

Making PDF Files With Samba

J. Paul Richardson

john.p.richardson@worldnet.att.net

v1.2.1, 20 May 2006

Table of Contents

1. Introduction	2
1.1 Copyright, Disclaimer, Trademarks and Credits	2
1.2 Changes and Errata	2
2. PDF Documents	3
2.1 What Is A PDF Document?	3
2.2 Why PDF Documents?	3
2.3 Why Not Just Buy Adobe Acrobat?	3
2.4 What About Open Office?	4
3. How It Works	4
4. Requirements	4
4.1 The Linux Computer	5
4.2 The Windows Computer	5
4.3 Network Connection	5
5. Making It Work	6
5.1 Adding the Pseudo-Printer	6
5.2 Creating the Script File	7
5.3 Restart Samba	8
5.4 Add the Pseudo-Printer to Samba	8
5.5 Do a Test Print	8
6. Go Farther	9
7. Bibliography	9
8. Revision History	9

1. Introduction

This How-To covers how to make PDF documents from Windows-based applications via Samba. A working Linux installation with Samba and Ghostscript is required.

1.1 Copyright, Disclaimer, Trademarks and Credits

Copyright © 2006 by John P. Richardson.

E-mail: paul@reverendlinux.com

Please freely copy and distribute this document as you see fit. However, if you make any changes in its content, you must submit these revisions to the author before doing so.

If you wish to include information, in whole or in part, contained in this document in one of your own, you may do so as long as you reference it properly (cite document title and author).

Disclaimer: The information contained within in this document, although checked thoroughly for accuracy, is not “the gospel.” The author is in no way responsible for any errors within this document nor is he responsible for anything that happens to your hardware or software as a result of applying the information contained herein.

Trademarks: PDF and Portable Document Format are copyrighted and trademarked by Adobe Corporation. Use of these terms here are without their consent so if they want me to stop, let them contact me.

Credits: See the bibliography at the end.

1.2 Changes and Errata

If you find a mistake or would like to add something to this document, please contact the author via the e-mail address above. Any suggestions are greatly appreciated.

2. PDF Documents

2.1 What is a PDF Document?

PDF stands for Portable Document Format. Rather than coming up with my own definition of that, I will let Barron's Dictionary of Computer and Internet Terms do it for me: *a document that maintains its formatting and displays correctly on any output device*. This means the document can be viewed as it was intended to be on any device, from a computer to a PDA.

PDF files are normally created using a standard application (word processor, spreadsheet, web browser) by filtering them through the Distiller portion of Adobe Acrobat. Basically, you print the document to Distiller rather than your standard printer. This print is translated by Distiller from PostScript format (the pretty much standard language understood by most printers) to the Adobe PDF format.

2.2 Why PDF Documents?

Most people come in contact with PDF files on the Internet. These are usually in the form of downloadable forms or manuals on web sites. The PDF format is widely used due to the fact that it keeps the formatting of the document true to the original version and provides a way to distribute documents in a read-only (meaning you can't edit it) form. Adobe's Acrobat or Acrobat Reader are required to read a PDF file.

2.3 Why not just buy Adobe Acrobat?

Well, the average cost for Acrobat is about \$200. If you want to create PDF files that are complex with things like hyper links and drop-down selection boxes, then Acrobat is the way to go. But, if all you need to do is simply convert documents to PDF format for archiving or downloading, why not do it on the cheap with Samba? That's how this document was created. It was written and formatted using Microsoft Word 2000 on a Windows 2000 Professional-based computer. It was then converted to PDF format using the Samba setup described here. Cheap and effective!

2.4 What About Open Office?

True, beginning with OpenOffice 1.1 you can save any document straight to PDF. But there are three limitations here. The first is only documents created in OpenOffice can be converted to PDF. That means things like web pages and Quicken print-outs can't. Second, in a commercial or government setting, just loading up OpenOffice isn't all that feasible. In these environments (especially government) only certain software is allowed on workstations. All others are considered forbidden. Finally, the third drawback would be that OpenOffice would have to be loaded on each workstation requiring the capability to make PDF documents. In an office of 10 workstations, that may not be so bad. In an office of 100, that could get pretty tedious. In all three cases, a server-based PDF generator would be the best solution.

3. How It Works

Basically, it works like this. Via Samba, you create a share on the Linux computer that can be seen by the Windows computer. You then create a pseudo-printer in Samba that is really a link to a utility that converts files to PDF documents (see next section). Once that is done, you connect to the pseudo-printer from the Windows computer, load up some drivers and you're done.

4. Requirements

To make PDF documents on a Linux-based computer using Samba, you will need the following:

- 1 computer with Linux, Samba and Ghostscript installed
- 1 computer with Windows, Adobe Acrobat Reader and document creation software installed (aka MS Office)
- A network to connect the two computers together
- Some skill at using Samba and a text editor

4.1 The Linux Computers

To make PDFs via Samba, you must have Samba running and a share available to your Windows-based computer. If you don't, now would be a good time to do so. You can download the latest Samba software at <http://www.samba.org>. You can find plenty of documentation on how to get Samba up and running at that site and also at The Linux Documentation Project, <http://www.tldp.org>.

Once you have Samba working, you will need to ensure you have Ghostscript loaded on your Linux System. Ghostscript is an application for creating and manipulating Postscript files. It includes a utility called ps2pdf that converts the PostScript data stream to PDF format in much the same way as Adobe's Distiller does for Adobe Acrobat. Most Linux distributions install Ghostscript by default. So unless you chose otherwise, it's probably there. The ps2pdf utility normally resides in /usr/bin so check there to see if you have it. If you do not have Ghostscript, you can get it from <http://www.ghostscript.com>.

4.2 The Windows Computer

To create PDF files, the only thing you will need on this computer is some sort of program to create your documents. This can be something rudimentary like a word processor (such as Microsoft Wordpad), something of medium complexity (such as Powerpoint) or something complex (like MS Publisher or Quicken).

To view the PDF files, you will need Adobe's Acrobat Reader. This is a free application and can be downloaded from Adobe's website at www.adobe.com

4.3 Network Connection

Of course it goes without saying that to get your document from the Windows-based application to ps2pdf, you will need some sort of network connection. Without it, you're dead in the water. So if you don't have your two machines networked, now is a good time. Again, some good resources can be found on the aforementioned Linux Documentation Project site (<http://www.tldp.org>) and in any of a myriad of books covering networking computers.

5. Making It Work

5.1 Adding the Pseudo-Printer

To add the pseudo-printer, you will need to edit the Samba configuration file (`smb.conf`) and add a section similar to the following:

```
[smbpdf]
comment = PDF File Generator
path = /var/spool/samba
printable = yes
print command = /usr/bin/printpdf %s %J
lpq command =
lprm command =
```

The `[smbpdf]` defines the share that will be available. This can be any name you like, but try to keep it short.

The `comment =` is the general description of the share. Something short that tells the user what it is.

The `path =` is the path to the directory where Samba will queue up the print job. Make sure this directory is writable to all who will be generating PDF files.

The `printable =` is self-explanatory.

The `print command =` is the important one. This is where you link the pseudo-printer to the `ps2pdf` utility via a shell script called `printpdf` and located in the `/usr/bin` directory. The `%s` and `%J` pass the file contents and the file name to the script.

The `lpq command =` and `lprm command =` are left blank. This is because we are sending the print job to the `ps2pdf` utility and therefore don't need to make use of Linux's printer controls. However, they do seem to increase response times between the two computers a bit.

Once you have made the above changes, save them then go on to the next section.

5.2 Creating the Script File

Now cd to the /usr/bin directory and create a new file called printpdf. This file should have the following as a minimum:

```
#!/bin/sh
OUTDIR=/temp
SPDIR=/var/spool/samba
ps2pdf13 $1 $OUTDIR/$2.pdf
rm $SPDIR/smbprn*
```

The first line tells Linux this is a shell script.

The second line defines where the PDF file will end up when it is created. This can be any directory on the Linux computer that you want. One caution: make sure the file is saved someplace that is shared by Samba so that you can get to it from the Windows computer.

The third line defines where the Samba print spool directory is located. This is the same location as defined by `path =` in the `smb.conf` file (see section 5.1 above).

The fourth line runs the `ps2pdf` utility specifying PDF format version 1.3 and higher (`ps2pdf13`). It takes the data stream (`$1`), converts it to PDF then places the pdf file in the directory designated by `$OUTDIR`. The PDF file name is the same as the originating file's name plus `pdf`. For instance: if you “printed” `john.doc` to the pseudo-printer, the output file would be located in `/temp` and be called `john.doc.pdf`. NOTE: You may have to play around with the variables a bit depending on your version of Windows. Although `$2` works on Windows 2000, there are reports that a combination of `$2 $3 $4 & $5` are needed on Windows XP machines.

The fifth line removes the spooled print job from the spool directory defined in line three (no, Samba does not do this automatically). This bit of garbage collection will keep the partition where the spool directory resides from filling up with old print jobs. All of Samba's print jobs start with `smbprn` so we can remove all files starting with `smbprn` to quickly do away with them. You can structure this as needed for a more elegant solution.

Once you have created this file, save it and make it executable (a `chmod 755` should do it).

5.3 Restart Samba

Now restart Samba. This will make the new pseudo-printer available to your Windows-based computer(s). You can check to see if the pseudo-printer is available by navigating to the Linux computer (via Network Neighborhood/My Network Places) and looking for a printer icon labeled smbpdf (or whatever you called it).

If the printer is not there, double-check the Samba config file changes you made as detailed in 5.1 above the restart Samba again.

5.4 Add the Pseudo-Printer to Windows

Once you have the pseudo-printer visible to Windows, right-click on it. From the pop-up menu select Connect. You will get a message saying that there are no drivers loaded for this printer. Click on OK so that you can load them.

Next you will see the Add Printer dialog box. You can choose any type of PostScript capable printer you want as your source for drivers. However, I have had the best results from the the HP LaserJet 2100 Series PS drivers.

Once you have the printer chosen, click OK. The drivers will now be loaded. Once they are, the printer is now available for printing. You can see it in Start->Settings->Printers.

5.5 Do a Test Print

Start up an application you can print from. As mentioned before, this can be almost anything, even your web browser. Once you have started the app, open a document and print it via the standard File->Print menu choices. When the Print dialog box appears, change the printer to the smbpdf printer, then click on OK. Your document will “print” to the pseudo-printer where it will be converted to PDF and placed in /temp (or whatever directory you chose to deposit it in).

Once printing is done, navigate to the /temp folder on the Linux computer. There you will find a PDF file with the file name of the file you printed. Open it by double-clicking on it. The file will open in Adobe Acrobat Reader. You are now capable of creating PDF files.

NOTE: If the file fails to open and Acrobat Reader gives an error then you probably chose a printer driver that does not support PostScript well or at all. In this case, delete the printer from the Windows computer and repeat the actions in 5.4 using a different printer driver.

6. Go Further

For those of you with a high level of scripting competency, you can make the printpdf script do pretty much anything you want. How about having the script place the completed PDF file in a user's home directory? Or e-mail it to the creator? The possibilities are endless.

7. Bibliography

The following sources were used to compile this document:

- Roderick W. Smith. *Using Samba Pseudo-Printer*. Linux Magazine, January 2003.
- John Bright. *PDF Service With Samba*. Linux Gazette, Issue 72. Available online at <http://www.linuxgazette.com>.
- Dictionary of Computer and Internet Terms, 7th Edition*. Barrons Educational Series, Inc. 2000.
-

8. Revision History

Version	Date	Changes
1.0	13 Feb 03	- Original version
1.1	22 Apr 04	- Updated printpdf script to reflect name of actual file being converted to PDF. Thanks to Obi Bob Godwin for this one. - Added info about OpenOffice.
1.2	7 May 04	- Revised the printpdf script to clean up

1.2.1

20 May 06

Samba print spool directory. Thanks to Jim Richardson for bringing this to my attention.
- Corrected e-mail address.
- Minor format changes.