



Roy Costa
Michael Maclsaac

Printing with Linux on IBM *@*server zSeries Using CUPS and Samba

Introduction and overview

This Redpaper discusses print serving on Linux using the Common UNIX® Printing System (CUPS) and Samba. Print serving is much more complex than file serving. With file serving, the “back end” is a Linux file system. All the specifics of hardware, such as the disk, the DASD, the controller and so on are abstracted away by the Linux kernel and operating system. With printing, this is not possible because the back end is hardware that is much more varied; printers have many more options and characteristics such as paper trays, duplex printing, cover pages, and so on.

CUPS an open source Linux print server that supports LPD and the newer Internet Printing Protocol (IPP). CUPS seems to be the new, de facto standard print server on Linux, moving ahead of the older lpr and the new LPRng.

Samba is also an open source package that lets Linux look like a Windows® file and print server. It enables Windows desktops to access CUPS-defined printers, using a familiar desktop printing interface. This paper is based on work done on Linux on zSeries® (the mainframe), but contains little information that is specific to zSeries. Therefore, it should be useful for Linux running on any hardware platform.

Two Linux distributions are addressed: SUSE Linux Enterprise Server 8 (SLES-8) with Service Pack 3 (SP3), and Red Hat Enterprise Linux 3 (RHEL-3). Customizing and running CUPS and Samba are also addressed. The following aspects of a production print serving solution are discussed:

- ▶ Printing job cover sheets with adequate information
- ▶ Printing on both sides of the paper (duplex)
- ▶ Setting up print classes or pools (groups of physical printers in a single queue)
- ▶ Setting up automatic driver download from Windows desktops (also known as *point and print*)
- ▶ Setting up a PDF *drop box*

The team that wrote this Redpaper

This Redpaper was produced by a team of specialists working with the International Technical Support Organization (ITSO) in Poughkeepsie, NY.

Roy Costa is an Advisory Systems Programmer at the International Technical Support Organization, Poughkeepsie Center. He has 18 years of experience in VM systems programming. Roy has worked with Linux for zSeries for more than three years, and has provided technical advice and support to numerous IBM Redbooks™ for the past eight years.

Michael Maclsaac works in the zSeries New Technology Center in Poughkeepsie, NY, providing technical support for Linux and z/VM®. He is a former ITSO project leader who contributes to ITSO projects.

Thanks to the following people for their contributions to this project:

- ▶ Kurt Pfeifle, Emmett O'Grady
- ▶ Michael Sweet
Primary developer of CUPS, co-owner of Easy Software Products

Overall approach

This paper is organized into the following high level steps:

- ▶ “Install Linux with CUPS and Samba” on page 3
- ▶ “Configure and start CUPS” on page 3
- ▶ “Create and test CUPS printers” on page 7
- ▶ “Create and test a CUPS class” on page 11
- ▶ “Configure and start Samba” on page 13
- ▶ “Set up printer drivers and automatic download” on page 16
- ▶ “Set up a PDF drop box” on page 24

Note: The steps outlined in this redpaper have been tested on 31-bit SLES-8 and RHEL-3 distributions. Your results might vary for other distributions. At least one difference is noted in “Set up a PDF drop box” on page 24.

Install Linux with CUPS and Samba

Obtain SLES-8 from SUSE, and obtain RHEL-3 from Red Hat. Installing the distributions is not discussed in this paper, but you can find information on installing SLES-8 and service packs in the IBM Redbook *IBM Lotus Domino 6.5 on Linux for zSeries Implementation*, SG24-7021.

Using the CUPS and Samba RPMs that come with the two distributions is recommended. (Often there are recommendations that you upgrade to the latest packages off the Internet and build them yourselves, but we do not recommend this unless there is a function you are *sure* that you need, because your Linux distributor will probably not support custom-built applications that they have not tested.)

In general, a root file system of a single 3390-3 does not leave sufficient disk space for a print server. There are many possible configurations for laying out file systems. One suggested configuration is three 3390-3s, one for each of / (root), /usr and /var.

The remainder of this paper gives instructions common to both the SUSE and Red Hat distributions. Where there are differences between the two distributions, they are pointed out.

Configure and start CUPS

Following is a list of CUPS system and user commands:

System commands (in /usr/sbin)

accept	accept print jobs to the specified destinations
cupsaddsmb	export printers to samba for Windows clients
cupsd	the cups daemon - Web browser listening on port 631 (ipp)
lpadmin	set default, create or delete cups printers and classes
lpinfo	show available devices or drivers
lpmove	move a job to a new destination
reject	symbolic link to accept
lpc	line printer control program

User commands (in /usr/bin)

cups-config	query various CUPS configuration values
cancel	cancel jobs

disable	Symbolic link to accept
enable	Symbolic link to accept
lp	print files
lpoptions	display or set printer options and defaults
lppasswd	add, change, or delete digest passwords
lpstat	print CUPS status information
lpq	show printer queue status
lpr	print files
lprm	cancel print jobs

The CUPS configuration directory is `/etc/cups/`. The `printers.conf` file contains the definitions of CUPS printers, and the `classes.conf` file contains the definitions of the classes. These files are analogous to the file `/etc/printcap` on more traditional UNIX and Linux systems that use the `lpd` print server, though the syntax is more straightforward. The `cups.conf` file contains configuration data for the CUPS Web interface that is shown in Figure 1 on page 7.

The directory `/etc/cups/ppd/` contains PostScript Printer Definition (PPD) files. These files describe the capabilities of PostScript printers, whether they are physical or virtual. There will be one PPD file in this directory for each printer defined to CUPS. Each file will contain a unique description of the printer's PostScript capabilities and physical attributes.

Red Hat-specific: The default `/etc/cups/cupsd.conf` file does not allow connection to the CUPS Web server. It is recommended that the modified file below be copied.

The default CUPS configuration file contains so many comments that it is difficult to read. We recommend that you make a backup of the original and remove comments via the following commands:

```
# cd /etc/cups
# cp cupsd.conf cupsd.conf.orig
# egrep -v '^#|^$' cupsd.conf.orig > cupsd.conf
# wc cupsd*
      16      35      269 cupsd.conf
     761     3171    19969 cupsd.conf.orig
```

Edit the file `cupsd.conf` and comment out the two lines with **Deny from all**. This will allow any client to connect to the CUPS Web interface. If you are security conscious, you may want to leave these lines in and “Allow” connections only from specific machines:

```
# vi cupsd.conf    -> comment out two lines
DocumentRoot /usr/share/cups/doc/
LogLevel info
Port 631
<Location />
Order Deny,Allow
#Deny From All
Allow From 127.0.0.1
Allow From 127.0.0.2
</Location>
<Location /admin>
AuthType Basic
AuthClass System
Order Deny,Allow
#Deny From All
Allow From 127.0.0.1
</Location>
```

If you are having problems with CUPS and need aid in debugging, you may wish to change the LogLevel to debug rather than the default value of info in order to get more output in the CUPS logs (in the /var/log/cups/ directory).

If you wish to limit the amount of Completed Jobs that are displayed from the Web browser, you can add the line MaxJobs (default is no maximum) to cupsd.conf:

```
MaxJobs 50
```

Determine if the Postscript Printer Definition (PPD) file for your printer is installed. You want to be sure the appropriate PPD file is in place before starting the CUPS daemon. SLES-8 comes standard with many PPD files, with a separate directory for most printer manufacturers:

```
# ls /usr/share/cups/model
Alps/      Epson/      Mitsubishi/ Samsung/     jp/
Anitech/   Fujitsu/    NEC/        Seiko/      laserjet.ppd.gz
Apollo/    HP/         Oce/        Sharp/      okidat24.ppd.gz
Apple/     Heidelberg/ Okidata/    Sony/       okidata9.ppd.gz
Avery/     Hitachi/    Olivetti/   Star/       stcolor.ppd.gz
Brother/   IBM/        PCPI/       Tally/      stcolor2.ppd.gz
CItoh/     Imagen/     Panasonic/  Tektronix/  stp/
CalComp/   Infotec/    Pentax/     Xerox/      stphoto.ppd.gz
Canon/     Kodak/      Postscript.ppd.gz deskjet.ppd.gz stphoto2.ppd.gz
Citizen/   Kyocera/    Printrex/   deskjet2.ppd.gz
Compaq/    LaserMaster/ QMS/        dymo.ppd.gz
DEC/       Lexmark/    Raven/      epson24.ppd.gz
Dymo-CoStar/ Minolta/    Ricoh/      epson9.ppd.gz
```

The RHEL-3 distribution has far fewer PPD files. However, neither SLES-8 nor RHEL-3 had the PPD file for the printer model used in this paper, which was the IBM 4324 (Network Printer 24). An example of the PPD file was found on the Web at:

<http://cvs.sourceforge.net/viewcvs.py/cups/cups-drivers/ppd/>

Red Hat-specific: Red Hat does not have a subdirectory structure below /usr/share/CUPS/model, so the subdirectory IBM must first be created:

```
# mkdir IBM
```

We downloaded the ibm43241.ppd file to a download server, and copied it via FTP to the /usr/share/cups/model/IBM/ directory on SLES-8 and the /usr/share/cups/model/ directory on RHEL-3. It was then compressed via the **gzip** command:

```
# cd IBM
# ls
3853_JetPrinter-jetp3852.ppd.gz
4019-gimp-print.ppd.gz
...
# ftp <your.Download.Server>
ftp> cd <your/Download/Directory>
ftp> get ibm43241.ppd
ftp> quit
# gzip ibm43241.ppd
```

An internal database of PPD files is created by CUPS when it is started. Therefore, if you add additional PPD files later, CUPS must be restarted in order for them to be recognized.

It is now time to start the CUPS daemon. Do this for both this session and across reboots. On SLES-8, the rc service commands are symbolic links to the service scripts in the

/etc/init.d/ directory. For example, the **rc cups** command is really a symbolic link to **/etc/init.d/cups** service script:

```
# rc cups status
Checking for cupsd:                unused
# rc cups start
Starting cupsd                     done
# rc cups status
Checking for cupsd:                running
```

Red Hat-specific: Rather than having a set of symbolic links in `/usr/sbin/rc*` to the service scripts, RHEL-3 uses a single prefix command, **service**, to start and stop services. To check the status of CUPS and start it on RHEL-3, use the following commands:

```
# service cups status
cups: scheduler is not running.
# service cups start
cups: started scheduler.
```

Now CUPS is running for this session. You will also probably want it to start up across reboots. For both distributions this is done with the **chkconfig** command, though the syntax is slightly different. For SLES-8, the command and output are simple:

```
# chkconfig cups
cups off
```

For RHEL-3, the **--list** flag is needed and the output shows settings for various run levels:

```
# chkconfig --list cups
cups          0:off  1:off  2:on   3:on   4:on   5:on   6:off
```

For either distribution, the command to set CUPS to come up at boot time is:

```
# chkconfig cups on
```

Now that CUPS is started and is set to start at boot time, verify that the PPD file was added via the **lpinfo** command:

```
# lpinfo -m | grep 4324
ibm43241.ppd.gz IBM Network Printer 24
```

Now that CUPS is running, test the Web interface. Point a browser to the well-known IPP port 631 of your Linux server. You should see the screen shown in Figure 1:

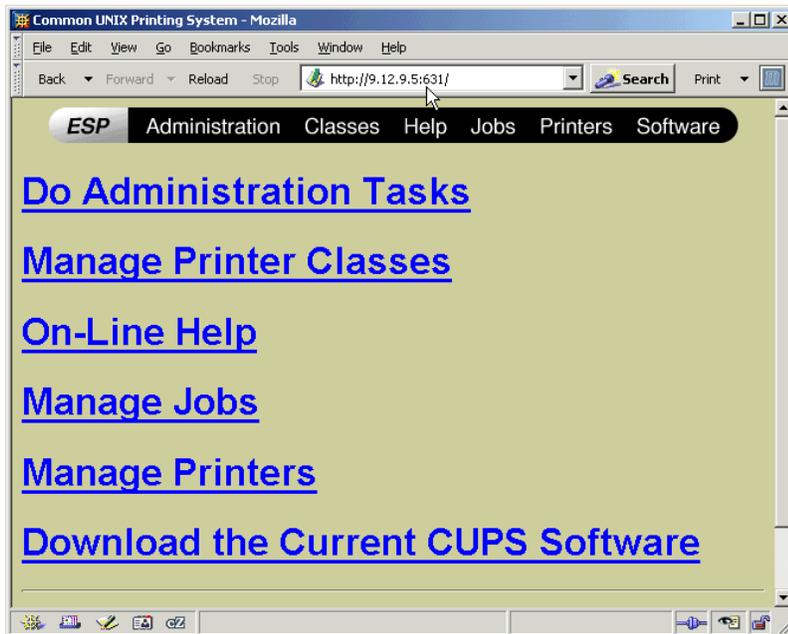


Figure 1 CUPS Web interface

From this Web interface, along with monitoring print jobs, you can define, configure and delete printers and printer classes. To perform these administrative tasks, you will be prompted for an authorized userid and password.

The line commands are used for the initial printer and class setups. This should make it easier to understand what needs to be done and which configuration files are being changed.

To learn more, refer to *CUPS Software Administrator's Manual*, which is available from the Web interface in the file `sam.pdf`:

<http://9.117.99.38:631/sam.pdf>

Create and test CUPS printers

Now that the PPD file is in the CUPS PPD database, two other pieces of information are needed, the TCP/IP address of the printer (network printers are assumed) and the LPD queue name. (CUPS can speak other protocols to the printer, including IPP, but only LPD is addressed in this paper.)

In the following example, the TCP/IP address is 9.12.2.6 and the LPD queue name is PASS. The queue name is often built into the printer and should be documented in the printer's technical manual.

With all this information, the printer can be added via the `lpadmin` command:

```
# lpadmin -p poknp24a -E -v lpd://9.12.2.5/PASS -m IBM/ibm43241.ppd.gz
# lpadmin -p poknp24b -E -v lpd://9.12.2.8/PASS -m IBM/ibm43241.ppd.gz
# lpadmin -p poknp24c -E -v lpd://9.12.2.6/PASS -m IBM/ibm43241.ppd.gz
```

The flags used have the following meaning:

- p The printer will be named poknp24.
- E The printer will be enabled.

- v The printer IP address and LPD queue name in the form //<host>/<queue name>.
- m PPD file to use relative to directory /usr/share/cups/model/.

A cover sheet attribute can also be added to the printer via the **-o** flag (options) of the **lpadmin** command:

```
# lpadmin -p poknp24a -o job-sheets-default=standard
# lpadmin -p poknp24b -o job-sheets-default=standard
# lpadmin -p poknp24c -o job-sheets-default=standard
```

Often printers are used by many people in a shared environment. Therefore, a banner page printed on colored paper from a different paper tray is required to delineate the jobs. A modified version of the standard cover sheet and a tip on setting options, such as paper tray, is described in “Banner customization” on page 26.

Now look at the files that were modified to get an idea of what is going on behind the scenes. The file /etc/cups/printers.conf contains the information for the three printers just added:

```
# cd /etc/cups
# cat printers.conf
# Printer configuration file for CUPS v1.1.15
# Written by cupsd on Tue Feb 10 21:13:05 2004
<DefaultPrinter poknp24a>
Info poknp24a
DeviceURI lpd://9.12.2.5/PASS
State Idle
Accepting Yes
JobSheets standard none
QuotaPeriod 0
PageLimit 0
KLimit 0
</Printer>
<Printer poknp24b>
Info poknp24b
DeviceURI lpd://9.12.2.8/PASS
...
```

A PPD file for each of the newly defined printers can be found in the /etc/cups/ppd/ directory. They are copied from the generic file for the printer type that was added earlier. As modifications are made to a printer, the PPD file for that printer is modified.

```
# cd ppd
# ls
poknp24a.ppd poknp24b.ppd poknp24c.ppd
```

Set up printer options

The **lpoptions** command with no flags shows the options set on the printer. If you have defined more than one printer, **lpoptions** will show you options on the first printer defined. Use the **-p <printername>** for specific printer options.

```
# lpoptions -p poknp24a
job-sheets=standard,none
```

All the options available to the printer can be shown with the **lpoptions -l** command:

```
# lpoptions -p poknp24b -l
IBJamRecovery/Jam recovery: *False True
Jog/Jog: *None EndOfJob EndOfSet
IBPageProtect/Page protect: *False True
TraySwitch/Printer select tray: False *True
IBStaple/Staple location: DoubleLandscape DoublePortrait SingleLandscape SingleP
```

```

ortrait *None
Duplex/Duplex (Two-sided printing): *DuplexNoTumble None DuplexTumble
ManualFeed/Manual feed: False True
PageSize/Media Size: Tabloid A3 A4 A5 B4 ISOB5 B5 EnvC5 Env10 EnvDL Executive Fa
nFoldGermanLegal DoublePostcardRotated Legal *Letter EnvMonarch Statement
InputSlot/Media Source: Auto MPT Envelope *Tray1 Tray2 Tray3
OutputBin/Output bin: Face-Up BinContinue BinLowerFD BinLowerFU *Main BinMiddleF
D BinMiddleFU BinUpperFD BinUpperFU
PageRegion/PageRegion: Tabloid A3 A4 A5 B4 ISOB5 B5 EnvC5 Env10 EnvDL Executive
FanFoldGermanLegal DoublePostcardRotated Legal Letter EnvMonarch Statement
OutputMode/Print quality: *TruRes None TonerSaver
OptionDuplexUnit/Duplex unit: True *False
OptionEnvelope1/Envelope feeder: True *False
OptionFinisher1/Finisher-stapler: True *False
OptionFlash1/Flash: True *False
OptionHardDrive1/Hard drive: True *False
InstalledMemory/Total Printer Memory: *8Meg 10Meg 12Meg 14Meg 16Meg 20Meg 22Meg
24Meg 28Meg 36Meg 38Meg 40Meg 44Meg 52Meg 68Meg
OptionTray3/Tray 3: True *False

```

As you can see, there are many features that can be modified. You may need to make specific modifications for your printers. Using the Web interface for modification works well and is recommended.

Before you go any further, it is good practice to create a backup of each of your printer's PPD files:

```

# cp poknp24a.ppd poknp24a.ppd.orig
# cp poknp24b.ppd poknp24b.ppd.orig
# cp poknp24c.ppd poknp24c.ppd.orig

```

You could also combine the previous commands into one:

```

# for i in a b c
> do
> cp poknp24$i.ppd poknp24$i.ppd.orig
> done

```

Some of the options that you may want to modify are:

- ▶ Enabling duplex printing
- ▶ Enabling additional paper trays (Installed)
- ▶ Selecting the default paper tray (Media Source)
- ▶ Setting starting and ending banners - cover sheets

Set up duplex printing and enable tray 3

The printers are modified via the Web interface. From the main page, go to **Manage Printers**. All the printers that you have defined should be displayed. Select **Configure Printer**; you should see the screen shown in Figure 2. In this example, Duplex is set to Long edge and the options Duplex Unit and Tray 3 are set to Installed.

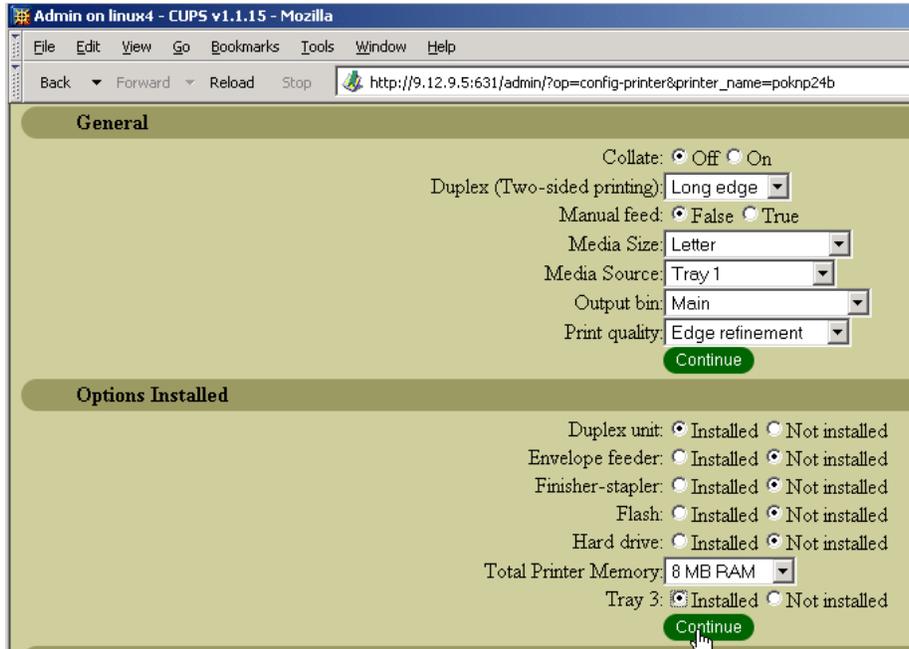


Figure 2 Setting Duplex via CUPS Web interface

Scrolling down further, you can make any other needed modification. After you are finished, click **Continue** to save your changes.

You should see the following message:

Printer poknp24a has been configured successfully.

See what changed in one printer's PPD file after the duplex option was set to "long edge" and duplex and tray3 were enabled:

```
# diff poknp24a.ppd poknp24a.ppd.orig
30c30
< *DefaultOptionTray3: True
---
> *DefaultOptionTray3: False
42c42
< *DefaultOptionDuplexUnit: True
---
> *DefaultOptionDuplexUnit: False
709c709
< *DefaultDuplex: DuplexNoTumble
---
```

Modify cover sheet layout

The standard cover sheet that will be printed before each main print job will be the PostScript file `/usr/share/cups/banners/standard`. It will contain the printer job ID, the file name and the user name, but not the date and time of printing. It may be desirable to modify this default file so that the date and time are printed, and to have the user name printed in a large font at the top. For details on how to modify the PostScript file, see "Banner customization" on page 26.

Once the modification has been made, simply copy it to the `/usr/share/cups/banners/` directory with the name `standard`. For example:

```
# cd /usr/share/cups/banners/
# cp standard standard.orig
```

```
# ftp <your.ftp.server>
...
ftp> get standard
...
```

Test printing

You should now be ready to test a print job from Linux with the **lpr** command. In the following example, the file `foo.txt` was created with more than 72 lines to test the duplex feature. After the job is sent, you can use the **lpq** command to see the job in the CUPS queue. If the printer is not busy, it will move off the CUPS queue into the printer queue in a few seconds. Note the difference in the output of the two **lpq** commands:

```
# lpr foo.txt
# lpq
poknp24 is ready and printing
Rank  Owner  Job   File(s)                Total Size
active root    2     foo.txt                8192 bytes
# lpq
poknp24 is ready
no entries
```

You can also print a test page from the Web interface; use **Print Test Page** within the **Printers** view.

Now you have CUPS printers defined and working. The next step is to create a pool of multiple physical printers into a single print queue.

Create and test a CUPS class

A printer pool or class is advantageous for several reasons. A printer class uses a round robin approach for sending jobs to the printers in the class, therefore print jobs should get distributed among printers in the class. If a single printer goes down due to mechanical failure or lack of paper or ink, other jobs can still be printed on the remaining functioning printers.

To create printer classes, you will need at least two identical printers. The following example uses three.

Printer classes can be defined using the line commands, but this is a good opportunity to show the Web interface.

From the Web interface, select **Classes** on the top tool bar and click **Add Class**. Give the class a name, describe the location, and give it a description. Click **Continue**, then add the printers that will comprise the class. Use Shift-click to select multiple printers. Finally, click **Continue** to save. This is shown in Figure 3.

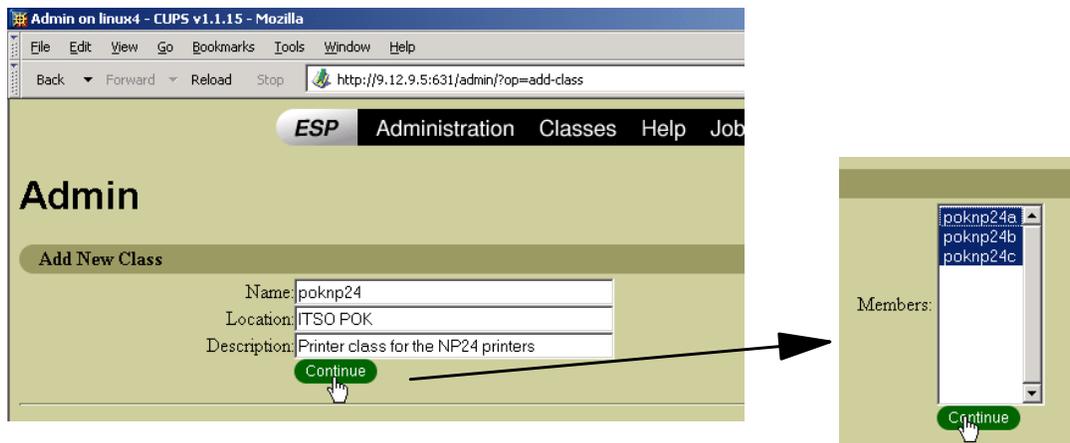


Figure 3 Defining a CUPS printer class

You should see the following message:

Class poknp24 has been added successfully.

Tip: If you want to use the line command rather than the Web interface, the following commands will accomplish the same task:

```
# lpadmin -p poknp24a -c poknp24
# lpadmin -p poknp24b -c poknp24
# lpadmin -p poknp24c -c poknp24
# /usr/bin/enable poknp24
```

Because the Web interface does not have a way to set the JobSheet option, this should be done from the `lpadmin` command as follows:

```
# lpadmin -p poknp24 -o job-sheets-default=standard,none
```

Note: If you get an error with this command, you may have to edit the `classes.conf` file; the file `/etc/cups/classes.conf` should be similar to the following example.

```
# Class configuration file for CUPS v1.1.15
# Written by cupsd on Thu Feb 12 17:06:32 2004
<Class poknp24>
Info Printer class for the NP24 printers
Location ITSO Pok
State Idle
Accepting Yes
JobSheets standard none
Printer poknp24a
Printer poknp24b
Printer poknp24c
QuotaPeriod 0
PageLimit 0
KLimit 0
</Class>
```

Configure and start Samba

Now that printing is working from Linux to the printer via CUPS, it is time to configure Samba. Samba is not really a print server; instead it can be viewed as a “middleman” between a print server and Windows clients that want a familiar printing interface.

Samba is configured for the most part by one configuration file that is usually named `/etc/samba/smb.conf`. Like CUPS, Samba has a Web interface called the Samba Web Administration Tool (SWAT). Again, editing the configuration file directly is recommended. A side effect of using SWAT to maintain the `smb.conf` file is that all comments in the file are deleted. However, if you like GUIs for administration, you will probably prefer SWAT.

The issue of the many types of authentication possible with Samba is complex and is beyond the scope of this paper. This example uses the default Samba authentication scheme of obtaining credentials from the `smbpasswd` file (when the parameter `encrypt passwords = yes`).

SLES-8 comes with a useful default `smb.conf` file. Make a backup copy of the original and then make a few modifications as described. This same configuration file will be used on both SLES-8 and RHEL-3.

Red Hat-specific: The default `smb.conf` file that Red Hat ships is significantly different from SUSE's. The following `smb.conf` file was used on both.

```
# cd /etc/samba
# cp smb.conf smb.conf.orig
# vi smb.conf -> add three lines, comment out one and add to the write list
# smb.conf is the main Samba configuration file. You find a full commented
# version at /usr/share/doc/packages/samba/examples/smb.conf.SuSE
# Date: 2003-10-27
[global]
    workgroup = TUX-NET
    os level = 2
    time server = Yes
    unix extensions = Yes
    encrypt passwords = Yes
    map to guest = Bad User
    printing = CUPS
    printcap name = CUPS
    socket options = SO_KEEPALIVE IPTOS_LOWDELAY TCP_NODELAY
    wins support = No
    veto files = /*.eml/*.*.nws/riched20.dll/*.*}/
    netbios name = linux4 1
    printer admin = mikem 2
[homes]
    comment = Home Directories
    # valid users = %S 3
    browseable = No
    read only = No
    create mask = 0640
    directory mask = 0750
[printers]
    comment = All Printers
    path = /var/tmp
    printable = Yes
    create mask = 0600
    browseable = No
    print command = lpr -r -P'%p' -J'%J' -U'%U' %s 4
```

```

    guest ok = yes 5
    guest account = ftp 5

[print$]
    comment = Printer Drivers
    path = /var/lib/samba/drivers
    write list = @ntadmin root mikem 6
    force group = ntadmin
    create mask = 0664
    directory mask = 0775
    guest ok = yes 5
    guest account = ftp 5

```

The lines that were changed are as follows:

- 1** Give the Samba server a name. Use the DNS name, if possible.
- 2** Allow the user mikem to manipulate printers.
- 3** This parameter appears to break the [homes] section under RHEL-3.
- 4** Adds **-J %J** to the **lpr** command so the correct cover sheet file name is used, and **-U %U** so the correct user name is used.
- 5** Allows any user to access printers as the user ftp.
- 6** Adds the user mikem to upload printer drivers. You could also have added mikem to the group ntadmin due to the fact that the entire ntadmin group (@ntadmin) is on the write list.

After you modify the smb.conf file, check the syntax with the **testparm** command. Watch for any errors in the first few lines of output:

```

# testparm smb.conf
Load smb config files from smb.conf
Processing section "[homes]"
Processing section "[printers]"
Processing section "[print$]"
Loaded services file OK.
Press enter to see a dump of your service definitions
...

```

Before you start Samba, there are a few more tasks you need to do. Create a user (in this example, mikem) in both the /etc/passwd and the /etc/samba/smbpasswd files. The /etc/passwd file contains the standard Linux hash of the user's password, and the /etc/samba/smbpasswd file contains the NT4 and LANMAN hashes of the same password needed to authenticate Windows users.

The following commands create the user mikem with an empty home directory, and set both the Linux and Samba passwords. It is recommended that you use the same user ID and password that is used to access your Windows desktop.

```

# useradd -m -k /dev/null mikem
# passwd mikem
Changing password for mikem.
New password:
Re-enter new password:
Password changed
# smbpasswd -a mikem
New SMB password:
Retype new SMB password:
Added user mikem.

```

Red Hat-specific: Red Hat does not have subdirectories below `/var/lib` for drivers, so the correct structure must first be created. Also, the group named `ntadmin` must be added:

```
# cd /var/lib
# mkdir -p samba/drivers
# groupadd ntadmin
# chgrp ntadmin samba/drivers
# chmod g+w samba/drivers
```

Now add a secondary group `ntadmin` for the user `mikem`. After you add the group, verify that it was added by using the `id` command:

```
# usermod -G ntadmin mikem
# id mikem
uid=501(mikem) gid=100(users) groups=100(users),71(ntadmin)
```

One other item is worth pointing out. The *sticky bit* is set on the Samba print file directory, and is denoted by a trailing “t”. This allows each user’s print job to be deleted after it has been printed.

```
# ls -ld /var/tmp
drwxrwxrwt  3 root  root  4096 Jan 23 13:07 /var/tmp/
```

For a quick test, switch user to the new printer admin user to test that you can write to the drivers directory, which will have to be done later. For example:

```
# su - mikem
# cd /var/lib/samba/drivers/W32X86
# touch foo
# rm foo
# exit
```

Start Samba

Start Samba for this session and across reboots. On SLES-8, it is started with both the `rcnmb` and `rcsmb` commands. Previous versions of SLES and other Linux distributions combine the two services into the `smb` script.

```
# rcnmb start
Starting Samba classic NMB daemon                               done
# rcsmb start
Starting Samba classic SMB daemon                               done
```

Red Hat-specific: Samba services are started with the `service smb` command:

```
# service smb start
Starting SMB services: [ OK ]
Starting NMB services: [ OK ]
```

Again, the services are set to start at boot time with the `chkconfig` command:

```
# chkconfig nmb on
insserv: script boot.proc.orig: service boot.proc already provided!
# chkconfig smb on
insserv: script boot.proc.orig: service boot.proc already provided!
```

It is helpful but not necessary to test the Samba setup and connectivity by first trying to get a file share. Because the `[homes]` section is present, a share name equal to the user name is

available for all users with home directories. The share can be obtained from the Windows Explorer **Map Network Drive** function, or from DOS via the **net use** command as shown:

```
C:\>net use * \\9.12.9.5\mikem
Drive E: is now connected to \\9.12.9.5\mikem.
```

The command completed successfully.

Getting the share shows that Samba connectivity and authentication are working.

You can also use the **net view** command to see the available shares:

```
E:\>net view \\9.12.9.5
Shared resources at \\9.12.9.5
```

Samba 2.2.8a-SuSE

Share name	Type	Used as	Comment
------------	------	---------	---------

mikem	Disk		Home Directories
poknp24a	Print		poknp24a
poknp24b	Print		poknp24b
poknp24c	Print		poknp24c
poknp24	Print		Printer class for the NP24 printers

The command completed successfully.

Note that the CUPS-defined printers are present. Just as the [homes] section automatically creates a file share for each user, so the [printers] section automatically makes all printers defined to Linux appear as Windows printer shares.

Set up printer drivers and automatic download

The printer poknp24a could now be used from Windows for the user mikem. But what if the desktop machine does not have the right drivers installed? A common course of action is for the user adding the printer to try to find the right drivers. One of the functions of SMB print serving allows for automatic downloading of printer drivers; this function is sometimes called *point and print*.

Both Windows print servers and Linux systems with Samba running support this function. Most would agree it is a better and more standardized solution for the correct drivers to exist on the server and relieve the end user of the chore of finding the correct drivers.

A critical step in this process is to first get the correct Windows printer drivers for your printer onto the server. It is possible that valid printer drivers for your printer will be available on the Windows installation CD. However, you may need to upload multiple sets of drivers to support different versions of Windows clients, and therefore multiple Windows Operating Systems CDs.

There are two basic approaches to loading printer drivers onto the CUPS/Samba server:

- ▶ On Linux, by using the **cupsdmsmb** command with generic Adobe PostScript drivers
- ▶ From a Windows desktop, by using the Add Printer Driver Wizard

Loading drivers on Linux using the **cupsdmsmb** command

An option is to not use the manufacturer's printer drivers, but instead to use the generic Adobe PostScript drivers, which can be installed via the CUPS **cupsdmsmb** command on the

Linux server. This command runs Samba `rpcclient` commands behind the scenes to populate the printer's drivers directory. The overall steps for this process are:

- ▶ Download Adobe drivers from the Web.
- ▶ Install/extract drivers on a Windows desktop.
- ▶ Move certain driver files from Windows to Linux.
- ▶ Run `cupsaddsmb` to populate the Samba drivers directory.

Download Adobe drivers from the Web

The Adobe PostScript generic drivers, both versions 4 and 5 (though version 5 is now numbered 1.0.6), must be downloaded from the Web. Start at the Adobe Web download site:

<http://www.adobe.com/support/downloads/>

Click **Windows** under the heading **PostScript printer drivers**. This should bring you to the Web site:

<http://www.adobe.com/support/downloads/product.jsp?product=44&platform=Windows>

Click the two links for version 1.0.6 and 4.2.6, corresponding to the language you want. For English, click:

- ▶ **Adobe Universal PostScript Windows Driver Installer 1.0.6 - English**
- ▶ **PostScript Printer Driver AdobePS 4.2.6 for Windows 95 and Windows 98 -- U.S. English**

Each of these two links will bring you to a download screen. Click **Download** for each of them; this should enable you to download the files `winsteng.exe` and `aps426eng.exe`.

Install/extract drivers on a Windows desktop

After the files are downloaded, they must be run from a Windows desktop. Execute the file **winsteng.exe**, which will unpack files then kick off the Add Printer Wizard. For the Printer Connection Type panel, accept the default of **Local Printer**. For the Local Port Selection panel choose **FILE: Local Port**. Accept all default values, but do not bother to print a test file.

The installation of the version 4 drivers is slightly different. Execute the file **aps426eng.exe**. Uncheck **When done unzipping run setup.exe**. This will enable you to extract the files, but not actually install them on the Windows desktop. The default location to copy the drivers is `C:\temp\adobe\AdobePS 4.2.6 - U.S. English`.

Move certain driver files from Windows to Linux

The new drivers from the version 1.0.6 (version 5) package should be in the directory `C:\WINNT\system32\spool\drivers\w32x86\2` or `w32x86\3`. Bring up a DOS prompt and change to that directory.

Then FTP the following files from DOS to your Linux CUPS/Samba server into the directory `/usr/share/cups/drivers/`, making sure that the files remain in upper case:

```
ADOBEPS5.DLL
ADOBEPSU.DLL
ADOBEPSU.HLP
DEFPRTR2.PPD
```

The new drivers from the version 4 package should be in the directory `C:\temp\adobe\AdobePS 4.2.6 - U.S. English`. Again start a DOS session and change to that directory. Again FTP certain files to the `/usr/share/cups/drivers/` directory on Linux. However, this time the last character of the file names has to be modified. Following is a synopsis of the FTP session:

```
# ftp <your.linux.server>
```

```
ftp> put ADfonts.mf_ ADFONTS.MFM
ftp> put adobe4.dr_ ADOBEPS4.DRV
ftp> put adobe4.hl_ ADOBEPS4.HLP
ftp> put iconlib.dl_ ICONLIB.DLL
ftp> put psmon.dl_ PSMON.DLL
```

Now that you have FTP'd the two sets of drivers to Linux, there should be nine files in the directory /usr/share/cups/drivers:

```
# cd /usr/share/cups/drivers
# ls -l
total 2600
-rw-r--r-- 1 root root 284394 Apr 7 1999 ADFONTS.MFM
-rw-r--r-- 1 root root 462184 Apr 7 1999 ADOBEPS4.DRV
-rw-r--r-- 1 root root 75293 Apr 7 1999 ADOBEPS4.HLP
-rw----- 1 root root 457600 Feb 10 15:53 ADOBEPS5.DLL
-rw----- 1 root root 135680 Feb 10 15:53 ADOBEPSU.DLL
-rw----- 1 root root 18451 Feb 10 15:54 ADOBEPSU.HLP
-rw----- 1 root root 13492 Feb 10 15:54 DEFPRTR2.PPD
-rw-r--r-- 1 root root 22343 Apr 7 1999 ICONLIB.DLL
-rw-r--r-- 1 root root 13020 Apr 7 1999 PSMON.DLL
```

After you have these drivers in one place, you can make a compressed tar file of them:

```
# tar czf samba-cups-PSdrivers.tgz *
```

Copy this new tar file to a safe place, as it will come in handy if you have to set up a CUPS print server again.

Run cupsaddsmb to populate Samba drivers directory

Now run the `cupsaddsmb` command for all printers as the printer admin (in this example, mikem). Note that the subdirectory W32X86 under the printer drivers directory /var/lib/samba/drivers is populated with two DLLs, a help file, and a PPD file for each of the three defined printers. First verify that the W32X86 directory is empty:

```
# ls /var/lib/samba/drivers
W32ALPHA/ W32MIPS/ W32PPC/ W32X86/ WIN40/
# ls /var/lib/samba/drivers/W32X86/
```

Now invoke `cupsaddsmb` with the `-v` (verbose) flag, `-a` (all printers) and pipe to `less` to see what is going on behind the scenes:

```
# cupsaddsmb -U mikem -v -a | less
Password for mikem required to access localhost via SAMBA: <mikem-password>
added interface ip=9.12.9.5 bcast=9.12.9.31 nmask=255.255.255.224
Domain=[TUX-NET] OS=[Unix] Server=[Samba 2.2.8a-SuSE]
NT_STATUS_OBJECT_NAME_COLLISION making remote directory \W32X86
putting file /var/spool/cups/tmp/4048a566e3be0 as \W32X86/poknp24a.PPD (777.9 kb
/s) (average 777.9 kb/s)
putting file /usr/share/cups/drivers/ADOBEPS5.DLL as \W32X86/ADOBEPS5.DLL (8593.
7 kb/s) (average 4082.2 kb/s)
putting file /usr/share/cups/drivers/ADOBEPSU.DLL as \W32X86/ADOBEPSU.DLL (26499
.5 kb/s) (average 4957.9 kb/s)
putting file /usr/share/cups/drivers/ADOBEPSU.HLP as \W32X86/ADOBEPSU.HLP (18016
.8 kb/s) (average 5059.1 kb/s)
added interface ip=9.12.9.5 bcast=9.12.9.31 nmask=255.255.255.224
Domain=[TUX-NET] OS=[Unix] Server=[Samba 2.2.8a-SuSE]
NT_STATUS_OBJECT_NAME_COLLISION making remote directory \WIN40
...
```

Watch for errors of files that do not exist, and correct as needed.

This will only enable point and print for the individual printers, not CUPS classes. If you want to print to printer classes, run `cupsaddsmb` for the specific class:

```
# cupsaddsmb -U mikem -v poknp24 | less
```

You should now see that the driver directory is populated, probably under the subdirectory `2/`. You should see the required drivers and a PPD file for each printer defined and for each class that you specifically ran `cupsaddsmb` against:

```
# ls /var/lib/samba/drivers/W32X86/  
2/  
# ls /var/lib/samba/drivers/W32X86/2  
ADOBEPS5.DLL ADOBEPSU.HLP poknp24b.PPD  
ADOBEPSU.DLL poknp24a.PPD poknp24c.PPD poknp24.PPD
```

Loading drivers from a Windows desktop

Perhaps a more common method of obtaining printer drivers is to get them from the printer manufacturer and upload them from a Windows desktop.

Note: If you performed the previous `cupsaddsmb` process on Linux, you do not have to do this step.

In this example, the drivers for the IBM Network Printer 24 were obtained from the Internet. A good starting point for IBM printer drivers is on the Web at:

http://www.printers.ibm.com/internet/wwwsites.nsf/vwwebpublished/supportdownloads_ww

The Windows XP and Windows 2000 drivers for the IBM® NP 24 were downloaded, the .exe files were unzipped, and **Extract to disk** was chosen (versus Install, which would install the drivers directly onto the client). The location of the extracted drivers was noted.

From a Windows desktop, go to any Windows Explorer window and type: `\\<yourServer>` in the address bar. You should see the resources that your Samba server has made available. An example screen is shown in Figure 4 on page 19.

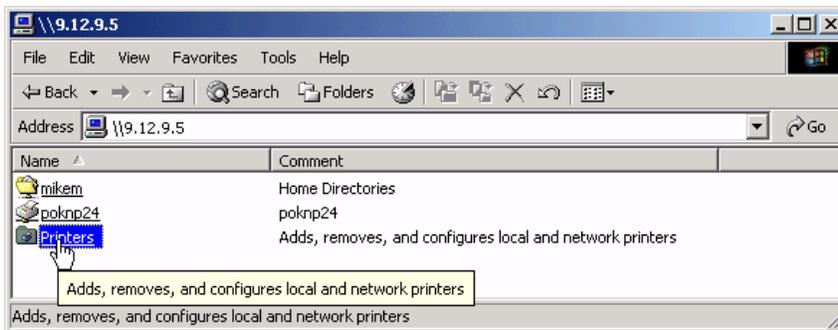


Figure 4 Samba server resources from Windows Explorer

Note that the printer (poknp24) is shown, as well as the folder named Printers. To do administration with printers, you must go into the Printers folder. If you do not see this folder, it is possible that the user you are connected as is not in the `smb.conf` "printer admin" list.

Right-click the printer for which you want to upload the drivers. You should see the prompt shown in Figure 5. If you do not see this prompt, it is possible that the Windows desktop from which you are connected already has the correct printer drivers installed. If this is the case, either delete them or start from a different machine that does not have the drivers installed.

Choose **No**, because you do not want to install the printer drivers on the Windows desktop; instead, you want to upload them to the Samba server.

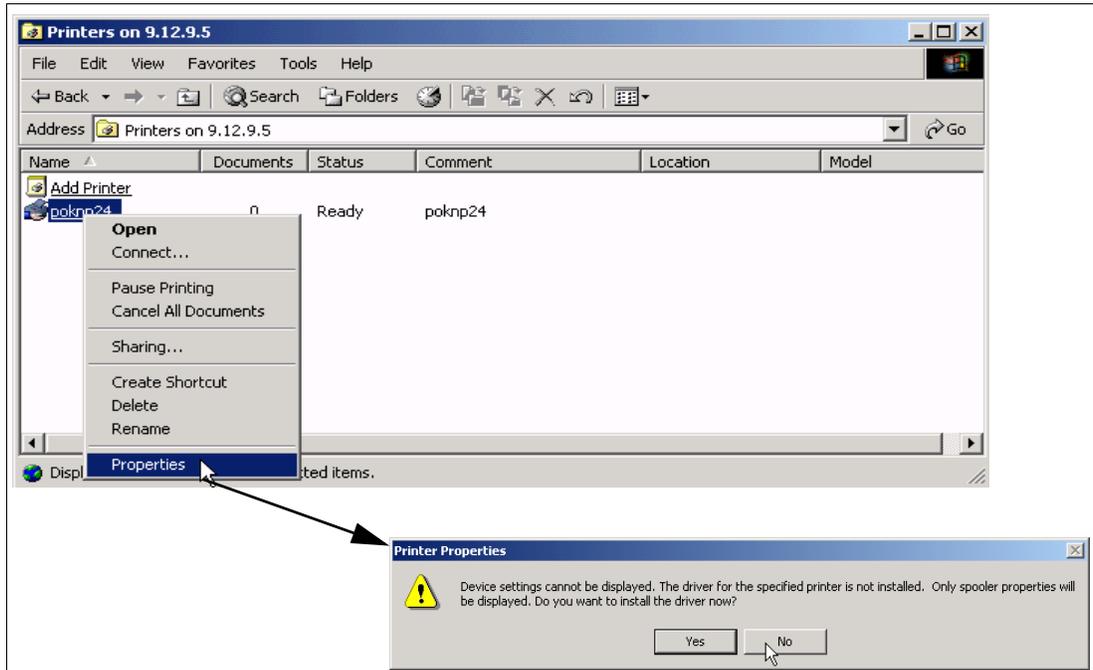


Figure 5 Starting upload of Windows printer drivers

Click the **Advanced** tab and then click **New Driver**. This should invoke the Add Printer Driver Wizard, as shown in Figure 6 on page 20.

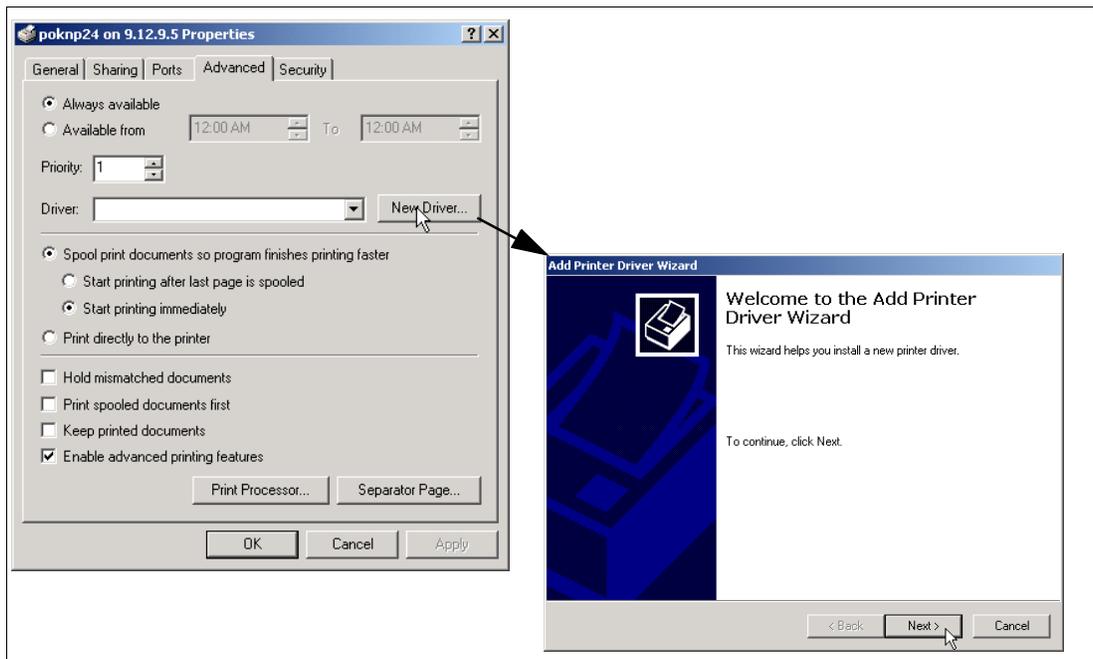


Figure 6 Invoking the Add Printer Driver Wizard

Click **Next**, and you will be presented with the panel shown in Figure 7. If you have obtained the printer drivers as discussed earlier, click **Have Disk** to bring up a file chooser to get to those drivers.

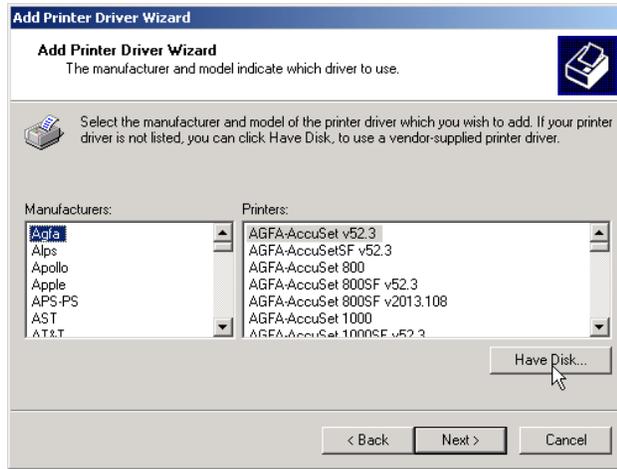


Figure 7 Bringing up a file chooser to point to drivers

When the file chooser comes up, click **Browse** to traverse to the directory where the drivers are and click **OK**. If you have the right drivers, you should see a list of printers, or perhaps only your printer. Select it and click **Next**. Doing so should bring up the Completing the Add Printer Driver Wizard window, as shown in Figure 7 on page 21. Click **Finish**.

You may get a warning panel with the message: Digital Signature Not Found. If this happens, click **Yes** to continue. This should bring up a Copying Files status window which shows the drivers being uploaded to your Samba server. If you get an error here, check the Samba log files (usually found in /var/log/samba/) for possible reasons.

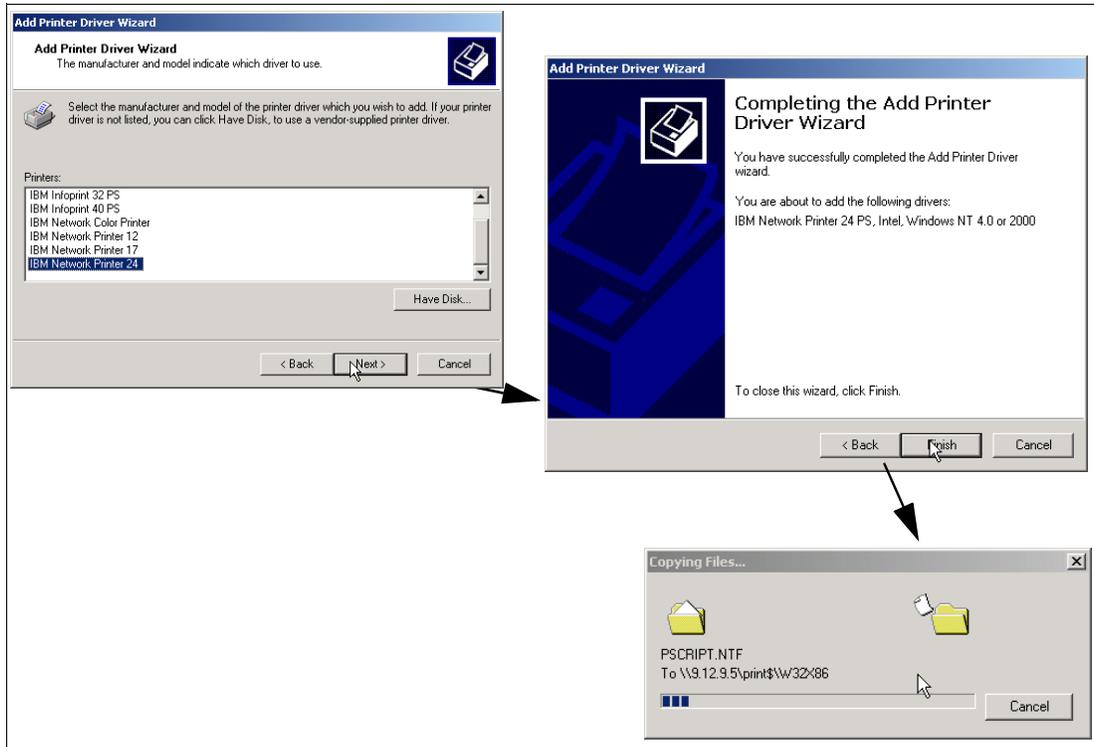


Figure 8 Beginning the Windows printer driver upload

Adding drivers for other versions of Windows

If you need to support multiple versions of Windows, you may have to upload different drivers. To do so, click the **Sharing** tab and then **Additional Drivers**. This will bring up a window offering a choice of various architectures and Windows versions; see Figure 9. A suggestion is to do this from a Windows XP desktop, because it will have the largest number of other Windows versions.

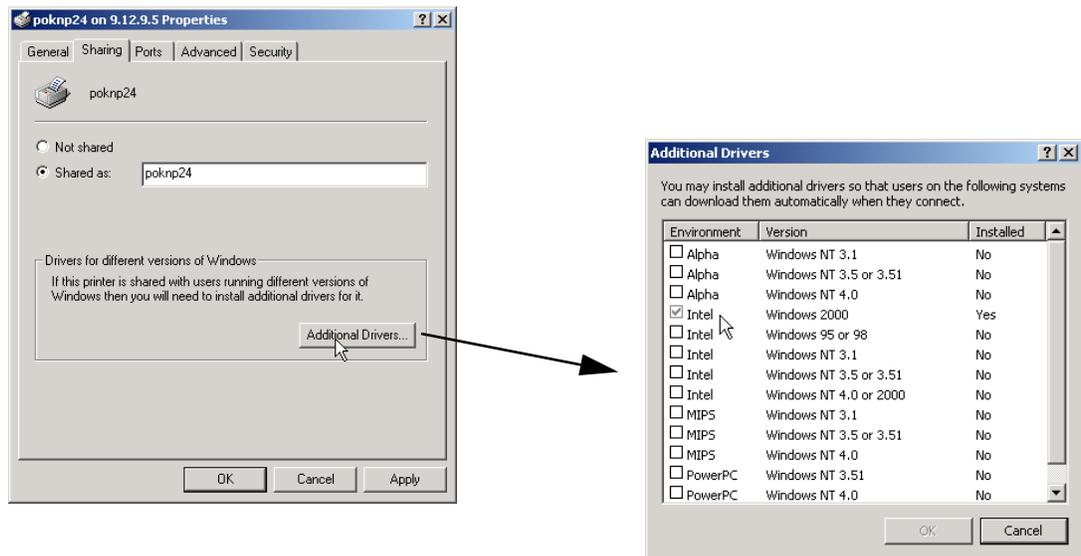


Figure 9 Uploading drivers for other versions of Windows

Automatically downloading printers drivers

If the printer drivers were uploaded properly, they should download automatically. When another user tries to use the Samba/CUPS printer, a pop-up as shown in Figure 10 should appear. Click **Yes**; the drivers should be automatically downloaded in just a few seconds.

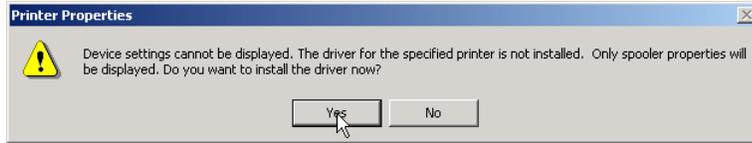


Figure 10 A good message to see

Test printing from Windows desktop

There are a number of ways to add printers from Windows desktops.

You should now be ready to test printing from Windows through Samba and CUPS to the printer. First add the printer to the Windows desktop. From the Windows Explorer you opened, click **Printers** and then click **Add Printer**. This should bring up the Add Printer Wizard as shown in Figure 11.

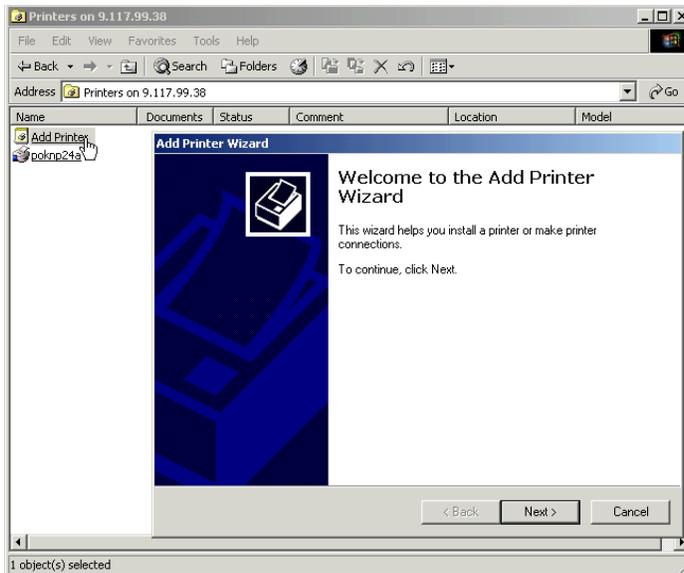


Figure 11 Adding a Samba/CUPS printer from a Windows desktop

Click **Next** and you should see that the remote print server is the only option. Click **Next** again and you should be able to add the new printer; see Figure 12 on page 24.

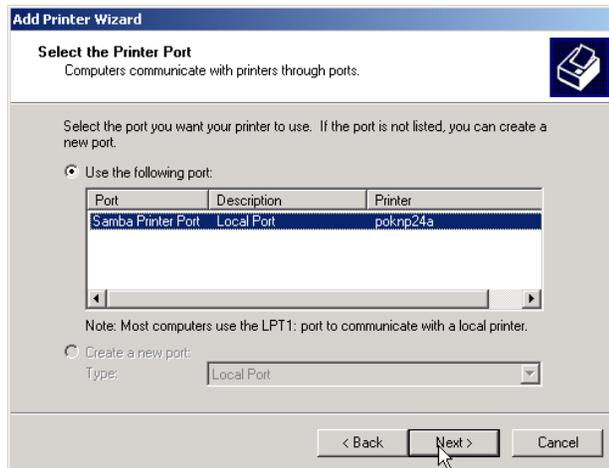


Figure 12 Selecting the Samba/CUPS printer

Set up a PDF drop box

Ghostscript can be used in conjunction with CUPS and Samba so users from Windows desktops can easily generate PDFs. This is done via defining a “PDF writer” printer to CUPS that “prints” PDFs to a directory exported by Samba. The Windows users send files to this printer, and the PDFs arrive in the directory after processing.

To set this service up, you will need ghostscript (the `gs` command, standard with SLES-8 and RHEL-3), which is the `pdfdistiller` script available from the KDE printing site and an Acrobat distiller PPD file.

Get the PDF distiller script from the KDE printing site and download it to an FTP server:

```
ftp://ftp.kde.org/pub/kde/printing/pdfdistiller
```

Copy it to the directory `/usr/lib/cups/backends/`, name it `pdf`, make it executable, and test that it runs:

```
# cd /usr/lib/cups/backends
# ftp ftp.kde.org
Name (ftp.kde.org:root): anonymous
Password: <anything>
ftp> cd pub/kde/printing
ftp> get pdfdistiller pdf
ftp> quit
# chmod +x pdf
# ./pdf
direct pdf "Unknown" "PDF Writing"
```

Note: Testing on 64-bit SUSE SLES8 indicates the `pdf` script should be copied to and executed from the `/usr/lib64/cups/backend` directory.

The last step invokes the `pdf` script without any parameters. The error message confirms that the script at least runs. Now get an Acrobat distiller PPD file; you can find one at:

<http://www.pentondigitalads.com/downloads/adist4.zip>

In this example, the zip file was downloaded and unzipped on Windows via a Samba share. The resulting PPD file was FTP'd to the `/usr/share/cups/model/` directory and compressed with the `gzip` command:

```
# cd /usr/share/cups/model
# ftp <your.download.server>
ftp> get adist4.ppd
ftp> quit
# gzip adist4.ppd
```

Verify that there is no PDF back end via the `lpinfo -v` and `grep` commands. CUPS can now be restarted. Verify that both the PDF back end and the distiller PPD file were added via the `lpinfo -v` and `-m` flags, respectively:

```
# lpinfo -v | grep pdf
# rccups restart
Shutting down cupsd                               done
Starting cupsd                                   done
# lpinfo -v | grep pdf
direct pdf
# lpinfo -m | grep Adist
Adist4.ppd.gz Acrobat Distiller 3010.106
```

Add a Samba share for the directory where the PDF files will be written to. In the following example, the share name is `PDFfiles` and the PDFs will be written to the directory `/var/PDFfiles/`. Then Samba is restarted:

```
# cd /etc/samba
# vi smb.conf -> add the following section
[PDFfiles]
    path = /var/PDFfiles/
    read only = no
# rcsmb restart
Shutting down Samba classic SMB daemon           done
Samba SMB: Waiting for cupsd to get ready (29s)  done
Starting Samba classic SMB daemon               done
```

The PDFwrite printer is added via the `lpadmin` command. The URI is:

```
# lpadmin -p PDFwrite -E -v pdf:/var/PDFfiles/ -m Adist4.ppd.gz
```

Everything should now be ready for Windows users to distill files into PDFs by sending the file to the PDFwrite printer.

Managing printer administration

When your printers are in a network pool, you may want to designate specific people to have various administrative control over the printers. In our installation, we have a computer operations group that manages the daily running of the printers. We needed to give them the ability do some basic printer and job functions, such as to start and stop the printers, but not the authority to modify, add or delete the printers.

We created linux user ids (`op1` and `op2`) and a new group (`printops`) for the printer operators. We continued to use the `ntadmin` group for userids that will have full admin authority.

Next, we modified the `cuspd.conf` file to set up this authority:

```
<Location /admin/?op=delete-printer>
AuthType Basic
AuthClass Group
AuthGroupName ntadmin
```

```

Order Deny,Allow
Allow From 127.0.0.1
</Location>
<Location /admin/?op=add-printer>
AuthType Basic
AuthClass Group
AuthGroupName ntadmin
Order Deny,Allow
Allow From 127.0.0.1
</Location>
<Location /admin/?op=config-printer>
AuthType Basic
AuthClass Group
AuthGroupName ntadmin
Order Deny,Allow
Allow From 127.0.0.1
</Location>
<Location /admin/?op=modify-printer>
AuthType Basic
AuthClass Group
AuthGroupName ntadmin
Order Deny,Allow
Allow From 127.0.0.1
</Location>
<Location /admin/?op=delete-class>
AuthType Basic
AuthClass Group
AuthGroupName ntadmin
Order Deny,Allow
Allow From 127.0.0.1
</Location>
<Location /admin/?op=add-class>
AuthType Basic
AuthClass Group
AuthGroupName ntadmin
Order Deny,Allow
Allow From 127.0.0.1
</Location>
<Location /admin/?op=modify-class>
AuthType Basic
AuthClass Group
AuthGroupName ntadmin
Order Deny,Allow
Allow From 127.0.0.1
</Location>

```

Since we wanted **printops** to have a majority of the functions and views, we authorized that group to all admin and then overrode specific admin functions to allow only **ntadmin**.

Banner customization

You may want to customize the default banner provided with CUPS. Example 1 illustrates a customized banner definition file named `itso_mod`.

Example 1 The itso CUPS banner

```

%!PS-Adobe-3.0
%%BoundingBox: 0 0 612 792
%%Pages: 1
%%LanguageLevel: 1

```

```

%%DocumentData: Clean7Bit
%%DocumentSuppliedResources: procset bannerprint/1.0
%%DocumentNeededResources: font Helvetica Helvetica-Bold Times-Roman
%%Creator: Michael Sweet, Easy Software Products
%%CreationDate: May 10, 2000
%%ModifiedBy: Roy Costa IBM International Technical Support Organization
%%ModificationDate: March 01, 2004
%%Title: ITS0 Custom cover page
%%EndComments
%%BeginProlog
%%BeginResource procset bannerprint 1.1 0
%
% PostScript banner page for the Common UNIX Printing System ("CUPS").
%
% Copyright 1993-2002 Easy Software Products
%
% These coded instructions, statements, and computer programs are the
% property of Easy Software Products and are protected by Federal
% copyright law. Distribution and use rights are outlined in the file
% "LICENSE.txt" which should have been included with this file. If this
% file is missing or damaged please contact Easy Software Products
% at:
%
% Attn: CUPS Licensing Information
% Easy Software Products
% 44141 Airport View Drive, Suite 204
% Hollywood, Maryland 20636-3111 USA
%
% Voice: (301) 373-9603
% EMail: cups-info@cups.org
% WWW: http://www.cups.org
%
/CUPSLOGO {% Draw the CUPS logo
            % height CUPSLOGO
% Start with a big C...
/Helvetica findfont 1 index scalefont setfont
0 setgray
0 0 moveto
(C) show

% Then "UNIX Printing System" much smaller...
/Helvetica-Bold findfont 1 index 9 div scalefont setfont
0.25 mul
dup dup 2.0 mul moveto
(UNIX) show
dup dup 1.6 mul moveto
(Printing) show
dup 1.2 mul moveto
(System) show
} bind def
/ESPLOGO {% Draw the ESP logo
           % height ESPLOGO
% Compute the size of the logo...
0 0
2 index 1.5 mul 3 index

% Do the "metallic" fill from 10% black to 40% black...
1 -0.001 0 {
  dup % loopval
  -0.15 mul% loopval * -0.15

```

```

0.9 add% 0.9 - loopval * 0.15
setgray% set gray shade

0 % x
1 index neg% loopval
1 add% 1 - loopval
3 index% height
mul % height * (1 - loopval)
moveto% starting point

dup % loopval
3 index% width
mul % loopval * width
2 index% height
lineto% Next point

0 % x
2 index% height
lineto% Next point

closepath
fill

dup % loopval
0.15 mul% loopval * 0.15
0.6 add% 0.6 + loopval * 0.15
setgray

dup % loopval
neg 1 add% 1 - loopval
3 index% width
mul % (1 - loopval) * width
0 % y
moveto% Starting point

2 index% width
exch% loopval
2 index% height
mul % loopval * height
lineto% Next point

1 index% width
0 % y
lineto% Next point

closepath
fill
} for

0 setgray rectstroke

/Helvetica-BoldOblique findfont 1 index 3 div scalefont setfont
dup 40 div

dup 4 mul 1 index 25 mul moveto (E) show
dup 10 mul 1 index 15 mul moveto (S) show
dup 16 mul 1 index 5 mul moveto (P) show

/Helvetica-BoldOblique findfont 2 index 5 div scalefont setfont
dup 14 mul 1 index 29 mul moveto (asy) show

```

```

dup 20 mul 1 index 19 mul moveto (oftware) show
dup 26 mul 1 index 9 mul moveto (roducts) show

pop
} bind def
%%EndResource
%%EndProlog
%%Page: 1 1
gsave

% Determine the imageable area and device resolution...
initclip newpath clippath pathbbox% Get bounding rectangle
72 div /pageTop exch def% Get top margin in inches
72 div /pageRight exch def% Get right margin in inches
72 div /pageBottom exch def% Get bottom margin in inches
72 div /pageLeft exch def% Get left margin in inches

/pageWidth pageRight pageLeft sub def% pageWidth = pageRight - pageLeft
/pageHeight pageTop pageBottom sub def% pageHeight = pageTop - pageBottom

/boxWidth % width of text box
pageWidth pageHeight lt
{ pageWidth 54 mul }
{ pageHeight 42 mul }
ifelse def

newpath % Clear bounding path

% Create fonts...

/largeFont /Helvetica findfont % largeFont = Helvetica
pageHeight 7 mul scalefont def % size = pageHeight * 3 (nominally 33)

/mediumFont /Helvetica findfont% mediumFont = Helvetica
pageHeight 1.5 mul scalefont def% size = pageHeight * 1.5 (nominally 16.5)

/bigFont /Helvetica findfont % bigFont = Helvetica
pageHeight 3 mul scalefont def % size = pageHeight * 3 (nominally 33)

% Offset page to account for lower-left margin...
pageLeft 72 mul
pageBottom 72 mul
translate

% Job information box...

largeFont setfont

pageWidth 4 mul % x = pageWidth * 1/2 * 72
pageHeight 58 mul % y = pageHeight * 1/2 * 72
pageHeight 5 mul add % y -= 1 line
2 copy % Copy X & Y
moveto
(({job-originating-user-name}) show 1

bigFont setfont % Big sized font

pageWidth 4 mul % x = pageWidth * 1/2 * 72

```

```

pageHeight 58 mul
pageHeight -5 mul add
2 copy
moveto
({job-name}) show
2

pageWidth 4 mul
pageHeight 58 mul
pageHeight -15 mul add
2 copy
moveto
({printer-name}-{job-id}) show
3

pageWidth 4 mul
pageHeight 58 mul
pageHeight -25 mul add
2 copy
moveto
({time-at-creation}) show
4

% Then the CUPS logo....
gsave
  pageWidth 4 mul
  pageWidth 6 mul
  translate
  pageWidth 9 mul CUPSLOGO
grestore

% And the ESP logo....
gsave
  pageWidth 59 mul
  pageWidth 6 mul
  translate
  pageWidth 6 mul ESPLOGO
grestore
% Show the page...
grestore
showpage
%
% End of "$Id: standard,v 1.8 2002/01/31 14:19:37 mike Exp $".
%
%%EOF

```

The customizations create a banner containing the following information:

- ▶ The user name
- ▶ The job name
- ▶ The printer name
- ▶ The time of creation

The resultant banner is shown in Figure 13 on page 31.

costa01

Linux Printing

poknp24c-486

[14/Apr/2004:11:19:29 -0400]



Figure 13 The itso banner page

Without knowing the details of Post Script and your printer's PPD file, it is difficult to modify other options within the banner for your specific type of printer.

Here is a way you can do further banner customization, such as changing to the colored paper tray to print your banner.

Define a printer that prints to a file, using a special file:/ back end. You must first enable this in the `/etc/cups/cupsd.conf` file. Add this line *before* any of the `<Location/>` statements:

```
FileDevice Yes
```

Create a filewriter printer which will route output to a file in the `/tmp` directory, using the attributes of the appropriate printer .ppd file.

Enter the line command:

```
lpadmin -p filewriter -v file:/tmp/my_banner_4_poknp24 -E -P /etc/cups/ppd/poknp24a.ppd
```

From the Web interface, configure the filewriter printer with the option you want.

To change the paper for the banner, modify Media Source to the desired tray and click **Continue** to save.

Print your customized banner with the filewriter printer:

```
lpr -P filewriter /usr/share/cups/banners/itso_mod
```

Your original customized banner and customized PPD options are now incorporated into a new banner file, my_banner_4_poknp24.

Copy that new banner to /usr/share/cups/banners as your new banner:

```
cp /tmp/my_banner_4_poknp24 /usr/share/cups/banners/itso
```

Now modify the banner selection for the printers (via the Web browser) or the printer class (via the **lpadmin** command), designating itso as your new banner. If you are using printer classes, make sure that the new banner is properly identified in the /etc/cups/classes.conf file.

If you wish to have different banners for different printers or classes, you could follow the same process, naming the banners differently and making sure that you modify the printer or class banner selections accordingly.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.



Send us your comments in one of the following ways:

- ▶ Use the online **Contact us** review redbook form found at:
ibm.com/redbooks
- ▶ Send your comments in an Internet note to:
redbook@us.ibm.com
- ▶ Mail your comments to:
IBM Corporation, International Technical Support Organization
Dept. HYJ Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400 U.S.A.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

@server®
@server®
ibm.com®

IBM®
Redbooks (logo) ™
Redbooks™

z/VM®
zSeries®

The following terms are trademarks of other companies:

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Windows and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.