7

Printing and Name Resolution

This chapter tackles two Samba topics: setting up printers for use with a Samba server and configuring Samba to use or become a Windows Internet Name Service (WINS) server. Samba allows client machines to send documents to printers connected to the Samba server. In addition, Samba can also assist you with printing Unix documents to a printer on a Windows machine. In the first part of this chapter, we will discuss how to get printers configured to work on either side.

In the second half of the chapter, we will introduce the Windows Internet Name Service, Microsoft's implementation of a NetBIOS Name Server (NBNS). As mentioned in Chapter 1, *Learning the Samba*, an NBNS allows machines to perform name resolution on a NetBIOS network without having to rely on broadcasts. Instead, each machine knows exactly where the WINS server is and can query it for the IP addresses of other machines on the network.

Sending Print Jobs to Samba

A printer attached to the Samba server shows up in the list of shares offered in the Network Neighborhood. If the printer is registered on the client machine and the client has the correct printer driver installed, the client can effortlessly send print jobs to a printer attached to a Samba server. Figure 7-1 shows a Samba printer as it appears in the Network Neighborhood of a Windows client.

To administer printers with Samba, you should understand the basic process by which printing takes place on a network. Sending a print job to a printer on a Samba server involves four steps:

1. Opening and authenticating a connection to the printer share
2. Copying the file over the network

3. Closing the connection
4. Printing and deleting the copy of the file

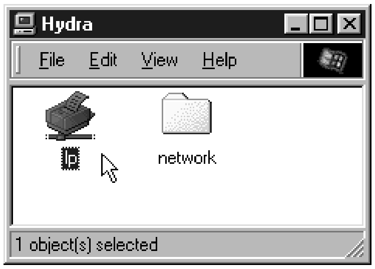


Figure 7-1. A Samba printer in the Network Neighborhood

When a print job arrives at a Samba server, the print data is temporarily written to disk in the directory specified by the `path` option of the printer share. Samba then executes a Unix print command to send that data file to the printer. The job is printed as the authenticated user of the share. Note that this may be the guest user, depending on how the share is configured.

Print Commands

In order to print the document, you'll need to tell Samba what the command is to print and delete a file. On Linux, such a command is:

```
lpr -r -Pprinter file
```

This tells `lpr` to copy the document to a spool area, usually */var/spool*, retrieve the name of the printer in the system configuration file (*/etc/printcap*), and interpret the rules it finds there to decide how to process the data and which physical device to send it to. Note that because the `-r` option has been listed, the file specified on the command line will be deleted after it has been printed. Of course, the file removed is just a copy stored on the Samba server; the original file on the client is unaffected.

Linux uses a Berkeley (BSD) style of printing. However, the process is similar on System V Unix. Here, printing and deleting becomes a compound command:

```
lp -dprinter -s file; rm file
```

With System V, the */etc/printcap* file is replaced with different set of configuration files hiding in */usr/spool/lp*, and there is no option to delete the file. You have to do it yourself, which is why we have added the `rm` command afterward.

Printing Variables

Samba provides four variables specifically for use with printing configuration options. They are shown in Table 7-1.

Table 7-1. Printing Variables

Variable	Definition
`%s`	The full pathname of the file on the Samba server to be printed
`%f`	The name of the file itself (without the preceding path) on the Samba server to be printed
`%p`	The name of the Unix printer to use
`%j`	The number of the print job (for use with `lprm`, `lppause`, and `lpresume`)

A Minimal Printing Setup

Let's start with a simple but illustrative printing share. Assuming that you're on a Linux system and you have a printer called `lp` listed in the printer capabilities file, the following addition to your *smb.conf* file will make the printer accessible through the network:

```
[printer1]
    printable = yes
    print command = /usr/bin/lpr -r  %s
    printer = lp
    printing = BSD
    read only = yes
    guest ok = yes
```

This configuration allows anyone to send data to the printer, something we may want to change later. For the moment, what's important to understand is that the variable `%s` in the `print command` option will be replaced with the name of the file to be printed when Samba executes the command. Changing the `print command` to reflect a different style of Unix machine typically involves only replacing the right side of the `print command` option with whatever command you need for your system and changing the target of the `printing` option.

Let's look at the commands for a System V Unix. With variable substitution, the System V Unix command becomes:

```
print command = lp -d%p -s %s; rm %s
```

As mentioned earlier, the `%p` variable resolves to the name of the printer, while the `%s` variable resolves to the name of the file. After that, you can change the `printing` option to reflect that you're using a System V architecture:

```
printing = SYSV
```

If you are using share-level security, pay special attention to the guest account used by Samba. The typical setting, nobody, may not be allowed to print by the operating system. If that's true for your operating system, you should place a guest account option under the printing share (or even perhaps the global share) specifying an account that can. A popular candidate with the Samba authors is the ftp account, which is often preconfigured to be safe for untrusted guest users. You can set it with the following command:

```
guest account = ftp
```

Another common printing issue is that clients may need to request the status of a print job sent to the Samba server. Samba will not reject a document from being sent to an already busy printer share. Consequently, Samba needs the ability to communicate not only the status of the current printing job to the client, but also which documents are currently waiting to be printed on that printer. Samba also has to provide the client the ability to pause print jobs, resume print jobs, and remove print jobs from the printing queue. Samba provides options for each of these tasks. As you might expect, they borrow functionality from existing Unix commands. The options are:

- lpq command
- lprm command
- lppause command
- lpresume command

We will cover these options in more detail below. For the most part, however, the value of the printing configuration option will determine their values, and you should not need to alter the default values of these options.

Here are a few important items to remember about printing shares:

- You must put printable = yes in all printer shares (even [printers]), so that Samba will know that they are printer shares. If you forget, the shares will not be usable for printing and will instead be treated as disk shares.
- If you set the path configuration option in the printer section, any files sent to the printer(s) will be copied to the directory you specify instead of to the default location of */tmp*. As the amount of disk space allocated to */tmp* can be relatively small in some Unix operating systems, many administrators opt to use */var/spool* or some other directory instead.
- The read only option is ignored for printer shares.
- If you set guest ok = yes in a printer share and Samba is configured for share-level security, it will allow anyone to send data to the printer as the guest account user.

Using one or more Samba machines as a print server gives you a great deal of flexibility on your LAN. You can easily partition your available printers, restricting some to members of one department, or you can maintain a bank of printers available to all. In addition, you can restrict a printer to a selected few by adding the trusty `valid users` option to its share definition:

```
[deskjet]
    printable = yes
    path = /var/spool/samba/print
    valid users = gail sam
```

All of the other share accessibility options defined in the previous chapter should work for printing shares as well. Since the printers themselves are accessed through Samba by name, it's also simple to delegate print services among several servers using familiar Unix commands for tasks such as load balancing or maintenance.

The [printers] Share

Chapter 4, *Disk Shares*, briefly introduced `[printers]`, a special share for automatically creating printing services. Let's review how it works: if you create a share named `[printers]` in the configuration file, Samba will automatically read in your printer capabilities file and create a printing share for each printer that appears in the file. For example, if the Samba server had `lp`, `pcl` and `ps` printers in its printer capabilities file, Samba would provide three printer shares with those names, each configured with the options in the `[printers]` share.

Recall that Samba obeys following rules when a client requests a share that has not been created through the *smb.conf* file:

- If the share name matches a username in the system password file and a `[homes]` share exists, a new share is created with the name of the user and is initialized using the values given in the `[homes]` and `[global]` sections.
- Otherwise, if the name matches a printer in the system printer capabilities file, and a `[printers]` share exists, a new share is created with the name of the printer and initialized using the values given in the `[printers]` section. (Variables in the `[global]` section do not apply here.)
- If neither of those succeed, Samba looks for a `default service` share. If none is found, it returns an error.

This brings to light an important point: be careful that you do not give a printer the same name as a user. Otherwise, you will end up connecting to a disk share when you may have wanted a printer share instead.

Here is an example [printers] share for a Linux (BSD) system. Some of these options are already defaults; however, we have listed them anyway for illustrative purposes:

```
[global]
    printing = BSD
    print command = /usr/bin/lpr -P%p -r %s
    printcap file = /etc/printcap
    min print space = 2000

[printers]
    path = /usr/spool/public
    printable = true
    guest ok = true
    guest account = pcguest
```

Here, we've given Samba global options that specify the printing type (BSD), a print command to send data to the printer and remove a temporary file, our default printer capabilities file, and a minimum printing space of 2 megabytes.

In addition, we've created a [printers] share for each of the system printers. Our temporary spooling directory is specified by the path option: */usr/spool/public*. Each of the shares is marked as printable—this is necessary, even in the [printers] section. The two guest options are useful in the event that Samba is using share-level security: we allow guest access to the printer and we specify the guest user that Samba should use to execute print commands.

Test Printing

Here is how you can test printing from the Samba server. Let's assume the most complex case and use a guest account. First, run the Samba *testparm* command on your configuration file that contains the print shares, as we did in Chapter 2, *Installing Samba on a Unix System*. This will tell you if there are any syntactical problems with the configuration file. For example, here is what you would see if you left out the path configuration option in the previous example:

```
# testparm
Load smb config files from /usr/local/samba/lib/smb.conf
Processing configuration file "/usr/local/samba/lib/smb.conf"
Processing section "[global]"
Processing section "[homes]"
Processing section "[data]"
Processing section "[printers]"
No path in service printers - using /tmp
Loaded services file OK.
Press enter to see a dump of your service definitions
Global parameters:
    load printers: Yes
    printcap name: /etc/printcap
Default service parameters:
    guest account: ftp
    min print space: 0
    print command: lpr -r -P%p %s
    lpq command: lpq -P%p
    lprm command: lprm -P%p %j
lppause command:
    lpresume command:
 Service parameters [printers]:
    path: /tmp
    print ok: Yes
    read only: true
    public: true
```

Second, try the command `testprns` *`printername`*. This is a simple program that verifies that the specified printer is available in your *printcap* file. If your *printcap* file is not in the usual place, you can specify its full pathname as the second argument to the *testprns* command:

```
# testprns lp /etc/printcap
Looking for printer lp in printcap file /etc/printcap
Printer name lp is valid.
```

Next, log on as the guest user, go to the spooling directory, and ensure that you can print using the same command that *testparm* says Samba will use. As mentioned before, this will tell you if you need to change the guest account, as the default account may not be allowed to print.

Finally, print something to the Samba server via `smbclient`, and see if the following actions occur:

- The job appears (briefly) in the Samba spool directory specified by the path.
- The job shows up in your print systems spool directory.
- The job disappears from the spool directory that Samba used.

If *smbclient* cannot print, you can reset the `print command` option to collect debugging information:

```
print command = /bin/cat %s >>/tmp/printlog; rm %s
```

or:

```
print command = echo "printed %s on %p" >>/tmp/printlog
```

A common problem with Samba printer configuration is forgetting to use the full pathnames for commands; simple commands often don't work because the guest account's PATH doesn't include them. Another frequent problem is not having the correct permissions on the spooling directory.

There is more information on debugging printers in the Samba documentation (*Printing.txt*). In addition, the Unix print systems are covered in detail in AEleen Frisch's *Essential Systems Administration* (published by O'Reilly).

Setting Up and Testing a Windows Client

Now that Samba is offering a workable printer, you need to set it up on a Windows client. Look at the Samba server in the Network Neighborhood. It should now show each of the printers that are available. For example, in Figure 7-1, we saw a printer called `lp`.

Next, you need to have the Windows client recognize the printer. Double-click on the printer icon to get started. If you try to select an uninstalled printer (as you just did), Windows will ask you if it should help configure it for the Windows system. Respond "Yes," which will open the Printer Wizard.

The first thing the wizard will ask is whether you need to print from DOS. Let's assume you don't, so choose No and press the Next button to get to the manufacturer/model window as shown in Figure 7-2.

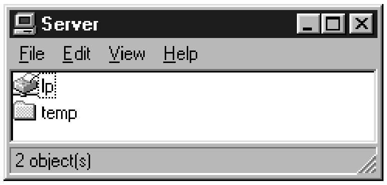
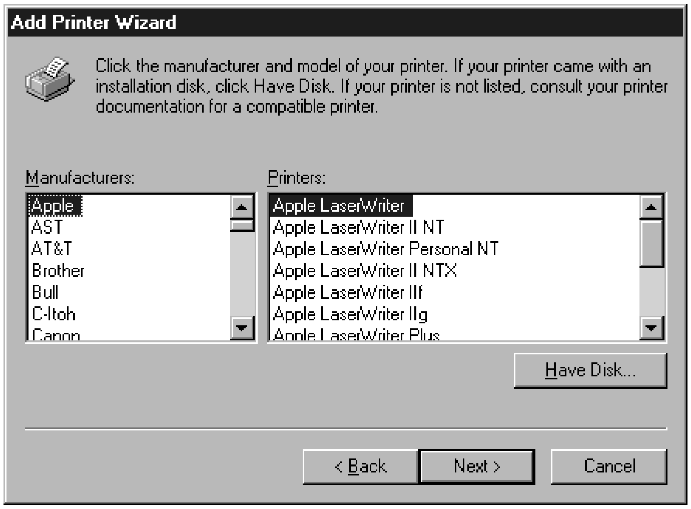


Figure 7-2. A printer in the Network Neighborhood

In this dialog box, you should see a large list of manufacturers and models for almost every printer imaginable. If you don't see your printer on the list, but you know it's a PostScript printer, select Apple as the manufacturer and Apple LaserWriter as the model. This will give you the most basic Postscript printer setup, and arguably one of the most reliable. If you already have any Postscript printers attached, you will be asked about replacing or reusing the existing driver. Be aware that if you replace it with a new one, you may make your other printers fail. Therefore, we recommend you keep using your existing printer drivers as long as they're working properly.

Following that, the Printer Wizard will ask you to name the printer. Figure 7-3 shows this example, where the name has defaulted to our second laserwriter.

Here, you rename it from Apple Laserwriter (Copy 2) to “ps on Samba server,” so you know where to look for the printouts. In reality, you can name the printer anything you want.


Add Printer Wizard

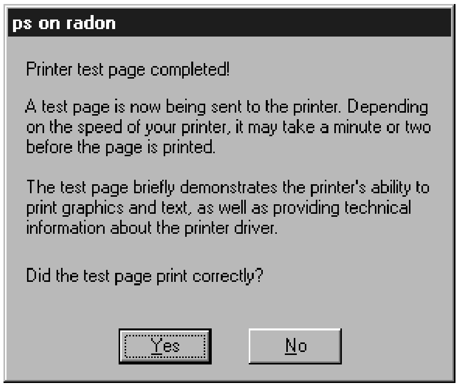
Click the manufacturer and model of your printer. If your printer came with an installation disk, click Have Disk. If your printer is not listed, consult your printer documentation for a compatible printer.

Manufacturers:
- Apple
- AST
- AT&T
- Brother
- Bull
- C-Itoh
- Canon

Printers:
- Apple LaserWriter
- Apple LaserWriter II NT
- Apple LaserWriter Personal NT
- Apple LaserWriter II NTX
- Apple LaserWriter IIf
- Apple LaserWriter IIg
- Apple LaserWriter Plus

Figure 7-3. Printer manufacturers and models

Finally, the Printing Wizard asks if it should print a test page. Click on Yes, and you should be presented with the dialog in Figure 7-4.


ps on radon

Printer test page completed!

A test page is now being sent to the printer. Depending on the speed of your printer, it may take a minute or two before the page is printed.

The test page briefly demonstrates the printer's ability to print graphics and text, as well as providing technical information about the printer driver.

Did the test page print correctly?

Figure 7-4. Printing successfully completed

If the test printing was unsuccessful, press the No button in Figure 7-4 and the Printing Wizard will walk you through some debugging steps for the client side of the process. If the test printing does work, congratulations! The remote printer will now be available to all your PC applications through the File and Print menu items.

Automatically Setting Up Printer Drivers

The previous section described how to manually configure a printer driver for your Windows system. As a system administrator, however, you can't always guarantee that users can perform such a process without making mistakes. Luckily, however, you can ask Samba to automatically set up the printer drivers for a specific printer.

Samba has three options that can be used to automatically set up printer drivers for clients who are connecting for the first time. These options are `printer driver`, `printer driver file`, and `printer driver location`. This section explains how to use these options to allow users to skip over the Manufacturer dialog in the Add Printer Wizard above.

For more information on how to do this, see the *PRINTER_DRIVER.TXT* file in the Samba distribution documentation.

There are four major steps:

1. Install the drivers for the printer on a Windows client (the printer need not be attached).
2. Create a printer definition file from the information on a Windows machine.
3. Create a `PRINTER$` share where the resulting driver files can be placed.
4. Modify the Samba configuration file accordingly.

Let's go over each of the four steps in greater detail.

Install the drivers on a windows client

Use Windows 95/98 for this step. It doesn't matter which client you choose, as long as it has the ability to load the appropriate drivers for the printer. In fact, you don't even need to have the printer attached to the machine. All you're interested in here is getting the appropriate driver files into the Windows directory. First, go to the Printers window of My Computer and double-click on the Add Printer icon, as shown in Figure 7-5.

At this point, you can follow the Add Printer Wizard dialogs through to select the manufacturer and model of the printer in question. If it asks you if you want to

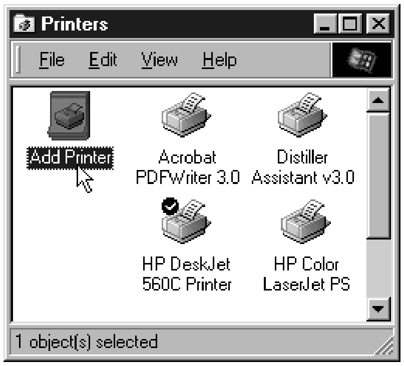


Figure 7-5. The Printers window

print from MS-DOS, answer No. Windows should load the appropriate driver resources from its CD-ROM and ask you if you want to print a test page. Again, respond No and close the Add Printer Wizard dialog.

Create a printer definition file

You can create a printer definition file by using the *make_printerdef* script in the */usr/local/samba/bin* directory. In order to use this script, you need to copy over the following four files from a Windows client:*

C:\WINDOWS\INF\MSPRINT.INF
C:\WINDOWS\INF\MSPRINT2.INF
C:\WINDOWS\INF\MSPRINT3.INF
C:\WINDOWS\INF\MSPRINT4.INF

Once you have the four files, you can create a printer definition file using the appropriate printer driver and its .INF file. If the printer driver starts with the letters A–K, use either the *MSPRINT.INF* file or the *MSPRINT3.INF* file. If it begins with the letters L-Z, use the *MSPRINT2.INF* file or the *MSPRINT4.INF* file. You may need to *grep* through each of the files to see where your specific driver is. For the following example, we have located our driver in *MSPRINT3.INF* and created a printer definition file for a HP DeskJet 560C printer:

```
$grep "HP DeskJet 560C Printer" MSPRINT.INF MSPRINT3.INF
MSPRINT3.INF: "HP DeskJet 560C Printer"=DESKJETC.DRV,HP_DeskJet_ ...

$make_printerdef MSPRINT3.INF "HP DeskJet 560C Printer" >printers.def
FOUND:DESKJETC.DRV
```

* Older Windows 95 clients may have only the first two files.

```
End of section found
CopyFiles: DESKJETC,COLOR_DESKJETC
Datasection: (null)
Datafile: DESKJETC.DRV
Driverfile: DESKJETC.DRV
Helpfile: HPVDJC.HLP
LanguageMonitor: (null)

Copy the following files to your printer$ share location:
DESKJETC.DRV
HPVCM.HPM
HPVIOL.DLL
HPVMON.DLL
HPVRES.DLL
HPCOLOR.DLL
HPVUI.DLL
HPVDJCC.HLP
color\HPDESK.ICM
```

Note the files that the script asks you to copy. You'll need those for the next step.

Create a PRINTER$ share

This part is relatively easy. Create a share called `[PRINTER$]` in your *smb.conf* that points to an empty directory on the Samba server. Once that is done, copy over the files that the *make_printerdef* script requested of you into the location of the `path` configuration option for the `[PRINTER$]` share. For example, you can put the following in your configuration file:

```
[PRINTER$]
    path = /usr/local/samba/print
    read only = yes
    browsable = no
    guest ok = yes
```

The files requested by the *make_printerdef* script are typically located in the *C:\WINDOWS\SYSTEM* directory, although you can use the following commands to find out exactly where they are:

```
cd C:\WINDOWS
dir filename /s
```

In this case, each of the files needs to be copied to the */usr/local/samba/print* directory on the Samba server. In addition, copy the *printers.def* file that you created over to that share as well. Once you've done that, you're almost ready to go.

Modify the Samba configuration file

The last step is to modify the Samba configuration file by adding the following three options:

- `printer driver`

- printer driver file
- printer driver location

The printer driver file is a global option that points to the *printers.def* file; place that option in your [global] section. The other options should be set in the printer share for which you wish to automatically configure the drivers. The value for printer driver should match the string that shows up in the Printer Wizard on the Windows system. The value of the printer driver location is the pathname of the PRINTER$ share you set up, not the Unix pathname on the server. Thus, you could use the following:

```
[global]
    printer driver file = /usr/local/samba/print/printers.def
[hpdeskjet]
    path = /var/spool/samba/printers
    printable = yes

    printer driver = HP DeskJet 560C Printer
    printer driver location = \\%L\PRINTER$
```

Now you're ready to test it out. At this point, remove the Windows printer that you "set up" in the first step from the list of printers in the Printers window of My Computer. If Samba asks you to delete unneeded files, do so. These files will be replaced shortly on the client, as they now exist on the Samba server.

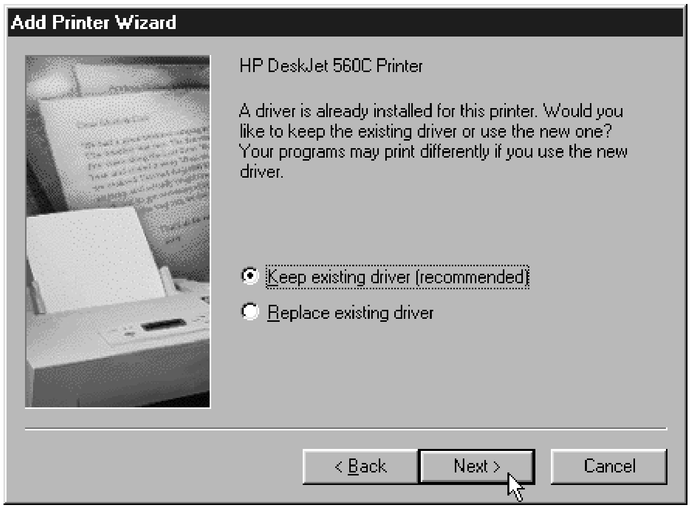
Testing the configuration

Restart the Samba daemons and look for the [hpdeskjet] share under the machine name in the Network Neighborhood. At this point, if you click on the printer icon, you should begin the printer setup process and come to the dialog shown in Figure 7-6.

This is different from the dialog you saw earlier when setting up a printer. Essentially, the dialog is asking if you wish to accept the driver that is "already installed"—in other words, offered by Samba. Go ahead and keep the existing driver, and press the Next button. At this point, you can give the printer a name and print out a test page. If it works, the setup should be complete. You should be able to repeat the process now from any Windows client.

Printing to Windows Client Printers

If you have printers connected to clients running Windows 95/98 or NT 4.0, those printers can also be accessed from Samba. Samba comes equipped with a tool called *smbprint* that can be used to spool print jobs to Windows-based printers. In order to use this, however, you need to set up the printer as a shared resource on



HP DeskJet 560C Printer

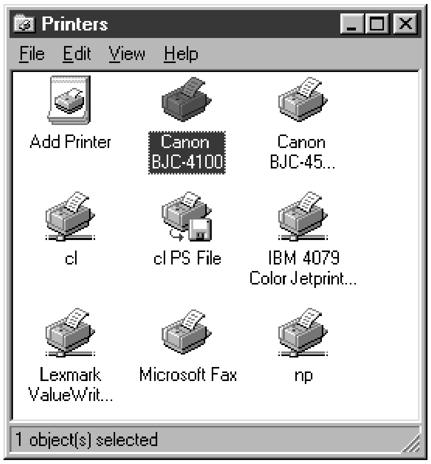
A driver is already installed for this printer. Would you like to keep the existing driver or use the new one? Your programs may print differently if you use the new driver.

● Keep existing driver (recommended)

○ Replace existing driver

Figure 7-6. Automatically configuring the printer driver

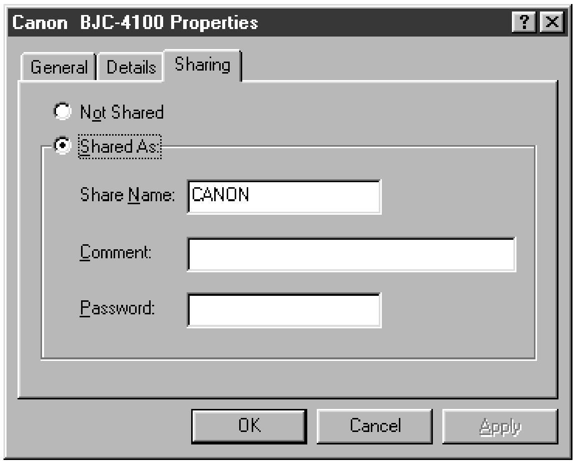
the client machine. If you haven't already done this, you can reset this from the Printers window, reached from the Start button, as shown in Figure 7-7.



1 object(s) selected

Figure 7-7. The Printers window

Select a printer that's locally connected (for example, ours is the Canon printer), press the right mouse button to bring up a menu, and select Sharing. This will give you the Sharing tab of the Printer Properties frame, as shown in Figure 7-8. If you want it available to everybody on your LAN as the Windows guest user, enter a blank password.



○ Not Shared
● Shared As:

Share Name: CANON

Comment:

Password:

Figure 7-8. The Sharing tab of the printer

Once you've got this working, you can add your printer to the list of standard printers and Samba can make it available to all the other PCs in the workgroup. To make installation on Unix easier, the Samba distribution provides two sample scripts: *smbprint* and *smbprint.sysv*. The first works with BSD-style printers; the second is designed for System V printers.

BSD printers

There are two steps you need to have a BSD Unix recognize a remote printer:

1. Place an entry for the printer in the */etc/printcap* file (or equivalent).
2. Place a configuration file in the */var/spool* directory for the printer.

First, edit your */etc/printcap* file and add an entry for the remote printer. Note that the input filter (`if`) entry needs to point to the *smbprint* program if the machine is on Windows 95/98. The following set of lines will accomplish on a Linux machine, for example:

```
laserjet:\
  :sd=/var/spool/lpd/laser:\         # spool directory
  :mx#0:\                            # maximum file size (none)
  :sh:\                              # surpress burst header (no)
  :if=/usr/local/samba/bin/smbprint: # text filter
```

After that, you need to create a configuration file in the spool directory that you specified with the sd parameter above. (You may need to create that directory.) The file must have the name *.config* and should contain the following information:

- The NetBIOS name of the Windows machine with the printer
- The service name that represents the printer
- The password used to access that service

The last two parameters were set up in the Sharing dialog for the requested resource on the Windows machine. In this case, the *.config* file would have three lines:

```
server = phoenix
service = CANON
password = ""
```

After you've done that, reset the Samba server machine and try printing to it using any standard Unix program.

System V printers

Sending print jobs from a System V Unix system is a little easier. Here, you need to get obtain the *smbprint.sysv* script in the */usr/local/samba/examples/printing* directory and do the following:

1. Change the server, service, and password parameters in the script to match the NetBIOS machine, its shared printer service, and its password, respectively. For example, the following entries would be correct for the service in the previous example:

```
server = phoenix
service = CANON
password = ""
```

2. Run the following commands, which create a reference for the printer in the printer capabilities file. Note that the new Unix printer entry canon_printer is named:

```
# lpadmin -p canon_printer -v /dev/null -i./smbprint.sysv
# enable canon_printer
# accept canon_printer
```

After you've done that, restart the Samba daemons and try printing to it using any standard Unix program. You should now be able to send data to a printer on a Windows client across the network.

Samba Printing Options

Table 7-2 summarizes the Samba printing options.

Table 7-2. Printing Configuration Options

Option	Parameters	Function	Default	Scope
printing	bsd, sysv, hpux, aix, qnx, plp, softq, or lprng	Sets the print system type for your Unix system.	System dependent	Share
printable (print ok)	boolean	Marks a share as a printing share.	no	Share
printer (printer name)	string (Unix printer name)	Sets the name of the printer to be shown to clients.	System dependent	Share
printer driver	string (printer driver name)	Sets the driver name that should be used by the client to send data to the printer.	None	Share
printer driver file	string (fully-qualified pathname)	Sets the name of the printer driver file.	None	Global
printer driver location	string (network pathname)	Specifies the pathname of the share for the printer driver file.	None	Share
lpq cache time	numeric (time in seconds)	Sets the amount of time in seconds that Samba will cache the lpq status.	10	Global
postscript	boolean	Treats all print jobs sent as postscript by prepending %! at the beginning of each file.	no	Share
load printers	boolean	Automatically loads each of the printers in the *printcap* file as printing shares.	no	Global
print command	string (shell command)	Sets the Unix command to perform printing.	See below	Share
lpq command	string (shell command)	Sets the Unix command to return the status of the printing queue.	See below	Share
lprm command	string (shell command)	Sets the Unix command to remove a job from the printing queue.	See below	Share
lppause command	string (shell command)	Sets the Unix command to pause a job on the printing queue.	See below	Share

Table 7-2. Printing Configuration Options (continued)

Option	Parameters	Function	Default	Scope
`lpresume command`	string (shell command)	Sets the Unix command to resume a paused job on the printing queue.	See below	Share
`printcap name (printcap)`	string (fully-qualified pathname)	Specifies the location of the printer capabilities file.	System dependent	Global
`min print space`	numeric (size in kilobytes)	Sets the minimum amount of disk free space that must be present to print.	`0`	Share
`queuepause command`	string (shell command)	Sets the Unix command to pause a queue.	See below	Share
`queueresume command`	string (shell command)	Sets the Unix command to resume a queue.	See below	Share

printing

The `printing` configuration option tells Samba a little about your Unix printing system, in this case which printing parser to use. With Unix, there are several different families of commands to control printing and print statusing. Samba supports seven different types, as shown in Table 7-3.

Table 7-3. Printing Types

Variable	Definition
BSD	Berkeley Unix system
SYSV	System V
AIX	AIX Operating System (IBM)
HPUX	Hewlett-Packard Unix
QNX	QNX Realtime Operating System (QNX)
LPRNG	LPR Next Generation (Powell)
SOFTQ	SOFTQ system
PLP	Portable Line Printer (Powell)

The value for this optio.n will be one of these seven options. For example:

```
printing = SYSV
```

The default value of this option is system dependent and is configured when Samba is first compiled. For most systems, the *configure* script will automatically detect the printing system to be used and configure it properly in the Samba makefile. However, if your system is a PLP, LPRNG, or QNX printing system, you will need to explicitly specify this in the makefile or the printing share.

The most common system types are BSD and SYSV. Each of the printers on a BSD Unix server are described in the printer capabilities file—normally */etc/printcap*.

Setting the `printing` configuration option automatically sets at least three other printing options for the service in question: `print command`, `lpq command`, and `lprm command`. If you are running Samba on a system that doesn't support any of these printing styles, simply set the commands for each of these manually.

printable

The printable option must be set to `yes` in order to flag a share as a printing service. If this option is not set, the share will be treated as a disk share instead. You can set the option as follows:

```
[printer1]
    printable = yes
```

printer

The option, sometimes called `printer name`, specifies the name of the printer on the server to which the share points. This option has no default and should be set explicitly in the configuration file, even though Unix systems themselves often recognize a default name such as `lp` for a printer. For example:

```
[deskjet]
    printer = hpdkjet1
```

printer driver

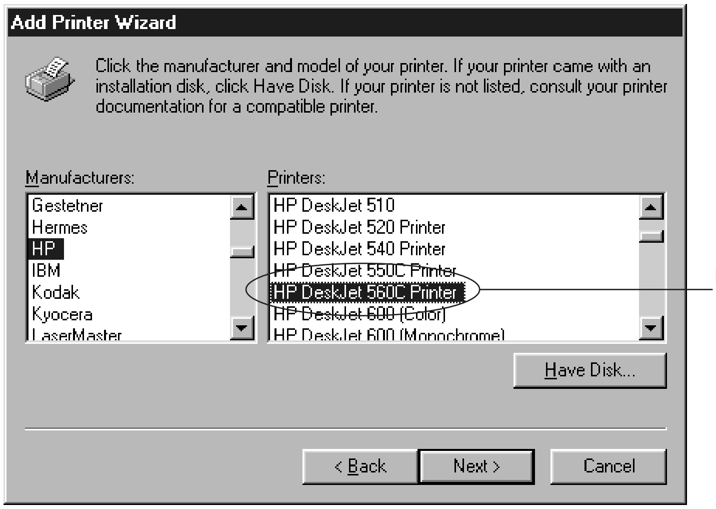
The `printer driver` option sets the string that Samba uses to tell Windows what the printer is. If this option is set correctly, the Windows Printer Wizard will already know what the printer is, making installation easier for end users by giving them one less dialog to worry about. The string given should match the string that shows up in the Printer Wizard, as shown in Figure 7-9. For example, an Apple LaserWriter typically uses `Apple LaserWriter`; a Hewlett Packard Deskjet 560C uses `HP DeskJet 560C Printer`.

Automatically configuring printer drivers with Samba is explained in greater detail in the section "Automatically Setting Up Printer Drivers," earlier in this chapter.

printer driver file

This global option gives the location of the Windows 95/98 printer driver definition file, which is needed to give printer drivers to clients using a Samba printer. The default value of this option is */usr/local/samba/lib/printers.def*. You can override this default as shown below:

```
[deskjet]
    printer driver file = /var/printers/printers.def
```



Click the manufacturer and model of your printer. If your printer came with an installation disk, click Have Disk. If your printer is not listed, consult your printer documentation for a compatible printer.

Use this as a value for `printer driver`

Figure 7-9. The Add Printer Wizard dialog box in Windows 98

This option is explained in greater detail in the section "Automatically Setting Up Printer Drivers," earlier in this chapter.

printer driver location

This option specifies a specific share that contains Windows 95 and 98 printer driver and definition files. There is no default parameter for this value. You can specify the location as a network pathname. A frequent approach is to use a share on your own machine, as shown here:

```
[deskjet]
    printer driver location = \\%L\PRINTER$
```

This option is also explained in greater detail in the section "Automatically Setting Up Printer Drivers," earlier in this chapter.

lpq cache time

The global `lpq cache time` option allows you to set the number of seconds that Samba will remember the current printer status. After this time elapses, Samba will issue an *lpq* command (or whatever command you specify with the `lpq command` option) to get a more up-to-date status. This defaults to 10 seconds, but can be increased if your `lpq command` takes an unusually long time to run or you have lots of clients. The following example resets the time to 30 seconds:

```
[deskjet]
    lpq cache time = 30
```

postscript

The `postscript` option forces the printer to treat all data sent to it as Postscript. It does this by prepending the characters `%!` at the beginning of the first line of each job. It is normally used with PCs that insert a `^D` (control-D or "end-of-file mark) in front of the first line of a PostScript file. It will not, obviously, turn a non-PostScript printer into a PostScript one. The default value of this options is `no`. You can override it as follows:

```
[deskjet]
    postscript = yes
```

print command, lpq command, lprm command, lppause command, lpresume command

These options tell Samba which Unix commands used to control and send data to the printer. The Unix commands involved are: *lpr* (send to Line PRinter), *lpq* (List Printer Queue), *lprm* (Line printer ReMove), and optionally *lppause* and *lpresume*. Samba provides an option named after each of these commands, in case you need to override any of the system defaults. For example, consider:

```
lpq command = /usr/ucb/lpq %p
```

This would set the `lpq command` to use */usr/ucb/lpq*. Similarly:

```
lprm command = /usr/local/lprm -P%p %j
```

would set the Samba printer remove command to */usr/local/lprm*, and provide it the print job number using the `%j` variable.

The default values for each of these options are dependent on the value of the `printing` option. Table 7-4 shows the default commands for each of the printing options. The most popular printing system is BSD.

Table 7-4. Default Commands for Various Printing Commands

Option	BSD, AIX, PLP, LPRNG	SYSV, HPUX	QNX	SOFTQ
`print command`	`lpr -r -P%p %s`	`lp -c -d%p %s; rm %s`	`lp -r -P%p %s`	`lp -d%p -s %s; rm %s`
`lpq command`	`lpq -P%p`	`lpstat -o%p`	`lpq -P%p`	`lpstat -o%p`
`lprm command`	`lprm -P%p %j`	`cancel %p-%j`	`cancel %p-%j`	`cancel %p-%j`
`lppause command`	`lp -i %p-%j -H hold (SYSV only)`	`None`	`None`	`None`
`lpresume command`	`lp -i %p-%j -H resume (SYSV only)`	`None`	`None`	`qstat -s -j%j -r`

It is typically not necessary to reset these options in Samba, with the possible exception of `print command`. This option may need to be explicitly set if your printing system doesn't have a `-r` (remove after printing) option on the printing command. For example:

```
/usr/local/lpr -P%p %s; /bin/rm %s
```

With a bit of judicious programming, these *smb.conf* options can also used for debugging:

```
print command = cat %s >>/tmp/printlog; lpr -r -P%p %s
```

For example, this configuration can verify that files are actually being delivered to the Samba server. If they are, their contents will show up in the */tmp/printlog* file.

After BSD, the next most popular kind of printing system is SYSV (or System V) printing, plus some SYSV variants for IBM's AIX and Hewlett-Packard's HP-UX. These system do not have an */etc/printcap* file. Instead, the `printcap file` option can be set to an appropriate *lpstat* command for the system. This tells Samba to get a list of printers from the *lpstat* command. Alternatively, you can set the global configuration option `printcap name` to the name of a dummy *printcap* file you provide. In the latter case, the file must contain a series of lines such as:

```
lp|print1|My Printer 1
print2|My Printer 2
print3|My Printer 3
```

Each line names a printer, and provides aliases for it. In this example, the first printer is called `lp`, `print1`, or `My Printer 1`, whichever the user prefers to use. The first name will be used in place of `%p` in any command Samba executes for that printer.

Two additional printer types are also supported by Samba: LPRNG (LPR New Generation) and PLP (Public Line Printer). These are public domain and Open Source printing systems, and are used by many sites to overcome problems with vendor-supplied software. In addition, the SOFTQ and QNX realtime operating systems are supported by Samba.

load printers

The `load printers` option tells Samba to create shares for all known printer names and load those shares into the browse list. Samba will create and list a printer share for each printer name in */etc/printcap* (or system equivalent). For example, if your *printcap* file looks like this:[*]

```
lp:\
  :sd=/var/spool/lpd/lp:\            # spool directory
  :mx#0:\                            # maximum file size (none)
```

[*] We have placed annotated comments off to the side in case you've never dealt with this file before.

```
  :sh:\                            # surpress burst header (no)
  :lp=/dev/lp1:\                   # device name for output
  :if=/var/spool/lpd/lp/filter:    # text filter

laser:\
  :sd=/var/spool/lpd/laser:\       # spool directory
  :mx#0:\                          # maximum file size (none)
  :sh:\                            # surpress burst header (no)
  :lp=/dev/laser:\                 # device name for output
  :if=/var/spool/lpd/lp/filter:    # text filter
```

and you specify:

```
load printers = yes
```

the shares `[lp]` and `[laser]` will automatically be created as valid print shares when Samba is started. Both shares will borrow the configuration options specified in the `[printers]` section to configure themselves, and will be available in the browse list for the Samba server.

printcap name

If the `printcap name` option (also called `printcap`) appears in a printing share, Samba will use the file specified as the system printer capabilities file. This is normally */etc/printcap*. However, you can reset it to a file consisting of only the printers you want to share over the network. The value must be a fully-qualified filename of a printer capabilities file on the server:

```
[deskjet]
    printcap name = /usr/local/printcap
```

min print space

The `min print space` option sets the amount of spool space that must be available on the disk before printing is allowed. Setting it to zero (the default) turns the check off; setting it to any other number sets the amount of free space in kilobytes required. This option helps avoid having print jobs fill up the remaining disk space on the server, which may cause other processes to fail:

```
[deskjet]
    min print space = 4000
```

queuepause command

This configuration option specifies a command that tells Samba how to pause a print queue entirely, as opposed to a single job on the queue. The default value depends on the printing type chosen. You should not need to alter this option.

queueresume command

This configuration option specifies a command that tells Samba how to resume a paused print queue, as opposed to resuming a single job on the print queue. The default value depends on the printing type chosen. You should not need to alter this option.

Name Resolution with Samba

Before NetBIOS Name Servers (NBNS) came about, name resolution worked entirely by broadcast. If you needed a machine's address, you simply broadcast its name across the network and, in theory, the machine itself would reply. This approach is still possible: anyone looking for a machine named `fred` can still broadcast a query and find out if it exists and what its IP address is. (We use this capability to troubleshoot Samba name services with the `nmblookup` command in Chapter 9, *Troubleshooting Samba.*)

As you saw in the first chapter, however, broadcasting—whether it be browsing or name registration and resolution—does not pass easily across multiple subnets. In addition, many broadcasts tend to bog down networks. To solve this problem, Microsoft now provides the Windows Internet Naming Service (WINS), a cross-subnet NBNS, which Samba supports. With it, an administrator can designate a single machine to act as a WINS server, and can then provide each client that requires name resolution the address of the WINS server. Consequently, name registration and resolution requests can be directed to a single machine from any point on the network, instead of broadcast.

WINS and broadcasting are not the only means of name resolution, however. There are actually four mechanisms that can be used with Samba:

- WINS
- Broadcasting
- Unix */etc/hosts* or NIS/NIS+ matches
- *LMHOSTS* file

Samba can use any or all of these name resolution methods in the order that you specify in the Samba configuration file using the `name resolve order` parameter. However, before delving into configuration options, let's discuss the one that you've probably not encountered before: the *LMHOSTS* file.

The LMHOSTS File

LMHOSTS is the standard LAN Manager *hosts* file used to resolve names into IP addresses on the system. It is the NBT equivalent of the */etc/hosts* file that is standard on all Unix systems. By default, the file is usually stored as */usr/local/samba/lib/LMHOSTS* and shares a format similar to */etc/hosts*. For example:

```
192.168.220.100    hydra
192.168.220.101    phoenix
```

The only difference is that the names on the right side of the entries are NetBIOS names instead of DNS names. Because they are NetBIOS names, you can assign resource types to them as well:

```
192.168.220.100    hydra#20
192.168.220.100    simple#1b
192.168.220.101    phoenix#20
```

Here, we've assigned the hydra machine to be the primary domain controller of the SIMPLE domain, as indicated by the resource type <1B> assigned to the name after hydra's IP address in the second line. The other two are standard workstations.

If you wish to place an *LMHOSTS* file somewhere other than the default location, you will need to notify the *nmbd* process upon start up, as follows:

```
nmbd -H /etc/samba/lmhosts -D
```

Setting Up Samba to Use Another WINS Server

You can set up Samba to use a WINS server somewhere else on the network by simply pointing it to the IP address of the WINS server. This is done with the global wins server configuration option, as shown here:

```
[global]
    wins server = 192.168.200.122
```

With this option enabled, Samba will direct all WINS requests to the server at 192. 168.200.122. Note that because the request is directed at a single machine, we don't have to worry about any of the problems inherent to broadcasting. However, though you have specified an IP address for a WINS server in the configuration file, Samba will not necessarily use the WINS server before other forms of name resolution. The order in which Samba attempts various name-resolution techniques is given with the name resolve order configuration option, which we will discuss shortly.

If you have a Samba server on a subnet that still uses broadcasting and the Samba server knows the correct location of a WINS server on another subnet, you can configure the Samba server to forward any name resolution requests with the wins proxy option:

```
[global]
    wins server = 192.168.200.12
    wins proxy = yes
```

Use this only in situations where the WINS server resides on another subnet. Otherwise, the broadcast will reach the WINS server regardless of any proxying.

Setting Up Samba as a WINS Server

You can set up Samba as a WINS server by setting two global options in the configuration file, as shown below:

```
[global]
    wins support = yes
    name resolve order = wins lmhosts hosts bcast
```

The `wins support` option turns Samba into a WINS server. Believe it or not, that's all you need to do! Samba handles the rest of the details behind the scenes, leaving you a relaxed administrator. The `wins support=yes` and the `wins server` option are mutually exclusive; you cannot simultaneously offer Samba as the WINS server and point to another system as the server.

If Samba is acting as a WINS server, you should probably get familiar with the `name resolve order` option mentioned earlier. This option tells Samba the order of methods in which it tries to resolve a NetBIOS name. It can take up to four values:

lmhosts
: Uses a LAN Manager *LMHOSTS* file

hosts
: Uses the standard name resolution methods of the Unix system, */etc/hosts*, DNS, NIS, or a combination (as configured for the system)

wins
: Uses the WINS server

bcast
: Uses a broadcast method

The order in which you specify them in the value is the order in which Samba will attempt name resolution when acting as a WINS server. For example, let's look at the value specified previously:

```
name resolve order = wins lmhosts hosts bcast
```

This means that Samba will attempt to use its WINS entries first for name resolution, followed by the LAN Manager *LMHOSTS* file on its system. Next, the hosts value causes it to use Unix name resolution methods. The word `hosts` may be misleading; it covers not only the */etc/hosts* file, but also the use of DNS or NIS (as configured on the Unix host). Finally, if those three do not work, it will use a broadcast to try to locate the correct machine.

Finally, you can instruct a Samba server that is acting as a WINS server to check with the system's DNS server if a requested host cannot be found in its WINS database. With a typical Linux system, for example, you can find the IP address of the

DNS server by searching the */etc/resolv.conf* file. In it, you might see an entry such as the following:

```
nameserver 127.0.0.1
nameserver 192.168.200.192
```

This tells us that a DNS server is located at 192.168.220.192. (The 127.0.0.1 is the localhost address and is never a valid DNS server address.)

Use the global `dns proxy` option to alert Samba to use the configured DNS server:

```
[global]
    wins support = yes
    name resolve order = wins lmhosts hosts bcast
    dns proxy = yes
```

Name Resolution Configuration Options

Samba's WINS options are shown in Table 7-5.

Table 7-5. WINS Options

Option	Parameters	Function	Default	Scope
`wins support`	boolean	If set to `yes`, Samba will act as a WINS server.	`no`	Global
`wins server`	string (IP address or DNS name)	Identifies a WINS server for Samba to use for name registration and resolution.	None	Global
`wins proxy`	boolean	Allows Samba to act as a proxy to a WINS server on another subnet.	`no`	Global
`dns proxy`	boolean	If set to `yes`, a Samba WINS server will search DNS if it cannot find a name in WINS.	`no`	Global
`name resolve order`	`lmhosts`, `hosts`, `wins`, or `bcast`	Specifies an order of the methods used to resolve NetBIOS names.	`lmhosts hosts wins bcast`	Global
`max ttl`	numerical	Specifies the maximum time-to-live in seconds for a requested NetBIOS names.	`259200` (3 days)	Global
`max wins ttl`	numerical	Specifies the maximum time-to-live in seconds for NetBIOS names given out by Samba as a WINS server.	`518400` (6 days)	Global
`min wins ttl`	numerical	Specifies the minimum time-to-live in seconds for NetBIOS names given out by Samba as a WINS server.	`21600` (6 hours)	Global

wins support

Samba will provide WINS name service to all machines in the network if you set the following in the `[global]` section of the *smb.conf* file:

```
[global]
    wins support = yes
```

The default value is `no`, which is typically used to allow another Windows NT server to become a WINS server. If you do enable this option, remember that a Samba WINS server currently cannot exchange data with any backup WINS servers. If activated, this option is mutually exclusive with the `wins server` parameter; you cannot set both to `yes` at the same time or Samba will flag an error.

wins server

Samba will use an existing WINS server on the network if you specify the `wins server` global option in your configuration file. The value of this option is either the IP address or DNS name (not NetBIOS name) of the WINS server. For example:

```
[global]
    wins server = 192.168.220.110
```

or:

```
[global]
    wins server = wins.example.com
```

In order for this option to work, the `wins support` option must be set to `no` (the default). Otherwise, Samba will report an error. You can specify only one WINS server using this option.

wins proxy

This option allows Samba to act as a proxy to another WINS server, and thus relay name registration and resolution requests from itself to the real WINS server, often outside the current subnet. The WINS server can be indicated through the `wins server` option. The proxy will then return the WINS response back to the client. You can enable this option by specifying the following in the `[global]` section:

```
[global]
    wins proxy = yes
```

dns proxy

If you want the domain name service (DNS) to be used if a name isn't found in WINS, you can set the following option:

```
[global]
    dns proxy = yes
```

This will cause *nmbd* to query for machine names using the server's standard domain name service. You may wish to deactivate this option if you do not have a permanent connection to your DNS server. Despite this option, we recommend using a WINS server. If you don't already have any WINS servers on your network, make one Samba machine a WINS server. Do not, however, make two Samba machines WINS servers (one primary and one backup) as they currently cannot exchange WINS databases.

name resolve order

The global `name resolve order` option specifies the order of services that Samba will use in attempting name resolution. The default order is to use the *LMHOSTS* file, followed by standard Unix name resolution methods (some combination of */etc/hosts*, DNS, and NIS), then query a WINS server, and finally use broadcasting to determine the address of a NetBIOS name. You can override this option by specifying something like the following:

```
[global]
    name resolve order = lmhosts wins hosts bcast
```

This causes resolution to use the *LMHOSTS* file first, followed by a query to a WINS server, the system password file, and finally broadcasting. You need not use all four options if you don't want to. This option is covered in more detail in the section "Setting Up Samba as a WINS Server," earlier in this chapter.

max ttl

This option gives the maximum time to live (TTL) during which a NetBIOS name registered with the Samba server will remain active. You should never need to alter this value.

max wins ttl

This option give the maximum time to live (TTL) during which a NetBIOS name resolved from a WINS server will remain active. You should never need to change this value from its default.

min wins ttl

This option give the minimum time to live (TTL) during which a NetBIOS name resolved from a WINS server will remain active. You should never need to alter this value from its default.