The Printing HOWTO

Grant Taylor

<gtaylor+pht@picante.com>

Dirk Allaert

<dirk.allaert@reachout.be>

v1.8, 2003−06−22

This is the Printing HOWTO, a collection of information on how to generate, preview, print and fax anything under GNU/Linux. Almost everything applies equally well to free software users using other Unix−like operating systems.

Table of Contents

Table of Contents

Table of Contents

Table of Contents

1. Introduction

The Printing HOWTO should contain everything you need to know to help you set up printing services on your GNU/Linux box(en). As life would have it, it's a bit more complicated than in the point−and−click world of Microsoft and Apple, but it's also a bit more flexible and certainly easier to administer for large LANs.

This document is structured so that most people will only need to read the first half or so. Most of the more obscure and situation−dependent information in here is in the last half, and can be easily located in the Table of Contents, whereas some information through section 10 or 11 is probably needed by most people.

If you find this document or the linuxprinting.org website useful, consider buying something (ink, for example) through the referral links on the site; such purchases support this effort.

The linuxprinting.org website is a good place to find the latest version; it is also, of course, distributed from tldp.org and your friendly local LDP mirror.

1.1. Terminology

I try to use consistent terminology throughout this document, so that users of all free Unix−like systems, and even users of non−Unix−like free software, can benefit. Unfortunately, there are many handy ambiguous names and many awkward unambiguous names, so just to be clear, here's a quick glossary of what each name means:

Unix	Unix is an operating system constructed at Bell Labs by various researchers. A variety of operating systems, mostly commercial, are based on this code and are also included in the name Unix.
*Un*x*	Un*x is an awkward word used to refer to every Unix−like operating system. A Unixlike operating system provides something similar to a POSIX programming interface as its native API. GNU/Linux, FreeBSD, Solaris, AIX, and even special−purpose systems like Lynx and QNX are all Un*x.
Linux	Linux is a Unixlike kernel and a small assortment of peripheral software written by Linus Torvalds and hundreds of other programmers. It forms the foundation of the most widely used Un*x operating system.
GNU	The GNU (GNU's Not Unix) project is a longtime development effort to produce an entirely free Unixlike operating system. The GNU Project is in many ways the father of most modern free software efforts.
GNU/Linux	A GNU/Linux operating system is a complete system comprised of the Linux kernel, its peripheral programs, and the GNU runtime environment of libraries, utilities, end−user software, etc. Red Hat, Debian, Caldera, SuSE, TurboLinux, and similar companies are all commercial vendors of complete GNU/Linux systems.

1.2. History

This have been severel generations of the Printing HOWTO. The history of the PHT may be chronicled thusly:

1. Grant Taylor wrote the printing HOWTO in 1992 in response to all the printing questions in comp.os.linux, and posted it. This predated the HOWTO project by a few months and was the first FAQlet called a `howto'. This edition was in plain ASCII.
2. After joining the HOWTO project, the Printing−HOWTO was merged with an Lpd FAQ by Brian McCauley `<B.A.McCauley@bham.ac.uk>`; Grant Taylor continued to co−author the PHT for two years or so. At some point he incorporated the work of Karl Auer `<Karl.Auer@anu.edu.au>`. This generation of the PHT was in TeXinfo, and available in PS, HTML, ASCII, and Info.
3. After letting the PHT rot and decay for over a year, and an unsuccessful attempt at getting someone else to maintain it, this rewrite happened. This generation of the PHT is written in SGML using the LinuxDoc DTD and the SGML−Tools−1 package. Beginning with version 3.27, it incorporated a summary of a companion printer support database; before 3.27 there was never a printer compatibility list in this HOWTO (!).
4. In mid−January, 2000, Grand found out about the PDQ print "spooler". PDQ provides a printing mechanism so much better than lpd ever did that he spent several hours playing with it, rewrote parts of this HOWTO, and bumped the version number of the document to 4.
5. In mid−2000, Grant moved his printing website to `www.linuxprinting.org`, and began offering more powerful configuration tools there. He also converted the HOWTO to DocBook, and initiated coverage of CUPS, LPRng, and GPR/libppd.
6. In early 2001, Grant began using the GNU Free Documentation License, which seems quite suitable. He also began an effort to clarify what is and isn't Linux−specific; there are several free Unixlike kernels out there, and they all use the same printing software.
7. In early 2003, after listening to a presentation from Till Kampeter at FOSDEM, I (Dirk) decided to update this HOWTO. Since Grant last edited the HOWTO, CUPS has gotten more mature and a lot more popular.

1.3. Copyright

2. Quick Start

The quickest way to get started is simply to use the setup tools provided by your vendor. Assuming that this includes support for your driver, and assuming that your vendor shipped the driver for your printer, then it should be easy to get a basic setup going this way. For information on vendor−provided setup tools, see Section 9.

If your vendor's tool doesn't work out, you should figure out if your printer is supposed to work at all. Consult the printer compatibility listings in Section 5.3.1 as well as the online version described there.

If your printer is known to work with a driver, check that you have that driver, and install if it not. Typically you will be able to find a contributed Ghostscript package including newer Ghostscript code and assorted third−party drivers. If not, you can compile it yourself; the process is not trivial, but it is well documented. See Section 10 for more information on Ghostscript.

After installing the proper driver, attempt again to configure your printer with your vendor's tools. If that fails, select a suitable third party tool from those described in Section 8. If that also fails, you'll need to construct your own setup; again see Section 8.

If you're still stuck, you've got a little troubleshooting to do. It's probably best to read most of this document first to get a feel for how things are supposed to work; then you'll be in a better position to debug.

2.1. Where to Get Help

The Usenet newsgroups comp.os.linux.hardware, comp.os.linux.setup, and comp.periphs.printers all have a share of general printing questions. These are well−trafficked newsgroups where an answer is sure to be found; check in the Google Groups archives, too. There are also the linuxprinting.foo newsgroups; these are available both as web−based forums and via NNTP; see the website.

Please also poke around the web looking for your answers. LinuxPrinting.org is an excellent place to start; other websites and projects are linked to from there.

If you need more help, please try newsgroups, mailing lists, your distribution's support line, and so forth. If do want to contact me, please do so via the discussion forums on LinuxPrinting.org; this will give others a chance to respond, and will archive your problem and any solution publicly for the next hapless user.

3. How to print

You actually use a different command to print depending on which spooling software you use.

3.1. With BSD LPD and the lpr command

If you've already got lpd setup to print to your printer, or your system administrator already did so, or your vendor did so for you, then all you need to do is learn how to use the lpr command. The Printing Usage HOWTO covers this, and a few other queue manipulation commands you should probably know. Or just read the lpr(1) man page.

In a nutshell, you specify the queue name with `−P`, and specify a filename to print a file, or nothing to print from stdin. Driver options are traditionally not controllable from lpr, but various systems accept certain options with `−o`, `−Z`, or `−J`.

Example 1. lpr

```
lpr /etc/hosts
lpr −J "my hosts file" /etc/hosts
lpr −P mylaserjet /etc/services
```

3.2. With System V LPD and the lp command

There are two sets of commands that you may encounter if you have to deal with several brands of Unix. The BSD based LPD print system (*BSD, Linux) uses lpr (print),lpq (display queue),lprm (remove jobs). System V based systems on the other hand use lp (print), lpstat (display queue), cancel (remove jobs). Solaris, SCO and others are System V Unix systems.

On SYSV systems, you can of course consult the man page of the lp command. To specify a queue you use the −d option and a filename to print a file, or nothing to print from stdin.

Example 2. lp

```
lp /etc/hosts
lp −d mylaserjet /etc/services
```

3.3. With CUPS

CUPS provides both the System V and Berkeley command−line interfaces. This means that you can use either lpr or lp to print. This is really nice if you have a bunch of scripts that already use eg. lp or you have prior experience with either a System V or a BSD flavor.

3.4. GUI Printing Tools

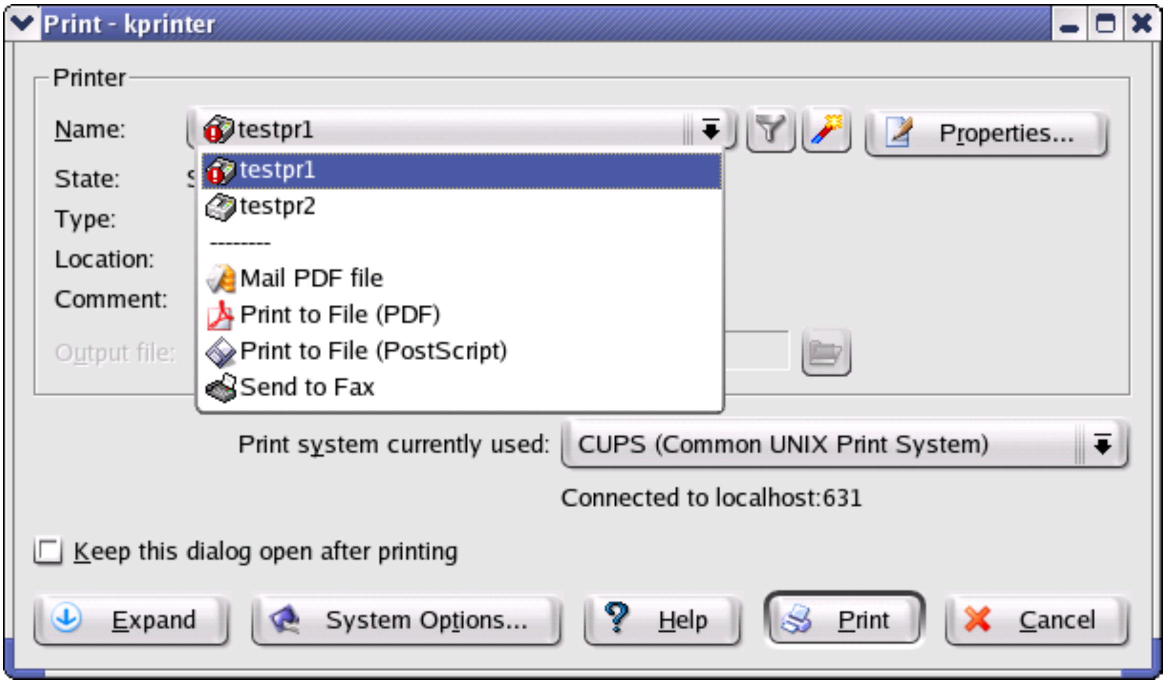
Most spooling systems alone offer only a rather basic command−line interface. Rather than use **lpr** directly, you may wish to obtain and use a front−end interface. These generally let you fiddle with various printing options (the printer, paper types, collation, n−up, etc) in an easy−to−use graphical way. Some may have other features, as well.

3.4.1. KDEPrint

KDEPrint allows users access to printing subsystems (CUPS, LPD, RLPR, LPRng etc.) through a KDE graphical user interface. With KDEPrint, you can easily print, administer jobs and printers and the printing daemon. KDEPrint is a replacement for the old QtCUPS and KUPS. It is easy to use for both developers and users. KDEPrint is already a part of KDE since 2.2.0 and has several nice features.

kprinter is the print dialog of KDEPrint which allows you to select the destination printer and change printer options. Among the destination printers, there are a few virtual printers allowing you to print to email, fax or pdf.

Figure 1. kprinter



Print - kprinter

Name: testpr1

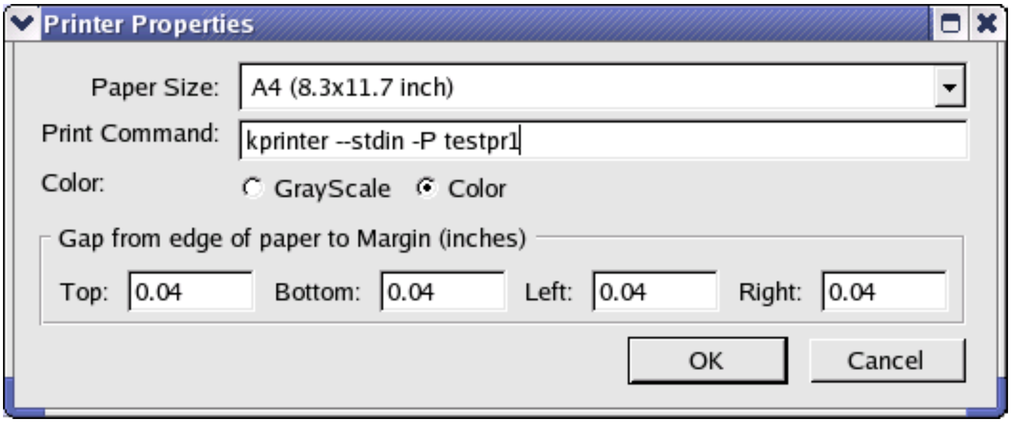
Print system currently used: CUPS (Common UNIX Print System)

Connected to localhost:631

☐ Keep this dialog open after printing

You can use KDEPrint's **kprinter** in any application that lets you configure your print command. Examples of these are Mozilla and OpenOffice.

Figure 2. Using kprinter with Mozilla

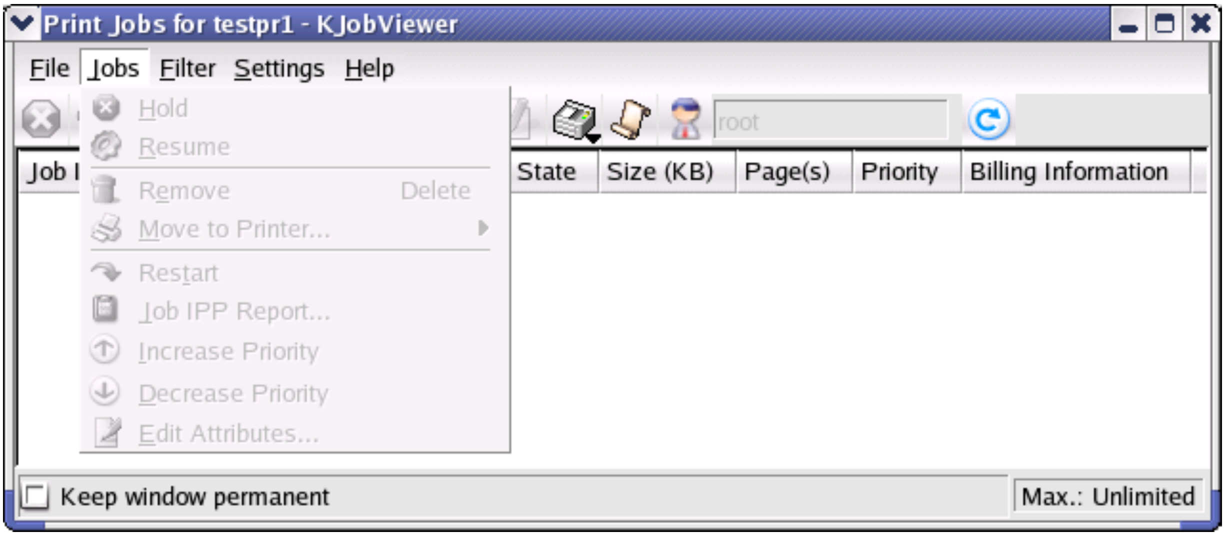


Paper Size: A4 (8.3x11.7 inch)
Print Command: kprinter --stdin -P testpr1
Color: ○ GrayScale ● Color
Gap from edge of paper to Margin (inches)
Top: 0.04 Bottom: 0.04 Left: 0.04 Right: 0.04

KDEPrint also features a Print Preview. that you can select from the Print Dialog. This is accomplished by passing the print file through the filters which make it suitable for displaying on screen using KGhostView or an external application like gv.

The KDEPrint Job Viewer **KJobViewer** allows you to view, move and cancel print jobs.

Figure 3. KJobViewer



☐ Keep window permanent

You can find more information about KDEPrint at http://printing.kde.org/.

3.4.2. XPP

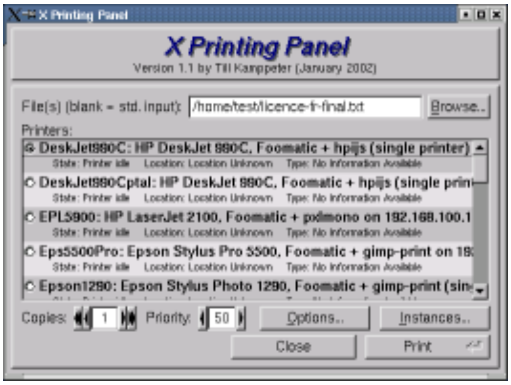
Another good choice for CUPS is the program **XPP** (see Figure 4). XPP is built from the FLTK library and is therefore desktop agnostic.

To print with XPP, simply run the xpp program, and specify a file (or nothing, if you're using xpp in place of lpr to print from stdin). Then select a printer from the list of configured printers, and select any options you'd like to apply from the various tabbed panels. See Figure 5 for an example options panel highlighting the standard CUPS options.

When used with Foomatic driver interface system, XPP will also let you control numeric parameters not normally supported by CUPS. This typically includes such things as extended color tuning, cartridge alignment, and so forth. See Figure 6 for an example of this.

You can save your selected printer and all the options with the `Save Settings' button.

Figure 4. XPP Main Window



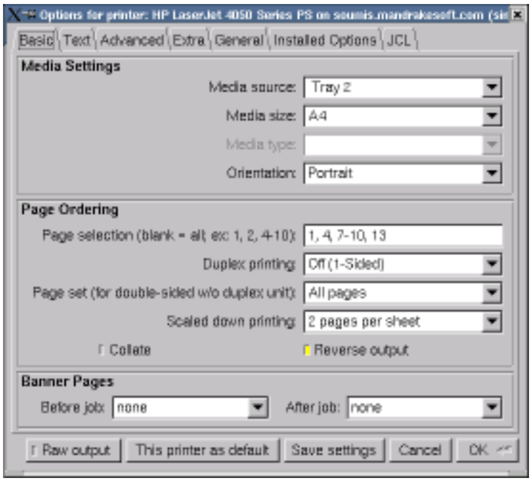
X Printing Panel
Version 1.1 by Till Kamppeter (January 2002)

File(s) (blank = std. input): /home/test/licence-fr-final.txt

Printers:
● DeskJet990C: HP DeskJet 990C, Foomatic + hpijs (single printer)
State: Printer idle Location: Location Unknown Type: No Information Available
○ DeskJet990Cptal: HP DeskJet 990C, Foomatic + hpijs (single prin
State: Printer idle Location: Location Unknown Type: No Information Available
○ EPL5900: HP LaserJet 2100, Foomatic + pxlmono on 192.168.100.1
State: Printer idle Location: Location Unknown Type: No Information Available
○ Eps5500Pro: Epson Stylus Pro 5500, Foomatic + gimp-print on 19
State: Printer idle Location: Location Unknown Type: No Information Available
○ Epson1290: Epson Stylus Photo 1290, Foomatic + gimp-print (sin

Copies: 1
Priority: 50

Figure 5. CUPS/XPP Options Window



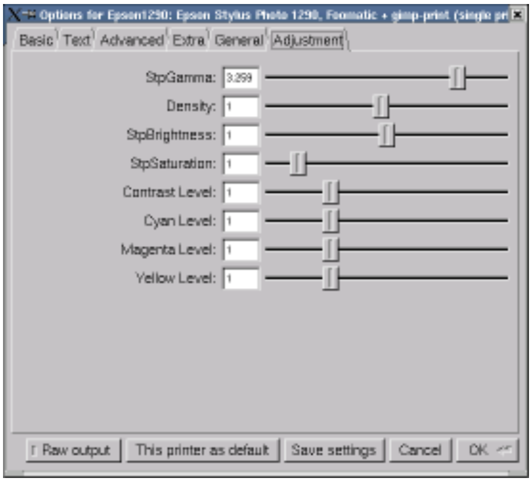
Options for printer: HP LaserJet 4050 Series PS on soumis.mandrakesoft.com (si

Media Settings
Media source: Tray 2
Media size: A4
Media type:
Orientation: Portrait

Page Ordering
Page selection (blank = all; ex: 1, 2, 4-10): 1, 4, 7-10, 13
Duplex printing: Off (1-Sided)
Page set (for double-sided w/o duplex unit): All pages
Scaled down printing: 2 pages per sheet
☐ Collate
☑ Reverse output

Banner Pages
Before job: none
After job: none

Figure 6. CUPS/XPP Foomatic Options Window



Options for Epson1290: Epson Stylus Photo 1290, Foomatic + gimp-print (single pr

StpGamma: 3.299
Density: 1
StpBrightness: 1
StpSaturation: 1
Contrast Level: 1
Cyan Level: 1
Magenta Level: 1
Yellow Level: 1

3.4.3. GPR

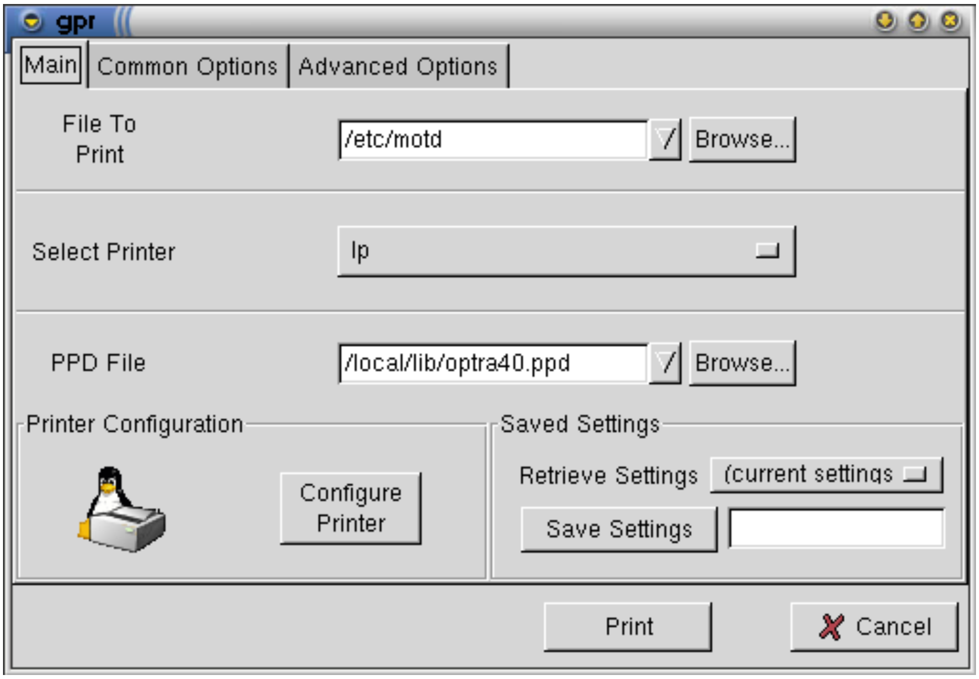
GPR, by Thomas Hubbell, uses code from CUPS to filter Postscript jobs and offer easy user control over job options. Some options (like n−way printing, page selection, etc) are implemented directly by GPR, while most others are implemented by the printer or by the spooler's filter system.

GPR works with LPD or LPRng; or can be compiled specifically for use with GNUlpr. When compiled normally, it uses VA's libppd directly to produce printer−specific PostScript which it will then submit to the lpr command. When compiled for GNUlpr, it will submit your unmodified Postscript job to the lpr command, along with the set of job options you specify. This is arguably the better route, since it allows the Postscript to be redirected to a different printer by the spooler when appropriate; unfortunately it requires GNUlpr, which is not in wide circulation (although it is of course trivial to install).

To use GPR, first select a printer (by LPD queue name) and check that GPR has loaded the proper PPD file. If it hasn't, you'll need to specify the PPD filename, and specify your printer's options in the Printer Configuration dialog (you get this dialog by pressing the Printer Configuration button; it contains assorted printer setup options defined by the PPD).

Once you've configured your printer in GPR, you can print jobs by specifying the filename and selecting the proper options from the `Common' and `Advanced' tabbed panels. The `Common' options are implemented directly by GPR for all printers, while the `Advanced' options are defined by the PPD file for your printer. You can see these option panels in Figure 8 and Figure 9.

Figure 7. GPR Main Options



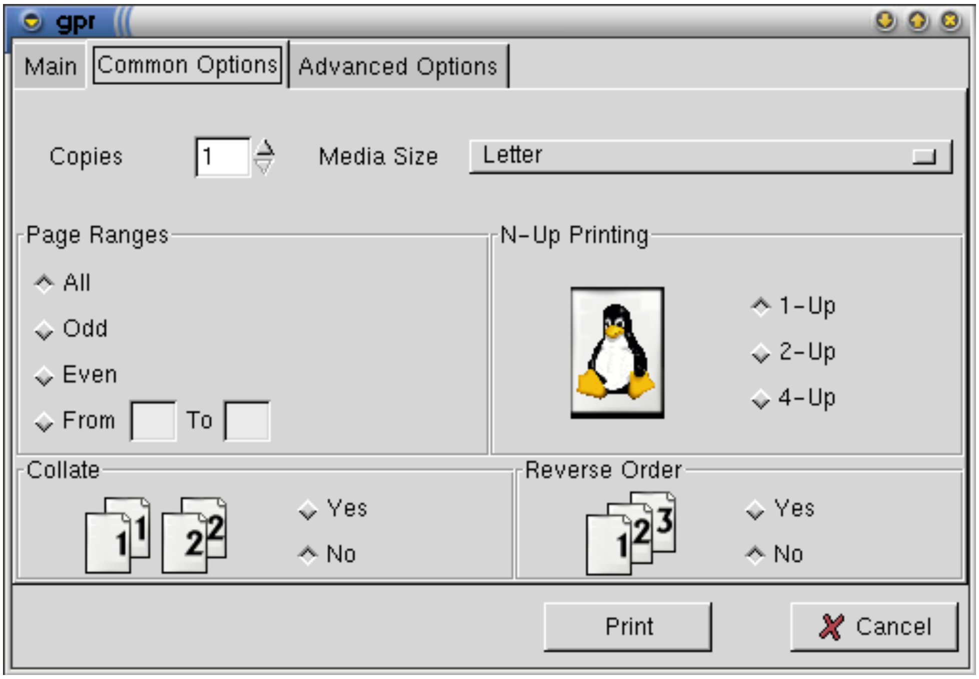
File To Print: /etc/motd

Select Printer: lp

PPD File: /local/lib/optra40.ppd

Retrieve Settings: (current settings

Figure 8. GPR Common Options



Copies: 1

Media Size: Letter

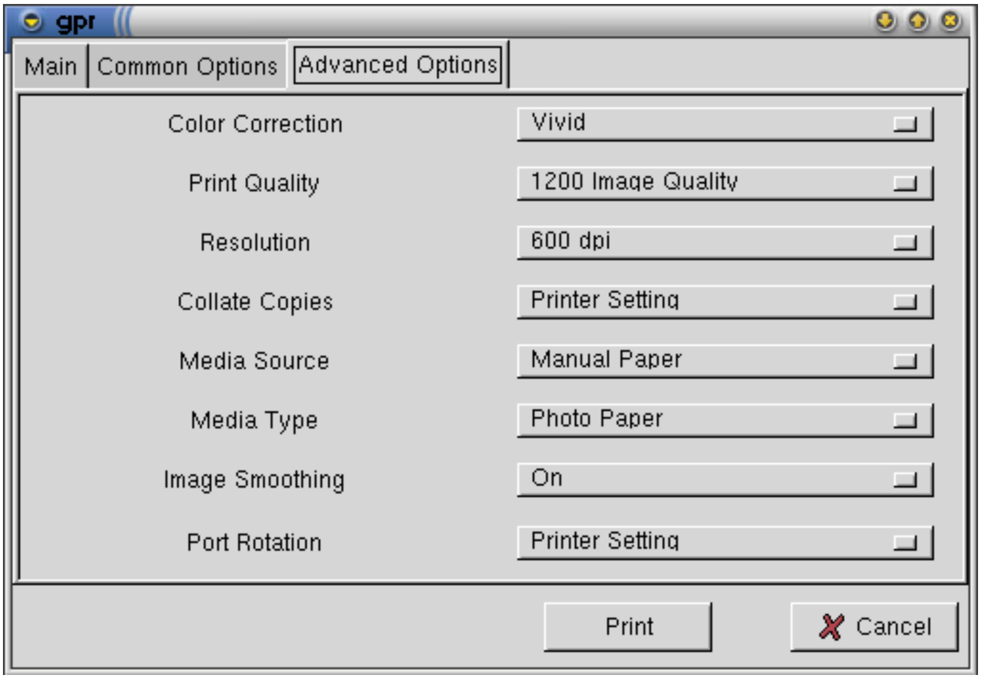
Page Ranges
● All
○ Odd
○ Even
○ From ☐ To ☐

N-Up Printing
● 1-Up
○ 2-Up
○ 4-Up

Collate
○ Yes
● No

Reverse Order
○ Yes
● No

Figure 9. GPR Printer Options



Color Correction: Vivid
Print Quality: 1200 Image Quality
Resolution: 600 dpi
Collate Copies: Printer Setting
Media Source: Manual Paper
Media Type: Photo Paper
Image Smoothing: On
Port Rotation: Printer Setting

---

4. Kernel printer devices

There are two completely different device drivers for the parallel port; which one you are using depends on your kernel version (which you can find out with the command `uname −a`). The driver changed in Linux 2.1.33; essentially all current systems will be running kernel 2.2 or later, so you'll probably want to skip ahead to the parport driver section.

A few details are the same for both styles of driver. Most notably, many people have found that Linux will not detect their parallel port unless they disable "Plug and Play" in their PC BIOS. (This is no surprise; the track record for PnP of non−PCI devices with Windows and elsewhere has been something of a disaster).

4.1. The lp device (kernels <=2.1.32)

The Linux kernel (<=2.1.32), assuming you have compiled in or loaded the lp device (the output of `cat /proc/devices` should include the device lp if it is loaded), provides one or more of */dev/lp0*,*/dev/lp1*, and */dev/lp2*. These are NOT assigned dynamically, rather, each corresponds to a specific hardware I/O address. This means that your first printer may be *lp0* or *lp1* depending on your hardware. Just try both.

A few users have reported that their bidirectional lp ports aren't detected if they use an older unidirectional printer cable. Check that you've got a decent cable.

One cannot run the plip and lp drivers at the same time on any given port (under 2.0, anyway). You can, however, have one or the other driver loaded at any given time either manually, or by kerneld with version 2.x (and later 1.3.x) kernels. By carefully setting the interrupts and such, you can supposedly run plip on one port and lp on the other. One person did so by editing the drivers; I eagerly await a success report of someone doing so with only a clever command line.

There is a little utility called `tunelp` floating about with which you, as root, can tune the Linux 2.0 lp device's interrupt usage, polling rate, and other options.

When the lp driver is built into the kernel, the kernel will accept an `lp=` option to set interrupts and io addresses:

```
When the lp driver is built in to the kernel, you may use the
LILO/LOADLIN command line to set the port addresses and interrupts
that the driver will use.

Syntax:      lp=port0[,irq0[,port1[,irq1[,port2[,irq2]]]]]

For example:   lp=0x378,0   or   lp=0x278,5,0x378,7 **

Note that if this feature is used, you must specify *all* the ports
you want considered, there are no defaults.  You can disable a
built−in driver with lp=0.
```

When loaded as a module, it is possible to specify io addresses and interrupt lines on the insmod command line (or in*/etc/conf.modules* so as to affect kerneld) using the usual module argument syntax. The parameters are`io=port0,port1,port2` and `irq=irq0,irq1,irq2`. Read the man page for*insmod* for more information on this.

**For those of you who can never find the standard port numbers when you need them, they are as in the second example above. The other port (*lp0*) is at 0x3bc. I've no idea what interrupt it usually uses.

The source code for the Linux 2.0 parallel port driver is in /usr/src/linux/drivers/char/lp.c.

4.2. The parport device (kernels >= 2.1.33)

Beginning with kernel 2.1.33 (and available as a patch for kernel 2.0.30), the lp device is merely a client of the new parport device. The addition of the parport device corrects a number of the problems that plague the old lp device driver − it can share the port with other drivers, it dynamically assigns available parallel ports to device numbers rather than enforcing a fixed correspondence between I/O addresses and port numbers, and so forth.

The advent of the parport device has enabled a whole flock of new parallel−port drivers for things like Zip drives, Backpack CD−ROMs and disks, and so forth. Some of these are also available in versions for 2.0 kernels; look around on the web.

The main difference that you will notice, so far as printing goes, is that parport−based kernels dynamically assign lp devices to parallel ports. So what was lp1 under Linux 2.0 may well be lp0 under Linux 2.2. Be sure to check this if you upgrade from an lp−driver kernel to a parport−driver kernel.

The most popular problems with this device seems to stem from misconfiguration:

The Distribution

Some GNU/Linux distributions don't ship with a properly setup /etc/modules.conf (or /etc/conf.modules), so the driver isn't loaded properly when you need it to be. With a recent modutils, the proper magical lines from modules.conf seem to be:

```
alias /dev/printers lp             # only for devfs?
alias /dev/lp*      lp             # only for devfs?
alias parport_lowlevel parport_pc  # missing in Red Hat 6.0−6.1
```

The BIOS

Many PC BIOSes will make the parallel port into a Plug−and−Play device. This just adds needless complexity to a perfectly simple device that is nearly always present; turn off the PnP setting for your parallel port ("LPT1" in many BIOSes) if your parallel port isn't detected by the Linux driver. The correct setting is often called "legacy", "ISA", or "0x378", but probably not "disabled".

You can also read the parport documentation in your kernel sources, or look at the parport web site.

4.3. Serial devices

Serial devices are usually called something like*/dev/ttyS1* under Linux. The utility `stty` will allow you to interactively view or set the settings for a serial port; `setserial` will allow you to control a few extended attributes and configure IRQs and I/O addresses for non−standard ports. Further discussion of serial ports under Linux may be found in the Serial−HOWTO.

When using a slow serial printer with flow control, you may find that some of your print jobs get truncated. This may be due to the serial port, whose default behavior is to purge any untransmitted characters from its buffer 30 seconds after the port device is closed. The buffer can hold up to 4096 characters, and if your printer uses flow control and is slow enough that it can't accept all the data from the buffer within 30 seconds after printing software has closed the serial port, the tail end of the buffer's contents will be lost. If the command `cat file > /dev/ttyS2` produces complete printouts for short files but truncated ones for longer files, you may have this condition.

The 30 second interval can be adjusted through the "closing_wait" command−line option of setserial (version 2.12 and later). A machine's serial ports are usually initialized by a call to setserial in the rc.serial boot file. The call for the printing serial port can be modified to set the closing_wait at the same time as it sets that port's other parameters.

4.4. USB Devices

4.4.1. USB 1.1

Linux supports USB pretty well. USB should work with any late−model 2.2 kernel, and any 2.4 kernel or newer. Of course you need kernel support for USB, either linked in or through a module (recommended).

If you have a modular kernel, the following modules need to be loaded:

- usb−core.o
- usb−uhci.o or uhci.o or usb−ohci.o
- printer.o

Which one of usb−uhci.o or uhci.o or usb−ohci.o you need depends on the kind of motherboard or adaptor you have. Intel and Via motherboards and Via based adaptors are UHCI (you can use either usb−uhci.o or uhci.o). You can find out which type of HCI (Host Controller Interface) you have with **lspci −v|grep HCI**

4.4.2. USB 2.0

To get high speed transfers out of a USB 2.0 capable device you must attach it to an USB 2.0 controller and use the EHCI driver (ehci−hcd.o). A recent 2.4 kernel or higher is recommended if you want to use USB 2.0.

4.4.3. Hints

One thing to remember is that USB devices are dynamically allocated. A USB printer gets assigned a device file (/dev/usb/lp*) when it is turned on or connected. This could mean that print jobs are sent to the wrong printer because you turned them on in a certain order. CUPS uses special Uri's containing manufacturer, model and printer serial number to keep sending the jobs to the correct physical printer.

Although most USB printers work fine on Linux, there are exceptions. For example the new MF devices from Epson (Stylus CX3200/CX5200) return garbage when one polls the IEEE−1284 ID string via IOCTL, for example with the code of the CUPS "usb" backend. Whereas one can poll the ID string via an Epson−proprietary method.

Till Kamppeter has written some tools to retrieve the device ID string from USB printers. getusbprinterid.pl and usb_id_test.c are the same thing but respectively in Perl and C. As mentioned above, the new MF devices from Epson are an exception, but the "Epson proprietary method" is implemented in the ttink tool of the MTink package.

More documentation about USB is available at the Linux USB Website.

5. Supported Printers

The Linux kernel will let you speak with any printer that you can plug into a serial, parallel, or usb port, plus any printer on the network. Unfortunately, this alone is insufficient; you must also be able to generate data that the printer will understand. Primary among the incompatible printers are those referred to as "Windows" or "GDI" printers. They are called this because all or part of the printer control language and the design details of the printing mechanism are not documented. Typically the vendor will provide a Windows driver and happily sell only to Windows users; this is why they are called Winprinters. In some cases the vendor also provides drivers for NT, OS/2, or other operating systems.

Many of these printers *do not work* with free software. A few of them do, and some of them only work a little bit (usually because someone has reverse engineered the details needed to write a driver). See the printer support list below for details on specific printers.

A few printers are in−between. Some of NEC's models, for example, implement a simple form of the standard printer language PCL that allows PCL−speaking software to print at up to 300dpi, but only NEC knows how to get the full 600dpi out of these printers.

Note that if you already have one of these Winprinters, there are roundabout ways to print to one, but they're rather awkward. SeeSection 12 in this document for more discussion of Windows−only printers.

5.1. Postscript

As for what printers *do* work with free software, the best choice is to buy a printer with native PostScript support *in firmware*. Nearly all Un*x software that produces printable output produces it in PostScript, so obviously it'd be nice to get a printer that supports PostScript directly. Unfortunately, PostScript support is scarce outside the laser printer domain, and is sometimes a costly add−on.

Un*x software, and the publishing industry in general, have standardized upon Postscript as the printer control language of choice. This happened for several reasons:

Timing

Postscript arrived as part of the Apple Laserwriter, a perfect companion to the Macintosh, the system largely responsible for the desktop publishing revolution of the 80s.

It's device−independent

Postscript programs can be run to generate output on a pixel screen, a vector screen, a fax machine, or almost any sort of printer mechanism, without the original program needing to be changed. Postscript output will look the same on any Postscript device, at least within the limits of the device's capabilities. Before the creation of PDF, people exchanged complex documents online as Postscript files. The only reason this standard didn't "stick" was because Windows machines didn't usually include a Postscript previewer, so Adobe specified hyperlinks and compression for Postscript, called the result PDF, distributed previewers for it, and invented a market for their "distiller" tools (the functionality of which is also provided by ghostscript's ps2pdf and pdf2ps programs).

It's a real programming language

Postscript is a complete programming language; you can write software to do most anything in it. This is mostly useful for defining subroutines at the start of your program to reproduce complex things over and over throughout your document, like a logo or a big "DRAFT" in the background. But there's no reason you couldn't compute À in a Postscript program.

It's open

> Postscript is fully specified in a publically available series of books (which you can find at any good bookstore) and also online at http://partners.adobe.com/asn/developer/technotes/postscript.html. Although Adobe invented it and provides the dominant commercial implementation, other vendors like Aladdin produce independently coded implementations as well.

5.2. Non−Postscript

Failing the (larger) budget necessary to buy a Postscript printer, you can use any printer supported by *Ghostscript*, the free Postscript interpreter used in lieu of actual printer Postscript support. Note that most GNU/Linux distributions can only ship a somewhat outdated version of Ghostscript due to the license. Fortunately, there is usually a prepackaged up to date Ghostscript made available in each distribution's contrib area.

Adobe now has a new printer language called "PrintGear". I think it's a greatly simplified binary format language with some Postscript heritage but no Postscript compatibility. And I haven't heard of Ghostscript supporting it. But some PrintGear printers seem to support another language like PCL, and these printers will work with GNU/Linux (if the PCL is implemented in the printer and not in a Windows driver).

Similarly, Adobe offers a host−based Postscript implementation called PressReady. This works much like Ghostscript does to provide Postscript support for a non−Postscript printer, but has the disadvantage that it runs only on Windows.

5.3. What printers work?

You can look in several places to see if a particular printer will work. The cooperatively maintained Printing HOWTO printer database aims to be a comprehensive listing of the state of GNU/Linux printer support. A summary of it is below; be sure to check online for more details and information on what driver(s) to use.

The best bet for new printer shoppers is to consult the list of suggested printers. These center around color inkjets and mono laser devices. You can even help support this document and the website by buying from one of affiliated vendors.

Ghostscript's printer compatibility page has a list of some working printers, as well as links to other pages.

Google groups contains hundreds of "it works" and "it doesn't work" testimonials. Try all three, and when you're done, check that your printer is present and correct in the database, so that it will be listed properly in this document in the future.

5.3.1. Printer compatibility list

This section is a summary of the online database. The online version includes device specifications, notes, driver information, user−maintained documentation, manufacturer web pages, and interface scripts for using drivers with several print spooling systems (including LPR, LPRng, PDQ, and CUPS). The online version of this list is also interactive; people can and do add printers all the time, so be sure to check it as well. Finally, if your printer isn't listed, add it!

Note that this listing is not gospel; people sometimes add incorrect information, which are eventually weeded out. Entries which have not been sanity−checked are marked with an asterisk (*). Verify from Google Groups that a printer works for someone before buying it based on this list.

Printers here are categorized into four types:

Perfectly
: Perfect printers work perfectly − you can print to the full ability of the printer, including color, full resolution, etc. In a few cases printers with undocumented "resolution enhancement" modes that don't work are listed as perfect; generally the difference in print quality is small enough that it isn't worth worrying about.

Mostly
: You can print fine, but there may be minor limitations of one sort or another in either printing or other features.

Partially
: You can print, but maybe not in color, or only at a poor resolution. See the online listing for information on the limitation.

Paperweight
: You can't print a darned thing; typically this will be due to lack of a driver and/or documentation on how to write one. Paperweights occasionally get "promoted", either when someone discovers that an existing driver works, or when someone creates a new driver, but you shouldn't count on this happening.

In all cases, since this information is provided by dozens of people, none of it is guaranteed to be correct; entries with an asterisk (*) are particularly suspect. The facts, however, should be easy to corroborate from the driver web pages and manufacturer web sites.

And without further ado, here is the printer compatibility list:

Table 1. Linux Printer Support

Manufacturer	Perfectly	Mostly	Partially	Paperv
Alps		MD−1000 MD−1300 MD−1500* MD−2000 MD−2010 MD−2300 MD−4000 MD−5000 MD−5500		
Anitech	M24			
Apollo	P−1200 P−1220 Barbie P−1250 P−2100 P−2150 P−2200 P−2250 P−2500 P−2550 P−2600 P−2650			

Apple	12/640ps Dot Matrix ImageWriter ImageWriter LQ LaserWriter 4/600* LaserWriter 16/600* LaserWriter IINTX* LaserWriter IIg LaserWriter Pro 630* LaserWriter Select 360*	Color StyleWriter 1500 Color StyleWriter 2200 Color StyleWriter 2400 Color StyleWriter 2500* ImageWriter II* LaserWriter NT StyleWriter 1200 StyleWriter I StyleWriter II		
Avery	Personal Label Printer+*	Personal Label Printer		
Brother	HL−4Ve HL−8 HL−10V HL−10h HL−630 HL−660 HL−720 HL−730 HL−760 HL−820 HL−960* HL−1020 HL−1040 HL−1070* HL−1250 HL−1260 HL−1270N HL−1440 HL−1450 HL−1470N HL−1650 HL−1660e HL−1670N HL−2060 HL−2400CeN HL−2460 HL−2460N HL−3400CN	HJ−400 HL−1030 HL−1050 HL−1060 HL−1240	DCP−1200 HJ−100i* HL−4V* HL−6* HL−6V* HL−630/631* HL−641/645/655M* HL−665* HL−730/730DX* HL−1270* HL−P2000* M−1309* M−1324* M−1809* M−1809 Color* M−1824L* M−1824L Color* M−1909* M−1909 Color* M−1924L* M−1924L Color* M−4309* MC−3000 MFC 7150C MFC−4000/4500/5500* MFC−4350 MFC−6550MC MFC−6550MC/7550MC* MFC−8300 MFC−9050 MFC−9100c* MFC−9500 MFC−9600 MFC−P2000* MFC−P2500	4550* MP−2
CItoh	M8510			
CalComp	Artisan 1023 penplotter*			
Canon	BJ−5	BJC−50*	BJ F100*	BJC−5

	BJ−10e BJ−10v* BJ−15v* BJ−20 BJ−30* BJ−35v* BJ−100 BJ−200 BJ−330 BJC−70 BJC−210 BJC−250* BJC−250ex BJC−255SP BJC−265SP BJC−600* BJC−610 BJC−620 BJC−680J* BJC−800 BJC−880J* BJC−4000 BJC−4100 BJC−4200 BJC−4300* BJC−4400* BJC−4550* GP 335* GP 405 LBP−4+ LBP−4U LBP−8A1 LBP−310* LBP−320 Pro* LBP−350* LBP−430 LBP−1000* LBP−1260 LBP−1760* LIPS−II+* LIPS−III* LIPS−IV* LIPS−IVv*	BJC−55* BJC−80* BJC−85* BJC−240* BJC−1000* BJC−2000* BJC−2010* BJC−2100 BJC−2110 BJC−3000 BJC−4310SP BJC−6000 BJC−7004* BJC−8200* LBP−4sx S100 S400* imageRunner 330s	BJ F200* BJ F600* BJ F800* BJ F6000* BJ−30v* BJ−200e* BJ−220JCII* BJ−220JSII* BJ−230* BJ−300 BJC−35v* BJC−35vII* BJC−50v* BJC−80v* BJC−210J* BJC−210SP BJC−240J* BJC−250J* BJC−400J* BJC−410J* BJC−420J* BJC−430J* BJC−430J Lite* BJC−430JD Lite* BJC−440J* BJC−455J* BJC−465J* BJC−600J* BJC−600e* BJC−610JW* BJC−620JW* BJC−700J* BJC−820* BJC−820J* BJC−4200 Photo* BJC−4304 Photo* BJC−4650* BJC−5500* BJC−5500J* BJC−6100* BJC−6200* BJC−6500* BJC−7000* BJC−7004 Photo* BJC−7100* BJC−8000* MultiPASS C2500* MultiPASS C3000* MultiPASS C3500* MultiPASS C5000*	BJC−5 BJC−8 LBP−4 LBP−6 LBP−6 LBP−8 Multip S200

			MultiPASS C5500 S300 S450* S500 S600 S630 S800 S4500*	
Citizen	ProJet II* ProJet IIc printiva600C printiva600U	printiva700 printiva1700		
Compaq		IJ750* IJ1200	IJ900	IJ300*
DEC	DECWriter 500i* DECwriter 110i* DECwriter 520ic* LA50* LA70* LA75* LA75 Plus* LJ250 LN03* LN07*	1800* LN17*		
Daewoo			DP−3630H* DP−7200H* DP−7400C*	
Dymo−CoStar	ASCII 250* ASCII+* EL40* EL60* LabelWriter II* LabelWriter XL* LabelWriter XL+* SE250* SE250+* Turbo*			
Epson	ActionLaser 1100* ActionLaser II* ActionPrinter 3250* AcuLaser C2000 AcuLaser C2000PS AcuLaser C4000 AcuLaser C4000PS AcuLaser C8500 AcuLaser C8500PS CL 700 CL 750 Dot Matrix	EPL−5700 MC 5000 MJ 520C MJ 5100C PM 790PT* PM 850PT PM 4000PX PM 5000C Stylus Color 200* Stylus Color 300* Stylus Color II Stylus Color IIs	9−pin 136 Col* 9−pin 80 Col* 24−pin 136 Col* 24−pin 80 Col* AP−800* AP−2250* AP−3250* AP−3260* AP−5000* AP−5500* ActionLaser 1000/EPL−5000* ActionLaser 1400*	AcuLa EPL−5 EPL−5 EPL−5 EPL−5 Stylus

	EM 900C EM 900CN EM 930C EM 930CN EPL−5200* EPL−5200+* EPL−5800 EPL−5800PS EPL−5900 EPL−5900PS EPL−7100 EPL−N1600 EPL−N1600PS EPL−N2050 EPL−N2050+ EPL−N2050PS EPL−N2050PS+ EPL−N2120 EPL−N2750 EPL−N2750PS L−1000* LP 8000 LP−2000* LP−3000* LP−7000* LP−7000G* LP−xx00* LQ−24 LQ−500 LQ−570+* LQ−850 LQ−2550 LX−1050* MC 7000 MJ 6000C MJ 8000C MachJet* PM 700C PM 730C* PM 750C PM 760C* PM 770C PM 780C* PM 800C PM 820C PM 880C* PM 2000C PM 2200C* PM 3000C PM 3300C	Stylus Photo 785* Stylus Photo 825* Stylus Photo 875* Stylus Photo 895* Stylus Photo 915 Stylus Photo 925 Stylus Photo 950 Stylus Photo 960 Stylus Photo 2100 Stylus Photo 2200 Stylus Pro 5000 Stylus Pro 5500 Stylus Pro XL	ActionLaser 1500/EPL−5200* ActionLaser 1600/EPL−5600* ActionLaser II/EPL−4000* DFX−5000* DFX−5000+* DFX−8000* DFX−8500* DLQ−2000 (360 dpi)* DLQ−3000+* EPL−7000/7100* EPL−8000/8100* EPL−N1200* EPL−N2000* EX−800* EX−1000* FX−286e* FX−850* FX−870* FX−880* FX−980* FX−1000* FX−1050* FX−1170* FX−1180* FX−2170* FX−2180* Generic 48 pin* Generic ESC2P* Generic ESC_P 24−J84* Generic ESC_P 24−J84C* JX−80* LQ−100* LQ−150* LQ−300* LQ−300 Color* LQ−400* LQ−510* LQ−550* LQ−570* LQ−670* LQ−850 (N9)* LQ−850+ (360 dpi)* LQ−860* LQ−870* LQ−950 (N9)* LQ−1010* LQ−1050* LQ−1050 (N9)* LQ−1050+ (360 dpi)* LQ−1060*

	PM 3500C PM 7000C SQ 1170 Stylus* Stylus 800* Stylus C20SX Stylus C20UX Stylus C40SX Stylus C40UX Stylus C41SX Stylus C41UX Stylus C42SX Stylus C42UX Stylus C60 Stylus C61 Stylus C62 Stylus C70 Stylus C80 Stylus C82 Stylus Color* Stylus Color 8 3 Stylus Color 400 Stylus Color 440* Stylus Color 460* Stylus Color 480 Stylus Color 500 Stylus Color 580 Stylus Color 600 Stylus Color 640 Stylus Color 660 Stylus Color 670* Stylus Color 680 Stylus Color 740 Stylus Color 760 Stylus Color 777 Stylus Color 800 Stylus Color 850 Stylus Color 860 Stylus Color 880 Stylus Color 900 Stylus Color 980 Stylus Color 1160 Stylus Color 1500 Stylus Color 1520 Stylus Color 3000 Stylus Color I Stylus Color PRO Stylus Photo Stylus Photo 700 Stylus Photo 720*		LQ−1070* LQ−1070+* LQ−1170* LQ−2070* LQ−2080* LQ−2170* LQ−2180* LQ−2500* LX−100* LX−300* LX−300 Color* LX−400* LX−800* LX−810* LX−850* MC 2000 MC 9000 MC 10000 MJ−500C* MJ−510C* MJ−700V2C* MJ−800C* MJ−900C* MJ−910C* MJ−3000C* MJ−3000CU* MJ−5000C* MJ−5100C* MX−80 (w_GRAFTRAX 80)* PM 950C PM 9000C PM 10000 PX 7000 PX 9000 SQ−850* SQ−2500* SQ−2550* Stylus 200* Stylus 300* Stylus 400* Stylus 500* Stylus 800+* Stylus 820* Stylus 1000* Stylus 1500* Stylus Color 600Q* Stylus Photo 2000P Stylus Pro* Stylus Pro 7600 Stylus Pro 9000

	Stylus Photo 750 Stylus Photo 780* Stylus Photo 790* Stylus Photo 810 Stylus Photo 820 Stylus Photo 830 Stylus Photo 870* Stylus Photo 890* Stylus Photo 1200 Stylus Photo 1270 Stylus Photo 1280 Stylus Photo 1290 Stylus Photo 1290S Stylus Photo EX Stylus Photo EX3* Stylus Pro 7000 Stylus Pro 7500 Stylus Scan 2000 Stylus Scan 2500		Stylus Pro 9500 Stylus Pro 9600 Stylus Pro 10000 Stylus Pro XL+* TLQ−4800* TSQ−4800*	
Fujitsu	1200* 2400* 3400* FMLBP2xx Page Printer* FMPR* PrintPartner 10V* PrintPartner 16DV* PrintPartner 20W* PrintPartner 8000*			
Generic	PostScript Printer		9−pin 136 Col* 9−pin 80 Col* 24−pin 136 Col* 24−pin 80 Col* 48 pin*	
GoldStar			GLP−1450* GLP−2050* GLP−5750*	
HP	2000C 2500C 2500CM 2563 Business Inkjet 2200 Business Inkjet 2230 Business Inkjet 2250 Business Inkjet 2250TN Business Inkjet 2280 Business Inkjet 2600 Business Inkjet 3000 Color Inkjet Printer CP1160 Color Inkjet Printer CP1700 Color LaserJet 2500	Color LaserJet 5 DesignJet 230 DesignJet 350C DesignJet 650C DesignJet 750C DesignJet 750C Plus DeskJet 350C DeskJet 420C DeskJet 520 DeskJet 560C LaserJet 2D LaserJet 3200 LaserJet 3200m LaserJet 3200se	C LaserJet 4500* C LaserJet 4550* C LaserJet 8500* C LaserJet 8550* Color LaserJet* Color LaserJet 5/5M* Color LaserJet 5000 DesignJet 5500 LaserJet 2P Plus* LaserJet 3P* LaserJet 3Si* LaserJet 4 Plus/4M Plus* LaserJet 4/4M* LaserJet 4LJ Pro*	DeskJ DeskJ LaserJ LaserJ PhotoS

	Color LaserJet 4500 Color LaserJet 4550 Color LaserJet 4600 Color LaserJet 5500 Color LaserJet 8550GN DesignJet 3500CP DesignJet 5500ps DesignJet ColorPro CAD DeskJet DeskJet 200 DeskJet 310 DeskJet 320 DeskJet 340C DeskJet 400 DeskJet 450 DeskJet 500 DeskJet 500C DeskJet 505J Plus DeskJet 510 DeskJet 540C DeskJet 550C DeskJet 600 DeskJet 610C DeskJet 610CL DeskJet 612C DeskJet 630C DeskJet 632C DeskJet 640C DeskJet 648C DeskJet 656C DeskJet 660C DeskJet 670C DeskJet 670TV DeskJet 672C DeskJet 680C DeskJet 682C DeskJet 690C DeskJet 692C DeskJet 693C DeskJet 694C DeskJet 695C DeskJet 697C DeskJet 710C DeskJet 712C DeskJet 720C DeskJet 722C DeskJet 810C DeskJet 812C DeskJet 815C DeskJet 816C	LaserJet 3330 MFP OfficeJet OfficeJet 300 OfficeJet 330 OfficeJet 350 OfficeJet 500 OfficeJet 600 OfficeJet 625 OfficeJet 635 OfficeJet 710 OfficeJet D135 OfficeJet D145 OfficeJet D155 OfficeJet G85 OfficeJet G95 OfficeJet K60 OfficeJet K80 OfficeJet LX OfficeJet T45 OfficeJet T65 OfficeJet V40 PSC 950 PSC 2210	LaserJet 4P/4MP* LaserJet 4PJ* LaserJet 4Si/4Si Mx* LaserJet 4V/4MV* LaserJet 5/5M* LaserJet 5P/5MP* LaserJet 5Si/5Si Mx/5Si Mopier* LaserJet 6L/6L Gold* LaserJet 6P/6MP* LaserJet 500 Plus* LaserJet 1000 LaserJet 2000* LaserJet 4000/4000N* LaserJet 4000T/TN* LaserJet Classic* LaserJet Plus* LaserJet Series 2* ThinkJet

	DeskJet 820C DeskJet 825C DeskJet 830C DeskJet 832C DeskJet 840C DeskJet 841C DeskJet 842C DeskJet 843C DeskJet 845C DeskJet 850C DeskJet 855C DeskJet 870C DeskJet 880C DeskJet 882C DeskJet 890C DeskJet 895C DeskJet 916C DeskJet 920C DeskJet 930C DeskJet 932C DeskJet 933C DeskJet 934C DeskJet 935C DeskJet 940C DeskJet 948C DeskJet 950C DeskJet 952C DeskJet 955C DeskJet 957C DeskJet 959C DeskJet 960C DeskJet 970C DeskJet 975C DeskJet 980C DeskJet 990C DeskJet 995C DeskJet 1000C DeskJet 1100C DeskJet 1120C DeskJet 1125C DeskJet 1200C DeskJet 1220C DeskJet 1600C DeskJet 1600CM DeskJet 3820 DeskJet 5550 DeskJet 5551 DeskJet 6122 DeskJet 6127 DeskJet Plus		

	DeskJet Portable LaserJet LaserJet 2 LaserJet 2 w/PS LaserJet 2P LaserJet 2P Plus LaserJet 3 LaserJet 3D LaserJet 3P w/ PCL5 LaserJet 3P w/PS LaserJet 4 LaserJet 4 Plus LaserJet 4L LaserJet 4M LaserJet 4ML LaserJet 4P LaserJet 4Si LaserJet 4V LaserJet 4V/4LJ Pro* LaserJet 5 LaserJet 5L LaserJet 5M LaserJet 5MP LaserJet 5P LaserJet 5Si LaserJet 6 LaserJet 6L LaserJet 6MP LaserJet 6P LaserJet 1100 LaserJet 1100A LaserJet 1200 LaserJet 1220 LaserJet 2100 LaserJet 2100M LaserJet 2200 LaserJet 3300 MFP LaserJet 3310 MFP LaserJet 3320 MFP LaserJet 3320N MFP LaserJet 4000 LaserJet 4050 LaserJet 4100 LaserJet 5000 LaserJet 5100 LaserJet 8000 LaserJet 8100 LaserJet 8150 LaserJet 9000 LaserJet Plus		

	Mopier 240 Mopier 320 OfficeJet D125 OfficeJet G55 OfficeJet Pro 1150C OfficeJet Pro 1170C OfficeJet Pro 1175C OfficeJet R45 OfficeJet R60 OfficeJet R65 OfficeJet R80 PSC 370 PSC 380 PSC 500 PSC 750 PSC 2110 PSC 2150 PaintJet PaintJet XL PaintJet XL300 PhotoSmart 7150 PhotoSmart 7350 PhotoSmart 7550 PhotoSmart P100 PhotoSmart P130 PhotoSmart P230 PhotoSmart P1000 PhotoSmart P1100 PhotoSmart P1115 PhotoSmart P1215 PhotoSmart P1218 PhotoSmart P1315 e−printer e20			
Hansum			BJ−330H* HS−650C*	
Heidelberg	Digimaster 9110*			
Hitachi	DDP 70 (with MicroPress)*			
IBM	3853 JetPrinter* 4019* 4029 10P* 4303 Network Color Printer* Execjet 4072* Infoprint 12* Page Printer 3112* ProPrinterII*		4029 030 LaserPrinter 10* 5183 Portable Printer* Infoprint 20* Infoprint 21* Infoprint 32* Infoprint 40* Infoprint 70* Infoprint 2085* Infoprint 2105* Network Printer 12* Network Printer 17* Network Printer 24*	

Imagen	ImPress*			
Infotec	4651 MF*			
Jepcom			JP−B330*	
Kodak	DigiSource 9110* IS 70 CPII*			
Kyocera	F−800T* F−1010* F−3300 FS−600* FS−600 (KPDL−2)* FS−680* FS−800* FS−1000* FS−1000+ FS−1010 FS−1200 FS−1700+* FS−1750* FS−1800 FS−1900 FS−3750* FS−3800 FS−5900C* FS−9100DN FS−9500DN P−2000*	FS−3500*	F−800A/F−800* F−820* F−1000A/F−1000* F−1200S* F−1800A/F−1800* F−2000A/F−2200S* F−3000A/F−3300* F−5000A/F−5000* FS−400A/FS−400* FS−850A/FS−850* FS−1500A/FS−1500* FS−1550+* FS−1550A/FS−1550* FS−1600+* FS−1600A/FS−1600* FS−1700* FS−1800/FS−1800N* FS−3400+* FS−3400A/FS−3400* FS−3500A/FS−3500* FS−3600+* FS−3600A/FS−3600* FS−3700* FS−3700+* FS−3800/FS−3800N* FS−5500A/FS−5500* FS−5800C* FS−6300* FS−6500/6500+* FS−6700* FS−6900* FS−7000* FS−7000+* FS−9000* LS 6550*	
LaserMaster				LM 10
Lexmark	4039 10plus E210 Optra C710 Optra Color 40 Optra Color 45 Optra Color 1200 Optra Color 1275 Optra E*	1020 Business* 3000* 3200 4076* Z31 Z42*	1000 1020* 1100 2030* 2050 2070 5000 5700*	Winwr Winwr Winwr Z13 Z23 Z33

	Optra E+* Optra E310 Optra E312* Optra Ep* Optra K 1220 Optra M410 Optra M412 Optra R+* Optra S 1250* Optra S 1855* Optra Se 3455* Optra T610 Optra T612 Optra T614 Optra T616 Optra W810 Valuewriter 300* Z52 Z53		7000* 7200 Winwriter 400* X73 Z11 Z12 Z22* Z32 Z43* Z51 Z82	
Minolta	PagePro 6* PagePro 6e* PagePro 6ex* PagePro 8* PagePro 1100		PagePro 8L*	PageP PageP
Mitsubishi	CP50 Color Printer*			
NEC	MultiWriter* P2X* PC−PR150* PC−PR201* PC−PR1000* PC−PR2000* PICTY180* PinWriter P6* PinWriter P6 plus* PinWriter P7* PinWriter P7 plus* PinWriter P60* PinWriter P70* Pinwriter P20* SilentWriter LC 890* Silentwriter 95f* Silentwriter2 S60P* Silentwriter2 model 290* SuperScript 660i* SuperScript 1800 SuperScript 4600N*		SuperScript 100C* SuperScript 150C* SuperScript 650C* SuperScript 750C* SuperScript 860* SuperScript 870* SuperScript 1260*	SuperS SuperS SuperS
Oce	3165* 9050*			
Okidata	ML 320	DP 5000	ML−184 Turbo*	Okijet

	ML 321 ML 380* Microline 600CL* Microline 620CL* Microline IBM compatible 9 pin* OL400 OL400e OL400ex OL600e* OL610e/PS OL800 OL810e/PS OL810ex OL830Plus Okipage 6e Okipage 6ex* Okipage 8c Okipage 8p Okipage 10e Okipage 10ex Okipage 12i Okipage 14ex Okipage 20DXn	Microline 182 OL400w* OL410e OL610e/S Okijet 2500* Okipage 4w Okipage 4w+* Okipage 6w Okipage 8w Okipage 8w Lite Okipage 8z Super 6e	ML−192* ML−193* ML−320* ML−320 Turbo* ML−321* ML−321 Turbo* ML−380* ML−390* ML−390 Turbo* ML−391* ML−391 Turbo* ML−393* ML−393C* ML−395B* ML−395C* ML−520* ML−521* ML−590* ML−591* ML−3410* ML−4410* Microline 192+* OL410ex* OL810e* OL820* OL1200* OkiPage 6e/6ex* OkiPos 425D* OkiPos 425S*	
Olivetti	JP350S* JP470* PG 306*		JP450	
Others			BJ−230* HDMF NONE−FF* LG GIP 3000Q_3000+* VP−6570K*	
PCPI	1030*			
Panasonic	KX−P1123* KX−P1124* KX−P1150* KX−P2023* KX−P2135* KX−P2150* KX−P4410* KX−P4450* KX−P5400* KX−P8420* KX−P8475*	KX−P2123* KX−P6150*	KX−P1124i* KX−P1180* KX−P1180i* KX−P1191* KX−P1624* KX−P1654* KX−P1695* KX−P2124* KX−P2130* KX−P2180* KX−P2624* KX−P3123* KX−P3124*	KX−P KX−P KX−P

			KX−P3624* KX−P6500* KX−PS600* KX−Pxxxx 24−pin*	
Pentax	PocketJet 200 PocketJet II			
Printrex			820 DL*	
QMS	2425 Turbo EX* LPK−100* magicolor 2+*	ps−810*		magico
Qnix			PICASSOI* PICASSOII* QBJ−3630E*	
Raven		LP−410		
Ricoh	4081* 4801* 6000* Aficio 220* Aficio 700 Aficio AP2000 RPDL I Laser Printer* RPDL II Laser Printer* RPDL III Laser Printer* RPDL IV Laser Printer*	Aficio 401*		Aficio Afico
Sambo			PageJet P1* PageJet P2* Stylus 800H+* Stylus 1000H+*	
Samsung	ML−85* ML−200 ML−210 ML−1000 ML−1010 ML−1020 ML−1200 ML−1210 ML−1220 ML−4500 ML−4600* ML−5000a* ML−5080 ML−6000/6100* ML−6040 ML−7000/7000P/7000N* ML−7050* QL−5100A* QL−6050* SI−630A*	ML−85G QL−85G	MJ 630V* SLB−3106G* SP−2417HW* SP−2421HW* SP−2433HW* SP−4017* SP−4021*	ML−5 SF/MS
Seiko	SpeedJET 200*			

		SLP* SLP 120* SLP 220* SLP EZ30* SLP Plus* SLP Pro* SLP−100* SLP−200* SLP−240*		
Sharp	AR−161*	AJ−1800 AJ−1805 AJ−2000 AJ−2005		AJ−21
Sindo			ExecJet 4072* ExecJet 4076*	
Sony			IJP−V100	
Star	JJ−100* LC24−100* LS−04 NL−10*	LC 90* LC24−200* LaserPrinter 8 NX−1001* StarJet 48*	LC 24−15 Multi−Font* LC 24−20 Multi−Font* LC 24−200 Colour* LC−15 Multi−Font* LC−20* LC−100 Colour* NX−1040R* NX−1500* NX−2415* NX−2420R* NX−2430* Starjet SJ−144* XB 24−200 Multi−Font* XB 24−250 Multi−Font* XB−2420* XB−2425* XR−1020* XR−1520* ZA−200 Multi−Font* ZA−250 Multi−Font*	WinTy
Tally	MT908*			
Tektronix	3693d color printer, 8−bit mode* 4693d color printer, 2−bit mode* 4693d color printer, 4−bit mode* 4695* 4696* 4697* Phaser 780 Phaser 850* Phaser IISX* Phaser PX*	Phaser 350*		
Xerox	2700 XES 3700 XES	Able 1406 DocuPrint C6*	DocuPrint C8* DocuPrint C11*	DocuP WorkC

	4045 XES* DocuPrint 4508 DocuPrint C20 DocuPrint C55* DocuPrint N17 DocuPrint N32* DocuPrint N4512 DocuPrint N4512 PS Document Centre 400*	DocuPrint M750* DocuPrint M760* DocuPrint P8e DocuPrint P12* DocuPrint P1202*	DocuPrint XJ6C DocuPrint XJ8C Document Homecentre WorkCentre 450cp* WorkCentre 470cx* WorkCentre XK35c	WorkC WorkC WorkC

* This entry has not been sanity−checked.

5.4. How to buy a printer

It's a bit difficult to select a printer these days; there are many models to choose from. Here are some shopping tips:

Cost

You get what you pay for. Most printers under $200−300 can print reasonably well, but printing costs a lot per page. For some printers, it only takes one or two cartridges to add up to the cost of a new printer! This is specially true for cheap inkjets. Similarly, the cheapest printers won't last very long. The least expensive printers, for example, have a MTBF of about three *months*; obviously these are poorly suited for heavy use.

Inkjets

Inkjet printheads will clog irreparably over time, so the ability to replace the head somehow is a feature. Inkjet printheads are expensive, with integrated head/ink cartridges costing ten times (!) what ink−only cartridges go for, so the ability to replace the head only when needed is a feature. Epson Styluses tend to have fixed heads, and HP DeskJets tend to have heads integrated into the cartridges. Canons have three−part cartridges with independently replaceable ink tanks; I like this design. OTOH, the HP cartridges aren't enormously more expensive, and HP makes a better overall line; Canon is often the third choice from the print quality standpoint. Epson Styluses and HP inkjets are the best supported by free software at the moment.

Lasers

Laser printers consume a drum and toner, plus a little toner wiping bar. The cheapest designs include toner and drum together in a big cartridge; these designs cost the most to run. The best designs for large volume take plain toner powder or at least separate toner cartridges and drums.

Photography

The best color photograph output is from continuous tone printers which use a silver halide plus lasers approach to produce surprise! actual photographs. Since these printers cost tens of thousands to buy, Ofoto.com offers inexpensive print−by−print jobs. The results are stunning; even the best inkjets don't compare.

The best affordable photo prints come from the dye−sublimation devices like some members of the Alps series (thermal transfer of dry ink or dye sublimation), or the few consumer−grade Sony photo printers. Unfortunately the Alps devices have poor free software support (the one report I have from a Alps user of the Ghostscript driver speaks of banding and grainy pictures), and even then it's unclear if the dye−sub option is supported. I have no idea if the Sonys work at all.

The more common photo−specialized inkjets usually feature 6 color CMYKcm printing or even a 7 color CMYKcmy process. All photo−specialized printers are expensive to run; either you always run out of blue and have to replace the whole cartridge, or the individual color refills for your high−end photo printer cost an arm and a leg. Special papers cost a bundle, too; you can expect top−quality photo inkjet output to run over a US dollar per page. See also the section on printing photographs later in this document, and the sections on color tuning (such as it is) in Ghostscript.

Lately color lasers have been getting a lot cheaper, these devices may be interesting for color reports. Color lasers are a lot cheaper per page than inkjets. However they may still not be suited for photographs. One day color lasers may become common and replace those boring monochrome laser printers.

Speed

Speed is proportional to processing power, bandwidth, and generally printer cost. The fastest printers will be networked Postscript printers with powerful internal processors. Consumer−grade printers will depend partly on Ghostscript's rendering speed, which you can affect by having a reasonably well−powered machine; full pages of color, in particular, can consume large amounts of host memory. As long as you actually*have* that memory, things should work out fine.

Forms

If you want to print on multicopy forms, then you need an impact printer; many companies still make dot matrix printers, most of which emulate traditional Epson models and thus work fine.

Labels

There are two supported lines of label printer; look for the Dymo−Costar and the Seiko SLP models. Other models may or may not work. Avery also makes various sizes of stick−on labels in 8.5x11 format that you can run through a regular printer.

Plotting

Big drafting formats are usually supported these days by monster inkjets; HP is a popular choice. Mid−sized (11x17) inkjets are also commonly used for smaller prints. Much plotting of this sort is done with the languages RTL, HP−GL, and HP−GL/2, all of which are simple HP proprietary vector languages usually generated directly by application software.

6. Spooling software

Until recently, the choice for free software users was simple − everyone ran the same old lpd lifted mostly verbatim out of BSD's Net−2 code. Even today, some vendors ship this software. But this is beginning to change. SVR4−like systems including Sun's Solaris come with a completely different print spooling package, centered around lpsched.

Today, there are a number of good systems to chose from. They are all described below; read the descriptions and make your own choice. CUPS is a good option and recommended for most users; it has excellent Postscript printer support, offers IPP support, a web interface, and a number of other features. For business environments with mainly networked Postscript printers, a front−end program like GPR with LPRng is another option; it handles PPD options directly and has a nice interface.

6.1. CUPS

CUPS has become the standard printing system in most distributions today. What makes CUPS different from the rest ? CUPS is an implementation of the Internet Printing Protocol (IPP), a new standard intended to solve some of the deficiencies of the old LPD protocol. CUPS also supports LPD, SMB and AppSocket (JetDirect) with reduced functionality. The implementation of CUPS has been driven by Michael Sweet of Easy Software Products; CUPS is distributed under the GPL. Being a new protocol, the IPP has a number of advantages on the ancient LPD protocol:

- the scheduler is a HTTP 1.1 web server and also delivers a web interface
- printer options, you can even ask the IPP device what options and document formats it supports.
- access control which restricts print jobs, job controls, and system administration commands coming from and to specified computers and printers. Like Apache, you can control access to CUPS using Allow and Deny directives.
- proxy support (since IPP uses HTTP)
- encryption support
-

Today, all major operating system vendors actively support IPP, as well as the major printer vendors. IPP is a standard printing protocol in Windows 2000 (IIS needs to be installed) which may be a better option for free software users than the proprietary SMB protocol. However, on Windows 2000 automatic printer driver downloading only works with SMB and not with IPP, this may be a reason for administrators with a lot of Windows clients to choose for SMB printer sharing using Samba and CUPS.

There are a number of very good features in it, including sensible option handling; web, GUI, and command−line interfaces; and a mime−based filtering system with strong support for Postscript.

There are several sets of PPDs which you can use with CUPS:

Built−in

The default CUPS installation contains generic PPDs for 9−pin and 24−pin Epson matrix printers, Epson Stylus Color, Stylus Photo printers, HP LaserJet, DeskJet printers and Dymo Label printers. These will enable you to print to a lot of printer models, but will not give you access to specific capacities of the models

Foomatic

Foomatic can generate a suitable PPD for use with any printer driver that has full details entered in the linuxprinting.org database. The PPD gets used together with a backend script named **foomatic−rip**.

foomatic−rip uses free software drivers. At the moment there is support for a rather large number of printers in this system. Foomatic forms a basis for non−Postscript printer support in most GNU/Linux distributions. CUPS and Foomatic are becoming quite popular and this is currently the recommended printing system for most situations.

Postscript PPDs

CUPS can use vendor−supplied PPD files for Postscript printers directly. Often these come with the Windows drivers for a printer, or can be found on the printer vendor's website. If you have a choice between a driver for Windows 9x and Windows NT/W2K, than select the driver for Windows NT. Adobe also distributes PPD files for many Postscript printers.

ESP Print Pro

Easy Software Products, Inc. sells CUPS bundled with a collection of proprietary drivers. Although they are not free software, they do drive many common printers. The bundle is somewhat expensive measured against the price of a single supported printer, but it certainly has a place. The package includes graphical front−end tools.

Gimp−Print

The Gimp−Print drivers are high quality drivers for Canon, Epson, Lexmark, and PCL printers for use with Ghostscript, CUPS, Foomatic, and the Gimp.

OMNI

Omni is a package made by IBM, now containing support for more than 450 printers. The OMNI printer driver model is distributed by IBM under LGPL License.

HPIJS

HPIJS supports around 150 of HP's own printers at excellent print quality now (currently only via the Foomatic path). As of Version 1.0.1 , the "hp Product Only" clause has been removed from the license and the drivers are distributed with a BSD license.

The third−party program **XPP** (see Figure 4) offers a very nice graphical interface to the user functionality of CUPS, including an marvelous interface to print−time options (shown in Figure 5). For information on using XPP, see Section 3.4.2.

6.2. LPD

LPD, the original BSD Unix Line Printer Daemon, has been the standard on Unix for years. It is available for every style of Unix, and offers a rather minimal feature set derived from the needs of timesharing−era computing. Despite this somewhat peculiar history, it is still useful today as a basic print spooler. To be really useful with modern printer, a good deal of extra work is needed in the form of companion filter scripts and front−end programs. But these exist, and it does all work.

LPD is also the name given to the network printing protocol by RFC 1179. This network protocol is spoken not only by the LPD daemon itself, but by essentially every networked print server, networked printer, and every other print spooler out there; LPD is the least common denominator of standards−based network printing.

LPRng(see Section 6.3) is a far better implementation of the basic LPD design than the regular one; if you must use LPD, consider using LPRng instead. There is far less voodoo involved in making it do what you want, and what voodoo there is is well documented. LPRng is essentially an enhanced LPD implementation with better security and extra features.

There are a large number of LPD sources floating around in the world. Arguably, some strain of BSD Unix is probably the official owner, but everyone implements changes willy−nilly, and they all cross−pollinate in unknown ways, such that it is difficult to say with certainty exactly which LPD you might have. Of the readily available LPDs, GNUlpr offers one with a few minor modifications that make the user interface much more flexible. The GNUlpr supports command−line option specification with a `−o` flag; options are then passed through to filters. This is similar to the features offered by a number of traditional Unix vendors, and similar to (although incompatible with) LPRng's `−z` option mechanism.

If you go with LPD, the best way to use it is via a front−end. There are several to chose from; KDEPrint, GPR (see Section 3.4) and XPP are perhaps the best. Others exist; tell me about them.

6.3. LPRng

Some GNU/Linux vendors provide LPRng, a far less ancient LPD print spooling implementation. LPRng is far easier to administer for large installations (read: more than one printer, any serial printers, or any peculiar non−lpd network printers) and has a less frightfully haphazard codebase than does stock lpd. It can even honestly claim to be secure − there are no SUID binaries, and it supports authentication via PGP or Kerberos.

LPRng also includes some example setups for common network printers − HP LaserJets, mainly − that include some accounting abilities. LPRng uses more or less the same basic filter model as does BSD lpd, so the LPD support offered by the linuxprinting.org website applies to LPRng as well. This can help you effectively use free software drivers for many printers.

LPRng is distributed under either the GPL or an Artistic license.

6.4. PPR

PPR is a Postscript−centric spooler which includes a rudimentary Postscript parsing ability from which it derives several nice features. It includes good accounting capabilities, good support for Appletalk, SMB, and LPD clients, and much better error handling than lpd. PPR, like every other spooler here, can call Ghostscript to handle non−Postscript printers.

PPR was written by, and is in use at, Trinity College. The license is BSD−style; free for all use but credit is due.

6.5. Others

6.5.1. PDQ

PDQ stands for "Print, Don't Queue", and the way it works reflects this design. PDQ is a non−daemon−centric print system which has a built−in, and sensible, driver configuration syntax. This includes the ability to declare printing options, and a GUI or command line tool for users to specify these options with; users get a nice dialog box in which to specify resolution, duplexing, paper type, etc.

Running all of the filters as the user has a number of advantages: the security problems possible from Postscript are mostly gone, multi−file LaTeX jobs can be printed effectively as dvi files, and so forth.

PDQ is not without flaws: most notably it processes the entire job before sending it to the printer. This means that, for large jobs, PDQ may simply be impracticalâ€”you can end up with hundreds of megs being copied back and forth on your disk. Even worse, for slow drivers like the better quality inkjet drivers, the job will not start printing until Ghostscript and the driver have finished processing. This may be many minutes after submission.

There's a real place for PDQ; it has a simple design that doesn't subtract user control. And the normal control path crosses no security boundaries, so it can't have the classes of security bug people are always finding in other systems. And to top it off, it's small.

However there is no active development done on PDQ. A new maintainer would be most welcome.

6.5.2. GNUlpr

GNUlpr began its life in some work that HP sponsored VA Linux to do. Unfortunately, GNUlpr is now pretty much dead.

6.5.3. CPS

The Coherent Printing System is a set of Perl scripts called "lpr", "lpd", "lprm", and "lpq". These replace the programs of the same name which come with many Linux systems.

6.5.4. CEPS

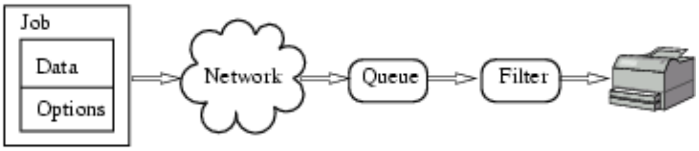
The Cisco Enterprise Print System was developed by Damian Ivereigh when he was a sysadmin at Cisco. He did more than he was hired to do, he developed a new printing system to improve the administrative hassle. Cisco authorized the release of the software for free under the GNU General Public License. Installing CEPS will however only pay off at large organisations.

---

7. How it all works

In order to get printing working well, you need to understand how your spooling software works. All systems work in essentially the same way, although the exact order might vary a bit, and some systems skip a step or two:

Figure 10. Spooling Illustration

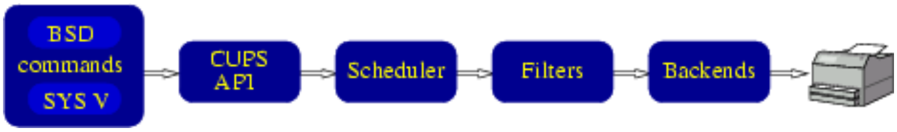


1. The user submits a job along with his selection of options. The job data is usually, but not always, Postscript.
2. The spooling system copies the job and the options over the network in the general direction of the printer.
3. The spooling system waits for the printer to be available.
4.
5. The spooling system applies the user's selected options to the job, and translates the job data into the printer's native language, which is usually not Postscript. This step is called *filtering*; most of the work in setting things up lies in getting the proper filtering to happen.
6. The job is done. The spooling system will usually do assorted cleanup things at this point. If there was an error along the way, the spooler will usually notify the user somehow (for example, by email).

7.1. CUPS

To print a job with CUPS, you can use both the BSD (see Section 5.3.1) and System V commands making it really easy for people with prior experience with either system.

Figure 11. Simplified CUPS illustration



Initially CUPS lacked an LPD backend. This was of course quickly added. Currently there are backends available for at least IPP, LPD, SMB, JetDirect, USB, Netatalk, parallel and serial printers. You may find others on the net or write your own.

There are only a handfull of built−in drivers, allowing you to print with most printers but probably not at the maximum resolution. A PPD file for a Postscript driver can be added to CUPS but if you want to print at best quality with your fancy new HP Deskjet you are out of luck. It is here that Foomatic comes to the rescue. You can use Foomatic in combination with CUPS. Foomatic uses a CUPS filter called foomatic−rip to do its magic. foomatic−rip uses PPD files to describe printer capabilities, even for non−Postscript printers. CUPS + Foomatic is currently the recommended printing system. Some Linux distributions already use it and the number that do will only grow.

The CUPS scheduler does not only accept jobs, it is also a administrative webinterface. Currently you can add/delete printers, cancel jobs, start/stop printers. Moving jobs will be available in a later release.

7.2. LPD

Lpd stands for Line Printer Daemon, and refers in different contexts to both the daemon and the whole collection of programs which run print spooling. These are:

lpd
: The spooling daemon. One of these runs to control everything on a machine, AND one is run per printer while the printer is printing.

lpr
: The user spooling command. Lpr contacts lpd and injects a new print job into the spool.

lpq
: Lists the jobs in a print queue.

lpc
: The Lpd system control command. With lpc you can stop, start, reorder, etc, the print queues.

lprm
: **lprm** removes a job from the print spool.

So how does it fit together? The following things happen:

1. At boot time, **lpd** is run. It waits for connections and manages printer queues.
2. A user submits a job with the **lpr** command or, alternatively, with an lpr front−end like GPR, PDQ, etc.**Lpr** contacts **lpd** over the network and submits both the user's data file (containing the print data) and a control file (containing user options).
3. When the printer becomes available, the main **lpd** spawns a child **lpd** to handle the print job.
4. The child **lpd** executes the appropriate filter(s) (as specified in the `if` attribute in`/etc/printcap`) for this job and sends the resulting data on to the printer.

The lp system was originally designed when most printers were line printers − that is, people mostly printed plain ASCII. By placing all sorts of magic in the `if` filter, modern printing needs can be met with **lpd** (well, more or less; many other systems do a better job).

There are many programs useful for writing LPD filters. Among them are:

gs
: Ghostscript is a host−based Postscript interpreter (aka a Raster Image Processor or RIP). It accepts Postscript and produces output in various printer languages or a number of graphics formats. Ghostscript is covered in Section 10.

ppdfilt
: **ppdfilt** is a standalone version of a CUPS component. It filters Postscript, executing a few basic transformations on it (n−up printing, multiple copies, etc) and adding in user option statements according to a Postscript Printer Definition (PPD) file usually included with Postscript printers.

: **ppdfilt** is best used together with an option−accepting LPD system (like the GNUlpr, or LPRng) and a filter script which parses user−provided options into the equivalent **ppdfilt** command. VA Linux and HP provide a modified rhs−printfilters package which does exactly this; it produces nice results if you have a Postscript printer. See Section 8.2.2 for information on this system.

ps2ps

ps2ps is a utility script included with Ghostscript. It filters Postscript into more streamlined Postscript, possibly at a lower Language Level. This is useful if you have an older Postscript printer; most modern software produces modern Postscript.

mpage

mpage is a utility which accepts text or Postscript, and generates n−up output that is, output with several page images on each piece of paper. There are actually several programs which do this, including**enscript**, **nenscript**, and**a2ps**.

a2ps

a2ps, aka any−to−ps, is a program which accepts a variety of file types and converts them to Postscript for printing.

---

8. How to set things up

For common configurations, you can probably ignore this section entirely − instead, you should jump straight to Section 9 below, or better yet, your vendor's documentation. Most GNU/Linux distributions supply one or more "idiot−proof" tools to do everything described here for common printers.

If your vendor's tool doesn't work out for you, or you'd like the ability to interactively control printing options when you print, then you should use some other system. APS Filter is another good system; it configures LPD queues and filters very easily on most any sort of Unix system.

You can also use the printing system interfaces from the linuxprinting.org website to connect many free drivers into several spooling systems. Once this project is complete, these interfaces will offer the best functionality: all styles of free software drivers are supported, user−settable options are available, and most common spooling systems are supported. Currently the foomatic print system is used in most modern distributions anyway. However, your distro may include a slightly outdated version of foomatic.

8.1. Configuring CUPS

If you are using a client with CUPS and a CUPS server has already been configured, installing the printers on your client can not get much easier than this: do nothing. Through broadcasting, the client should find the CUPS server and automatically configure the printers that are installed on that print server. This is one of the features of CUPS that will be really appreciated on large networks.

Manually configuring printers with CUPS, also is a peace of cake. If you are new to CUPS and/or Unix printing, the way to go is probably the web interface. If you have to configure lots of printers, using the command−line will probably be faster.

The URL to access the CUPS web interface is http://hostname:631/admin by default. The port can be changed in cupsd.conf if necessary.

To add a printer from the command−line the general syntax is **lpadmin −p printer −E −v device −m ppd** Lpadmin with the −p option adds or modifies a printer. The printers are saved in the file The −x option deletes the named printer. Read the lpadmin man page for available options.

Example 3. command−line examples

```
/usr/sbin/lpadmin −p testpr1 −E −v socket://192.168.1.9 −m deskjet.ppd
/usr/sbin/lpadmin −p testpr2 −E −v parallel:/dev/lp0 −m laserjet.ppd
/usr/sbin/lpadmin −x testpr1
```

More information about configuring printers and options can be found in the CUPS documentation. The Software Administrators Manual will teach you all you need to know about configuring printers with CUPS.

8.2. Configuring LPD

Until recently most GNU/Linux distributions shipped with LPD. This section describes a very basic setup for LPD; further sections detail the creation of complex filters and network configuration.

8.2.1. Basic LPD configuration

The minimal setup for lpd results in a system that can queue files and print them. It will not pay any attention to whether or not your printer will understand them, and will probably not let you produce attractive output. But we have to start somewhere.

To add a print queue to lpd, you must add an entry in`/etc/printcap`, and make the new spool directory under `/var/spool/lpd`.

An entry in `/etc/printcap` looks like:

```
# LOCAL djet500
lp|dj|deskjet:\
        :sd=/var/spool/lpd/dj:\
        :mx#0:\
        :lp=/dev/lp0:\
        :sh:
```

This defines a spool called *lp*,*dj*, or *deskjet*, spooled in the directory `/var/spool/lpd/dj`, with no per−job maximum size limit, which prints to the device`/dev/lp0`, and which does not have a banner page (with the name of the person who printed, etc) added to the front of the print job.

Go now and read the man page for *`printcap`*.

The above looks very simple, but there a catch − unless I send in files a DeskJet 500 can understand, this DeskJet will print strange things. For example, sending an ordinary Unix text file to a deskjet results in literally interpreted newlines, and gets me:

```
This is line one.
        This is line two.
                This is line three.
```

ad nauseam. Printing a PostScript file to this spool would get a beautiful listing of the PostScript commands, printed out with this "staircase effect", but no useful output.

Clearly more is needed, and this is the purpose of filtering. The more observant of you who read the printcap man page might have noticed the spool attributes `if` and`of`. Well, `if`, or the input filter, is just what we need here.

If we write a small shell script called **filter** that adds carriage returns before newlines, the staircasing can be eliminated. So we have to add in an `if` line to our printcap entry above:

```
lp|dj|deskjet:\
        :sd=/var/spool/lpd/dj:\
        :mx#0:\
        :lp=/dev/lp0:\
        :if=/var/spool/lpd/dj/filter:\
        :sh:
```

A simple filter script might be:

```
#!perl
# The above line should really have the whole path to perl
# This script must be executable: chmod 755 filter
while(<STDIN>){chomp $_; print "$_\r\n";};
# You might also want to end with a form feed: print "\f";
```

If we were to do the above, we'd have a spool to which we could print regular Unix text files and get meaningful results. (Yes, there are four million better ways to write this filter, but few so illustrative. You are encouraged to do this more efficiently.)

The only remaining problem is that printing plain text is really not too hot − surely it would be better to be able to print PostScript and other formatted or graphic types of output. Well, yes, it would, and it's easy to do. The method is simply an extension of the above linefeed−fixing filter.

Such a filter is called a *magic* filter. Don't bother writing one yourself unless you print strange things − there are a good many written for you already, and most have easy−to−use interactive configuration tools. You should simply select a suitable pre−written filter:

foomatic−rip
: foomatic−rip is a filter designed to use data from the LinuxPrinting.org printer database. It supports essentially all free software printer drivers, including regular Ghostscript drivers, Uniprint drivers, and the assorted filter programs floating around out there. foomatic−rip works with CUPS, LPRng, LPD, GNUlpr, PPR, PDQ, no spooler.

APS Filter
: apsfilter is a filter designed for use on a wide variety of Unices. It supports essentially all Ghostscript drivers. It, too, works with various strains of LPD, including stock BSD and LPRng.

RHS−Printfilters
: RHS−Printfilters is a filter system constructed by Red Hat. It shipped beginning, I think, in version 4 of Red Hat Linux, as the backend to the easy−to−use **printtool** GUI printer configuration tool.

: The rhs filter system is built on an ASCII database listing distributed with it. This listing supports many Ghostscript and Uniprint drivers, but not filter−style drivers. The filters constructed also do not support much in the way of user−controllable options at print time.

: The **printtool** places a configuration file named `postscript.cfg` in the spool directory. Inside this Bourne shell−style file, each setting is a variable. In unusual cases, you can make useful changes directly to the config file which the printtool won't allow; typically this would be the specification of an unusual Ghostscript driver, or a PPD filename for the VA rhs−printfilters version.

: VA Linux has made some enhancements to the rhs−printfilters system under contract from HP. With the proper versions, it is possible to select options for Postscript printers under control of Adobe PPD files. I cover this system inSection 8.2.2.

There's one catch to such filters: older version of lpd don't run the *if* filter for remote printers, while most newer ones do (although often with no arguments). The version of LPD shipped with modern GNU/Linux and FreeBSD distributions does; most commercial Unices that still ship LPD have a version that does not. See the section on network printing later in this document for more information on this. If you only have locally−connected printers, then this won't affect you.

8.2.2. LPD for PostScript Printers

While most versions of LPD don't gracefully handle PostScript (never mind user options), VA Linux modified LPD and Red Hat's filtering software to support PostScript printers fairly well. Because the intention was to donate the code to the gnu project, they called it GNUlpr

8.2.2.1. How it works

VA's system uses Postscript Printer Definition, or PPD, files. PPD files are provided by printer manufacturers and declare the available options on a printer, along with the Postscript code needed to activate them. With the VA system, the normal LPD scheme works a little differently:

1. The user can specify options with the `−o` flag. For example, you might specify `−o MediaType:Transparency` if you were about to print on overhead film. Alternatively, the front−end GPR can be used to specify options in a dialog box; you can see screenshots of GPR in Section 3.4.3.
2. LPR passes the options to LPD as an extended attribute in the LPD control file.
3. A modified version of the rhs−printfilters package is given the extended options data in an environment variable, and uses ppdfilt to add these options to the print data.

8.2.2.2. Obtaining and Installing

You can obtain RPM packages, or source tarballs, from the project's website on SourceForge. For installation details, consult the project's installation micro−HOWTO. In essence, you need to uninstall the Red Hat version of printtool, lpd, and rhs−printfilters entirely, and then install the VA versions, plus ppdfilt, gpr, and a few other utilities.

You will also need PPD files for your Postscript printers. PPD files are usually fairly easy to find. VA Linux and HP distribute PPD files for many Laserjet models. Other vendors provide PPDs for their own printers, and Adobe distributes PPD files for many printers.

At the moment, much of this is a bit difficult to install. But future installation tools will build upon the printer configuration library **libprinterconf**, which enables both the autodetection and rhs−printfilter configuration of both networked and local printers.

It is possible to use GPR alone, without the modified LPD or even rhs−printfilters. GPR can be compiled with all the logic needed to massage Postscript jobs directly. This may be an easier−to−install option suitable for people who never really need to print using lpr directly.

8.2.2.3. Controlling Postscript Options

Once you've setup VA's Postscript−capable LPD system (GNUlpr), you can control your printer's options in two ways:

With the GUI

To use GPR, you first make sure that you've specified the proper PPD file. Then the printer's options will be available on the `Advanced' panel. Basic **ppdfilt** options will be available on the `Common' panel.

With the command line

This **lpr** supports the `−o` option. You may specify any option/value pair from your printer's PPD file with `−o`. For example, consider this PPD file option clause:

```
*OpenUI *PrintQuality/Print Quality: PickOne
*DefaultPrintQuality: None
*OrderDependency: 150 AnySetup *PrintQuality
*PrintQuality None/Printer Setting: ""
```

```
*PrintQuality Quick/QuickPrint:  "<< /DeviceRenderingInfo ...
*PrintQuality Normal/Normal: "<< /DeviceRenderingInfo << /...
*PrintQuality Pres/Presentation: "<< /DeviceRenderingInfo ...
*PrintQuality Image/1200 Image Quality: "<< /DeviceRenderi...
*CloseUI: *PrintQuality
```

For the option `PrintQuality`, the possible values are `Quick`, `Normal`,`Pres`, or `Image`. You might give a command like:

```
% lpr −o PrintQuality:Image file.ps
```

There are a number of options common to all printers which will work in addition to the ones from your PPD. These include:

`page−ranges`
: You can specify a range of pages to print. For example,`page−ranges:2−3`.

`page−set`
: You can print only odd or even pages. For example,`page−set:odd`.

`number−up`
: You can print multiple pages on each piece of paper. For example, `number−up:2`.

Other options are detailed in the **ppdfilt** man page.

8.2.3. File Permissions

By popular demand, I include below a listing of the permissions on interesting files on my system. There are a number of better ways to do this, ideally using only SGID binaries and not making everything SUID root, but this is how my system came out of the box, and it works for me. (Quite frankly, if your vendor can't even ship a working lpd you're in for a rough ride).

```
−r−sr−sr−x   1 root     lp    /usr/bin/lpr*
−r−sr−sr−x   1 root     lp    /usr/bin/lprm*
−rwxr−−r−−   1 root     root  /usr/sbin/lpd*
−r−xr−sr−x   1 root     lp    /usr/sbin/lpc*
drwxrwxr−x   4 root     lp    /var/spool/lpd/
drwxr−xr−x   2 root     lp    /var/spool/lpd/lp/
```

Lpd must currently be run as root so that it can bind to the low−numbered lp service port. It should probably become UID lp.lp or something after binding, but I don't think it does. This is simply one more reason to avoid the stock BSD LPD.

8.3. Large Installations

Large installations, by which I mean networks including more than two printers or hosts, have special needs. Below are some tips.

CUPS has some nice features that make a good choice for a large network. Printer classes, access control and automatic client configuration to name a few.

If you use LPD, for really large environments, merely distributing printcap/filter information becomes a difficult problem; the Cisco Enterprise Print System addresses this and is probably either a good starting point or a nearly complete solution, depending on your needs. Medium to large environments can be well supported by native LPRng features.

- Each printer should have a single point of control, where an administrator can pause, reorder, or redirect the queue. To implement this, have everyone printing to a local server, which will then queue jobs and direct them to the proper printer. For large campuses or distributed networks, have one server per building or other suitable network subset.
- Use CUPS or LPRng, at least on servers; the BSD LPD is too buggy for "real" use. But don't take my word for it you should test a number of spoolers and see which suits you best.
- Client systems should not have unique printing configurations. CUPS provides automatic client configuration of printers on the same subnet. You can even configure CUPS (BrowsePoll) to poll servers on other subnets for available printers. These features limit the amount of configuration that needs to take place at the client. To implement a uniform printing configuration with LPRng, use LPRng's extended printcap syntax so that you have one printcap to use everywhere. CEPS provides for this by building atop a lightweight distributed database instead of traditional printcap files.
- Print queues should not be named for make or model; name print queues for something sensible like location (floor2_nw) or capability (color_transparency). Three years from now, when a printer breaks, you will be able to replace it with a different make or model without causing confusion.
- Operate a web page which shows detailed information on each printer, including location, capabilities, etc. Consider having it show the queue and include a button to remove jobs from the queue. Complex networked environments are unmanageable for users without proper documentation.
- On Windows and Apple systems, use either the platform−specific drivers *everywhere* (Samba supports the Windows automagical driver−download mechanism) or, better, use generic Postscript drivers *everywhere*. Do not mix and match; primitive word processors often produce different output when the installed printer driver changes; users cannot deal with output that varies depending on the particular client/printer pair.
- If at all possible, buy a large−volume printer for large−volume printing. If on a budget, use LPRng's multiple printers/one queue facility or CUPS printer classes and assign a babysitter; printers are complex mechanical devices that will often jam and run out of paper in such configurations.
- Do not feel that printers must be plugged into workstations; Ethernet "print servers" now cost under $100. The ability to locate printers anywhere you can network is a big improvement over forced location near a host; locate printers in sensible, central locations.
- Use any SNMP trap or other monitoring/alert facility available to you − someone should be tasked with running around and fixing printers with no ink or paper. Npadmin (see Section 11.10.1) can be used to do some management operations with SNMP printers.

8.4. Accounting

Regular LPD provides very little to help you with accounting. You can specify the name of an accounting file in the `af` printcap attribute, but this is merely passed as an argument to your `if` filter. It's up to you to make your `if` filter write entries to the accounting file, and up to you to process the accounting file later (the traditional format is mainly useful for line printers, and is nontrivial to parse in Perl, so there's no reason to preserve it). Also, if you're using **foomatic−rip** program as your filter, you'll need to make changes, since it depends on being given a configuration file as the ``accounting'' file name.

CUPS provides page accounting by passing jobs through the pstops filter. This filter expects Postscript input. If you use print "raw" jobs, this is always counted as 1 page. This means that accounting will not work, if you print from Windows client with the native printer driver.

Ghostscript provides a PageCount operator that you can use to count the number of pages in each job; basically you just tack a few lines of postscript onto the end of the job to write an accounting file entry; for the best example of this see the file`unix−lpr.sh` in the Ghostscript source distribution.

Note that the **unix−lpr** implementation of accounting writes to a file from the Ghostscript interpreter, and is thus incompatible with the recommended −dSAFER option. A better solution might be to query the printer with a PJL command after each job, or to write a postscript snippet that prints the pagecount on stdout, where it can be captured without having to write to a file.

The LPRng print spooler includes an HP−specific sample implementation of accounting; I assume that it queries the printer with PJL. This technique should work for most PJL, Postscript, or SNMP printers with which you have two−way communications.

If you have a networked printer that supports SNMP, you can use the npadmin program to query a pagecount after each job. This should work properly for all print jobs. See Section 11.10.1 for more information on npadmin.

9. Vendor Solutions

This section is, by definition, incomplete. Feel free to send in details of your favorite distribution.

There are a number of third−party packages out there designed to make printer configuration under Unix easy. These are covered in Section 8; see the subsection there for your particular spooling software for pointers.

9.1. Red Hat

Red Hat has a GUI printer administration tool called printtool which can add remote printers and printers on local devices. It lets you choose a ghostscript−supported printer type and Unix device file to print to, then installs a print queue in `/etc/printcap` and uses a filter program from the rhs−printfilters package to support postscript and other common input types. This solution works fairly well, and is trivial to setup for common cases.

Red Hat 6.x shipped a BSD LPD flavor; Red Hat 7.x and 8.0 appear to default to using LPRng.

Where Red Hat 6.x and 7.x fail is when you have a printer which isn't supported by their standard Ghostscript (which is GNU rather than Aladdin Ghostscript, and which supports fewer printers). Check in the printer compatibility list above (or online) if you find that you can't print properly with the stock Red Hat software. If your printer isn't supported by Red Hat's tools, you may need to install a contributed venison of Aladdin Ghostscript, and will probably also be better off if you use the lpdomatic or apsfilter packages, which know all about the printers supported by late−model Ghostscripts, and others besides.

Red Hat 8.0 still installs LPRng by default although you can select CUPS. But even if you explicitly select only CUPS, LPRng is still installed. In Red Hat 8.1 CUPS will finally be the default spooler.

Red Hat 9.0 finally uses CUPS as default spooler.

9.2. Debian

Debian offers a choice between plain LPD, LPRng, or CUPS; LPRng or CUPS are probably the better choices. PDQ is provided in the unstable tree (currently called sid). Debian also offers a choice of printer configuration tools; apsfilter version 5 or later is probably your best bet, since that venison adds support for LPRng and Ghostscript's uniprint driver scheme. Red Hat's printtool is also supported, for those who like GUI administration tools.

9.3. SuSE

The printing system on SuSE Linux is based on apsfilter, with some enhancements; SuSE's apsfilter will recognize all common file formats (including HTML, if html2ps is installed). There are two ways to setup printers on SuSE systems:

- YaST will let you configure "PostScript", "DeskJet" and "Other printers", supported by Ghostscript drivers; it's also possible to setup HP's GDI printers (DeskJet 710/720, 820, 1000, via the "ppa" package). YaST will provide `/etc/printcap` entries for every printer ("raw", "ascii", "auto" and "color", if the printer to configure is a color printer). YaST will create spool directories and it will arrange apsfilterrc files, where you're able to fine tune some settings (Ghostscript preloads, paper size,

paper orientation, resolution, printer escape sequences, etc.). With YaST it's also possible to setup network printers (TCP/IP, Samba, or Novell Netware Printer).
- In addition SuSE includes the regular SETUP program from the original apsfilter package (with some enhancements); run**lprsetup** to invoke this configuration script. Once you get accustomed to its GUI, you'll be able to configure local and network printers.

The SuSE installation manual explains both of these setup procedures.

Wolf Rogner reported some difficulties with SuSE. Apparently the following bugs may bite:

- Apsfilter's regular SETUP script is a bit broken, as are the KDE setup tools. Use YaST. [Ed: does this still apply? It's been some time Wolf's report.]
- For networked printers that need to be fed from Ghostscript, you'll need to first uncomment the line REMOTE_PRINTER="remote" in `/etc/apsfilterrc`. Then run YaST to configure the printer and, under Network configurations, set up a remote printer queue.
- YaST's setup doesn't allow color laser printers, so configure a mono printer and then change mono to color everywhere in the printcap entry. You may have to rename the spool directory, too.

9.4. Caldera

Caldera ships LPRng. I have no idea what sort of setup tools they offer.

I've just signed up a Caldera employee as a maintainer of the LinuxPrinting.org database; evidently they plan to ship a CUPS and Foomatic−based print system in future releases.

9.5. Corel

Corel is Debian−based, so all the Debian facts above should still apply. In addition, they've written their own setup tool, based on the sysAPS library which in turn uses my database. They've certainly done so as part of WordPerfect.

Corel operates a printing support newsgroup named corelsupport.linux.printing. The bulk of the traffic appears to be WordPerfect and Corel Linux related.

9.6. Mandrake

As of version 7.2b1, Mandrake ships with CUPS standard. The program QtCUPS is used to provide a clean GUI administration interface. Till went to some trouble to include as many drivers as possible, and they ship CUPS PPD files build with my own foomatic interface code. Mandrake was the first distro to ship CUPS.

I think Earlier Mandrake versions shipped with the Red Hat printtool.

9.7. Slackware

Slackware ships with APS Filter. The apsfilter SETUP script is installed as the command `apsfilterconfig'. You should be able to get a reasonable setup by simply running that.

As of Slackware 9.0, CUPS is included in the extras dir of slackware but the default is still LPRng + APSFilter.

9.8. Other Distributions

Please send me info on what other distributions do!

10. Ghostscript.

Ghostscript is an incredibly significant program for free software−driven printing. Most printing software under Unix generates PostScript, which is typically a $100 option on a printer. Ghostscript, however, is free, and will generate the language of your printer from PostScript.

Ghostscript is available in several forms. The commercial version of Ghostscript, called Aladdin Ghostscript, may be used freely for personal use but may not be distributed by commercial entities. It is generally a year or so ahead of the free Ghostscript; at the moment, for example, it supports many color inkjets that the older Ghostscripts do not and has rather better PDF support.

The main free version of Ghostscript is GNU Ghostscript, and is simply an aged version of Aladdin ghostscript. This somewhat awkward arrangement has allowed Aladdin to be a totally self−funded free software project; the leading edge versions are done by L Peter and a few employees, and are licensed to hardware and software vendors for use in commercial products. Unfortunately, while this scheme has provided for L Peter's continued work on Ghostscript for years, it has also inhibited the participation of the wider free software community. Driver authors, in particular, find the arrangement poor. L Peter's retirement plans mandate a larger community involvement in the project, so he is considering license changes, and has established a SourceForge project.

The third version of Ghostscript is ESP Ghostscript, maintained by Easy Software Products (authors of CUPS) under contract from Epson. ESP Ghostscript is a combination of the gimp−print driver project's drivers and GNU Ghostscript, plus assorted usability patches. This version is not yet in full swing, but it will be available soon, and will hopefully simplify life for owners of Gimp−print−driven printers.

Whatever you do with **gs**, be very sure to run it with the option for disabling file access (`−dSAFER`). PostScript is a fully functional language, and a bad PostScript program could give you quite a headache.

Speaking of PDF, Adobe's Portable Document Format (at least through 1.3) is actually little more than organized PostScript in a compressed file. Ghostscript can handle PDF input just as it does PostScript. So you can be the first on your block with a PDF−capable printer.

10.1. Invoking Ghostscript

Typically, Ghostscript will be run by whatever filter you settle upon (I recommend Foomatic if your vendor didn't supply anything that suits you), but for debugging purposes it is often handy to run it directly.

gs −helpwill give a brief listing of options and available drivers (note that this list is the list of drivers compiled in, not the master list of all available drivers).

You might run gs for testing purposes like: `**gs <options> −q −dSAFER −sOutputFile=/dev/lp1 test.ps**'.

10.2. Ghostscript output tuning

There are a number of things one can do if Ghostscript's output is not satisfactory (actually, you can do anything you darn well please, since you have the source).

Some of these options, and others are described in the Ghostscript User Guide (the file *`Use.htm`* in the

Ghostscript distribution; possibly installed under `/usr/doc` or `/usr/share/doc` on your system) are all excellent candidates for driver options in your filter system.

10.2.1. Output location and size

The location, size, and aspect ratio of the image on a page is controlled by the printer−specific driver in ghostscript. If you find that your pages are coming out scrunched too short, or too long, or too big by a factor of two, you might want to look in your driver's source module and adjust whatever parameters jump out at you. Unfortunately, each driver is different, so I can't really tell you what to adjust, but most of them are reasonably well commented.

10.2.2. Gamma, dotsizes, etc.

Most non−laser printers suffer from the fact that their dots are rather large. This results in pictures coming out too dark. If you experience this problem with an otherwise untunable driver, you could use your own transfer function. Simply create the following file in the ghostscript lib−dir and add its name to the gs call just before the actual file. You may need to tweak the actual values to fit your printer. Lower values result in a brighter print. Especially if your driver uses a Floyd−Steinberg algorithm to rasterize colors, lower values (0.2 − 0.15) are probably a good choice.

```
%!
%transfer functions for cyan magenta yellow black
{0.3 exp} {0.3 exp} {0.3 exp} {0.3 exp} setcolortransfer
```

It is also possible to mend printers that have some kind of color fault by tweaking these values. If you do that kind of thing, I recommend using the file `colorcir.ps`, that comes with ghostscript (in the `examples/` subdirectory), as a test page.

For many of the newer color inkjet drivers, there are command−line options, or different upp driver files, which implement gamma and other changes to adapt the printer to different paper types. You should look into this before playing with Postscript to fix things.

10.2.3. Color Printing in Ghostscript

Ghostscript's default color dithering is optimized for low−resolution devices. It will dither rather coarsely in an attempt to produce 60ppi output (not dpi, ppi − the "apparent" color pixels per inch you get after dithering). This produces rather poor output on modern color printers; inkjets with photo paper, in particular, are capable of much finer ppi settings.

To adjust this, use the Ghostscript option `−dDITHERPPI=x`, where `x` is the value to use. This may or may not have an effect with all drivers; many newer drivers (the Epson Stylus **stp** driver, for example) implement their own dithering and pay no attention to this setting. Some drivers can use either the regular Ghostscript or driver−specific dithering (the Canon Bubblejet **bjc600** driver, for example).

Ghostscript's dithering is in fact rather rudimentary. Many things needed for good output on modern printers are simply not available in the Ghostscript core. Various projects to fix this situation and the free software world does have the software to do so ready and waiting are hampered by Ghostscript's licensing situation and the resulting "cathedral" development style. Beginning at the Open Source Printing Summit 2000, however, all the necessary people are talking, so you can expect this situation to improve shortly.

11. Networks

One of the features of most spoolers is that they support printing over the network to printers physically connected to a different machine, or to the network directly. With the careful combination of filter scripts and assorted utilities, you can print transparently to printers on all sorts of networks.

11.1. Printing to a Unix/lpd host

To allow remote machines to print to your printer using the LPD protocol, you must list the machines in*/etc/hosts.equiv* or*/etc/hosts.lpd*. (Note that*hosts.equiv* has a host of other effects; be sure you know what you are doing if you list any machine there). You can allow only certain users on the other machines to print to your printer by using the *rs* attribute; read the `lpd` man page for information on this.

11.1.1. With `lpd`

To print to another machine, you make an*/etc/printcap* entry like this:

```
# REMOTE djet500
lp|dj|deskjet:\
        :sd=/var/spool/lpd/dj:\
        :rm=machine.out.there.com:\
        :rp=printername:\
        :sh:
```

Note that there is still a spool directory on the local machine managed by `lpd`. If the remote machine is busy or offline, print jobs from the local machine wait in the spool area until they can be sent.

11.1.2. With `rlpr`

You can also use *rlpr* to send a print job directly to a queue on a remote machine without going through the hassle of configuring lpd to handle it. This is mostly useful in situations where you print to a variety of printers only occasionally. From the announcement for*rlpr*:

Rlpr uses TCP/IP to send print jobs to lpd servers anywhere on a network.

Unlike lpr, it *does not* require that the remote printers be explicitly known to the machine you wish to print from, (e.g. through */etc/printcap*) and thus is considerably more flexible and requires less administration.

rlpr can be used anywhere a traditional lpr might be used, and is backwards compatible with traditional BSD lpr.

The main power gained by rlpr is the power to print remotely *from anywhere to anywhere* without regard for how the system you wish to print from was configured. Rlpr can work as a filter just like traditional lpr so that clients executing on a remote machine like netscape, xemacs, etc, etc can print to your local machine with little effort.

Rlpr is available from `Metalab`.

11.2. Printing to a Windows or Samba printer

There is a Printing to Windows mini−HOWTO out there which has more info than there is here.

11.2.1. From LPD

It is possible to direct a print queue through the `smbclient` program (part of the Samba suite) to a TCP/IP based SMB print service. Samba includes a script to do this called `smbprint`. In short, you put a configuration file for the specific printer in question in the spool directory, and install the `smbprint` script as the*if*.

The */etc/printcap* entry goes like this:

```
lp|remote−smbprinter:\
    :sh:\
    :lp=/dev/null:\
    :sd=/var/spool/lpd/lp:\
    :if=/usr/local/sbin/smbprint:
```

You should read the documentation inside the `smbprint` script for more information on how to set this up.

You can also use `smbclient` to submit a file directly to an SMB printing service without involving `lpd`. See the man page.

11.3. Printing to a NetWare Printer

The ncpfs suite includes a utility called `nprint` which provides the same functionality as `smbprint` but for NetWare. You can get ncpfs from Metalab. From the LSM entry for version 0.16:

"With ncpfs you can mount volumes of your NetWare server under Linux. You can also print to NetWare print queues and spool NetWare print queues to the Un*x print spooler. You need kernel 1.2.x or 1.3.54 and above. ncpfs does NOT work with any 1.3.x kernel below 1.3.54."

11.3.1. From LPD

To make `nprint` work via lpd, you write a little shell script to print stdin on the NetWare printer, and install that as the *if* for an lpd print queue. You'll get something like:

```
sub2|remote−NWprinter:\
        :sh:\
        :lp=/dev/null:\
        :sd=/var/spool/lpd/sub2:\
        :if=/var/spool/lpd/nprint−script:
```

The `nprint−script` might look approximately like:

```
#! /bin/sh
# You should try the guest account with no password first!
/usr/local/bin/nprint −S net −U name −P passwd −q printq−name −
```

11.4. Printing to an EtherTalk (Apple) printer

The netatalk package includes something like `nprint` and `smbclient`. Others have documented the procedure for printing to and from an Apple network far better than I ever will; see the Linux Netatalk−HOWTO.

11.5. Printing to a networked printer

Many printers come with an ethernet interface which you can print to directly, typically using the LPD protocol. You should follow the instructions that came with your printer or its network adaptor, but in general, such printers are "running" lpd, and provide one or more queues which you can print to. An HP, for example, might work with a printcap like:

```
lj−5|remote−hplj:\
        :sh:\
        :sd=/var/spool/lpd/lj−5:\
        :rm=printer.name.com:\
        :rp=raw:
```

HP Laserjet printers with JetDirect interfaces generally support two built in lpd queues − "raw" which accepts PCL (and possibly Postscript) and "text" which accepts straight ascii (and copes automatically with the staircase effect). If you've got a JetDirect Plus3 three−port box, the queues are named "raw1", "text2", and so forth.

Note that the ISS company has identified an assortment of denial of service attacks which hang HP Jetdirect interfaces. Most of these have been addressed beginning in Fall 98. These sorts of problems are common in embedded code; few appliance−style devices should be exposed to general Internet traffic.

In a large scale environment, especially a large environment where some printers do not support PostScript, it may be useful to establish a dedicated print server to which all machines print and on which all ghostscript jobs are run. This will allow the queue to be paused or reordered using the topq and lprm commands.

This also allows your GNU/Linux box to act as a spool server for the printer so that your network users can complete their print jobs quickly and get on with things without waiting for the printer to print any other job that someone else has sent. This is suggested too if you have unfixable older HP Jetdirects; it reduces the likelihood of the printers wedging.

To do this, set up a queue on your linux box that points at the ethernet equipped HP LJ (as above). Now set up all the clients on your LAN to point at the LPD queue (eg lj−5 in the example above).

Some HP network printers apparently don't heed the banner page setting sent by clients; you can turn off their internally generated banner page by telnetting to the printer, hitting return twice, typing "banner: 0" followed by "quit". There are other settings you can change this way, as well; type "?" to see a list.

The full range of settings can be controlled with HP's webJetAdmin software. This package runs as a daemon, and accepts http requests on a designated port. It serves up forms and Java applets which can control HP printers on the network. In theory, it can also control Unix print queues, but it does so using the rexec service, which is completely unsecure. I don't advise using that feature.

11.5.1. To AppSocket Devices

Some printers (and printer networking "black boxes") support only a cheesy little non−protocol involving plain TCP connections; this is sometimes called the "AppSocket" protocol. Notable in this category are early−model JetDirect (including some JetDirectEx) cards. Basically, to print to the printer, you must open a TCP connection to the printer on a specified port (typically 9100, or 9100, 9101 and 9102 for three−port boxes) and stuff your print job into it. LPRng has built−in support for stuffing print jobs into random TCP ports, but with BSD lpd it's not so easy. The best thing is probably to obtain and use the little utility called netcat.

Failing that, it can be implemented, among other ways, in Perl using the program below. For better performance, use the program netcat ("nc"), which does much the same thing in a general purpose way. Most distributions should have netcat available in prepackaged form.

11.6. Running an `if` for remote printers with old LPDs

One oddity of older versions of lpd is that the `if` is not run for remote printers. (Versions after 0.43 or so have the change originated on FreeBSD such that the `if` is always run). If you find that you need to run an`if` for a remote printer, and it isn't working with your lpr, you can do so by setting up a double queue and requeueing the job. As an example, consider this`printcap`:

```
lj−5:\
        :lp=/dev/null:sh:\
        :sd=/var/spool/lpd/lj−5:\
        :if=/usr/lib/lpd/filter−lj−5:
lj−5−remote:sh:rm=printer.name.com:\
        :rp=raw:sd=/var/spool/lpd/lj−5−raw:
```

in light of this **filter−lj−5** script:

```
#!/bin/sh
gs <options> −q −dSAFER −sOutputFile=− − | \
lpr −Plj−5−remote −U$5
```

The `−U` option to lpr only works if lpr is run as daemon, and it sets the submitter's name for the job in the resubmitted queue correctly. You should probably use a more robust method of getting the username, since in some cases it is not argument 5. See the man page for *`printcap`*.

11.7. From Windows.

Printing from a Windows (or presumably, OS/2) client to a Un*x server is directly supported over SMB through the use of the SAMBA package, which also supports file sharing of your Un*x filesystem to Windows clients.

Samba includes fairly complete documentation, and there is a good Samba FAQ which covers it, too. You can either configure a magic filter on the Un*x box and print PostScript to it, or run around installing printer−specific drivers on all the Windows machines and having a queue for them with no filters at all. Relying on the Windows drivers may in some cases produce better output, but is a bit more of an administrative hassle if there are many Windows boxes. So try Postscript first. Modern versions of Samba should support the automagical driver download mechanism offered by Windows NT servers to deal with this problem.

11.8. From an Apple.

Netatalk supports printing from Apple clients over EtherTalk. See the Netatalk HOWTO Page for more information.

Really, though, any modern Mac can print over TCP/IP using the LPD protocol just fine. UVa provides a very nice support page detailing how to set this up.

11.9. From Netware.

The ncpfs package includes a daemon named pserver which can be used to provide service to a NetWare print queue. From what I understand, this system requires a Bindery−based NetWare, eg 2.x, 3.x, or 4.x with bindery access enabled.

For more information on ncpfs and it's pserver program, see the ncpfs FTP site.

11.10. Networked Printer Administration

Most networked printers support some method of remote administration. Often there are easy−to−use web pages for configuration. More usefully, there is often support for SNMP management. Typically you can find out interesting information on printer status like ink and paper levels, print volumes, and so forth, and you can usually change certain settings. SNMP printer control, and a number of other printing−related things, are being standardized by the IEEE's Printer Working Group

11.10.1. npadmin

Npadminis a command−line program which offers an interface to the common SNMP functionality of networked printers. It implements the standard Printer MIB, as well as a few vendor−proprietary schemes used mainly for older devices. Both printer−discovery style actions and various printer status queries are supported.

npadmin has an excellent man page, and precompiled packages are distributed for a number of RPM and dpkg based distributions.

11.10.2. Other SNMP tools

Besides npadmin, there are a number of SNMP tools that will be useful. **snmptraplogd** can log SNMP trap events. This is useful for observing printer jams, out of paper events, etc; it would be straightforward to retransmit certain events to a pager, or to send an email.

While npadmin provides simplified support for many network printers' SNMP interfaces, some printers may have vendor extensions which npadmin doesn't know about. In this case, you can use the CMU SNMP tools, which support arbitrary SNMP GET and SET operations, as well as walks and the like. With these, and a bit of work, you can make use of any SNMP feature offered by your printer's MIB. You may need to obtain a MIB from your vendor to figure out what all the variables are; sometimes vendors think that people actually use the proprietary tools they ship.

VA Linux's **libprinterconf** includes code to perform network printer discovery. Printers are identified against a compiled−in library of printer signatures; at the moment the library is not large, but does cover many common networked printer models.

12. Windows−only printers

As I discussed earlier, some printers are inherently unsupported because they don't speak a normal printer language, instead using the computer's CPU to render a bitmap which is then piped to the printer at a fixed speed. In a few cases, these printers also speak something normal like PCL, but often they do not. In some (really low−end) cases, the printer doesn't even use a normal parallel connection but relies on the vendor's driver to emulate what should be hardware behavior (most importantly flow control).

In any case, there are a few possible workarounds if you find yourself stuck with such a lemon.

12.1. The Ghostscript Windows redirector

There is now a Ghostscript printer driver available (called **mswinpr2**) that will print using Windows GDI calls. There is also a port redirection tool called **redmon** which will run a print job through Ghostscript before finally printing it. (Rather like an `if` filter in Unix's LPD). Taken all together, this allows a Windows machine to print PostScript to a Windows−only printer through the vendor's driver.

If you have a host−based printer that can't be used directly, you can export it as a "Postscript" printer by using redmon, Ghostscript, and mswinpr2 on a Windows PC and print through the vendor's drivers.

12.2. HP Winprinters

Some HP printers use "Printing Performance Architecture" (marketing speak for "we were too cheap to implement PCL"). This is supported in a roundabout way via the pbm2ppa translator written by Tim Norman. Basically, you use ghostscript to render PostScript into a bitmapped image in pbm format and then use pbm2ppa to translate this into a printer−specific ppa format bitmap ready to be dumped to the printer. This program may also come in ghostscript driver format by now.

The ppa software can be had from the ppa home page; pbm2ppa supports some models of the HP 720, 820, and 1000; read the documentation that comes with the package for more details on ppa printer support.

12.3. Lexmark Winprinters

Most of the cheap Lexmark inkjets use a proprietary language and are therefore Winprinters. However, Henryk Paluch has written a program which can print on a Lexmark 7000. Hopefully he'll be able to figure out color and expand support to other Lexmark inkjets. See here for more info.

Similarly, there are now drivers for the 5700, 1000, 1100, 2070, 3200, and others. See the supported printers listing above, and my web site, for more information on obtaining these drivers.

13. How to print to a fax machine.

You can print to a fax machine with, or without, a modem.

13.1. Using a faxmodem

There are a number of fax programs out there that will let you fax and receive documents. One of the most powerful is Sam Leffler's HylaFAX. It supports all sorts of things from multiple modems to broadcasting.

SuSE ships a Java HylaFax client which allegedly works on any Java platform (including Windows and GNU/Linux). There are also non−Java fax clients for most platforms; GNU/Linux can almost certainly handle your network faxing needs.

Also available, and a better choice for smaller installations, is **efax**, a simple program which sends and receives faxes. The getty program **mgetty** can receive faxes using **efax** (and do voicemail or interactive logins).

13.2. Using the Remote Printing Service

There is an experimental service offered that lets you send an email message containing something you'd like printed such that it will appear on a fax machine elsewhere. Nice formats like postscript are supported, so even though global coverage is spotty, this can still be a very useful service. For more information on printing via the remote printing service, see the Remote Printing WWW Site.

13.3. Commercial Faxing Services

A number of companies operate web−based faxing services. EFax, in particular, offers free inbound faxes (to your own dedicated fax number, no less) via email, and fax transmission for a fee. Other companies offer similar services.

---

14. How to generate something worth printing.

Here we get into a real rat's−nest of software. Basically, Linux can run many types of binaries with varying degrees of success: Linux/x86, Linux/Alpha, Linux/Sparc, Linux/foo, iBCS, Win16/Win32s (with dosemu and, someday, with Wine), Mac/68k (with Executor), and Java. I'll just discuss native GNU/Linux and common Un*x software.

14.1. Markup languages

Most markup languages are more suitable for large or repetitive projects, where you want the computer to control the layout of the text to make things uniform.

nroff

This was one of the first markup languages on the original version of Unix. Man pages are the most common examples of things formatted in *roff macros; many people swear by them, but nroff has, to me at least, a more arcane syntax than needed (see Figure 12), and probably makes a poor choice for new works. It is worth knowing, though, that you can typeset a man page directly into postscript with groff. Most man commands will do this for you with **man −t foo | lpr**.

Figure 12. Example of roff Input

```
.B man
is the system's manual pager. Each
.I page
argument given to
.B man
is normally the name of a program, utility or function.
The
.I manual page
associated with each of these arguments is then found and
displayed. A
.IR section ,
if provided, will direct
.B man
to look
only in that
.I section
of the manual.
```

TeX

TeX, and the macro package LaTeX, are one of the most widely used markup languages on Un*x systems, although TeX did not originate on Unix and is available to run on a wide variety of systems. Technical works are frequently written in LaTeX because it greatly simplifies the layout issues and is*still* one of the few text processing systems to support mathematics both completely and well. TeX's output format is `dvi`, and is converted to PostScript or Hewlett Packard's PCL with **dvips** or **dvilj**. If you wish to install TeX or LaTeX, install the whole teTeX group of packages; it contains everything. Recent TeX installations include pdfTeX and pdfLaTeX, which produce Adobe PDF files directly. Commands are available do create hyperlinks and navigation features in the PDF file.

Figure 13. Example of LaTeX Input

```
\subsubsection{NAT}

Each real server is assigned a different IP address, and the NA
implements address translation for all inbound and outbound
packets.

  \begin{description}
  \item[Advantage] Implementation simplicity, especially if we
already implement other NAT capabilities.

  \item[Disadvantage] Return traffic from the server goes through
address translation, which may incur a speed penalty.  This
probably isn't too bad if we design for it from the
beginning.

  \item[Disadvantage] NAT breaks the end−to−end semantics of normal
internet traffic.  Protocols like ftp, H.323, etc would
require special support involving snooping and in−stream
rewriting, or complete protocol proxying; neither is likely
to be practical.
  \end{description}
```

SGML

There is at least one free SGML parser available for Un*x systems; it forms the basis of Linuxdoc−SGML's homegrown document system. It can support other DTD's, as well, most notably DocBook. This document is written in DocBook−DTD SGML; see Figure 14 for an example.

Figure 14. Example of DocBook SGML

```
<varlistentry>
<term>SGML</term>
<listitem>
<para>
There is at least one free SGML parser available for Un*x
systems; it forms the basis of Linuxdoc−SGML's homegrown
document system.  It can support other DTD's, as well, most
notably DocBook.  This document is written in DocBook−DTD
SGML.
</para>
</listitem>
</varlistentry>
```

14.2. WYSIWYG Word Processors

There is no shortage of WYSIWYG word processing software. Several complete office suites are available, including one that's free for personal use (StarOffice).

StarOffice

This full−blown office suite has all the features you'd expect, including both import and export of Microsoft Office file formats (including Word documents). There's a mini−HOWTO out there which describes how to obtain and install it. It generates PostScript, so should work with most any printer that works otherwise on GNU/Linux.

WordPerfect

Corel distributes a basic version of WordPerfect 8 free for GNU/Linux, and sells various packages of Word Perfect Office 2000 (which includes WordPerfect, Corel Draw and Quattro Pro Versions 9). The Linux WordPerfect Fonts and Printers page has information about configuring WordPerfect for use with either Ghostscript or its built−in printer drivers (which are apparently identical the DOS WordPerfect drivers, if your printer's driver isn't included in the distribution).

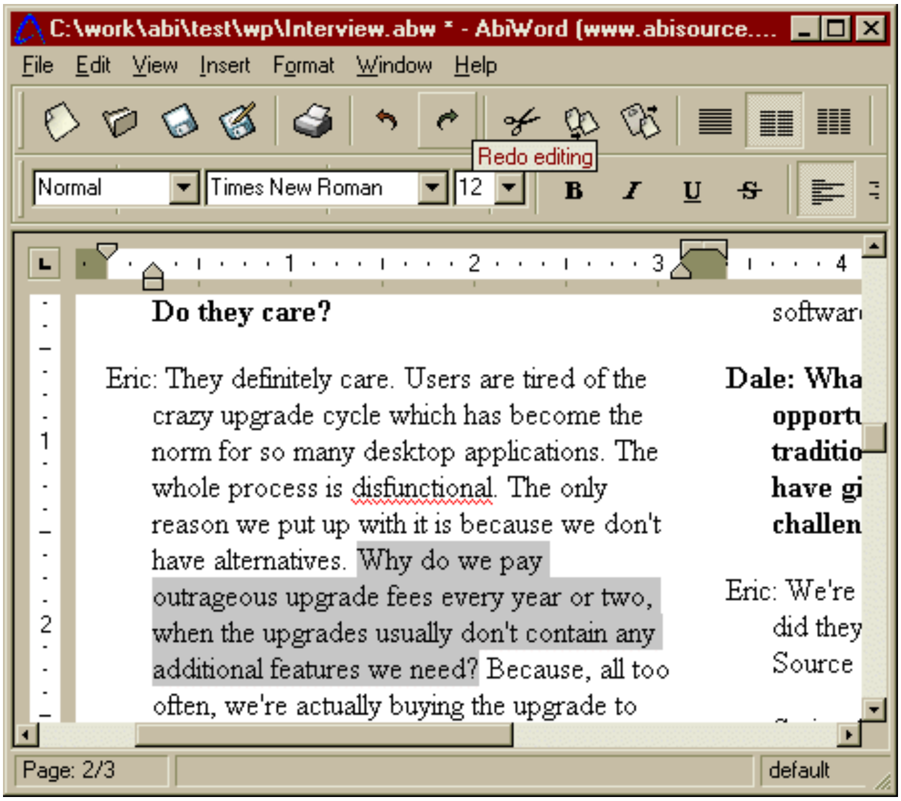
Applix

Applix is a cross−platform (eg, various Unices, Windows, and others) office suite sold by the Applix company. Red Hat and SuSE sold it themselves when it was the only game in town; now sales have reverted to Applix. This is the only native Unix−style application suite; it probably fits in better with the Unix way of doing things.

AbiWord

AbiWord is one of several GPL WYSIWYG word processor projects; this one has produced a very nice word processor based on an XML format. It is capable of Word file import. AbiWord is still a work in progress, although it is useful for small things now.

Figure 15. AbiWord



Do they care?

Eric: They definitely care. Users are tired of the crazy upgrade cycle which has become the norm for so many desktop applications. The whole process is disfunctional. The only reason we put up with it is because we don't have alternatives. Why do we pay outrageous upgrade fees every year or two, when the upgrades usually don't contain any additional features we need? Because, all too often, we're actually buying the upgrade to

LyX

LyX is a front−end to LaTeX which looks very promising. See the LyX Homepage for more information. There is a KDE−styled version of LyX, called Klyx; the author of LyX and the instigator of KDE are the same person.

Figure 16. LyX

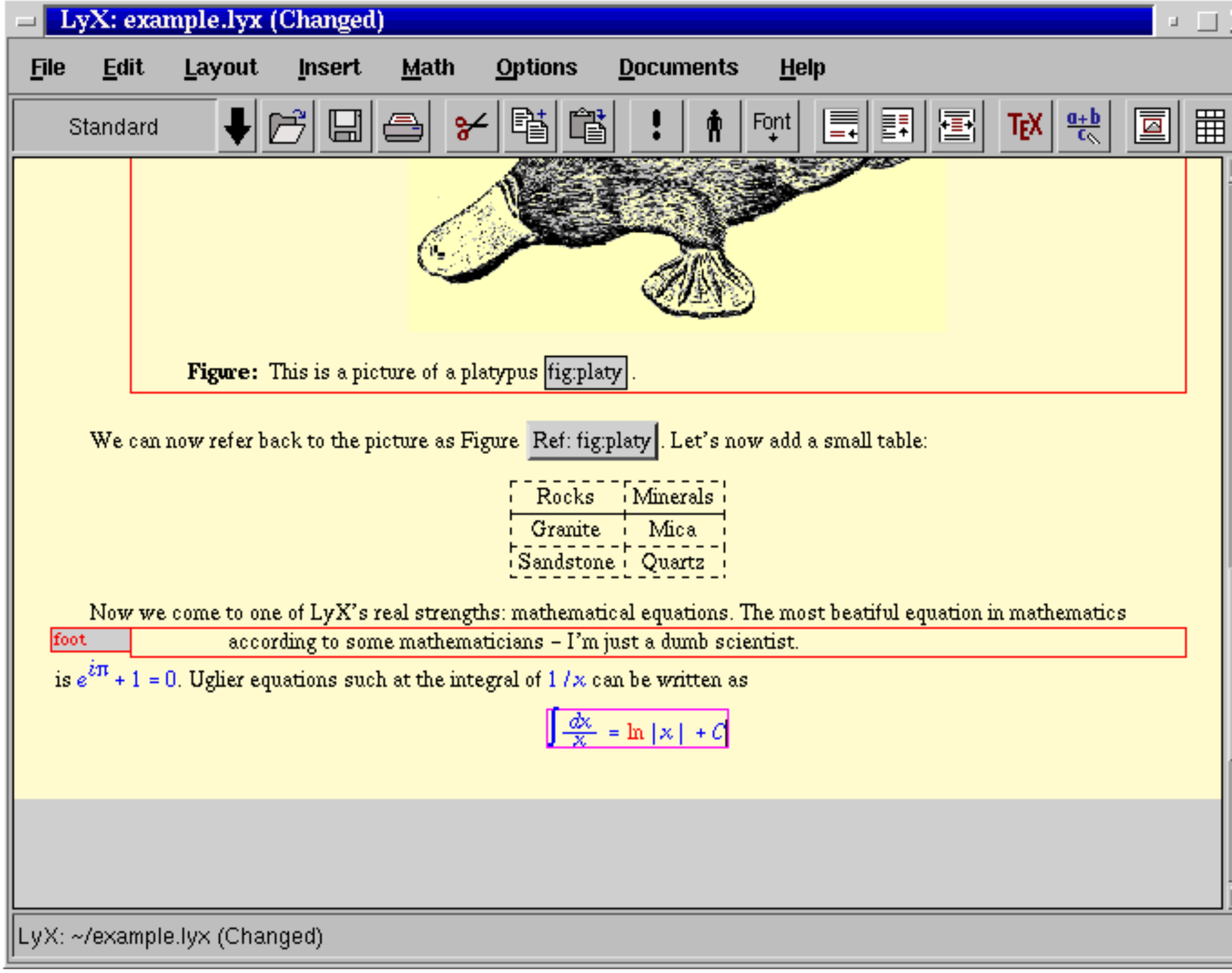


Figure: This is a picture of a platypus fig:platy .

We can now refer back to the picture as Figure Ref: fig:platy. Let's now add a small table:

Rocks	Minerals
Granite	Mica
Sandstone	Quartz

Now we come to one of LyX's real strengths: mathematical equations. The most beatiful equation in mathematics

according to some mathematicians – I'm just a dumb scientist.

is $e^{i\pi} + 1 = 0$. Uglier equations such at the integral of $1/x$ can be written as

$$\int \frac{dx}{x} = \ln |x| + C$$

Maxwell

Maxwell is a simple MS RTF−format based word processor which started as a commercial product but is now distributed under the GPL.

Other vendors should feel free to drop me a line with your offerings.

---

15. Printing Photographs

There are many details to getting decent photo output from common printers. If you haven't bought a photo printer yet, see the photo−related tips in Section 5.4.

15.1. Ghostscript and Photos

Ghostscript has some difficulties rendering color photographs through most drivers. The problems are several:

- Many drivers have poorly tuned color support. Often the colors don't match the Windows driver output or the screen. OTOH, all drivers, and Ghostscript as a whole, have readily adjustable color support; the "Gamma" settings (see Section 10.2.2) are one thing to play with, and there are others documented in Ghostscript's `Use.htm` documentation file.
- I'm only aware of one Ghostscript driver with support for 6 and 7 color printing; it's in beta at the moment and supports most Epson Stylus Photo models. It is rumored to produce better color than the Windows driver (!). The Ghostscript driver core itself provides no support for non CMYK or RGB colors; arguably, some work to put that there is needed.
- Ghostscript often ends up dithering coarsely, or generating printouts with artifacts like banding. The dithering can usually be corrected; see Section 10.2.3, and read the documentation for your driver.

You should be able to correct some of these problems by tuning Ghostscript; see Section 10 for more information on how to do this. Fiddling with Ghostscript options is much easier if you declare them as options in your spooling system.

That said, the obvious solution for now is to use non−Ghostscript software for printing photos, and indeed, such things do exist. The main contender is the print plugin in the Gimp, which supports pixel−for−pixel printing on Epson Styluses and Postscript printers (with basic PPD support). That Epson Stylus portion of that driver is available for Ghostscript, as well, as the **stp** driver. Also possible to use for this purpose are the assorted external pnm−to−foo programs used to print on printers like the cheap Lexmarks; these print attempt to print pixmaps pixel−for−pixel.

The best solution, of course, is to buy a Postscript printer; such printers can usually be completely controlled from available free software, and will print to the full capability of the printer.

15.2. Paper

Color inkjets are extremely dependent on the paper for good output. The expensive glossy coated inkjet papers will allow you to produce near−photographic output, while plain uncoated paper will often produce muddy colors and fuzzy details. Non−glossy coated inkjet papers will produce results in between, and are probably best for final prints of text, as well. Stiffer glossy coated "photo" papers will produce similar output to lighter−weight glossy papers, but will feel like a regular photo.

15.3. Printer Settings

For photo output on most color inkjets, you should use the most highly interlaced (and slowest) print mode; otherwise solid regions may have banding or weak colors. Generally with Ghostscript this is what will happen when you pick the highest resolution. With Postscript printers, you may need to add a snippet to the prologue based on the settings available in the PPD file. The Gimp's PPD support doesn't include (printer−specific) print quality settings, but I added one in an ugly way for my own use; contact me if you'd like that. If you use PDQ or CUPS, you can easily control all the printer settings you need. VA Linux's**libppd** and the GPR front−end can also add these options for Postscript printers.

15.4. Print Durability

Color inkjet printouts usually fade after a few years, especially if exposed to lots of light and air; this is a function of the ink. Printers with ink−only consumables like the Epsons and Canons can buy archival inks, which are less prone to this problem. Newer printers often use pigment−based inks, which don't fade as much as the older dye−based ink did. No inkjet output is really particularly good for long−term archival use. Write the bits to a CD−R and store that instead.

15.5. Shareware and Commercial Software

There's a program called xwtools which supports photo printing with all the bells and whistles on an assortment of Epson, HP, and Canon printers. Unfortunately, it was written under NDA, so comes without source. Unless you use it for the Epson Stylus Color 300 on GNU/Linux x86, it costs E15 for personal use; commercial pricing is unknown.

The ESP Print Pro package from Easy Software supports some printers which might otherwise be unsupported. These drivers are not reported to be very well−tuned for photos, but they do work.

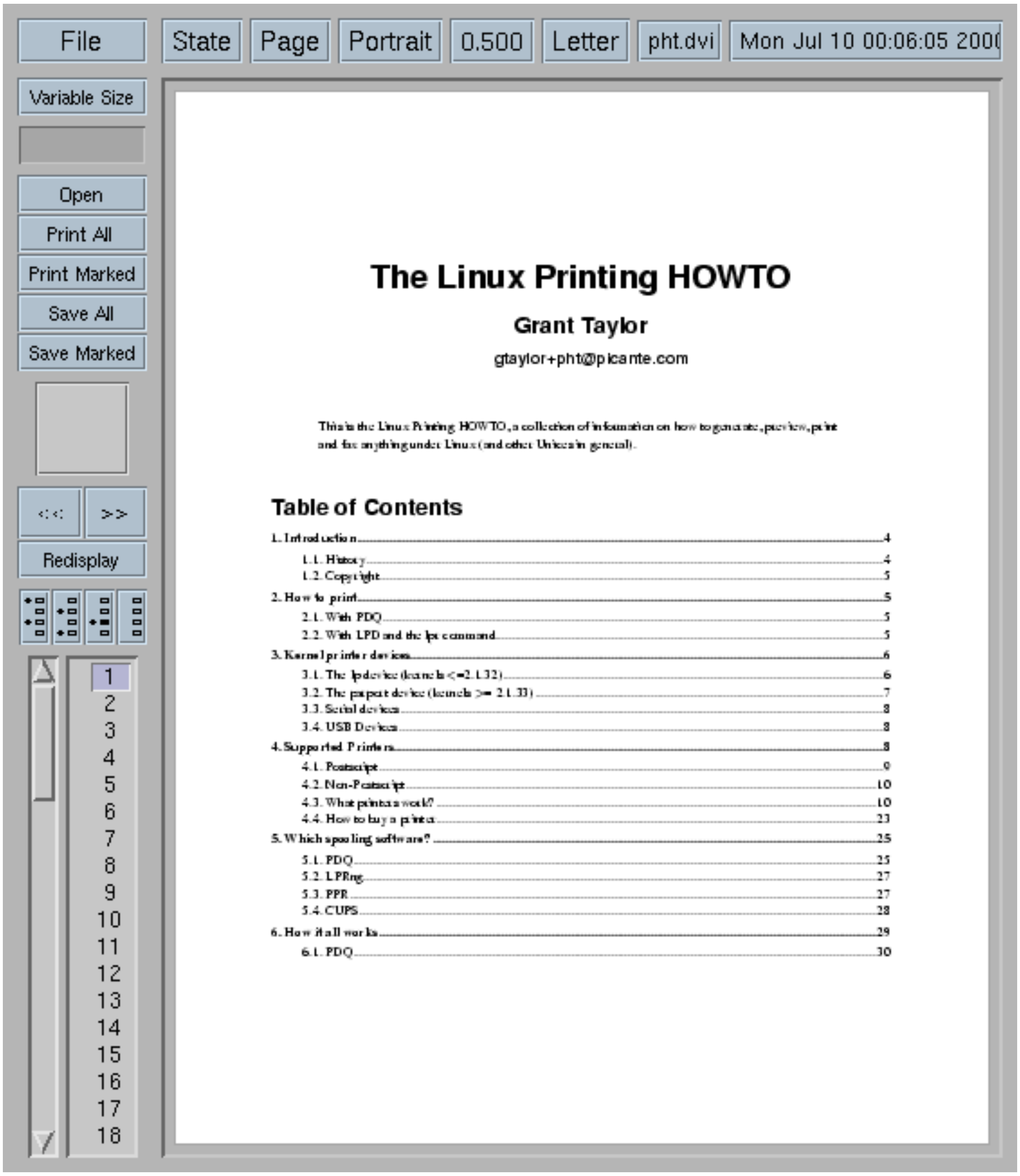
16. On−screen previewing of printable things.

Nearly anything you can print can be viewed on the screen, too.

16.1. PostScript

Ghostscript has an X11 driver best used under the management of the PostScript previewer gv. The latest versions of these programs should be able to view PDF files, as well. Note that gv has replaced the older previewer "Ghostview"; the new user interface is much prettier and featureful that ghostview's plain old Athena GUI.

Figure 17. Gv



The Linux Printing HOWTO

Grant Taylor

gtaylor+pht@picante.com

This is the Linux Printing HOWTO, a collection of information on how to generate, preview, print and fax anything under Linux (and other Unices in general).

Table of Contents

16.2. TeX dvi

TeX DeVice Independent files may be previewed under X11 withxdvi. Modern versions of xdvi call ghostscript to render PostScript specials.

A VT100 driver exists as well. It's called `dgvt`. `Tmview` works with GNU/Linux and svgalib, if that's all you can do.

16.3. Adobe PDF

Adobe's Acrobat Reader is available for GNU/Linux; just download it from the Adobe web site.

You can also use xpdf, which is free software, and I believe**gv** supports viewing PDF files with gs under X11.

17. Serial printers under lpd

Serial printers are rather tricky under lpd.

17.1. Setting up in printcap

Lpd provides five attributes which you can set in*/etc/printcap* to control all the settings of the serial port a printer is on. Read the *printcap* man page and note the meanings of*br#*, *fc#*,*xc#*, *fs#* and*xs#*. The last four of these attributes are bitmaps indicating the settings for use the port. The*br#* attribute is simply the baud rate, eg `` `br#9600` ``'.

It is very easy to translate from stty settings to printcap flag settings. If you need to, see the man page for stty now.

Use stty to set up the printer port so that you can cat a file to it and have it print correctly. Here's what `*stty −a*' looks like for my printer port:

```
dina:/usr/users/andy/work/lpd/lpd# stty −a < /dev/ttyS2
speed 9600 baud; rows 0; columns 0; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>;
eol2 = <undef>; start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R; werase = ^W;
lnext = ^V; min = 1; time = 0;
−parenb −parodd cs8 hupcl −cstopb cread −clocal −crtscts
−ignbrk −brkint −ignpar −parmrk −inpck −istrip −inlcr
−igncr −icrnl ixon −ixoff −iuclc −ixany −imaxbel
−opost −olcuc −ocrnl −onlcr −onocr −onlret −ofill −ofdel nl0 cr0 tab0
bs0 vt0 ff0
−isig −icanon −iexten −echo −echoe −echok −echonl −noflsh −xcase
−tostop −echoprt −echoctl −echoke
```

The only changes between this and the way the port is initialized at bootup are `−clocal`, `−crtscts`, and `ixon`. Your port may well be different depending on how your printer does flow control.

You actually use stty in a somewhat odd way. Since stty operates on the terminal connected to it's standard input, you use it to manipulate a given serial port by using the `` `<` ``' character as above.

Once you have your stty settings right, so that `*cat file > /dev/ttyS2*' (in my case) sends the file to the printer, look at the file /usr/src/linux/include/asm−i386/termbits.h. This contains a lot of #defines and a few structs (You may wish to cat this file to the printer (you do have that working, right?) and use it as scratch paper). Go to the section that starts out

```
/* c_cflag bit meaning */
#define CBAUD   0000017
```

This section lists the meaning of the *fc#* and *fs#* bits. You will notice that the names there (after the baud rates) match up with one of the lines of stty output. Didn't I say this was going to be easy?

Note which of those settings are preceded with a − in your stty output. Sum up all those numbers (they are octal). This represents the bits you want to clear, so the result is your*fc#* capability. Of course, remember that you will be setting bits directly after you clear, so you can just use `` `fc#0177777` ``' (I do).

Now do the same for those settings (listed in this section) which do not have a − before them in your stty output. In my example the important ones are CS8 (0000060), HUPCL (0002000), and CREAD (0000200). Also note the flags for your baud rate (mine is 0000015). Add those all up, and in my example you get 0002275. This goes in your *fs#* capability (``fs#02275`' works fine in my example).

Do the same with set and clear for the next section of the include file, "c_lflag bits". In my case I didn't have to set anything, so I just use ``xc#0157777`' and ``xs#0`'.

17.2. Older serial printers that drop characters

Jon Luckey points out that some older serial printers with ten−cent serial interfaces and small buffers*really* mean stop when they say so with flow control. He found that disabling the FIFO in his Linux box's 16550 serial port with `setserial` corrected the problem of dropped characters (you apparently just specify the UART type as an 8250 to do this).

18. What's missing?

Many of the parts for a complete printing system do not exist yet. Projects are underway to address most of these, although most have not yet produced running useful code, and efforts to standardize the necessary protocols and APIs are in their infancy.

18.1. Plumbing

There's a general problem with getting all the parts to talk to one another; especially in a spooler−independent way. This problem manifests itself most noticeably in the pathetic application support for control over all the "usual" printing features. There is simply no way for an application writer to get information about printers, jobs, etc; no standardized way to submit jobs; no good way to get job status back; nor even really a standardized way to generate print data (although most of the new desktop systems offer desktop−specific facilities for doing this).

18.2. Fonts

Font handling on free systems is rather awkward. The display, the printer, the application, and the data files should ideally all have access to the same fonts. Unfortunately this was simply not the case. With the advent of xft2 and fontconfig − which the newest distributions will start deploying − this should finally be solved. Redhat 8.0 is AFAIK the first distro that uses xft2.

18.3. Drivers

There is still some work to be done on free software drivers. Although the drivers have improved a lot the last several years, not all printers are supported.

Printer vendors have a major role to play in this area. With the increasing popularity of Linux it is getting really hard for them to simple ignore this userbase.

19. Credits

Special thanks to Grant Taylor for creating this HOWTO and to Till Kampeter for foomatic and his expert advice.

The `smbprint` information is from an article by Marcel Roelofs `<marcel@paragon.nl>`.

The `nprint` information for using Netware printers was provided by Michael Smith `<mikes@bioch.ox.ac.uk>`.

The serial printers under lpd section is from Andrew Tefft `<teffta@engr.dnet.ge.com>`.

The blurb about gammas and such for gs was sent in by Andreas`<quasi@hub−fue.franken.de>`.

The two paragraphs about the 30 second closing_wait of the serial driver was contributed by Chris Johnson `<cdj@netcom.com>`.

Robert Hart sent a few excellent paragraphs about setting up a print server to networked HPs which Grant used verbatim.

And special thanks to the dozens upon dozens of you who've pointed out typos, bad URLs, and errors in the document over the years.

A. GNU Free Documentation License

Version 1.2, November 2002

A.1. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

A.2. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world−wide, royalty−free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front−matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front−Cover Texts or Back−Cover Texts, in the notice that says that the Document is released under this License. A Front−Cover Text may be at most 5 words, and a Back−Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine−readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard−conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine−generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

A.3. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

A.4. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front−Cover Texts on the front cover, and Back−Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine−readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer−network location from which the general network−using public has access to download using public−standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

A.5. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
D. Preserve all the copyright notices of the Document.
E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
H. Include an unaltered copy of this License.
I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front−matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties−−for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front−Cover Text, and a passage of up to 25 words as a Back−Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front−Cover Text and one of Back−Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

A.6. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

A.7. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

A.8. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

A.9. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

A.10. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

A.11. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

A.12. ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front−Cover Texts, and no Back−Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front−Cover Texts and Back−Cover Texts, replace the "with...Texts." line with this:

> with the Invariant Sections being LIST THEIR TITLES, with the Front−Cover Texts being LIST, and with the Back−Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.