

The Common UNIX Printing System

PRINT-O-MATIC

ULRICH WOLF



CUPS is the printing system of the Internet era. Our overview shows what can now be done with it, and what's going on behind the scenes.

CUPS implements the Internet Printing Protocol (IPP) in Version 1.1. It can completely supersede, both locally as well as on the network, the ageing Line Printer Daemon. A few Linux distributions such as Mandrake, Caldera, EasyLinux or Connectiva have already stopped installing the LPD. And in the latest SuSE variants CUPS can now be configured with YaST.

On stand-alone workstations CUPS exhausts all the individual properties of the existing printer, but its real strengths only come into play on the network. This is where, on any free printer, it is possible to print using all its usual device options, without a driver having been installed on the client. More than that, the client automatically recognises which printers are installed in the network, provided these are attached to a CUPS server.

Architecture like a webserver

The beating heart of CUPS is the server daemon or scheduler *cupsd*, which is similar to an ordinary webserver. It can be addressed with any browser via the reserved port 631. Clients also communicate with the server via this port. But the exchange of data with the printers connected to the server can occur in a variety of ways. Ideally these can even cope with IPP, but this is only the case in some 200 models. Good laser printers such as the ones in our test are more likely to be able to use the Appsocket protocol, free printers on Windows computers can be addressed with Samba via SMB. For the rest LPD is used. Communication with the printers via these protocols is done by back-ends, which are separately configured.

Gimp Print for inkjet printers, a universal driver with special abilities

- About 120 inkjet printers from the companies Epson, HP, Canon and Lexmark are supported by the latest Alpha version (4.1.99-a2) of the Gimp Print Project (last stable version is the 4.0.5).
- Gimp Print includes a joint code base for three different target projects: GIMP, Ghostscript and CUPS.
- `./configure --help` displays the options. The CUPS system can be expanded using `--with-cups`. The option `--with-ghost --with-foomatic` generates the source code files which are merged with the Ghostscript source code. Then Ghostscript is recompiled. As a new component it now includes *stp*. The CUPS/Gimp Print filter together with the PPDs created and the normal Ghostscript filter, here under the name *stp*, can be used with CUPS.

The following points are important to remember:

- The PPDs must match the filter: the foomatic files belong to *stp* as can be seen as `GIMPPRINT+foomatic` or `stp+foomatic`
- The CUPS form of Gimp Print is called `CUPS + gimp-print`
- Only copy PPDs from other computers with the appropriate conversion filter
- Avoid Gimp Print, Foomatic or *stp* and associated PPDs in PostScript printers

CUPS and Postscript

Anyone printing on UNIX systems cannot avoid Postscript; CUPS is no exception to this rule. So the simplest way is to use Postscript-compatible printers. Their integrated processors process PS files into raster data in the highly calculation-intensive Raster Image Process (RIP). For printers which cannot cope with Postscript, a computer has to do it.

There are often dedicated RIP servers in the preliminary print stage. In a company network, where there are printers which are not Postscript-compatible being used regularly, the print server takes over these tasks and should therefore have sufficient computing power. CUPS is flexible enough to allow the preparation of print data on the client. But then the client obviously needs its own driver. The characteristics of each printer model are defined in a Postscript Printer Description (PPD) in ASCII format.

Printing without Postscript devices

CUPS filters, are based on the older Ghostscript version 5.5, but have been modified and improved. Some filters are in place after the standard installation: for PCL-compatible inkjet and laser printers from the Deskjet and Laserjet families of Hewlett-Packard, Epson Stylus Color and Stylus Photo and 9-pin and 24-pin printers from Epson. These are usually found, together with the filters derived from Ghostscript, in the directory `/usr/lib/cups/filter`.

For other, non-Postscript-compatible printers there is CUPS-O-Matic. This is a version of the Foomatic script collection adapted to CUPS by the Linux Printing HOWTO author Grant Taylor. It covers all Ghostscript filters for use with the CUPS system and thus requires that Ghostscript is installed on the print server. On the CUPS-O-Matic website, you first create a PPD for the existing printers in combination with a Ghostscript filter. From the point of view of the user, this is already the printer driver and can be installed in the CUPS system by the usual methods (command line, GUI or browser). Internally, however, you will need `cupsomatic`, a Perl script which must sit in the filter directory `/usr/lib/cups/filter`. With the aid of this script the printer-specific data from the PPD is sent to the relevant Ghostscript filter of the system, which then prepares the print data accordingly.

CUPS and Ghostscript

Cupsomatic passes the print data out of the usual CUPS environment to the Ghostscript filter specified in the CUPS-O-Matic-PPD and returns the result to CUPS. With CUPS-O-Matic, then, CUPS prints as good as any other Ghostscript systems, but all network features of CUPS for the printers supported by Ghostscript are available. Apart from CUPS-O-Matic there are other options for integrating filters in CUPS, namely for Turboprint and Gimp-Print.

Big print runs

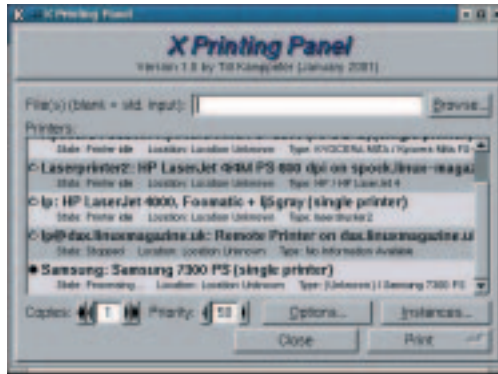
If there are several printers of the same type in a network, or if certain printers are only available for a short time, these can be combined into classes. If a print job is sent to a class, the job goes to the next free device. In this way, it is possible to achieve load balancing.

Configuration of CUPS

The CUPS print system can be monitored and administered via diverse command line commands. The main ones are `lpadmin`, `lpoptions` and `lpstat`. With `lpstat -p -d` the available printers are displayed. `lpadmin` configured printers, printer classes and user

Info

CUPS homepage	http://www.cups.org
IPP specification	http://www.ietf.org/html.charters/lipp-charter.html
Linux printing homepage/Cups-O-Matic	http://www.linuxprinting.org
Gimp Print	http://gimp-print.sourceforge.net/
CUPS documentation	http://www.cups.org/documentation.html

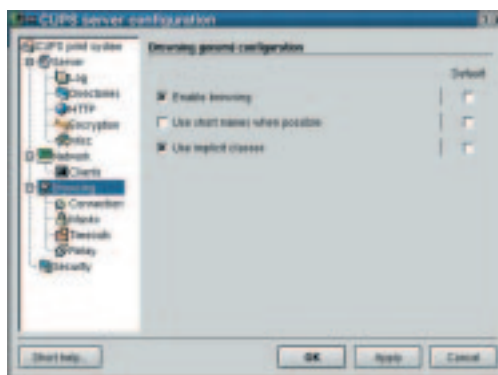


The graphical front-end Xpp is heavily based on the ESP PrintPro software of the inventor of CUPS, Michael Sweet

rights. CUPS has its own administrator, so needs no root rights. With `lpoptions` all the printer characteristics, which can be configured via the PPD, can be displayed and altered. But this is very tedious, since they vary from model to model. This is why in this case graphical tools like `xpp`, `gtklp`, `kups`, or the Web interface via `http://[computername]:631` are vital. These read out the respective PPDs and make them configurable with a mouse.

In KDE 2.2, KDEPrint is an intermediate layer between KDE applications and various printing subsystems. This means CUPS can be administered more simply in KDE applications. The former configuration tool `kups` is no longer being developed.

General settings for the CUPS print system can be found in `/etc/cups/cupsd.conf`. This is similar to the Apache configuration file `httpd.conf`. In `cupsd.conf` it is possible to announce printers to the whole network, using `Browsing On`, define browse relays for inter-network printing, options for authentication and encryption and much more. CUPS is largely ready to use with the default settings as soon as it has been installed. ■



In KDE 2.2 the CUPS configuration is integrated seamlessly, thanks to KDEPrint, into the system control