

POIPrinterManager

```
public class Printer {
    private static final String TAG = "Printer";

    public static void printerTest(Context context) {
        final POIPrinterManager printerManager = new POIPrinterManager(context);
        printerManager.open();
        int state = printerManager.getPrinterState();
        Log.d(TAG, "printer state = " + state);
        //printerManager.setPrintFont("/system/fonts/NotoSansTamil-Bold.otf");
        printerManager.setPrintGray(2000);
        printerManager.setLineSpace(5);
        //printerManager.cleanCache();
        String str1 = "This is an example of a receipt";
        PrintLine p1 = new TextPrintLine(str1, PrintLine.CENTER);
        printerManager.addPrintLine(p1);
        Bitmap bitmap = BitmapFactory.decodeResource(context.getResources(),
R.drawable.shopping_mall);
        printerManager.addPrintLine(new BitmapPrintLine(bitmap,
PrintLine.CENTER));
        printerManager.setPrintFont("/system/fonts/ComingSoon.ttf");
        String str2 = "Floor ** , Building **, No.*** LONG DONG Avenue, Pudong
New District, Shanghai, China";
        PrintLine p2 = new TextPrintLine(str2, PrintLine.LEFT, 20);
        printerManager.addPrintLine(p2);
        printerManager.setPrintFont("/system/fonts/DroidSansMono.ttf");
        List<TextPrintLine> list1 = printList("24 June 2025", "    Assistant
6", "815002", 18, false);
        printerManager.addPrintLine(list1);
        List<TextPrintLine> list2 = printList("Item", "Quantity", "Price", 24,
true);
        printerManager.addPrintLine(list2);
        List<TextPrintLine> list3 = printList("Tomato", "1", "$2.08", 24,
false);
        printerManager.addPrintLine(list3);
        List<TextPrintLine> list4 = printList("Orange", "1", "$1.06", 24,
false);
        printerManager.addPrintLine(list4);
        PrintLine p3 = new TextPrintLine("Total $3.14", PrintLine.RIGHT);
        printerManager.addPrintLine(p3);
        printerManager.addPrintLine(new TextPrintLine(""));
        // drawable/barcode.jpg 100*100
        bitmap = BitmapFactory.decodeResource(context.getResources(),
R.drawable.barcode);
        printerManager.addPrintLine(new BitmapPrintLine(bitmap,
PrintLine.CENTER));
        printerManager.addPrintLine(new TextPrintLine(""));
        String str3 = "Did you know you could have earned Rewards points on this
purchase?";
        PrintLine p4 = new TextPrintLine(str3, PrintLine.CENTER);
        printerManager.addPrintLine(p4);
        PrintLine p5 = new TextPrintLine("Simply sign up today for a Membership
Card!", PrintLine.CENTER);
```

```

        printerManager.addPrintLine(p5);
        bitmap = BitmapFactory.decodeResource(context.getResources(),
R.drawable.barcode_12345);
        printerManager.addPrintLine(new BitmapPrintLine(bitmap,
PrintLine.CENTER));
        printerManager.addPrintLine(new TextPrintLine(" ", 0, 100));
        POIPrinterManager.IPrinterListener listener = new
POIPrinterManager.IPrinterListener() {
            @Override
            public void onStart() {}

            @Override
            public void onFinish() {
                printerManager.close();
            }

            @Override
            public void onError(int code, String msg) {
                Log.e("POIPrinterManager", "code: " + code + "msg: " + msg);
                printerManager.close();
            }
        };
        if(state == 4){
            printerManager.close();
            return;
        }
        printerManager.beginPrint(listener);
    }

    private static List<TextPrintLine> printList(String leftStr, String
centerStr, String rightStr, int size, boolean bold){
        TextPrintLine textPrintLine1 = new TextPrintLine(leftStr,
PrintLine.LEFT, size, bold);
        TextPrintLine textPrintLine2 = new
TextPrintLine(centerStr,PrintLine.CENTER, size, bold);
        TextPrintLine textPrintLine3 = new TextPrintLine(rightStr,
PrintLine.RIGHT, size, bold);
        List<TextPrintLine> list = new ArrayList<>();
        list.add(textPrintLine1);
        list.add(textPrintLine2);
        list.add(textPrintLine3);
        return list;
    }
}

```

open

Open printer

```

void open(){...}

printerManager.open();

```

close

Close printer

```
void close(){...}  
  
printerManager.close();
```

getPrinterState

Get printer state

```
int getPrinterState(){...}  
  
public static final int STATUS_IDLE      = 0;  
public static final int STATUS_PRINTING  = 1;  
public static final int STATUS_OVERHEAT  = 2;  
public static final int STATUS_NO_PAPER  = 3;  
public static final int STATUS_NO_PRINTER = 4;  
  
int state = printerManager.getPrinterState();  
Log.d(TAG, "printer state = " + state); // printer state = 4
```

Return	Type	Value	Constant	Description
state	int	0	STATUS_IDLE	idle
		1	STATUS_PRINTING	printing
		2	STATUS_OVERHEAT	over heat
		3	STATUS_NO_PAPER	no paper
		4	STATUS_NO_PRINTER	no printer

setPrintGray

Set the print grayscale

```
void setPrintGray(int gray){...}  
  
printerManager.setPrintGray(2000);
```

Parameter	Type	Default	Description
gray	int	1200	The value smaller the lighter, the larger the thicker gray <= 0, not effective get value: gray & 0xffff It is recommended to adjust in units of 200

setPrintFont

Set the print font

```
void setPrintFont(String path){...}  
  
printerManager.setPrintFont("/system/fonts/DroidSansMono.ttf");
```

Parameter	Type	Description
path	String	System font path "/system/fonts/....."

setLineSpace

Set line space

```
void setLineSpace(int line){...}

printerManager.setLineSpace(5);
```

Parameter	Type	Description
line	int	line space Reference <code>textView.setLineSpacing(line, 1.0f);</code>

cleanCache

clean cache. It is used for multiple [beginPrint](#) between [open](#) and [close](#)

```
void cleanCache(){...}

printerManager.cleanCache();
```

addPrintLine

add print content

```
void addPrintLine(PrintLine line){...}
void addPrintLine(List<TextPrintLine> line){...}

printerManager.addPrintLine(new TextPrintLine("123", PrintLine.CENTER, 50));
```

Parameter	Type	Description
line	PrintLine <code>List<TextPrintLine></code>	TextPrintLine BitmapPrintLine <code>List<TextPrintLine></code>

class PrintLine

```

public class PrintLine {
    public static final int TEXT    = 0;
    public static final int BITMAP  = 1;

    public static final int LEFT    = 0;
    public static final int CENTER  = 1;
    public static final int RIGHT   = 2;

    protected int type;
    protected int position;
}

```

class TextPrintLine

```

public class TextPrintLine extends PrintLine {
    public static final int FONT_SMALL  = 16;
    public static final int FONT_NORMAL = 24;
    public static final int FONT_LARGE  = 36;

    public TextPrintLine() {...}
    public TextPrintLine(String content) {...}
    public TextPrintLine(String content, int position) {...}
    public TextPrintLine(String content, int position, int size) {...}
    public TextPrintLine(String content, int position, int size, boolean bold)
    {...}
    public TextPrintLine(String content, int position, int size, boolean bold,
    boolean italic) {...}
    public TextPrintLine(String content, int position, int size, boolean bold,
    boolean italic, boolean invert) {...}
}

new TextPrintLine("1234567890", PrintLine.CENTER, 24, false, false, true);

```

Attribute	Type	Value	Default	Description
type	int	PrintLine.TEXT	-/-	-/-
content	String	-/-	-/-	print content
position	int	PrintLine.LEFT PrintLine.RIGHT PrintLine.CENTER	PrintLine.LEFT	Align left Align right Align center
size	int	-/-	TextPrintLine.FONT_NORMAL	font size
bold	boolean	true false	false	Whether the printed text is bold
italic	boolean	true false	false	Whether the printed text is italicized
invert	boolean	true false	false	Black characters on white background White characters on black background

class BitmapPrintLine

```

public class BitmapPrintLine extends PrintLine {
    private Bitmap bitmap;

    public BitmapPrintLine() {...}
    public BitmapPrintLine(Bitmap bitmap) {...}
    public BitmapPrintLine(Bitmap bitmap, int position) {...}

    public Bitmap getBitmap() {...}
    public void setBitmap(Bitmap bitmap) {...}
}

// drawable/test.jpg
Bitmap bitmap = BitmapFactory.decodeResource(context.getResources(),
R.drawable.test);
printerManager.addPrintLine(new BitmapPrintLine(bitmap, PrintLine.CENTER));

```

Attribute	Type	Value	Default	Description
type	int	PrintLine.BITMAP	-/-	-/-
bitmap	Bitmap	-/-	null	-/-
position	int	PrintLine.LEFT PrintLine.RIGHT PrintLine.CENTER	PrintLine.LEFT	Align left Align right Align center

beginPrint

start to print

```
void beginPrint(IPrinterListener printerListener){...}  
  
printerManager.beginPrint(listener);
```

Parameter	Type	Description
printerListener	IPrinterListener	print listener

interface IPrinterListener

```
public interface IPrinterListener {  
    void onStart();  
  
    void onFinish();  
  
    void onError(int code, String msg);  
}  
  
public static final int ERROR_INIT      = -1;  
public static final int ERROR_PRINT     = -2;  
public static final int ERROR_OVERHEAT = -3;  
public static final int ERROR_NO_PAPER = -4;  
public static final int ERROR_OTHER    = -999;  
  
POIPrinterManager.IPrinterListener listener = new  
POIPrinterManager.IPrinterListener() {  
    @Override  
    public void onStart() {}  
  
    @Override  
    public void onFinish() {  
        printerManager.close();  
    }  
  
    @Override  
    public void onError(int code, String msg) {  
        Log.e("POIPrinterManager", "code: " + code + "msg: " + msg);  
        printerManager.close();  
    }  
};
```

Interface	Description
onStart	Called when printing starts
onFinish	Called at the end of printing
onError	Called to indicate an error

Parameter	Type	Value	Decription
code	int	POIPrinterManager.ERROR_INIT = -1 POIPrinterManager.ERROR_PRINT = -2 POIPrinterManager.ERROR_OVERHEAT = -3 POIPrinterManager.ERROR_NO_PAPER = -4 POIPrinterManager.ERROR_OTHER = -999	error code
msg	String	Printer init failed. In the case of a lock screen, printing is not support. Printer status failed. Printer bitmap failed. Printer overheat. Printer no paper. Printer other error.	ERROR_INIT ERROR_PRINT ERROR_PRINT ERROR_PRINT ERROR_OVERHEAT ERROR_NO_PAPER ERROR_OTHER

lineWrap

Print n blank lines at the beginning

```
void lineWrap(int value){...}

printerManager.lineWrap(3);
```

Parameter	Type	Description
value	int	n lines, fixed length

getPrinterLength

Total length of printed data

```
double getPrinterLength(){...}

double len = printerManager.getPrinterLength();
```

Return	Type	Description
len	double	Total length printed between open and close

getBeforePrinterLength

Total length of data printed before this print

```
double getBeforePrinterLength(){...}

double lenb = printerManager.getBeforePrinterLength();
```


Return	Type	Description
len	double	Current print length: <code>len - lenb</code> Except for the first print First print: <code>len == lenb</code>