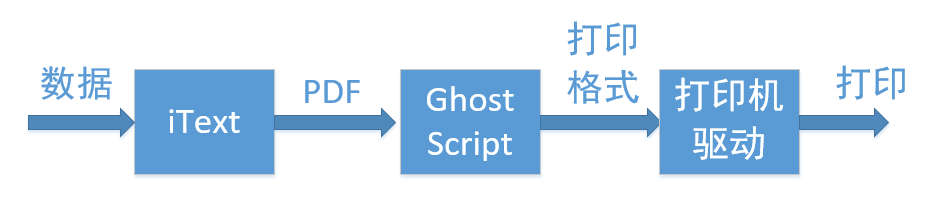
Android printer指导

2017.3.21周星宇

# 1.USB打印方案总览

USB打印采取如下流程

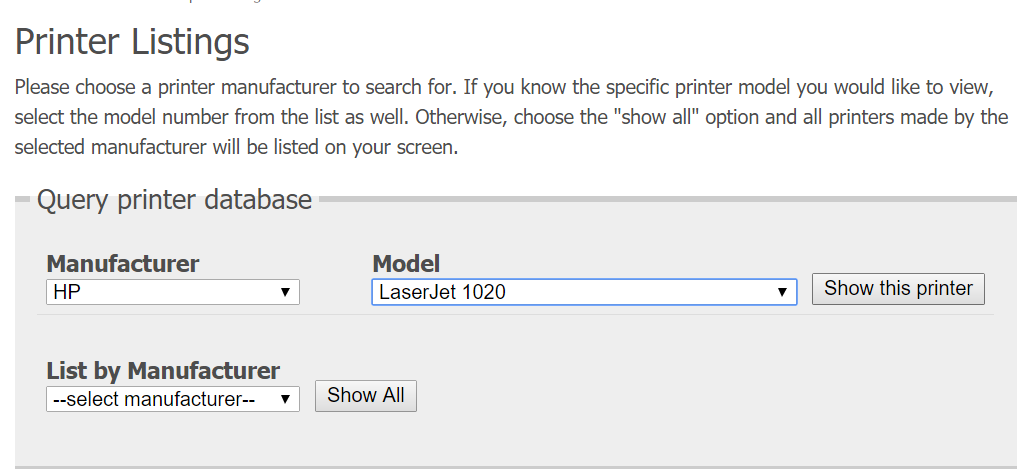


条件：

\*设备具有root权限

在已知打印机型号（如 HP laserjet 1020）的基础上，采用如下过程搜索并使用打印机驱动。

1. 在<http://www.openprinting.org/printers>（OpenPrinting）网站上查找相应打印机型号所使用的的驱动。下图搜索的型号为HP laserjet 1020



如果找到驱动就会进入以下的页面

1. 找到相应驱动的官网，查找编译和使用的方法。（具体驱动程序每个品牌的打印机都不同）。

Foo2zjs驱动编译方法请参考<>

3.交叉编译并安装GhostScript。学习GhostScript命令行参数的含义。参考：

gs静态编译：<http://blog.csdn.net/maohuiabc/article/details/6861692>

使用HPijs驱动打印机方案： Android系统USB打印(HPijs).doc

使用Foo2zjs驱动打印机方案：foo2zjs资料/ foo2zjs的静态编译与驱动LJ1020的方法.doc

4.学习iText PDF生成工具的使用方法。参考：

快速上手：docs/iText中文教程.pdf

官网iText例程：<http://developers.itextpdf.com/content/itext-5-examples>

官网FAQ：<http://developers.itextpdf.com/content/best-itext-questions-stackoverview/tables>

5.在android应用程序中运行shell语句来调用Ghostscript实现APP对打印机的驱动。

使用如下代码可以运行shell语句：（具体参考示例代码TestPrinter）

Process process = null;

DataOutputStream os = null;

try {

process = Runtime.getRuntime().exec("su"); //使用root权限

os = new DataOutputStream(process.getOutputStream());

os.writeBytes("你要执行的命令\n");

os.flush();

process.waitFor();

} catch (Exception e) {

return false;

} finally {

try {

if (os != null) {

os.close();

}

process.destroy();

} catch (Exception e) {

}

}

# 2.GhostScript使用说明

## 2.1 使用说明

在交叉编译完成GhostScript后，将生成的gs文件拷贝到开发板中/system/bin/目录下。之后在终端中输入 gs –h系统输出GhostScript相关信息在输出信息的最下面几行找到如下几行。GhostScript运行时在这些路径中查找初始化文件和字体等的资源文件。

Search path:

. : /usr/local/share/ghostscript/8.15/lib :

/usr/local/share/ghostscript/8.15/Resource :

/usr/local/share/ghostscript/fonts

运行GhostScript时应确保这些资源文件已经安装到位。

在确保安装无误之后即可在命令行运行GhostScript。

首先我们可以通过一个最简单的应用来了解GhostScript使用方法。

**gs -sDEVICE=epson -sOutputFile=ABC.xyz myfile.ps**

此命令指定设备模型为epson,输入文件为myfile.ps,输出文件为ABC.xyz

另外，gs命令和输入文件之间还可加入一些命令参数选项。（具体请参考GhostScript官网

<https://ghostscript.com/doc/9.21/Use.htm> ）

其中常用的命令行参数如下：

-sOutputFile：指定输出文件目录

-r<*number1>*x<*number2>* 指定分辨率

-dBATCH 当处理完所有文件之后退出GhostScript

-dNOPAUSE 每一页装换之间没有停顿

-sDEVICE 设定输出文件

-dFirstPage 第一页

-dLastPage 最后一页

-sPAPERSIZE 纸张大小

-g<number1>x<number2> 指定图片像素（一般不指定，使用默认值）

-q 忽略部分命令行输出信息（报错、警告信息）

使用命令行参数方法可参考如下实例：

gs -q -dBATCH -dSAFER -dNOPAUSE -sPAPERSIZE=a4 -r1200x600 -sDEVICE=pbmraw -sOutputFile=chinese.pbm chinese.pdf

在android系统中需要使用java输出流的形式执行shell脚本来调用GhostScript相关功能例如：

Process process = null;

DataOutputStream os = null;

try {

process = Runtime.getRuntime().exec("su"); //使用root权限

os = new DataOutputStream(process.getOutputStream());

os.writeBytes("gs -q -dBATCH -dSAFER -dNOPAUSE -sPAPERSIZE=a4 -r1200x600 -sDEVICE=pbmraw -sOutputFile=chinese.pbm chinese.pdf \n");

os.flush();

process.waitFor();

} catch (Exception e) {

return false;

} finally {

try {

if (os != null) {

os.close();

}

process.destroy();

} catch (Exception e) {

}

}

此段程序在android应用程序中使用java输出流的形式执行上面提到的GhostScript命令。

## 2.2 支持的设备

GhostScript支持一些内置驱动和可作为输出设备直接被GhostScript调用的驱动。

1. ijs驱动：支持喷墨打印机和一些其他的光栅设备
2. Rinkj驱动：华丽的喷墨打印机驱动程序
3. 以文件作为输出设备，以供第三方打印机驱动使用。

使用gs –h可查看GhostScript具体支持型号。如

*# gs -h*

*Input formats: PostScript PostScriptLevel1 PostScriptLevel2 PostScriptLevel3 PDF*

*Default output device: bmpmono*

*Available devices:*

*bbox bit bitcmyk bitrgb bj10e bj200 bjc600 bjc800 bmp16 bmp16m bmp256*

*bmp32b bmpgray bmpmono bmpsep1 bmpsep8 cdeskjet cdj550 cdjcolor cdjmono*

*cljet5 cljet5c deskjet devicen djet500 epswrite faxg3 faxg32d faxg4 ijs*

*jpeg jpeggray laserjet lj5gray lj5mono ljet2p ljet3 ljet3d ljet4 ljet4d*

*ljetplus nullpage pbm pbmraw pcx16 pcx24b pcx256 pcxcmyk pcxgray pcxmono*

*pdfwrite pgm pgmraw pgnm pgnmraw pj pjxl pjxl300 pkm pkmraw pksm pksmraw*

*pnm pnmraw ppm ppmraw psdcmyk psdrgb psgray psmono psrgb pswrite pxlcolor*

*pxlmono spotcmyk tiff12nc tiff24nc tiffcrle tiffg3 tiffg32d tiffg4*

*tifflzw tiffpack uniprint xcf*

GhostScript支持的输出设备可以是文件（如*pbmraw*）也可是驱动程序（如*bj200*）。

参考 <https://ghostscript.com/doc/9.21/Deprecated.htm>

<https://ghostscript.com/doc/9.21/Devices.htm>

GhostScript也可使用第三方驱动程序，如foo2zjs

具体方法见2.2节

## 2.3 字体支持

GhostScript运行时以如下流程在系统中找到它需要的字体

1. GhostScrip在lib/ Fontmap.GS文件中查找字体对应。例如，Fontmap.GS中如下语句将URWBookmanL-DemiBold字体对应到b018015l.pfb文件。

*/URWBookmanL-DemiBold (b018015l.pfb) ;*

1. GhostScript在字体路径这种查找对应的pfb文件。GhostScript查找的字体路径可以使用gs –h命令查看。例如：

/usr/local/share/ghostscript/fonts

（参考<https://ghostscript.com/doc/9.21/Fonts.htm> ）

刚刚编译好的GhostScript不包含字体文件，我们需要把字体文件从编译GhostScript的电脑中拷贝到GhostScript字体目录（例如：我使用的Ubuntu x64系统）。

我的Ubuntu中字体对应路径为：

/usr/share/fonts/type1/gsfonts/

安装好字体之后即可打印中文

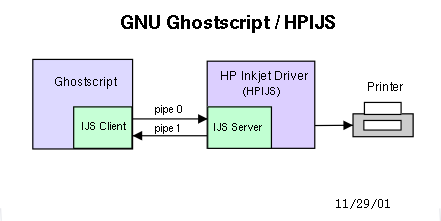
# 3几种通用打印机驱动

## 3.1 hpijs

此驱动支持惠普大部分打印机型号,主要支持较老型号的喷墨打印机。（具体支持型号见 打印机驱动支持.xlsx）

Hpijs驱动可以作为linux系统的服务端以进程间通信的方式和GhostScript进行数据交换。如下图所示，ijs服务端和客户端使用linux管道机制进行通信。

具体请参考官网<http://hplipopensource.com/node/125>

****

我们可以通过一个示例来学习hpijs的具体调用方法。

使用实例（选自GhostScript官网）

gs –dSAFER -sDEVICE=ijs -sIjsServer=hpijs -sDeviceManufacturer=HEWLETT-PACKARD -sDeviceModel='DESKJET 990' -dIjsUseOutputFD -sOutputFile=/dev/usb/lp0 -dNOPAUSE -- examples/tiger.ps

其中：

1) -sDeviceModel：根据IEEE-1284的定义选择打印设备模型。如果型号中存在空格，应该加上引号。具体参考<hpijs\_readme.html>

2) -dIjsUseOutputFD：传递输出设备的文件描述符

3) –r<number1>x<number2>：以dpi为单位的屏幕分辨率。通常

4) -sDEVICE=ijs

5) -sIjsServer=hpijs 所使用的ijs服务器（此处是打印机驱动程序hpijs）。GhostScript会产生一个新线程使用ijs协议和此设备进行通信。

6) -sDeviceManufacturer=HEWLETT-PACKARD

7) -sOutputFile=/dev/usb/lp0 指定ijs数据流输出到的设备节点，此选项不能更改

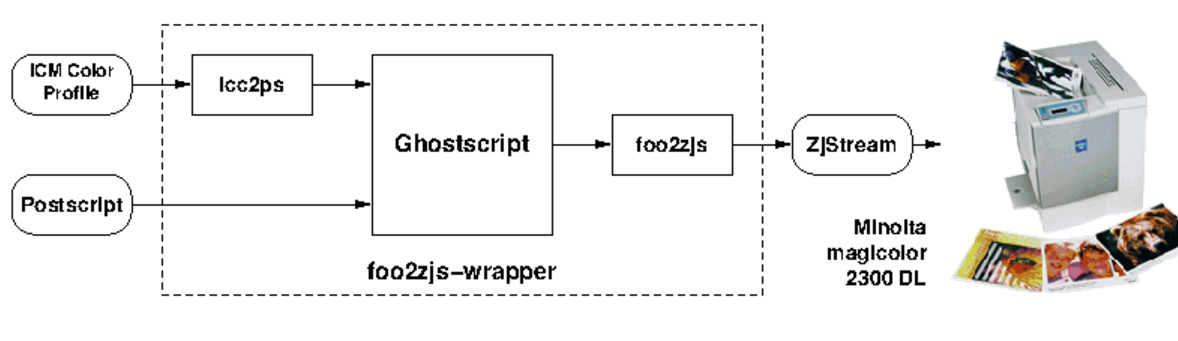
（参考<https://ghostscript.com/doc/9.21/Devices.htm#IJS> ）

## 3.2 foo2系列驱动

Foo2系列支持多种打印机输出协议格式，每种协议对应一个驱动文件，如foo2zjs/foo2xqx/foo2hp…….

以foo2zjs为例进行介绍:foo2zjs将GhostScript生成的pbm文件转化成符合ZJStream协议字节流，从而驱动打印机进行打印。（如下图所示）

具体参考官网<http://foo2zjs.rkkda.com/>



使用foo2系列驱动需要准备如下工具：

1. GhostScript
2. foo2系列对应驱动，对于具体打印机请参考openPrinting官网
3. 使用foo2中arm2hpdl打包生成的\*\*\*\*.dl驱动文件（如sihp1020.dl）

在已经编译好上述工具的情况下，以如下过程执行shell脚本即可实现打印机驱动：

1. cat sihp1020.dl > /dev/usb/lp0

此步骤使用cat命令将打印机驱动（此处使用hp laserjet 1020驱动作为示例）写入到/dev/usb/lp0设备节点上。（没有这个设备节点请重新编译内核）

1. gs -q -dBATCH -dSAFER -dNOPAUSE -sPAPERSIZE=a4 -r1200x600 -sDEVICE=pbmraw -sOutputFile=chinese.pbm /mnt/sdcard1/printer/chinese.pdf

具体参数含义请参考<<https://ghostscript.com/doc/9.21/Use.htm>>

1. -r1200x600根据foo2驱动设置的分辨率，具体参考<foo2zjs.pdf>
2. -sDEVICE=pbmraw 输出数据指定为pbm格式
3. -sOutputFile=chinese.pbm 输出文件
4. foo2zjs -z1 -p9 -r1200x600 chinese.pbm > /dev/usb/lp0
5. foo2zjs不同驱动见openPrinting官网
6. -z1 打印机所使用的的设备模型，具体参考<foo2zjs.pdf>
7. –p9 使用A4纸。（!!不是所有驱动都是p9, 具体参考<foo2zjs.pdf>）
8. -r1200x600 指定分辨率，一般情况可以空着使用默认值
9. /dev/usb/lp0 打印机字节流输出设备，这个不能改

（参考<foo2zjs.pdf> ）

## 3.3 BJC\*\*

**gs -sDEVICE=bjc600 gamma.ps myfile.ps**

## 3.4 Uniprint

支持很多厂家的小部分设备型号。

调用方法：

**gs @**{MODEL}**.upp -sOutputFile=**{printable file}**MyFile.ps -c quit**

如：**gs** **@stc.upp -sOutputFile=/dev/usb/lp0 tiger.eps -c quit**

**@stc.upp：**打印机设备使用的驱动模型，GhostScript支持如下模型

（摘自GhostScript官网）[https://ghostscript.com/doc/9.21/Deprecated.htm#Uniprint](https://ghostscript.com/doc/9.21/Deprecated.htm" \l "Uniprint)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Unified Printer Parameter files distributed with Ghostscript** | | | | | |
| **Canon BJC 610 (color, rendered)** | | | | | |
|  | **bjc610a0.upp** |  | 360×360dpi |  | plain paper, high speed |
|  | **bjc610a1.upp** |  | 360×360dpi |  | plain paper |
|  | **bjc610a2.upp** |  | 360×360dpi |  | coated paper |
|  | **bjc610a3.upp** |  | 360×360dpi |  | transparency film |
|  | **bjc610a4.upp** |  | 360×360dpi |  | back print film |
|  | **bjc610a5.upp** |  | 360×360dpi |  | fabric sheet |
|  | **bjc610a6.upp** |  | 360×360dpi |  | glossy paper |
|  | **bjc610a7.upp** |  | 360×360dpi |  | high gloss film |
|  | **bjc610a8.upp** |  | 360×360dpi |  | high resolution paper |
|  |  |  |  |  |  |
|  | **bjc610b1.upp** |  | 720×720dpi |  | plain paper |
|  | **bjc610b2.upp** |  | 720×720dpi |  | coated paper |
|  | **bjc610b3.upp** |  | 720×720dpi |  | transparency film |
|  | **bjc610b4.upp** |  | 720×720dpi |  | back print film |
|  | **bjc610b6.upp** |  | 720×720dpi |  | glossy paper |
|  | **bjc610b7.upp** |  | 720×720dpi |  | high-gloss paper |
|  | **bjc610b8.upp** |  | 720×720dpi |  | high resolution paper |
|  |  |  |  |  |  |
| **HP Ink-Printers** | | | | | |
|  | **cdj550.upp** |  | 300×300dpi |  | 32-bit CMYK |
|  | **cdj690.upp** |  | 300×300dpi |  | Normal mode |
|  | **cdj690ec.upp** |  | 300×300dpi |  | Economy mode |
|  | **dnj750c.upp** |  | 300×300dpi |  | Color -- also good for 450C |
|  | **dnj750m.upp** |  | 600×600dpi |  | Monochrome |
|  |  |  |  |  |  |
| **NEC P2X** | | | | | |
|  | **necp2x.upp** |  | 360×360dpi |  | 8-bit (Floyd-Steinberg) |
|  |  |  |  |  |  |
| **Any Epson Stylus Color** | | | | | |
|  | **stcany.upp** |  | 360×360dpi |  | 4-bit, PostScript halftoning |
|  | **stcany\_h.upp** |  | 720×720dpi |  | 4-bit, PostScript halftoning |
|  |  |  |  |  |  |
| **Original Epson Stylus and Stylus Pro Color** | | | | | |
|  | **stc.upp** |  | 360×360dpi |  | 32-bit CMYK, 15-pin |
|  | **stc\_l.upp** |  | 360×360dpi |  | 4-bit, PostScript halftoning, weaved noWeave |
|  | **stc\_h.upp** |  | 720×720dpi |  | 32-bit CMYK, 15-pin Weave |
|  |  |  |  |  |  |
| **Epson Stylus Color II** | | | | | |
|  | **stc2.upp** |  | 360×360dpi |  | 32-bit CMYK, 20-pin, Epson Stylus Color II(s) |
|  | **stc2\_h.upp** |  | 720×720dpi |  | 32-bit CMYK, 20-pin, Epson Stylus Color II |
|  | **stc2s\_h.upp** |  | 720×720dpi |  | 32-bit CMYK, 20-pin, Epson Stylus Color IIs |
|  |  |  |  |  |  |
| **Epson Stylus Color 200** | | | | | |
|  | **stc200.upp** |  | 360×720dpi |  | Plain Paper |
|  |  |  |  |  |  |
| **Epson Stylus Color 300** | | | | | |
|  | **stc300.upp** |  | 360×360dpi |  | 32-bit CMYK, plain paper |
|  | **stc300bl.upp** |  | 180×180dpi |  | black only, plain paper |
|  | **stc300bm.upp** |  | 360×360dpi |  | black only, plain paper |
|  |  |  |  |  |  |
| **Epson Stylus Color 500 (good transfer curves for plain paper)** | | | | | |
|  | **stc500p.upp** |  | 360×360dpi |  | 32-bit CMYK, noWeave, plain paper |
|  | **stc500ph.upp** |  | 720×720dpi |  | 32-bit CMYK, noWeave, plain paper |
|  |  |  |  |  |  |
| **Epson Stylus Color 600, 32/90-inch weaving** | | | | | |
|  | **stc600pl.upp** |  | 360×360dpi |  | 32-bit CMYK, 32-pin, plain paper |
|  | **stc600p.upp** |  | 720×720dpi |  | 32-bit CMYK, 32-pin, plain paper |
|  | **stc600ih.upp** |  | 1440×720dpi |  | 32-bit CMYK, 30-pin, inkjet paper |
|  |  |  |  |  |  |
| **Epson Stylus Color 640** | | | | | |
|  | **stc640p.upp** |  | 720×720dpi |  | plain paper? |
|  | **st640p.upp** |  | 720×720dpi |  | CMYK, plain paper |
|  | **st640pg.upp** |  | 720×720dpi |  | grayscale, plain paper |
|  | **st640pl.upp** |  | 360×360dpi |  | CMYK, plain paper |
|  | **st640plg.upp** |  | 360×360dpi |  | grayscale, plain paper |
|  | **st640ih.upp** |  | 1440×720dpi |  | CMYK, inkjet paper |
|  | **st640ihg.upp** |  | 1440×720dpi |  | grayscale, inkjet paper |
|  |  |  |  |  |  |
| **Epson Stylus Color 800, 64/180-inch weaving** | | | | | |
|  | **stc800pl.upp** |  | 360×360dpi |  | 32-bit CMYK, 64-pin, plain paper |
|  | **stc800p.upp** |  | 720×720dpi |  | 32-bit CMYK, 64-pin, plain paper |
|  | **stc800ih.upp** |  | 1440×720dpi |  | 32-bit CMYK, 62-pin, inkjet paper |
|  | **stc1520.upp** |  | 1440×720dpi |  | 32-bit CMYK, 62-pin, inkjet paper |
|  |  |  |  |  |  |
| **Sun raster file** | | | | | |
|  | **ras1.upp** |  | 1-bit |  | monochrome (Ghostscript) |
|  | **ras3.upp** |  | 3-bit |  | RGB (Ghostscript) |
|  | **ras4.upp** |  | 4-bit |  | CMYK (Ghostscript) |
|  | **ras8m.upp** |  | 8-bit |  | grayscale (Floyd-Steinberg) |
|  | **ras24.upp** |  | 24-bit |  | RGB (Floyd-Steinberg) |
|  | **ras32.upp** |  | 32-bit |  | CMYK (CMYK-Floyd-Steinberg) |

# 4.Itext使用指南

## 4.1 itext使用

将itext以jar包的形式包含到java或android工程中即可使用。

Android studio中导入jar包方法参考

<http://jingyan.baidu.com/article/e6c8503c7190b7e54f1a1893.html>

个人感觉使用itext简单方便的方法是在itext官方例程上进行修改，从而插入自己的PDF内容。

快速上手：docs/iText中文教程.pdf

官网iText例程：<http://developers.itextpdf.com/content/itext-5-examples>

官网FAQ：<http://developers.itextpdf.com/content/best-itext-questions-stackoverview/tables>

## 4.2 输出中文

itext中加入中文有两种方法：

1. 使用itext-assian.jar

itext-assian.jar是itext官方提供的亚洲字体支持包。使用此支持包需要

1. 将itext-assian.jar和itext.jar一同包含入java/android工程。注意：itext-assian.jar包中CMAP的路径要和itext.jar包中字体CMAP路径一致，否则会报错提示字体未找到。
2. 导入itext-assian.jar成功之后使用如下代码加载中文字体。

BaseFont bfChinese = BaseFont.createFont("STSong-Light", "UniGB-UCS2-H",

            BaseFont.NOT\_EMBEDDED);

Font FontChinese = **new** Font(bfChinese, 12, Font.NORMAL);

注意！！使用itext-assian.jar生成的PDF在做为GhostScript输入文件时会报错：找不到资源。所以建议使用方法2

2.使用字库

1)在windows电脑C：/windows/Fonts/中选择字体（最好选择比较常用的字体，不要选择微软字体）

2)将字体放入android平台的SD卡中（如/storage/sdcard0/printer\_tools/fonts）

3)在itext中调用字体文件。例：黑体字体文件 black.ttf

BaseFont bfChinese = BaseFont.createFont(PrinterParam.ToolDir+"black.ttf",BaseFont.IDENTITY\_H,BaseFont.NOT\_EMBEDDED);

Font FontChinese = new Font(bfChinese, 30, Font.NORMAL);