Linuxprinting + CUPS
Mini Tutorial

Till Kamppeter <till.kamppeter@gmx.net>, Mandriva SA, Paris, linuxprinting.org
(C) 2002 Till Kamppeter –– Distribute freely without change –– leave Copyright Note intact

Welcome Dinner, generously hosted by Ricoh Corporation USA, which also sponsored travel and hotel costs for various team members of the Linuxprinting.org booth. From left to right: Kurt Pfeifle (CUPS, Samba, KDEPrint and Linuxprinting documentation; FreeNX Development Team member); George Liu (Ricoh Inc.); Patrick Powell (LPRng & Foomatic developer); Uli Wehner (Lanier Inc., USA); ?? (??); Hitoshi Sekine (Ricoh Inc.); Cristian Tibirna (KDEPrint developer); Charles Samuels (KDE developer); Till Kamppeter (Mandriva + Linuxprinting.org + Foomatic + XPP Developer); David Suffield (HP Printing Software HPLIP developer); David's wife; ??(??)

Table of Contents

CUPS Printing Setup Mini−HUWTO 1

Table of Contents 3

CUPS Printing Setup Mini−HUWTO 5
- 1. Setting up a Print Queue with CUPS 5
  - 1. 1. Connecting your printer 5
  - 1. 2. Your distribution's printer setup tool 6
  - 1. 3. General printer setup with CUPS 6
  - 1. 4. PPDs for PostScript printers 7
  - 1. 5. Non−PostScript printers, GhostScript, and Foomatic 7
  - 1. 6. Multi−function devices from HP and Epson 9
- 2. Ethernet−Connected Printers 10
  - 2. 1. General 10
  - 2. 2. Assigning an IP address to a printer 10
  - 2. 3. Discovering the printer's network protocols 10
  - 2. 4. Setting up a print queue 11
- 3. Networked Printing with CUPS 12
  - 3. 1. Basic CUPS configuration for remote printer auto−discovery 12
  - 3. 2. High availability 14
  - 3. 3. Print quotas and accounting 15
  - 3. 4. LPD clients 15
- 4. Accessing Printers from Windows Clients via Samba 16
  - 4. 1. Sharing the CUPS queues to Windows clients 16
  - 4. 2. Emulate PostScript printers 18
  - 4. 3. More possibilities 19
- 5. CUPS Troubleshooting 19
  - 5. 1. RTFM − Read The F... Manual! 19
  - 5. 2. Supply all important information 19
  - 5. 3. Check device file permissions 19
  - 5. 4. Send print jobs with CUPS in debug mode 19
  - 5. 5. Where to ask 19
  - 5. 6. More about troubleshooting ... 20

CUPS Printing Setup Mini–HUWTO

Till Kamppeter <till.kamppeter@gmx.net>, Mandriva SA, Paris, linuxprinting.org
(C) 2002 Till Kamppeter –– Distribute freely without change –– leave Copyright Note intact

Table of Contents

CUPS Printing Setup Mini−HUWTO

- 1. Setting up a Print Queue with CUPS
  - 1. 1. Connecting your printer
  - 1. 2. Your distribution's printer setup tool
  - 1. 3. General printer setup with CUPS
  - 1. 4. PPDs for PostScript printers
  - 1. 5. Non−PostScript printers, GhostScript, and Foomatic
  - 1. 6. Multi−function devices from HP and Epson
- 2. Ethernet−Connected Printers
  - 2. 1. General
  - 2. 2. Assigning an IP address to a printer
  - 2. 3. Discovering the printer's network protocols
  - 2. 4. Setting up a print queue
- 3. Networked Printing with CUPS
  - 3. 1. Basic CUPS configuration for remote printer auto−discovery
  - 3. 2. High availability
  - 3. 3. Print quotas and accounting
  - 3. 4. LPD clients
- 4. Accessing Printers from Windows Clients via Samba
  - 4. 1. Sharing the CUPS queues to Windows clients
  - 4. 2. Emulate PostScript printers
  - 4. 3. More possibilities
- 5. CUPS Troubleshooting
  - 5. 1. RTFM − Read The F... Manual!
  - 5. 2. Supply all important information
  - 5. 3. Check device file permissions
  - 5. 4. Send print jobs with CUPS in debug mode
  - 5. 5. Where to ask
  - 5. 6. More about troubleshooting ...

CUPS Printing Setup Mini–HUWTO

Till Kamppeter, linuxprinting.org, Mandriva
till.kamppeter at gmx.net

1. Setting up a Print Queue with CUPS

1. 1. Connecting your printer

Connect your printer to the parallel port or USB. then restart the CUPS daemon with

```
killall -HUP cupsd
```

and check with

```
lpinfo -v
```

whether CUPS gas recognized your printer. You should get something like

```
...
direct parallel:/dev/lp0
...
direct usb://hp/dj450?serial=SG31K210C23S
...
```

If this is not the case check whether the printer is correctly recognized. On USB use the "`lsusb`" command:

```
[root@majax root]# lsusb
Bus 001 Device 016: ID 0924:3d53 Xerox
Bus 001 Device 014: ID 03f0:0512 Hewlett-Packard
Bus 001 Device 007: ID 03f0:1e11 Hewlett-Packard PSC-950
Bus 001 Device 005: ID 04b8:0112 Seiko Epson Corp. Perfection 2450
Bus 001 Device 003: ID 0451:2077 Texas Instruments, Inc. TUSB2077 Hub
Bus 001 Device 001: ID 0000:0000 [root@majax root]#
```

Your printer should lead to an entry here, also in the `/proc/bus/usb/devices` file it should appear. Check with "`lsmod`" whether the correct kernel modules (kernel 2.4.x: "`printer`", kernel 2.6.x: "`usblp`") are loaded and load them with "`modprobe <module>`" if needed.

For parellel printers check with "`/proc/sys/dev/parport/parport?/autoprobe*`":

```
[till@huxley till]$ cat /proc/sys/dev/parport/parport?/autoprobe*
CLASS:PRINTER;
MODEL:dj450;
MANUFACTURER:hp;
DESCRIPTION:hp dj450;
COMMAND SET:MLC,PCL,PML,BIDI-ECP,ECP18,DW-PCL,DYN,DESKJET;
[till@huxley till]$
```

If nothing is listed, reload the relevant kernel modules

```
modprobe -r lp parport_pc parport
modprobe lp
```

For USB printers the `/dev/` file system should contain

```
crw-rw---- 1 lp sys 180, 0 Jan 4 19:34 /dev/usb/lp0
crw-rw---- 1 lp sys 180, 1 Jan 4 19:34 /dev/usb/lp1 ...
```

for parallel

```
crw-rw---- 1 lp sys 6, 0 Jan 13 16:34 /dev/lp0
...
```

1. 2. Your distribution's printer setup tool

First, try to go the way your distribution does. Most printers can simply be set up by the tools coming with the distributions: "`printerdrake`" (or "`mcc`") on Mandriva Linux, "`yast2`" on SuSE, or "`printconf-gui`" on Red Hat or Fedora. They lead you step–by–step to the configuration of your printers, they auto–detect new printers, suggest the best driver, and they are even sometimes able to fully automatically set up all locally connected printers without any user interaction.

1. 3. General printer setup with CUPS

CUPS itself has two possibilities to add new print queues. The easier way is the built–in web interface of the CUPS system. You reach it with any web browser accessing the address:

```
http://localhost:631/
```

Choose "Do Administration Tasks", enter "`root` " and the root password in the login/password window, and then choose "Add Printer". Then you will be led step–by–step through the set up of the printer.

The name of the print queue should not contain other characters than letters, numbers, or underscores and not more than 12 characters. This assures compatibility with non–CUPS clients, especially Windows, and makes it easier to access the printer via the command line. "Location" and "Description" need not to be filled in, they are only comments, but filling them in makes it easier for users to find the most suitable printer in a network.

After clicking "Continue" you are asked for the "Device" for your queue. Choose the appropruate parallel or USB port. The entry should contain the model name of your printer.

On the next two pages choose manufacturer and model. If your model is not listed, follow the steps in the next sections.

If you come to the step "Printer ... has been added successfully" your queue is set up, click on the printer's name to get to the printer's administration page. Now click on "Configure Printer" and choose the default settings for it. Make sure that the default paper size is A4 when you are outside the US or Canada. Click on "Print Test Page" to check whether the printer really works.

Alternatively, you can set up a printer by the command line with the "`lpadmin`" command. See "man lpadmin" for more info.

NOTE:

- On SuSE systems another authentication method is used ("Digest"). To use the web interface, you must give a digest password to root at first, using the command "lppasswd":

  ```
  lppasswd -a -g sys root
  ```

  and enter a password (not necessarily the normal root password. This password has to be used for the web interface (or any other CUPS frontend) then.
- In certain error cases a print queue can get disabled and jobs stay waiting in the queue. To enable it, you need to run the command "`enable`" of CUPS:

  ```
  /usr/bin/enable <queue name>
  ```

  to re–enable the printer or click the green "Start Printer" button on the printer's page in the web interface.
- A very good graphical CUPS administration interface is the KDE Printing Manager. On every machine with installed KDE you find it in the KDE Control Center under "Peripherals" –> "Printers". Take care that at the bottom of the Window CUPS is chosen as your printing system. The "Administrator Mode" gives you the possibility to do several administration tasks without entering the root login and password again and again.

Operation should be intuitive, right−click on the printer queue entries and on the free area around them to get menues with all functions. On the web you will find information on

`http://printing.kde.org`

Further info you get in the CUPS documentation on

`http://localhost:631/documentation.html`

especially:

`http://localhost:631/sam.html`

1. 4. PPDs for PostScript printers

PostScript is a platform−independent page description language which is used by many mid−range and high−end laser printers and even by some high−end inkjets. Adobe has not only created PostScript, but also a way how to describe the properties and special commands of the different PostScript printers: The PPD (Postscript Printer Description) files, which also CUPS uses to describe printers. All this is well documented by Adobe:

`http://partners.adobe.com/public/developer/ps/index_specs.html`

PostScript printers are always shipped with appropriate PPD files. And these files describe their properties exactly. The files are also used by the drivers for Windows and Mac OS. Using these files with CUPS makes the full functionality of the printer (trays, resolutions, staplers, ...) available and therefore this is highly recommended.

To use a PPD simply copy it to `/usr/share/cups/model`, make it readable for everyone, let CUPS register doing "`killall -HUP cupsd`", and it will appear as a printer model in the web interface of CUPS. Set up your print queue as shown above then.

You find the PPDs either on the driver CDs which come with printer (but unfortunately often compressed in a way that one cannot uncompress them with free software), on installed Windows boxes, on the manufacturer's web site, at Adobe (mainly older models):

`http://www.adobe.com/products/printerdrivers/winppd.html`

or on linuxprinting.org (these files are released as free software by the manufacturers):

`http://www.linuxprinting.org/download/PPD/`

If there are PPDs for different Windows and Mac OS systems, the NT4 or Mac OS X versions should work best.

More info about PPDs you can find on:

`http://www.linuxprinting.org/ppd-doc.html`

Check the integrity of the PPD files with the " `cupstestppd`" utility. Often the output helps you fixing small flaws by editing the PPD file. CUPS only accept PPD files which pass the test.

1. 5. Non−PostScript printers, GhostScript, and Foomatic

Under Unix and Linux applications usually produce PostScript when one sends a document to a printer and all printers are treated as PostScript printers. If a printer does not understand PostScript by itself the software PostScript interpreter GhostScript is used to translate the incoming PostScript into the printer's own language (PCL, ESC/P 2, ... or something proprietary).

The code which generates the printer's native language is the so−called printer driver. There are more or less sophisticated drivers for many printers and languages.

There are different types of drivers:

- **GhostScript built−in**

  This driver concept is the oldest one. Drivers are pieces of code compiled into the GhostScript executable. One has to rebuild GhostScript to add a driver and so adding drivers for new printers is not trivial for most users. It is not recommended to create new drivers of this type. To determine which drivers are available, do "`gs −h`"and see the "Devices" section of the output. The current CVS of ESP GhostScript 7.07 contains all known drivers of this type.
- **Filter**

  Filters are separate executables which convert a printer−independent raster image format produced by GhostScript (PNM, PPM, ...) into the printer's native language. They appeared because some people wanted to create a printer driver quickly without needing to study the internals of GhostScript. The advantage is that you do not need to patch and compile GhostScript to install them. So they are very easy to install. To find out whether the driver you need is already installed, check whether the driver's executable is in the execution path (Use "`which <driver name>`").
- **CUPS Raster**

  CUPS raster drivers are a special form filter type drivers. The concept is developed as a part of CUPS to make driver installation easy. CUPS calls GhostScript to translate the PostScript input into the CUPS raster format and an additional CUPS filter to translate the CUPS raster format into the printer's language. These drivers come always with PPDs which do not only contain options and printer properties but also instruct CUPS to call the correct CUPS filter. Which of these drivers are installed you see by the PPD files in `/usr/share/cups/model` (and its subdirectories) and by the printer models showing up in the lists when setting up a print queue with the web interface of CUPS (entries contain "CUPS").
- **IJS plug−in**

  Like filters IJS plug−ins are also separate executables and so they also do not require GhostScript to be patched and recompiled to add support for a new printer. But in contrary to filters IJS plug−ins communicate bi−directionally with GhostScript and so GhostScript can ask them for certain printer properties and adapt its rendering appropriately. The executables have to be in the execution path, like filters. They usually contain "`ijs`" somewhere in their names and some of them can be called with the " `−h`" command line option to get version information.
- **Uniprint**

  This method is not common any more. GhostScript has a built−in driver named "uniprint" which is a universal raster driver. One can supply all comands which have to be sent to a printer to do raster printing on the GhostScript command line or by a file. For some printers there are several such files with the extension ".upp" in the `/usr/share/ghostscript/<version>/lib/` directory.

As the drivers come from many different sources (often students who simply needed their printers for their work) they are of so many different types and are also invoked by very different and cumbersome command lines. There are also usually no PPD files supplied with them to easily use them with CUPS.

Foomatic is a layer between the GhostScript/driver combo and the printing system. It provides a PPD file for every valid printer/driver pair and a universal filter, which can be easily integrated into the well−known printing systems (CUPS, LPRng, LPD, PPR, PDQ) or be used stand−alone for direct, spooler−less printing.

The filter, foomatic−rip, is then called by the printing system to convert the job data from PostScript to the printer's language. It reads in the PPD file which contains all information about the driver and how it is used, the command line prototype, all user−settable options and whet has to be inserted into the command line or into the job data to execute the users's choices. With this information and the user−supplied settings foomatic−rip builds the correct GhostScript command line and inserts PostScript or PJL commands into the job data stream. It finally executes GhostScript using the command line and pipes the PostScript data through it.

Using Foomatic with CUPS is easy. You look up your printer in the database on linuxprinting.org, choose a driver and then doenload the appropriate PPD fiile. Put the PPD file into /usr/share/cups/model and make it readable for everyone.

If your printer is not listed on linuxprinting.org, you could either try a driver for one of its predecessor models or you can check in the manual whether it uses a standard language (PostScript, PCL,ESC/P, ...) and use one of the "Generic" printer entries in the database.

Verify that the appropriate driver is installed. See the driver type descriptions above to learn how to determine whether a driver is installed. Install the driver if needed.

Download also foomatic−rip and foomatic−gswrapper, put them into /usr/bin and make them readable and executable for everyone. Create a link to make foomatic−rip visible for CUPS:

```
ln -s /usr/bin/foomatic-rip /usr/lib/cups/filter/
```

Now set up a print queue with CUPS as described above. In the web interface choose an entry which contains your printer's model name and "Foomatic".

NOTE:

- Do not use Foomatic PPDs for PostScript printers when you have a manufacturer−supplied PPD. The Foomatic PPDs for PostScript printers are generic and support only a few options, the manufacturer−supplied PPDs give access to the full functionality of your printer.

Some links:

linuxprinting.org printer database:

```
http://www.linuxprinting.org/printer_list.cgi
```

Instructions for setting up printers with CUPS and handling PPDs:

```
http://www.linuxprinting.org/cups-doc.html
http://www.linuxprinting.org/ppd-doc.html
```

1. 6. Multi−function devices from HP and Epson

On multi−function devices you probably want to use not only the printer but also the scanner and memory card reader. Using fax facilities is not supported in any device with free software, unfortunately.

On Epson devices printer, scanner, and card reader behave as independent USB devices. The card reader appears as a USB mass−storage device. The scanner is usually automatically discovered by SANE, via the "epson" backend. See

```
http://www.sane-project.org/
```

for whether and how the scanner in your device is supported.

On HP it is not so easy, but they are very well supported, too. Some distributions set them up with HPOJ for you. You can also scan with SANE then. The card readers of some newer models are USB mass−storage, on the other models they work with HPOJ, but they are so slow that buying a USB−2.0 card reader for around 10 EUR is recommended.

If your distro does not set up HPOJ for you, or if you want to have the newest software, get HPLIP from

```
http://hpinkjet.sourceforge.net/
```

to be able to scan with SANE. Card reader support is slow here, too. But the newest devices are supported and you get printer maintenance, as nozzle cleaning or head alignment. Note thet HPOJ is not maintainmed any more. So switch to HPLIP in case of problems.

Note that we are in a server tutorial: SANE does excellent network scanning via saned on the server and the "net" backend on the clients. And SANE−TWAIN on the Windows clients.

With non−HP/Epson multi−function devices you can be lucky to get printing to work.

2. Ethernet−Connected Printers

2. 1. General

Ethernet−connected printers (or conventional printers connected to print boxes like HP JetDirect or so) are very convenient in networks, as they do not rely on a single server. In small, especially home networks every PC has its own queue to the printer and so no PC needs to keep running day and night to have the printer accessible. In big networks several redundant servers have queues to the printer. If all are running they form a cluster and share the workload, if one fails, the jobs are simply done by the others.

2. 2. Assigning an IP address to a printer

On some more sophisticated devices you can enter the IP via the front panel, on others it has to be done by software running on a computer and accessing the printer or printbox through the network.

If you look into the manual of your printer or ethernet printing adapter, in many cases they simply tell you to install a Windows program from the supplied CDs to configure the ethernet interface. Do not think you have bought a ver expensive paperweight then, you can usually configure the IP of such a device with the "arp" (Address Resolution Protocol) utility, giving the command

```
arp -s <ip-address> <ethernet-address>
```

for example

```
arp -s 192.0.2.2 08:00:69:00:12:34
```

The ethernet address you find on the sticker on the device or by printing a self−test/configuration page with the device. Alternatively, you can install and run "`rarpd`" on your machine and enter all IP assignments in the `/etc/ethers` file like:

```
08:00:69:00:17:e3 myprinter.mydomain.com
08:00:69:00:12:34 192.0.2.2
```

Then the IP will be assigned whenever any device listed in the file is turned on. This way one has a central management for the IPs, there are no problems with devices not remembering their IPs when power−cycling, and if you have temporarily used a device at another place, it gets its IP back automatically.

Also bootp can be used to assign IP addresses to printers.

Check whether the printer accepted the IP address with

```
ping <ip-address>
```

2. 3. Discovering the printer's network protocols

To find out which protocols a printer supports, simply look which ports are open on it. This is done with the "`nmap` " command. Do

```
nmap <ip-address>
```

for example

```
[root@majax root]# nmap 192.168.100.20

Starting nmap 3.78 ( http://www.insecure.org/nmap/ ) at 2005-01-13 20:56 EST
Interesting ports on printer.mandrakesoft.com (192.168.100.20):
(The 1657 ports scanned but not shown below are in state: closed)
PORT STATE SERVICE
21/tcp open ftp
23/tcp open telnet
```

```
80/tcp open http
280/tcp open http-mgmt
515/tcp open printer
631/tcp open ipp
9100/tcp open jetdirect
MAC Address: 00:10:83:BB:B1:29 (Hewlett-packard Company)

Nmap run completed -- 1 IP address (1 host up) scanned in 12.671 seconds
[root@majax root]#
```

Now you can see by the ports which protocols are supported:

port	protocol
23	printer admin via telnet
80	printer admin via web
139	printing via SMB
515	printing via LPD
631	printing via IPP
9100	printing via TCP/Socket (can be also other ports > 1023)

your printer has an admin interface via telnet (do " `telnet <printer IP>`") or web (use URL "`http:/<printer IP>/` " in your browser) you can probably activate or deactivate the possible communication protocols of your printer. So open the desired protocol and close the others for example to avoid that a print quota system on your server will be surrounded.

See

`http://localhost:631/sam.html#COMMON_NETWORK`

for more information about setting up network printing devices.

2. 4. Setting up a print queue

In principle, the print queues are set up the same way as for local printers, via the printer setup tool of your distribution, the CUPS web interface or the "`lpadmin`" command. The only difference is what has to be entered for the device to which the jobs should go.

CUPS does not auto−detect the presence of network−connected printers. You have to know its IP and its protocol, in some cases even more information. In the web interface you choose on the "Device" page which protocol should be used to access your printer, on the next page you have to enter the so−called device URI (Unified Resource Identifier) then. If you use "`lpadmin`" you have to supply the device URI on the command line ("`-v`" option). The choice of printer model and driver is exactly the same as for a USB or parallel printer.

Easiest to set up is the access through the TCP/Socket protocol (also known as AppSocket or JetDirect). One only needs the IP of the printer and the port number. The IP you have assigned to your device and the port number you get from the "nmap" output. Most common is 9100, for multi−port print boxes 9101, 9102, and 9103, but other port numbers greater than 1023 are possible. So use TCP/Socket whenever it is possible.

The device URI is

`socket://<printer IP>:<port number>`

Example for a printer with IP 192.158.100.20 and port 9100:

`socket://192.158.100.20:9100`

If your device does not provide access via TCP/Socket, the second choice is to try SMB (Windows NT protocol). This will work if the printer's port 139 is listed as open by "`nmap` ". SMB is not supported by the CUPS software itself, but by Samba. It is enough to install the Samba client software coming with your distribution. You will not need running Samba

daemons. Restart the CUPS daemon with

```
killall -HUP cupsd
```

after installing Samba. "`lpinfo -v`" should list "`network smb`" then.

The information which you need to supply is the printer's IP and the printer share name. The share name varies between the different printer models and can be determined with the " `smbclient`" command:

```
smbclient -N -L <printer IP>
```

You should have at least one share of the type "Printer". The device URI looks as follows then:

```
smb://<printer-ip>/<share>
```

The third and less recommended way to print on a network printer is the LPD protocol. Its problem is that besides the IP you have assigned you need the name of the LPD queue on the network device and this you cannot discover by software running on your computer. You can either find the queue name in the manual (if you are lucky), guess common names as "LPD", "LPR", "ps", "PORT", "LPT", ... or these name with an added number, or find it on

```
http://localhost:631/sam.html#COMMON_NETWORK
```

The device URI is then

```
lpd://<printer-ip>/<queue-name>
```

where "queue−name" is the name of the LPD queue on your printer or print box.

More information you can find on

```
http://localhost:631/sam.html#PRINTING_OTHER
http://localhost:631/sam.html#COMMON_NETWORK
```

3. Networked Printing with CUPS

3. 1. Basic CUPS configuration for remote printer auto−discovery

Sharing printers between computers running CUPS (Linux, Unix, Mac OS X) is very easy, when the correct CUPS configuration is used. Once having the correct configuration, you set up a print queue on the machine where the printer is attached to and the other machines will have the printer vailable automatically within the next 30 seconds. And if you move the printer to another box, no change has to be done on the other machines, they will discover the move by themselves.

For the automatic discovery of remote printers every machine with locally defined print queues (CUPS server) broadcasts its host name and the names of the queues into the network every 30 seconds. Every machine running a CUPS daemon listens to such broadcasts, collects this information, and treats these queues like its own local queues (CUPS client). So a CUPS client lists all remote CUPS queues it gets knowledge of with "`lpstat -p`" command. Jobs sent to these remote queues the CUPS client passes automatically to the CUPS server so that the server can print them. Also the user−settable options of remote CUPS queues can be accessed on the CUPS cleint. " `lpoptions`" and the graphical printing frontends like "`kprinter` " or "`xpp`" show all the options as defined in the PPD file on the CUPS server and the user can set them. The processing of the jobs is done always on the server, so neither drivers nor PPD files need to be installed on the CUPS clients. And nothing needs to be removed from the cleints when a printer is removed from a server. A machine with a running CUPS daemon can be both a CUPS server and a CUPS client.

Do not think the broadcasts apply a high load to your networks. Only a few bytes are broadcasted every 30 seconds, not more than the host name and the queue names. All other info which you see about remote printers on a CUPS client is polled from the server on demand, only when an application requests it.

To get this working as easy as possible and with a basic security for small office or home networks edit the `/etc/cups/cupsd.conf` file as follows:

```
LogLevel info
TempDir /var/spool/cups/tmp
Port 631

BrowseAddress @LOCAL
BrowseDeny All
BrowseAllow 127.0.0.1
BrowseAllow @LOCAL
BrowseOrder deny,allow

<Location />
Order Deny,Allow
Deny From All
Allow From 127.0.0.1
Allow From @LOCAL
</Location>

<Location /admin>
AuthType Basic
AuthClass System
Order Deny,Allow
Deny From All
Allow From 127.0.0.1
</Location>
```

Comment lines and most blank lines are not shown here. All other lines not shown here can be (or stay) commented out (" `#`" added at the beginning of the line).

NOTE:

- Do not forget to restart the CUPS daemon with " `killall -HUP cupsd`" whenever you edit anything in /etc/cups/cupsd.conf or in any file in /etc/cups or its subdirectories.

The "`LogLevel`" setting tells how much should be logged in `/var/log/cups/error_log` by default. Leave it on " `info`" as long as you do not try to troubleshoot some problem. The " `TempDir`" can be different on your system. Do not modify or delete this line. 631 is the default port of CUPS (and IPP, the Internet Printing Protocol, the communication protocol CUPS is using), so do not change it.

The "`Browse...`" settings control the broadcasting of printer information between CUPS daemons.

"`BrowseAdress`" tells to which CUPS clients information about the queues on your machine is broadcasted. " `@LOCAL`" means all local networks, but not PPP, or dialed connections, so you printers will not get broadcasted into the internet and no costly dial–on–demand connections will be triggered. Yo can also specify an address range ("`192.168.100.*`") or several " `BrowseAddress`" lines with address ranges or even the IP addresses of single machines.

"`BrowseDeny`", "`BrowseAllow`", and " `BrowseOrder`" define from which CUPS servers broadcasted printers will be accepted and from where not. In our case we have a " `BrowseOrder deny,allow`" so the "`BrowseDeny`" are treated first and then the "`BrowseAllow`". Without any such line all broadcasted printers are accepted, the "`BrowseDeny all`" leads to no printer being accepted, but after that line the "`BrowseAllow` " lines are treated so the broadcasts from the IP 127.0.0.1 (the local machine) and from the local networks ("`@LOCAL`") are accepted again. The configuration of the example prevents queues appearing from faulty sites which broadcast into the internet. Using a more restrictive configuration you can get rid of long printer lists in your printing frontends so that you only see the printers you really need.

The "`<Location ...> ... </Location>`" blocks control the access to your machine by CUPS clients. Possible locations are:

Location	Purpose
/	The overall CUPS server

/admin	Adding/modifying/removing print queues
/printers	Printing on all queues
/printers/<queue name>	Printing on a particular queue
/jobs	Modifying/removing jobs

The locations behave like a directory hierarchy, if a sublocation does not exist, the access rules of the parent location apply. For example if there are the queues "bwlaser", " colorlaser", and "inkjet" and the locations "/printers " and "/printers/inkjet", the rules of "/printers" are valid only for "bwlaser" and "colorlaser", for " inkjet" the rules of "/printers/inkjet" apply.

The "Deny From", "Allow From", and "Order" rules work the same way for client access permission as "BrowseDeny", "BrowseAllow", and "BrowseOrder " for accepting broadcasted printers. So in the example general access (Location "/") is allowed for the local machine and all clients on the local networks. This applies especially for printing on all print queues as there is no "/printers" location. Print queue manipulation is even only allowed from the local machine.

Access cannot only be restricted per−network or per−machine but also per−user. For user−wise access restrictions the " AuthType" and "AuthClass" rules are used.

"AuthType" can be "None" (no per−user access restriction, no login/password required), "Basic " (permitted users can access with their usual Unix login/password on the server), or "Digest" (CUPS has its own user accounts defined with the "lppasswd" command).

"AuthClass" can be "System", then only users of the system group (usually "sys") can access. It can also be "Basic", to give all Unix or Digest users on the server access or "Group" to give access to the users in the group specified by the "AuthGroupName" rule.

In our example everyone can print, as there are no per−user restrictions in the "/" location. But queue manipulation can only be done by the members of the Unix System group, usually this is only "root". Therefore you need to log in as "root" in the web interface to set up queues.

3. 2. High availability

In CUPS it is also easy to set up high availability by having redundant printers or servers.

For example you can connect three (preferrably equal) printers via USB to one CUPS server. You define a queue with the same configuration to each of these printers, for example with the names "laser1", "laser2", and "laser3". Now you set up a so−called printer class. This is a queue which does not point to a printer, but to a set of other queues. In this case you let it point to outr three queues "laser1", "laser2", and "laser3". Let us call the class "laser". To define the class one uses again the web interface of CUPS, simply clicking the "Add Class" button on the Administration page. " lpadmin" also allows to define classes.

If a user would send a job to "laser2" the job has to wait until all other jobs on "laser2" are ready and "laser2" must be ready to print (turned on, paper and toner loaded). Otherwise the user's job waits in the queue, even if "laser1" and "laser3" are doing nothing.

If the user sends the job to the class "laser". The first printer which gets free will take the job. So if "laser1" is printing a 100−page job and "laser2" is out of paper, the user's one−page letter goes to "laser3" and so it comes out quickly.

The problem would be if the server fails. Then none of the printers is available and the users cannot print. So we connect each printer to another computer, which means that if one of the computers fails, the other two printers are still available. But there is one problem: On which server do we define the class "laser"? If this server fails, the printers are all not available again.

The solution is a so called "implicit class", a class automatically defined by CUPS. Therefore we do not call the three queues on the three servers "laser1", "laser2", and "laser3". It is much simpler, we call them all "laser" and let them get broadcasted into our network. Now the clients receive broadcasts of three equally−named queues. For the users of the clients only one queue named "laser" appears. And this queue behaves as a class: the first of the three printers which gets free takes the job. So independent which of the three server/printer combos are working, if at least one is working the jobs sent to "laser" get printed. Also there is no single point of failure which defines the class "laser". "laser" is defined

on every client and only if at least one of the three servers is working.

If you have an ethernet−connected printer you could define a queue for it on every client, so the clients are their own CUPS servers and do not depend on a single server. But imagine the number of print queues you see in graphical printing frontends when every queue has another name, and that in a network with only one printer. And imagine also the administration nightmare when the IP of the printer has to be changed.

So here you better choose three machines and define equally−named queues pointing to this one printer. You get an implicit class which will be the only one visible queue on the CUPS clients. And you do not need to rely on one server, you can print if at least one of the three is running.

Also combinations of this are possible: For example two servers and three ethernet printers, where on each of the two servers is exactly the same CUPS configuration: Three queues pointing to the three printers and a class putting together the three queues. The two classes on the two servers form an implicit class and so printing into this implicit class works with any arbitrary pair of server and printer.

3. 3. Print quotas and accounting

CUPS has also basic page accounting and quota capabilities.

Every printed page is logged in the file `/var/log/cups/page_log` So one can everytime read out this file and determine who printed how many pages.

In addition, one can restrict the amount of pages (or kBytes) which a user is allowed to print in a certain time frame. Such restrictions can be applied to the print queues with the "`lpadmin` " command.

```
lpadmin -p printer1 -o job-quota-period=604800 -o job-k-limit=1024
lpadmin -p printer2 -o job-quota-period=604800 -o job-page-limit=100
```

The first command means that within the " `job-quota-period`" (time always given in seconds, in this example we have one week) users can only print a maximum of 1024 kBytes (= 1 MByte) of data on the printer "printer1". The second command restricts printing on "printer2" to 100 pages per week. One can also give both " `job-k-limit`" and "`job-page-limit`" to one queue. Then both limits apply so the printer rejects jobs when the user already reaches one of the limits, either the 1 MByte or the 100 pages.

This is a very simple quota system: Quotas cannot be given per−user, so a certain user's quota cannot be raised independent of the other users, for example if the user pays his pages or gets a more printing−intensive job. Also counting of the pages is not very sophisticated. The pages are counted by the CUPS filters, simply by analyzing the PostScript data stream. This does not work with arbitrary PostScript data or with jobs already arriving in the printer's native language (for example from a Windows client). Another problems are paper jams or similar hardware problems of the printer. The pages do not get printed but were already counted by the filters.

So for more sophisticated accounting it is recommended to use add−on software which is specialized for this job. This software can limit printing per−user, can create bills for the users, use hardware page counting methods of laser printers, and even estimate the actual amount of toner or ink needed for a page sent to the printer by counting the pixels.

The most well−known free software packages are

PyKota (recommended):

`http://www.librelogiciel.com/software/PyKota/`

PrintBill:

`http://ieee.uow.edu.au/~daniel/software/printbill/`

3. 4. LPD clients

If you have some older boxes, especially with commercial Unix systems, you have probably LPD on them and no CUPS. You do not need to install LPD or LPRng on your print servers to get print jobs from these boxes printed. CUPS has an

interface to LPD clients, the so−called CUPS−LPD mini−daemon.

The CUPS−LPD mini−daemon maps the CUPS queues known to your server's CUPS daemon to the LPD protocol. The queues will have the same name for LPD as they have for CUPS.

To activate the CUPS−LPD mini−daemon, you have to get it invoked on demand by inetd or xinetd. For inetd add the line

```
printer stream tcp nowait lp /usr/lib/cups/daemon/cups-lpd cups-lpd
```

to `/etc/inetd.conf` and give a kick to the inetd with "`killall -HUP inetd`". For xinetd create a file named `/etc/xinetd.d/cups-lpd` containing

```
service printer
{
socket_type = stream
protocol = tcp
wait = no
user = lp
group = sys
server = /usr/lib/cups/daemon/cups-lpd
server_args = -o document-format=application/octet-stream
passenv =
env = TMPDIR=/var/spool/cups/tmp
disable = no
}
```

and kick the xinetd daemon with "`killall -HUP xinetd`". From now on LPD clients can print on your queues the classical LPD way. Note that any access restrictions in `/etc/cups/cupsd.conf`, in `/etc/hosts.allow` and in `/etc/hosts.deny` are ignored. So every user and every machine can print via LPD on your server.

4. Accessing Printers from Windows Clients via Samba

4. 1. Sharing the CUPS queues to Windows clients

At first check whether the needed Samba server packages are installed and that your Samba version is linked against the CUPS library (should be the case in most modern Linux distributions). To check whether Samba is linked against the CUPS library, use the "`ldd`" command:

```
root# ldd `which smbd`
libssl.so.0.9.6 => /usr/lib/libssl.so.0.9.6 (0x4002d000)
libcrypto.so.0.9.6 => /usr/lib/libcrypto.so.0.9.6 (0x4005a000)
libcups.so.2 => /usr/lib/libcups.so.2 (0x40123000)
[...]
```

If "`libcups`" is listed as shown in this example, your Samba is OK. If not, try to get CUPS−enabled packages of Samba for your distro or compile Samba from source. If you compile Samba from source, you need to have the header/development package of the CUPS library ("`libcups-devel`", "`cupsys-dev`", or similar) installed.

You also need to take care that the names of all your printers are not longer than 12 characters. If a queue name is longer, edit `/etc/cups/printers.conf` to modify the queue name and then activate the change with the usual "`killall -HUP cupsd`". CUPS has no on−board tool to rename print queues, but some other printer setup tools, like "`printerdrake`" in Mandriva Linux, have.

Now do the following modifications on `/etc/samba/smb.conf`:

```
[global]
load printers = yes
printing = cups
printcap name = cups
```

```
printer admin = @ntadmins

[printers]
comment = All Printers
path = /var/spool/samba
browseable = no
public = yes
guest ok = yes
writable = no
printable = yes
printer admin = root, @ntadmins

[print$]
comment = Printer Driver Download Area
path = /etc/samba/drivers
browseable = yes
guest ok = yes
read only = yes
write list = @ntadmins, root
```

All lines not supposed to be modified are not shown. Do not touch that lines, otherwise you could loose other Samba services, like file sharing or so.

Restart Samba (usually "`/etc/init.d/smb restart`") to get your changes active.

Create the "ntadmins" group in /etc/group and add users which should be allowed to to Samba printer administration. "root" is always allowed to do so and does not need to be added to this group.

If some of the directories referred to in the lines shown above does not exist, create it. Let the ownerships of these directories be root and the ntadmins group. Set permissions for everyone to read and execute and for user and group to write into the directories.

Create Samba accounts for "root" and every user in the "ntadmins" group with "`smbpasswd`":

```
root# smbpasswd -a root
New SMB password: secret
Retype new SMB password: secret
```

Samba has no access to the Unix passwords in `/etc/shadow`. Therefore Samba needs its own passwords. Use always the Samba passwords for authentication against Samba.

Most important in the `smb.conf` is "`printing = cups`". This makes Samba use the functions of the CUPS library to access printers, instead of shell command lines defined in `/etc/samba/smb.conf` So if your `smb.conf` contains lines like "`print command`", "`lpq command`", ... in the "`[global]`" or "`[printers]`" sections, they get ignored.

The "`[print$]`" section defines a file share where Windows printer drivers can be supplied so that Windows clients can download them automatically ("`Point'n'Print`").

The shown configuration makes all printers known to the local CUPS daemon being shared via Samba. This means they will all be detected in the network Neighborhood or in the Add Printer Wizard under Windows.

To really print with these printers under Windows one can use the Windows drivers which came with the printers and set up queues with them on each client. Or one uploads the drivers into the print$ share on the Samba server (can be done with the Add Printer Wizard under Windows). This requires also uncommenting the "`octet-stream`" lines in the end of the `/etc/cups/mime.types` and `/etc/cups/mime.convs` files and then restarting CUPS, so that CUPS accepts already completely processed print data on regular print queues and does not try to filter it. For more about this, see the printing sections of the Samba HOWTO collection:

http://de.samba.org/samba/docs/man/Samba-HOWTO-Collection/printing.html
http://de.samba.org/samba/docs/man/Samba-HOWTO-Collection/CUPS-printing.html

This approach of printing has the disadvantage that once many different printer drivers are on each Windows client and this can compromise the stability of Windows. Another problem is that CUPS does not understand the job data which passes through it and so it cannot count the printed pages. This prevents print accounting from working correctly.

The better way is to let our server emulate PostScript printers, as it does for Unix applications. For that a PostScript driver for Windows and the PPD file of the CUPS queue is put up in the print$ share and so Windows will use the PostScript driver for all printers and send PostScript, as a Unix client. This way every Windows client has only one printer driver installed and CUPS gets data on which it can do accounting and other types of processing.

4. 2. Emulate PostScript printers

At first, you need to get a PostScript driver for Windows, from CUPS (recommended) for Windows NT/2000/XP or newer, from Microsoft for Windows NT/2000/XP or newer or from Adobe for Windows 95/98/Me. You can also use both the CUPS or Microsoft and the Adobe driver to support all Windows versions on your clients.

The CUPS driver you get in the download section of

http://www.cups.org/

the Adobe driver on

http://www.adobe.com/

and the Microsoft driver on a machine with Windows installed (NT/2000/XP or newer), in the %WINDOWS%\SYSTEM32\SPOOL\DRIVERS\W32X86\3 directory.

Create a directory named drivers/ in /usr/share/cups and put the following files of the PostScript drivers for Windows there:

Driver	Files
CUPS:	cupsui6.dll cupsdrv6.dll ps5ui.dll pscript.hlp pscript.ntf pscript5.dll
Microsoft:	ps5ui.dll pscript.hlp pscript.ntf pscript5.dll
Adobe:	ADFONTS.MFM ADOBEPS4.DRV ADOBEPS4.HLP ICONLIB.DLL PSMON.DLL

The CUPS driver is the recommended driver as it supports several CUPS−specific printing options.

Use "unzip" or "cabextract" to uncompress the Adobe driver package. If you distribution does not include cabextract, get it from here:

http://www.kyz.uklinux.net/cabextract.php3

Now run either

```
cupsaddsmb -U root -v -a
```

to set up the PostScript driver in print$ for all printers known by the local CUPS daemon, or

```
cupsaddsmb -U root -v laser inkjet
```

to only set op the driver for the queues "laser" and "inkjet".

If you get asked for a password, enter the Samba password for "root".

Now you can set up print queues on your Windows clients without taking care of drivers. The Add Printer Wizard will download the PostScript driver automatically from your server.

See also the man page of "cupsaddsmb".

4. 3. More possibilities

There is excellent documentation about printing via Samba written by Kurt Pfeifle. It is part of the Samba HOWTO collection:

```
http://de.samba.org/samba/docs/man/Samba-HOWTO-Collection/printing.html
http://de.samba.org/samba/docs/man/Samba-HOWTO-Collection/CUPS-printing.html
```

5. CUPS Troubleshooting

5. 1. RTFM − Read The F... Manual!

Before trying to get support on mailing lists, newsgroups, or web forums, read the documetation of the software you are using. There is a lot of documentation for CUPS, Foomatic, and Samba, simply follow the links in this document. Most command line tools have also man pages.

5. 2. Supply all important information

Which distribution are you using? Mandriva Linux 2005, Fedora Core 2, but NOT Linux 7.2.

Which kernel? Do "uname −a" and you get the version number, for example 2.6.9.

Which version of CUPS, Gimp−Print, HPIJS, Samba, ...

Which exact command lines did you use? What was the output? Cut and paste from your terminal Window.

What did you fill into the fields in the web interface of CUPS?

Supply also config files like `/etc/cups/printers.conf`, `/etc/cups/cupsd.conf`, `/etc/cups/ppd/*.ppd`, `/etc/samba/smb.conf`, ...

Strip away the comments and blank lines in the config files when posting them, for example with a command like this:

```
cat /etc/cups/cupsd.conf|grep -v ^#|grep [:alnum:] > cupsd.conf-stripped
```

Then it is much easier to find the important things.

Run diagnostic commands like "`lsmod`", " `lsusb -vvv`", ...

Check the integrity of manufacturer−supplied PostScript PPD files with the "`cupstestppd`" utility. Often the output helps you fixing small flaws by editing the PPD file. CUPS only accept PPD files which pass the test. Unfortunately often printer manufacturers do not comply well with Adobe's standards.

5. 3. Check device file permissions

Device files for printers (like `/dev/lp0`, `/dev/usb/lp0`, ...) must be readable and writable for the user "root" and the group "sys".

5. 4. Send print jobs with CUPS in debug mode

Edit `/etc/cups/cupsd.conf` switching the " `LogLevel`" to "`debug`". Kick the CUPS daemon with " `killall -HUP cupsd`" and send a print job. Then have a look into `/var/log/cups/error_log` and look for error messages, GhostScript command lines, option settings. Post relevant parts of the file if you didn't find a solution for your problem.

5. 5. Where to ask

And finally, after reading what is important when asking for help, the places where you can ask:

`http://www.cups.org/newsgroups.php`

The home of the CUPS−centered newsgroups and mailing lists.

`http://www.linuxprinting.org/forums.cgi`

The ressource to look for answers if you have vendor− or model specific questions. There is a seperate forum for all majur manufacturers. here you can also post your questions concerning Foomatic.

`http://gimp-print.sourceforge.net/p_Mailing_Lists.php3`

This is the place of choice if you are suffering from difficulties with using Gimp−Print drivers.

NOTE:

- DO NOT send questions to the private mail addresses of people who answer questions on the lists regularly as long as they do not ask you explicitly to do so. Post always on the lists and always use "Reply to all" or "Reply to list" in your e−mail when answering. From every problem discussed and solved on the list everyone can learn. And everyone can read the discussion and help, not only a single person to whom you mail privately. And the more people read you questions, the quicker someone helps.

5. 6. More about troubleshooting ...

... you can read in Kurt Pfeifle's HOWTO:

`http://www.cups.org/cups-help.html`