false 失败
-----	------------------

5.7 设置浓度

函数接口	void printConcentration(int density)
功能说明	设置浓度
参数说明	浓度值【1~39】
返回值	无

5.8 重置打印机

函数接口	int resetPrint()
功能说明	重置打印机
参数说明	无
返回值	int

5.9 设置字体大小

函数接口	void printFontSize(int fontSize)
功能说明	设置字体大小
参数说明	int(详细值参考 6.1 章，常量类)
返回值	无

5.10 获取字体大小

函数接口	int getFontSize()
功能说明	获取字体大小
参数说明	无
返回值	int

5.11 设置粗体字

函数接口	void printTextBold(boolean fontBold)
功能说明	设置粗体
参数说明	boolean
返回值	无

5.12 是否粗体字

函数接口	boolean isFontBold()
功能说明	是否粗体
参数说明	无
返回值	boolean

5.13 设置黑标检测

函数接口	void printEnableMark(boolean enable)
功能说明	设置黑标
参数说明	boolean true:开、false : 关
返回值	无

5.14 检测黑标状态

函数接口	boolean isBlackLabel()
功能说明	是否打开黑标
参数说明	无
返回值	boolean

5.15 设置下划线

函数接口	void setUnderLine(boolean enable)
------	--

功能说明	设置下划线
参数说明	boolean : true 开、false 关
返回值	无

5.16 检测下划线

函数接口	boolean isUnderLine()
功能说明	检测下划线
参数说明	无
返回值	boolean

5.17 设置走纸距离

函数接口	void setFeedPaperSpace(int space)
功能说明	设置走纸距离
参数说明	int:距离 (标签模式下生效 , 最大值 2000 , 1000 约等 12.5cm)
返回值	无

5.18 获取走纸距离

函数接口	int getFeedPaperSpace()
功能说明	获取走纸距离
参数说明	无
返回值	int

5.19 设置回纸距离

函数接口	void setUnwindPaperLen(int length)
功能说明	设置回纸距离 (打印标签需开黑标才有实际效果)
参数说明	int
返回值	无

5.20 获取回纸距离

函数接口	<code>int</code> getUnwindPaperLen()
功能说明	获取回纸距离
参数说明	无
返回值	int

5.21 添加打印文字

函数接口	<code>void</code> printText(int offset, int fontSize, boolean isBold, boolean isUnderLine, String content)
功能说明	添加打印文字
参数说明	offset : PrintConfig.Align.ALIGN_LEFT 居左 PrintConfig.Align.ALIGN_CENTER 居中 PrintConfig.Align.ALIGN_RIGHT 居右 fontSize : PrintConfig.FontSize 字体大小 isBold : 是否粗体 isUnderLine : 是否下划线 Content : 文本内容
返回值	无

5.22 添加打印文字

函数接口	<code>void</code> printText(String content)
功能说明	添加打印文字
参数说明	Content : 文本内容
返回值	无

5.23 添加一维码

函数接口	<code>void</code> printBarcode(int offset, int height, String content, int barcodeType, int hriPosition)
------	--

功能说明	添加一维码
参数说明	offset : PrintConfig.Align.ALIGN_LEFT 居左 PrintConfig.Align.ALIGN_CENTER 居中 PrintConfig.Align.ALIGN_RIGHT 居右 height : 一维码高度 content : 一维码内容 barcodeType : 一维码类型 hriPosition : 一维码文本内容位置 PrintConfig.HRIPosition
返回值	无

5.24 添加一维码

函数接口	void printBarcode(int height, String content, int barcodeType)
功能说明	添加一维码
参数说明	height : 一维码高度 content : 一维码内容 barcodeType : 一维码类型
返回值	无

5.25 添加二维码

函数接口	void printQR(int offset, int height, String content)
功能说明	添加二维码
参数说明	offset : PrintConfig.Align.ALIGN_LEFT 居左 PrintConfig.Align.ALIGN_CENTER 居中 PrintConfig.Align.ALIGN_RIGHT 居右 height : 二维码高度 content : 二维码内容
返回值	无

5.26 添加二维码

函数接口	void printQR(String content)
------	--------------------------------------

功能说明	添加二维码
参数说明	content : 二维码内容
返回值	无

5.27 添加图片

函数接口	void printBitmap(int offset, Bitmap image)
功能说明	添加图片
参数说明	offset : PrintConfig.Align.ALIGN_LEFT 居左 PrintConfig.Align.ALIGN_CENTER 居中 PrintConfig.Align.ALIGN_RIGHT 居右 image : Bitmap 图片
返回值	无

5.28 添加图片

函数接口	void printBitmap(Bitmap image)
功能说明	添加图片
参数说明	image : Bitmap 图片
返回值	无

5.29 添加图片

函数接口	void printBitmap(int offset, String imagePath)
功能说明	添加图片
参数说明	offset : PrintConfig.Align.ALIGN_LEFT 居左 PrintConfig.Align.ALIGN_CENTER 居中 PrintConfig.Align.ALIGN_RIGHT 居右 imagePath : 图片路径
返回值	无

5.30 添加图片

函数接口	void printBitmap(String imagePath)
功能说明	添加图片
参数说明	imagePath : 图片路径
返回值	无

5.31 添加空白行

函数接口	void printLine(int lines)
功能说明	添加空白行
参数说明	lines : 行数
返回值	无

5.32 开始走纸

函数接口	void start()
功能说明	开始执行打印（热敏使用）
参数说明	无
返回值	无

5.33 设置黑白反转

函数接口	void setReverse(boolean reverse)
功能说明	设置黑白反转
参数说明	Reverse : true、false
返回值	无

5.34 是否黑白反转

函数接口	boolean isReverse()
------	----------------------------

功能说明	是否黑白反转
参数说明	无
返回值	Boolean

5.35 开始走纸

函数接口	void printGoToNextMark()
功能说明	走纸（标签使用）
参数说明	无
返回值	无

5.36 开始走纸

函数接口	void printGoToNextMark(int distance)
功能说明	走纸（标签使用）
参数说明	Distance:走纸距离
返回值	无

5.37 设置文本行间距

函数接口	void setPrintLineSpacing (float spacing)
功能说明	设置文本行间距
参数说明	spacing:单位倍数 例：spacing=1 1 倍行间距 spacing=3 3 倍行间距
返回值	无

5.38 获取文本行间距

函数接口	float getPrintLineSpacing ()
功能说明	获取文本行间距
参数说明	无
返回值	float 行间距倍数

5.38 获取当前支持模块

函数接口	<code>int</code> getSupportPrint()
功能说明	获取当前支持模块
参数说明	无
返回值	int 1：新模块、0：旧模块

6 附录

6.1 常量类

常量类：android.bld.print.configuration.PrintConfig

//IErrorCode：错误码

```
public class IErrorCode {  
    public static final int ERROR_CMD = 19;//指令错误  
  
    public static final int ERROR_DATA_INPUT = 165;//输入参数错误  
  
    public static final int ERROR_DATA_INVALID = 18;//数据非法  
  
    public static final int ERROR_DEV_BMARK = 7;//黑标检测异常  
  
    public static final int ERROR_DEV_FEED = 5;//正在走纸  
  
    public static final int ERROR_DEV_IS_BUSY = 1;//设备忙  
  
    public static final int ERROR_DEV_NOT_OPEN = 16;//设备未打开  
  
    public static final int ERROR_DEV_NO_BATTERY = 4;//低电  
  
    public static final int ERROR_DEV_PRINT = 6;//正在打印  
  
    public static final int ERROR_GRAY_INVALID = 20;//浓度非法  
  
    public static final int ERROR_NO_DATA = 17;//打印数据不能为空  
  
    public static final int ERROR_NO_ERROR = 0;//打印成功
```

```

    public static final int ERROR_PRINT_BARCODE = 162;//打印条码错误

    public static final int ERROR_PRINT_BITMAP = 161;//打印位图错误

    public static final int ERROR_PRINT_BITMAP_WIDTH_OVERFLOW = 164;//打印位图宽度溢出

    public static final int ERROR_PRINT_DATA_MAC = 167;//Mac 校验错误

    public static final int ERROR_PRINT_HOT = 2;//高温

    public static final int ERROR_PRINT_ILLEGAL_ARGUMENT = 166;//参数错误

    public static final int ERROR_PRINT_NOPAPER = 3;//缺纸

    public static final int ERROR_PRINT_QRCODE = 163;//打印二维码错误

    public static final int ERROR_PRINT_TEXT = 160;//打印文本错误

    public static final int ERROR_PRINT_UNKNOWN = 255;//未知错误

    public static final int ERROR_RESULT_EXIST = 168;//结果已存在

    public static final int ERROR_TIME_OUT = 169;//超时
}

//StateType : 状态类型
public class StateType {
    public static final int CHECK_ALL = 1;
    public static final int CHECK_BMASK = 7;
    public static final int CHECK_BUSY = 2;
    public static final int CHECK_FEED = 5;
    public static final int CHECK_PAPER = 4;
    public static final int CHECK_PRINT = 6;
    public static final int CHECK_TEMP = 3;
}

//HRIPosition : 一维码文本位置
public static class HRIPosition {
    public static final int POSITION_ABOVE = 2;

```

```

        public static final int POSITION_BELOW = 3;
        public static final int POSITION_BOTH = 4;
        public static final int POSITION_NONE = 1;
    }

    //BarCodeType : 条码类型
    public static class BarCodeType {
        public static final int TOP_TYPE_CODABAR = 71;
        public static final int TOP_TYPE_CODE128 = 73;
        public static final int TOP_TYPE_CODE39 = 69;
        public static final int TOP_TYPE_CODE93 = 72;
        public static final int TOP_TYPE_EAN13 = 67;
        public static final int TOP_TYPE_EAN8 = 68;
        public static final int TOP_TYPE_ITF = 70;
        public static final int TOP_TYPE_UPCA = 65;
        public static final int TOP_TYPE_UPCE = 66;

    }

    //ALIGN : 对齐方式
    public static class Align {
        public static final int ALIGN_CENTER = 2;//居中
        public static final int ALIGN_LEFT = 1;//居左
        public static final int ALIGN_RIGHT = 3;//居右
    }

    //FontSize : 字符字体
    public static class FontSize {
        public static final int TOP_FONT_SIZE_LARGE = 5;
        public static final int TOP_FONT_SIZE_MIDDLE = 3;
        public static final int TOP_FONT_SIZE_SMALL = 1;
        public static final int TOP_FONT_SIZE_SUPER = 7;
    }

```

```
public static final int TOP_FONT_SIZE_XLARGE = 6;
public static final int TOP_FONT_SIZE_XMIDDLE = 4;
public static final int TOP_FONT_SIZE_XSMALL = 2;
public static final int TOP_FONT_SIZE_XSUPER = 8;

}
```

7 Q&A

3.1 Q:如果已有老的 SDK , 如何兼容升级

A:获取新旧模块差异接口, 添加判断兼容处理。(区分新旧模块参考章节 5.38)

3.2 Q:新旧依赖包冲突怎么办

A:需要取消 core_3.4.0.jar 包的依赖。

3.3 Q:已有旧的 SDK 如何迁移到新的 SDK

A:可以参考《[S60 与 S60A 打印 SDK 接口差异.docx](#)》文档。

先了解下 sdk 的差异, 新旧打印的打印流程基本差别不大, 就个别方法有变动, 替换使用就可以了。