Debian and Windows Shared Printing mini−HOWTO

Ian Ward

<ian at excess dot org>

2005−07−01

Revision History		
Revision 1.6	2005−07−01	Revised by: iw
Clarified hpijs requirement, added lpinfo and lpoptions commands		
Revision 1.5	2005−06−19	Revised by: iw
Added note about becoming root to execute commands		
Revision 1.4	2004−01−05	Revised by: iw
Wording corrections		
Revision 1.3	2003−11−18	Revised by: iw
Removed incorrect lpadmin −h usage		
Revision 1.2	2003−10−03	Revised by: iw
Note about woody and gs−esp, conflict with bash's enable command and fix for XP/2000 clients		
Revision 1.1	2003−06−26	Revised by: iw
Added passwords on windows shared printers, Corrections		
Revision 1.0	2003−05−15	Revised by: tmm
Initial release, reviewed by LDP		
Revision 0.8	2003−04−11	Revised by: iw
converted from LaTeX		

Table of Contents

1. Introduction

Debian GNU/Linux (http://www.debian.org) is the premier volunteer−supported Linux distribution. Unfortunately, setting up printers in Debian can be difficult. Also, simple step−by−step instructions for sharing printers between Windows and Linux using the latest tools are hard to find. This HOWTO was written to address both problems.

This HOWTO will demonstrate how to use command−line tools to configure your Debian system for printing. It will explain how to send documents from Linux to Windows printers and how to share Linux printers with Windows PCs. Some troubleshooting examples are also given.

The primary url for this document is http://excess.org/docs/linux_windows_printing.html. The source Docbook/XML and EPS files for this document may be downloaded from http://excess.org/docs/src/. Please forward bug reports, corrections and suggestions regarding this document to ian at excess dot org.

---

2. Getting Started

2.1. Linux Printing Components

The main components we will be using include:

- CUPS

  The Common UNIX Printing System (http://www.cups.org) is a print spooler and a set of support programs for using and administering printers.
- Samba

  Samba (http://www.samba.org) is software that allows non−Windows computers to act like Windows computers on a network by implementing Windows file and printer sharing protocols.
- Printer Drivers

  LinuxPrinting.org (http://www.linuxprinting.org) offers the largest number of printer drivers and maintains a database of printers supported under Linux. You must download a printer driver for each model of printer you want to use in Linux. A printer driver consists of a PPD file and a filter program, or only a PPD file for PostScript printers.

2.2. Required Packages

All of the required programs and libraries are part of the standard Debian archive. You may download and install these packages with the usual Debian packaging tools. The following is a list of packages you need:

cupsys
: CUPS server

cupsys−bsd
: CUPS BSD commands

cupsys−client
: CUPS client programs

foomatic−bin
: LinuxPrinting.org printer support programs

samba
: Samba SMB/CIFS server for UNIX

smbclient
: Samba SMB/CIFS client for UNIX

gs−esp
: ESP Ghostscript (http://www.cups.org/ghostscript.php)

  Not available as a Debian GNU/Linux 3.0 (a.k.a. woody) package, use "gs" instead.

a2ps
: GNU A2PS (http://www.gnu.org/software/a2ps/)

The following commands will install these packages. You will have to become root or use **sudo** to execute these commands:

```
apt-get update
apt-get install cupsys cupsys-bsd cupsys-client foomatic-bin samba smbclient gs-esp a2ps
```

Additional packages may be required for specific printers. For example, the *hpijs* package must be installed for many HP InkJet, DeskJet and LaserJet printers to function properly. The PPD files for these printers are identified by the string `hpijs` in their filenames.

2.3. CUPS Local Printer Configuration

The **lpadmin** command is used to configure printers. The following is an example of setting up a laser printer with CUPS. You will have to become root or use **sudo** to execute these commands:

```
/usr/sbin/lpadmin -p Laser -v parallel:/dev/lp0 -P /root/laser.ppd
/usr/bin/enable Laser
/usr/sbin/accept Laser
/usr/sbin/lpadmin -d Laser
```

Please note that bash has a builtin command called **enable**, so bash users must use the full path (**/usr/bin/enable**) to enable printers.

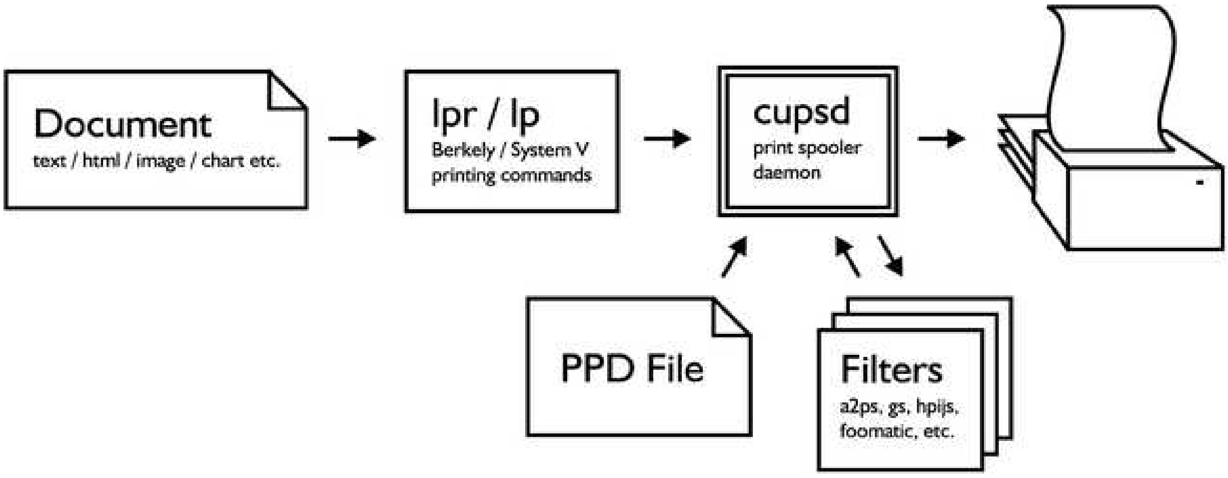
The first command creates a new printer called "Laser" that is connected to the first parallel port and is using the PPD file `/root/laser.ppd`. "Laser" is then enabled and told to accept jobs with the **enable** and **accept** commands. The last command sets "Laser" as the default printer.

If your printer is connected to a USB port or if you do not know the correct device−uri for your printer try running **/usr/sbin/lpinfo −v** to get a list of available printer devices.

Make sure your printer's page size and other options are set correctly by running **/usr/bin/lpoptions −l**. More detailed information about printer configuration is available in the CUPS documentation.

2.4. Linux Printing Basics

Figure 1. Printing Locally



Documents are spooled by using either **lpr** or **lp** followed by the file name. You may view the printer queue and check the printer status with the command **lpstat −o** or **lpstat −p**. To cancel a print job use either **cancel** or **lprm** followed by the job id.

The CUPS spooler daemon is called *cupsd*. It converts documents to PostScript, then converts them to a format native to the printer Figure 1. Printers that do not understand PostScript use a rasterized, or bitmap, format for documents. Rasterized formats can be much larger than the original PostScript, and will take longer to send to the printer.

Filters are programs used to convert documents from one format to another. The CUPS spooler will do its best to find a suitable filter for the documents you send. If no filter suitable for converting your document is installed you will receive an error similar to `lpr: unable to print file: client−error−document−format−not−supported.`

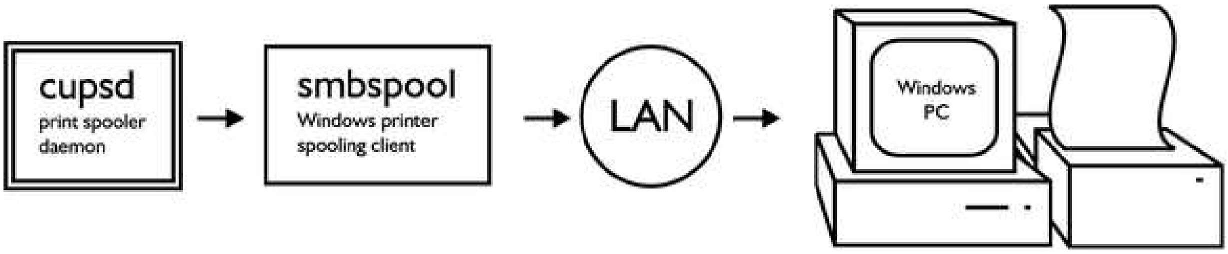
Many applications do not include filters for their documents formats. Documents created with these applications can only be printed from within the application itself, unless the document is exported to PostScript or another standard format.

---

3. Printing To Windows PCs

3.1. Connecting To Windows

Figure 2. Network Printing



SMB and CIFS are the Windows file and printer sharing protocols. We use Samba to speak to the Windows PCs using these protocols. Before configuring CUPS we should make sure we can connect to the Windows PC with **smbclient**, the Samba SMB/CIFS client Figure 2.

The following is an example of creating a connection to a Windows PC:

```
/usr/bin/smbclient -L rice -U fred

added interface ip=10.6.7.234 bcast=10.6.7.255 nmask=255.255.255.0
Got a positive name query response from 10.6.7.8 ( 10.6.7.8 )
Password: (not shown)

Sharename  Type  Comment
PRINTER$   Disk
INKJET     Printer
STUFF      Disk
IPC$       IPC    Remote Inter Process Communication
```

The command shown asks for a list of shares on a Windows PC named "rice", with the user id "fred". The result shows a printer named "INKJET".

If Windows naming service is unavailable you will need to specify the IP address of the Windows PC with the `-I` switch as in:

```
/usr/bin/smbclient -I 10.6.7.8 -L rice -N
```

For more information see the Samba documentation about **smbclient** usage.

3.2. CUPS Configuration

Once you have found a Windows printer you may configure CUPS. First verify that your installation of CUPS has the smb backend with the following command:

```
ls -l /usr/lib/cups/backend/smb
```

If this file does not exist create it by issuing the following:

```
ln -s `which smbspool` /usr/lib/cups/backend/smb
```

The following is an example of setting up the printer shown above. You will have to become root or use **sudo** to execute these commands:

```
/usr/sbin/lpadmin -p RicePrinter -v smb://fred:mypass@rice/INKJET -P /root/inkjet.ppd
/usr/bin/enable RicePrinter
/usr/sbin/accept RicePrinter
/usr/sbin/lpadmin -d RicePrinter
```

As mentioned above, bash has a builtin command called **enable**, so bash users must use the full path (**/usr/bin/enable**) to enable printers.

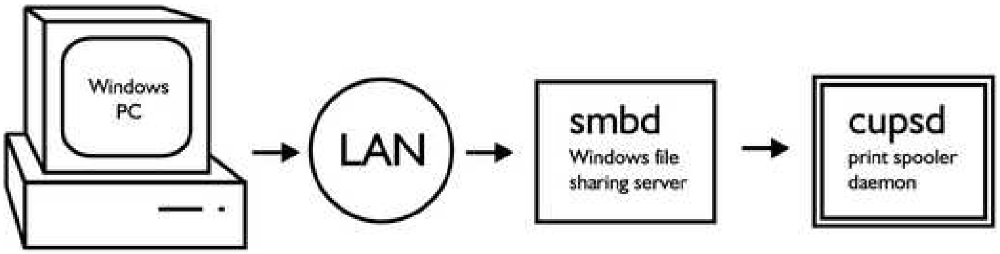
The "lpadmin" command sets up a the shared Windows printer by giving the username, password, netbios name and printer name as a single parameter. See Section 2.3 for a further explanation of the commands above.

Your printer is now ready to test. Send a file to the printer with the **lp** command followed by a filename, or by printing a document from within an application.

4. Sharing Printers With Windows PCs

4.1. Sharing Basics

Figure 3. Printer Sharing



Samba uses *nmbd* and *smbd* daemons to share files and printers with Windows PCs. *nmbd* acts as a Windows naming service, broadcasting your computer's name to Windows PCs on the LAN. *smbd* accepts file and printer requests from Windows PCs Figure 3.

You will need to download and install Windows printer drivers for each Linux printer you are sharing. Windows printer drivers can be found by searching the web site of your printer manufacturer.

4.2. Samba Configuration

If you are allowing anonymous access to your printer you will need to create a user account for remote print jobs:

```
/usr/sbin/adduser --system --disabled-password smbprint
```

This command adds a user called "smbprint" to your system. Make sure there is enough disk space in `/home/smbprint`, the "smbprint" user's home directory, to spool files. Check that the "smbprint" user does not have permission on your system to read or modify sensitive files and directories. If you have configured CUPS to restrict printing to certain users on your system, you must allow the "smbprint" user to access printers you want to share.

The Samba configuration file is `/etc/samba/smb.conf`. The following is an example configuration file set up to use CUPS with the "smbprint" user:

```
[global]
  printcap name = cups
  printing = cups
  security = share
[printers]
  browseable = yes
  printable = yes
  public = yes
  create mode = 0700
  guest only = yes
  use client driver = yes
  guest account = smbprint
  path = /home/smbprint
```

Please note that this configuration will allow printing by anyone that can make a network connection to your computer and is not recommended for computers on untrusted networks, such as computers with direct Internet connections. If you need to implement access control, set *security = user* or *security = domain* and read the Samba man pages for further information.

Once you have added the above settings to your Samba configuration file you must restart Samba with the command:

```
/etc/init.d/samba restart
```

4.3. CUPS Configuration

Windows printer drivers format their output for the printer before sending it across the network. You must configure CUPS to accept the pre−formatted output by uncommenting the following line from `/etc/cups/mime.convs`:

```
application/octet-stream   application/vnd.cups-raw   0   -
```

Also uncomment the following line from `/etc/cups/mime.types`:

```
application/octet-stream
```

Now CUPS must be told to allow connections from other machines on the network. Add these lines to `/etc/cups/cupsd.conf`:

```
<Location /printers>
 AuthType None
 Order Deny,Allow
 Deny From None
 Allow From All
</Location>
```

As in the Samba configuration, this configuration allows any computer to connect to your printers and is not recommended for computers on untrusted networks. For information about tightening access control to your printers, see the `cupsd.conf` man page and the CUPS documentation.

Finally, restart cups with the following command:

```
/etc/init.d/cupsys restart
```

Your Linux printers should now be shared to Windows PCs on the LAN. Follow the usual steps for adding a network printer to your Windows PCs, and remember to print a test page.

5. Troubleshooting

5.1. Failing To Connect To Windows Printers

When *smbspool*, the **smbclient** utility CUPS uses, fails to connect properly it emits error messages that are humorous but not very helpful. One such message is `Unable to connect to SAMBA host: Success.` Another sign of connection failures is when documents seem to get stuck on the queue when printing to Windows printers.

View the most recent entries in the CUPS log with the following command:

```
/usr/bin/tail /var/log/cups/error_log
```

If you see a message similar to `cli_connect() failed...` then *smbspool* could not find the Windows PC you are trying to connect to. Check the spelling of the Windows PC's host name. Check that the Windows PC is turned on and that its network connection is functioning properly. Make sure you can connect to it using **smbclient** as shown in Section 3.1.

If you see a message similar to `SMB tree connect failed: ERRSRV - ERRinvnetname` then **smbclient** connected to the Windows PC but could not connect to the printer you requested. Check the spelling of the shared printer using **smbclient** as shown in Section 3.1.

5.2. Other Failures

Other failures include being unable to print to a local printer and having your print jobs disappear from the queue without being printed. You may also see vague error messages such as `Child process 2384 exited with status 32.`

Increase CUPS' logging level to "debug" to see more messages about what happened before the print job failed.

1. Open the main CUPS configuration file `/etc/cups/cupsd.conf` in a text editor.
2. Change the line that reads "LogLevel warn" to "LogLevel debug".
3. Save the configuration file and exit the text editor.
4. Restart the CUPS server with the command:
   ```
   /etc/init.d/cupsys restart
   ```

You can follow the CUPS log with the following command:

```
/usr/bin/tail -f /var/log/cups/error_log
```

You should see a line that reads `Scheduler shutting down due to SIGTERM`. This indicates that the CUPS server was stopped successfully.

Send your print job again and watch for useful debug messages that appear. One example of a useful debug message is `GNU Ghostscript 7.05: Can't start ijs server 'hpijs'.` In this case the solution is to install the "hpijs" package.

If you cannot determine the cause of the failure, do an Internet search for key terms in error messages you see; it is likely that someone has solved your problem before. You may also try upgrading the packages listed in Section 2.2 to their latest versions.

6. License