



8

Linux Network Services

CERTIFICATION OBJECTIVES

8.01 Samba Networking

8.02 Print Services

✓ Two-Minute Drill
Q&A Self Test

Red Hat Enterprise Linux 3 supports direct communication with the various Microsoft Windows operating systems. Microsoft networking is based on the Common Internet File System (CIFS), which was developed from the Server Message Block (SMB) protocol. Samba was developed as a freely available SMB server for all Unix-related operating systems, including Linux, and has been upgraded to support CIFS.

Samba interacts with CIFS so transparently that Microsoft clients cannot tell your Linux server from a genuine Windows NT/2000/XP/2003 server, and with Samba there are no server, client, or client access licenses to purchase. If you can learn to edit the main Samba configuration file from the command line interface, you can configure Samba quickly. RHEL 3 includes a GUI alternative, the Samba Server Configuration utility.

INSIDE THE EXAM

Samba

If you're taking the RHCE exam, you definitely have to know how to configure Samba as a server and as a client. While there is no direct reference to Samba in the RHCT requirements, the spirit of the exam suggests that you need to know how to set up a Linux computer as a client on a network. While Samba clients are relatively easy to use, there is no reference to Samba client

software in the Red Hat curricula related to the RHCT exam.

Once you configure a Samba server, you can use Samba clients on Linux just as easily as you can use Microsoft's Network Neighborhood or My Network Places. While I can't tell you what I saw on the RHCE exam, you don't need a Microsoft Windows computer to confirm that you've configured Samba properly.

CUPS

CUPS is something that you may expect to configure on the RHCT and RHCE exams. You'll need to know how to configure a CUPS printer locally and to be shared on

a network. You need to know how to connect to a remote printer through CUPS. And finally, you'll want to learn how to manage CUPS print queues.

Printing is a fundamental service for all operating systems. The default for RHEL 3 is CUPS, which has replaced the Line Print Daemon. CUPS supports autoconfiguration of shared network printers and includes a Web interface configuration tool. Red Hat has also customized its own graphical configuration tool. It also provides connectivity to several other network print services via their native protocols.

As for the RHCE exam, you may have to configure or troubleshoot any of the services discussed in this chapter. So as you read this chapter and look through the configuration files and exercises, be willing to experiment. And practice, practice, practice what you learn.

CERTIFICATION OBJECTIVE 8.01

Samba Networking

Microsoft's CIFS was built on the Server Message Block (SMB) protocol. SMB was developed in the 1980s by IBM, Microsoft, and Intel as a way to share files and printers over a network.

As Microsoft has developed SMB into CIFS, the Samba developers have upgraded Samba accordingly. Samba services provide a stable, reliable, fast, and highly compatible file and print sharing service that allows your computer to act as a client, a member server, or even a Primary Domain Controller (PDC) on Microsoft-based networks. While Samba does not include every feature built into the latest Microsoft networks, I have confidence that it will in the near future.

SMB network communication over a Microsoft-based network is also known as NetBIOS over TCP/IP. Through the collective works of Andrew Tridgell and the Samba team, Linux systems provide transparent and reliable SMB support over TCP/IP via a package known as Samba. You can do four basic things with Samba:

- Share a Linux directory tree with Windows and Linux/Unix computers
- Share a Windows directory with Linux/Unix computers
- Share a Linux printer with Windows and Linux/Unix computers
- Share a Windows printer with Linux/Unix computers

4 Chapter 8: Linux Network Services

Samba emulates many of the advanced network features and functions associated with the Win9x/ME and NT/2000/XP operating systems through the SMB protocol. Complete information can be found at the official Samba Web site at www.samba.org. It is easy to configure Samba to do a number of things on a Microsoft-based network, for example:

- Participate in a Microsoft Windows 9x-style Workgroup or an NT/2000/XP Domain as a client, member server, or even a Primary Domain Controller
- Share user home directories
- Act as a WINS (Windows Internet Name Service) client or server
- Link to or manage a workgroup browse service
- Act as a Master Browser
- Provide user/password and share security databases locally, from another Samba server, or from a Microsoft NT 4 Primary Domain Controller
- Configure local directories as shared SMB filesystems
- Synchronize passwords between Windows and Linux systems
- Support Microsoft Access Control Lists

While Samba can do more, you get the idea. Samba features are configured through one very big file, `smb.conf`, in the `/etc/samba` directory. As this file may intimidate some users, Red Hat's Samba Server Configuration tool (**redhat-config-samba**) provides an easier interface. RHEL 3 does not include the Samba Web Administration Tool, so don't expect it to be available on the Red Hat exams.

RHEL 3 includes the newly released Samba version 3.0. This includes a number of features over and above the version of Samba included with Red Hat Linux 9 (2.2.7). Despite this change, you can use the same basic steps (for the most part) to configure a Samba client and or a Samba server on your computer.

exam

Watch

I believe that Red Hat's Samba Server Configuration utility is an effective tool. But remember, time is of the essence on the RHCE exam. If you know how to edit the `/etc/samba/smb.conf`

configuration file in a text editor, you're more likely to have time to configure the other elements you need to pass the exam. But don't be afraid to use the method that is fastest for you.

Installing Samba Services

If you selected the Windows File Server package when you installed RHEL 3, then the Samba RPM packages should already be installed. These are the four Samba RPM packages that you need:

- The `samba` RPM package includes the basic SMB server software for sharing files and printers.
- The `samba-client` RPM package provides the utilities needed to connect to shares from Microsoft computers.
- The `redhat-config-samba` package installs the Red Hat Samba Server Configuration utility.
- The `samba-common` RPM package contains common Samba configuration files. While it isn't a part of the Windows File Server package group, it is installed as a dependency if you install that package group.

It's easy to start the Samba Server Configuration tool. You can do so from a command line interface in the GUI with the **redhat-config-samba** command. Alternatively, you could click Main Menu | System Settings | Server Settings | Samba. Either command opens the utility as shown in Figure 8-1.

Configuring Samba to Start

Configuring Samba to start when Linux boots is a straightforward process. You'll want to configure Samba to start in runlevels 3 and 5, and then start the Samba server. You can do so with the following two commands:

```
# chkconfig --level 35 smb on
# service smb start
```

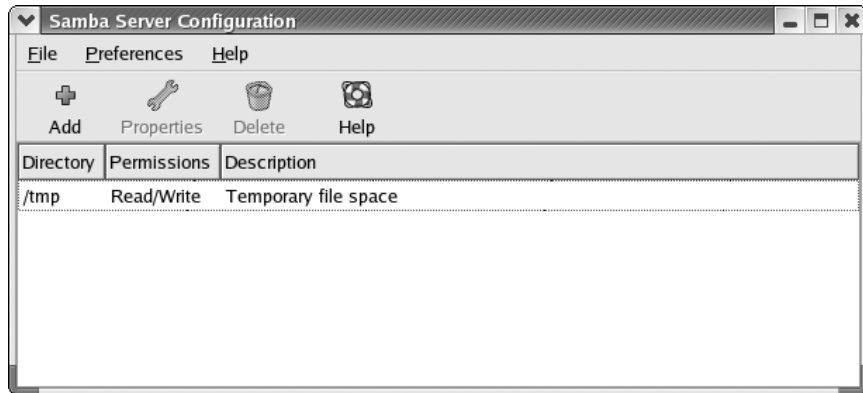
exam **watch**

If you're using Red Hat Linux 9 to study for the exam, you'll want to use an upgraded version of Samba. You can set it up on Red Hat Linux 9 using the RHEL 3 source RPMs available online

from ftp.redhat.com (or mirrors). Use the commands described in Chapter 4 to compile binary RPMs from source. You can then install the RHEL 3 version of Samba on your Red Hat Linux 9 computer.

FIGURE 8-1

Samba Server
Configuration
utility



Some Samba Background

Samba services provide interoperability between the Microsoft Windows and Linux/Unix computers. Before you begin configuring Samba, you need a basic understanding of how Microsoft Windows networking works with TCP/IP.

The original Microsoft Windows networks were configured with computer hostnames, known as NetBIOS names, limited to 15 characters. These unique hostnames provided a simple, flat hostname system for the computers on a LAN. All computer identification requests were made through broadcasts. This overall network transport system is known as NetBEUI, which is not “routable.” In other words, it does not allow communication between two different LANs. As a result, the original Microsoft-based PC networks were limited in size to 255 nodes.

While Microsoft networks could use the Novell IPX/SPX protocol stack to route messages between networks, that was not enough. As the Internet grew, so did the dominance of TCP/IP. Microsoft adapted its NetBIOS system to TCP/IP with SMB. Since Microsoft published SMB as an industry-wide standard, anyone could set up their own service to work with SMB.

One of the nice features of Windows networks is the browser service. All computers register their NetBIOS names with one “elected” master browser, the keeper of the database of network-wide services. In fact, a browse database is maintained by some “elected” host for every protocol running on the network. For instance, if the NetBEUI, IPX/SPX, and TCP/IP protocols were installed on a host, then three duplicate browse databases were required—one per protocol, as the services available may differ between protocols.

Name Resolution: WINS

WINS was designed as a dynamic, centralized, and robust service. It was supposed to become a viable alternative to DNS. Each WINS server maintained a central database with multiple records for all machines. On a large network, this was a big data file.

WINS needs about three to ten data records for each computer. Naturally, this is cumbersome for larger networks. This is another reason why Microsoft is phasing out WINS.

What About Samba?

This is where Samba fits in. Samba on Linux provides all the Windows networking services available on any Windows TCP/IP client or server. To configure Samba, you simply need to know the name of your NT/2000/XP/2003 Domain or Windows 9x/ME-style Workgroup, and configure the parameters accordingly for your Linux workstation or server to match the settings on the local Microsoft network.

Fortunately, Samba comes with extensive online documentation (with examples) available in the `smb.conf` configuration file. The following lists some of the key Samba commands and files:

<code>/usr/sbin/smbd</code>	- main SMB service daemon
<code>/usr/sbin/nmbd</code>	- NetBIOS name service daemon
<code>/etc/samba/smb.conf</code>	- SAMBA's primary configuration file
<code>/usr/bin/smbclient</code>	- connects to SMB shares, ftp-like syntax
<code>/usr/bin/smbmnt</code>	- mounts SMB shares on a designated directory
<code>/usr/bin/smbumount</code>	- unmounts a SMB shared directory
<code>/usr/bin/testparm</code>	- tests validity of <code>/etc/samba/smb.conf</code> file
<code>/etc/rc.d/init.d/smb</code>	- daemon start and stop control script
<code>smbfs</code>	- file system extension to mount SMB shares on directories; use with the mount -t command.
<code>/usr/bin/smbprint</code>	- a script to print to a printer on an SMB host
<code>/usr/bin/smbstatus</code>	- lists current SMB connections for the local host

Samba Has Two Daemons

You need two daemons to run Samba: `smbd` and `nmbd`, both located in `/usr/sbin`. Both are configured through the `/etc/samba/smb.conf` configuration file. It's easy to check the syntax of this large configuration file with the **testparm** command. If problems arise, this program produces error messages to help you correct them.

Configuring Samba as a Client

There are two types of clients that you can configure through Samba. One connects to directories shared from Microsoft Windows servers or Samba servers on Linux/Unix. The second connects to shared printers from one of the same two types of servers.

When you have installed the samba-common RPM package, you've installed the Samba client commands that you need to find browse lists and mount shared directories locally.

Checking Samba File and Print Services

If you want to browse shared directories from a Linux computer, you'll want to know how to use **smbclient**. This can help you test connectivity to any SMB host on a Windows- or Samba-based Linux/Unix computer. You can use **smbclient** to check the directories and printers that shared from remote computers on your network. For example, the **smbclient** command shown in Figure 8-2 checks shared directories and printers from a remote Linux-based Samba PDC.

As you can see, I've specified two arguments with the **smbclient** command: **-L** allows you to specify the name of the Samba server, and **-U** allows you to specify a username

FIGURE 8-2

List of shared directories and printers from a remote PDC

```
[root@Enterprise3 root]# smbclient -L //cosmicc -U donna
Password:

      Sharename      Type      Comment
      -----
netlogon            Disk      Network Logon Service
tmp                 Disk      Temporary file space
registry            Disk      Disabling Encryption
inst                Disk      Red Hat Linux 9 CD files
inst-1              Disk      Enterprise Inst Directory
IPC$                IPC       IPC Service (PDC - from a Samba Server)
ADMIN$              Disk      IPC Service (PDC - from a Samba Server)
WinPrint1           Printer
LinuxHP             Printer
WinClass1           Printer
WinPrint2           Printer
donna               Disk      Home Directories

      Server          Comment
      -----
ALLACCESS            Windows XP Pro Workstation
COSMICC              PDC - from a Samba Server

      Workgroup       Master
      -----
GRATEFUL             COSMICC
[root@Enterprise3 root]#
```

on the remote computer. If the command reaches the Samba server, you're then prompted for the appropriate password.

Alternatively, you can browse shared Samba directories using the graphical Nautilus client. In the Linux desktop, click Main Menu | Home Folder. This opens the Nautilus file manager with a list of files in your home directory. In the Location text box, enter

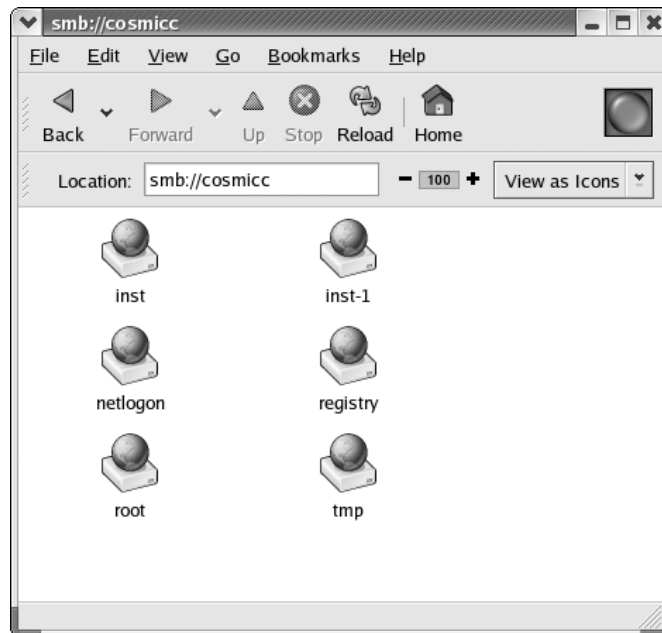
```
smb:///
```

If you're connected to a network with Windows and or Samba-enabled Linux computers, you'll see an icon associated with connected workgroups and or domains. Click on the workgroup or domain of your choice. You should see a list of Windows and Samba-enabled Linux computers on that workgroup or domain.

Select the computer of your choice. You'll be prompted for a username and password on the remote computer. Once entered, you'll get a list of shared directories as shown in Figure 8-3.

FIGURE 8-3

Browsing remote
shared
directories



Mounting Shared Samba Directories During Login

You can also configure automated mounting during the login process. In the first lab of Chapter 5, you saw how to configure connections to shared NFS directories. You can use the same process with shared Samba directories.

Specifically, you can configure commands to mount a shared directory for an individual user in the `.bashrc` file, and set up a command to unmount that directory in the `.bash_logout` file. Both files are stored in each user's home directory.

There's one drawback: unless you're willing to add the shared directory to your `/etc/fstab` file, or you're limiting yourself to the root user, you can't use the **mount** command. There is an option: the **smbmnt** and **smbumount** commands. Once you've set them up to be executable for all with the following commands, you can add them to individual users' `.bashrc` and `.bash_logout` files:

```
# chmod u+s /usr/bin/smbmnt
# chmod u+s /usr/bin/smbumount
```

These commands set the SUID bit, which I described briefly in Chapter 1 and use in Chapter 11. Once the SUID bit is set, you can mount shared Samba directories as a regular user. Now when you log in as a regular (not root) user, you can mount and unmount a Samba share named `inst` from a computer named `cosmic` on the `/home/michael/shared` directory with the following commands:

```
$ smbmnt //cosmicc/inst /home/michael/shared -o username=michael
$ smbmount /home/michael/shared
```

With the first command, you're prompted for a password for user `michael` on the `cosmicc` computer. In other words, user `michael` gets to enter two passwords: first to log into his local account, second to authorize a connection to the shared directory. Alternatively, if user `michael`'s password is `a2b3c4d5`, you could use the following command:

```
$ smbmnt //cosmicc/inst /home/michael/shared -o username=michael%a2b3c4d5
```

Once you've verified that they work, you can add these commands to user's `.bashrc` and `.bash_logout` files. If you're configuring this for the root user, you'll need to use the corresponding **mount** commands:

```
# mount -o username=michael //cosmicc/inst /root/shared
# umount /root/shared
```

You can add user michael's password in the same way as shown for the **smbmnt** command. Once you're satisfied with the result, you can add these commands to the root user's `.bashrc` and `.bash_logout` files.

Alternatively, if your users log in through GNOME, you can set up the same **mount** command through the Sessions utility. As discussed in Chapter 6, you can start this utility by clicking Main Menu | Preferences | More Preferences | Sessions. Figure 8-4 illustrates the previous command, with a password, added to the graphical startup programs list for the root user.

Client Configuration for Print Services

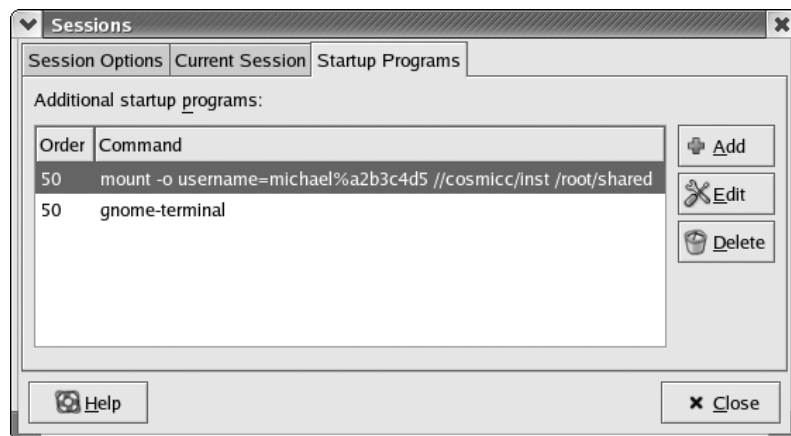
There is a simple option line in the `/etc/samba/smb.conf` file that shares all local printer systems as if this were another Windows host.

In `/etc/samba/smb.conf`, printer configurations start with the section heading named `[printers]`. Using the same share options used for directories, Samba can create a shared print service for each installed print queue. These print shares are available to Microsoft clients when users install network printers. For more information on the `[printers]` share, please read the next section.

Once shared, you can use a service such as CUPS to connect to these shared printers. In fact, I'll show you how this is done with the Red Hat Printer Configuration utility later in this chapter.

FIGURE 8-4

Using Startup Programs to connect to a shared Samba directory



Configuring a Samba Server

If you want to configure a Samba server, you'll need to edit the main Samba configuration file, `/etc/samba/smb.conf`. This file is long and includes a number of commands that require a good understanding of Microsoft Windows networking. Fortunately, the default version of this file also includes helpful documentation with suggestions and example configurations that you can use.

You can edit this file directly, or you can create directory shares using Red Hat's Samba Server Configuration utility. It's useful to study the original `/etc/samba/smb.conf` file. Once you see how the file is structured, back it up. Try editing the file directly. Try changing the file with the Samba Server Configuration utility (which I describe in the next section). Test the result by restarting the Samba server with the following command:

```
# service smb restart
```

To help you with this process, I've copied the RHEL 3 version of this file. The following code is essentially a complete view of this file. I've replaced the comments in the file with my own explanations. You might want to browse your own `/etc/samba/smb.conf` file as well.

exam

watch

As stated in the Red Hat exam guide, RHCEs must be able to configure various services, including

Samba, for basic operation. I go into detail on the `smb.conf` file that, in my opinion, goes beyond basic operation.

The `smb.conf` file includes two types of comment lines. The hash symbol (`#`) is used for a general text comment. This is typically verbiage that describes a feature. The second comment symbol is the semicolon (`;`), used to comment out Samba directives (which you may later wish to uncomment in order to enable the disabled feature).

on the job

Because of the dimensions of the book, the length of code lines is limited. In a few cases, I've modified the code lines slightly to meet this limit, without changing the intent of any command in this configuration file.

```
# This is the main Samba configuration file. You should read the
# smb.conf(5) manual page in order to understand the options listed
# here. Samba has a huge number of configurable options (perhaps
```

```
# too many!) most of which are not shown in this example
#
# comments deleted
# NOTE: Whenever you modify this file you should run the command
# "testparm" to check that you have not made any basic syntactic
# errors.
```

Global Settings

Remember, time is of the essence on the RHCE exam. While you should be aware of what you can do with all of the different global settings, change as little as possible. The less you change, the less that can go wrong. Perfect configuration files are not required. Configuration files that meet the specific requirements of your exam are.

In `smb.conf`, the global settings, which define the overall attributes of your server, follow the first set of comments. This section starts with the following two lines:

```
===== Global Settings =====
[global]
```

Now we'll examine the critical global settings. First, with respect to the **workgroup** variable, this Samba server will become a member of that Microsoft Workgroup or Domain. The default Samba workgroup is Workgroup. If you know Microsoft Windows, you'll recognize it as the name of the default peer-to-peer workgroup. The same variable is used if you're joining this computer to a Microsoft-style Domain.

```
# workgroup = NT-Domain-Name or Workgroup-Name
workgroup = MYGROUP
```

Next, it's a good idea to add a NetBIOS name for your computer to this file. This becomes what other clients see in network browse lists such as those shown from Network Neighborhood/My Network Places and the **smbclient** command. The command that follows becomes the comment shown with the browse list:

```
# local computer NetBIOS name
netbios name = cosmicc
# server string is the equivalent of the NT Description field
server string = Samba Server
```

If you activate the **hosts allow** command, you can limit access to the specified network. The following default would limit access to the networks with the 192.168.1.0 and 192.168.2.0 network IP addresses, as well as the local computer (127.):

```
; hosts allow = 192.168.1. 192.168.2. 127.
```

These default printer settings are required to share printers from this Samba server. You'll want to change the printing command as shown to enable cooperation with CUPS, as shown. (CUPS is an allowed value despite the comments in the default RHEL 3 version of this file.)

```
printcap name = /etc/printcap
load printers = yes
;   printing = bsd
    printing = cups
```

If you want to configure a special guest account, you can activate this command, as long as you add a pguest user with the **useradd** command described in Chapter 4.

```
;   guest account = pguest
```

This command sets up log files for every computer that connects to this Samba server. For example, if a computer named allaccess connects to this Samba server, you can find a log of its access problems in `/var/log/samba/allaccess.log`. The log file is limited to 50Kb.

```
log file = /var/log/samba/%m.log
max log size = 50
```

The **security** command may be a bit confusing. This command means that connections check the local password database. It is appropriate if you're configuring this computer as a PDC.

```
security = user
```

If you want to configure this computer as a member server on a Domain, you'll want to use a password database from a PDC. Strangely enough, in that case, you would substitute the following command:

```
security = domain
```



If you just want to set up this computer as a workstation that happens to share directories on a Microsoft Domain, you'll need to set up the computer as a member server on that Domain.

Alternatively, if you just want to use a database from another computer that is not a PDC, you'd substitute the following command:

```
security = server
```

Finally, if you're configuring this computer on a peer-to-peer workgroup, you want to substitute the following command:

```
security = share
```

There are four basic authentication options: **share**, **user**, **server**, and **domain**. The default is **user**; in this case, you'll want to make sure the Samba usernames and passwords that you create match those on individual Windows NT/2000/XP systems on your network.

If you use the **server** authentication option, you can name another Samba server to carry the database of usernames and passwords.

If you use the **domain** authentication option, you can name an NT/2000/XP domain controller. You can set up a Samba server as a domain controller as well, emulating the functionality of a Windows domain controller.



With Samba version 3.0, you can now configure a Samba-enabled Linux computer as a member server on an Active Directory network. If that's what you want, set up security = ads.

If you've set up **security = domain** or **share**, you'll want to activate this command with the name of the password server:

```
; password server = <NT-Server-Name>
```

Linux is case sensitive. If you want to disable the case sensitivity for usernames and passwords of up to eight characters, activate these commands:

```
; password level = 8
; username level = 8
```

Encryption is enabled by default for Microsoft Windows computers, unless you have a network with operating systems older than Windows 95 (pre-OSR2) or Windows NT 4 (before Service Pack 3). Therefore, you'll want to activate these commands. I'll describe how you can set up the Samba password file later in this chapter.

```
; encrypt passwords = yes
; smb passwd file = /etc/samba/smbpasswd
```

If you've configured the same usernames for your Microsoft and Linux computers, activate the following commands. They help synchronize the passwords on the different accounts.

```
; unix password sync = Yes
; passwd program = /usr/bin/passwd %u
```

```
; passwd chat = *New*UNIX*password* %n\n \
; *ReType*new*UNIX*password* %n\n \
*passwd:*all*authentication*tokens*updated*successfully*
```

If the usernames on your Linux computers aren't identical to your Windows usernames, you'll need to activate a database of matching usernames in the following database:

```
; username map = /etc/samba/smbusers
```

You can customize the local Samba configuration by the computer (machine) client that connects by activating the following command. Configuring these computer-specific command files is complex, and I suspect is beyond configuring “the service for basic operation.”

```
; include = /etc/samba/smb.conf.%m
```

Configuring socket options also takes trial and error and requires more than the “basic operation” configuration skills specified in the Red Hat Exam Prep guide.

```
socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
```

If your computer is connected to more than one network, you can specify the networks served by your Samba server here:

```
; interfaces = 192.168.12.2/24 192.168.13.2/24
```

If you want to keep your browse list synchronized with other LANs, you can specify those computers or networks here.

```
; remote browse sync = 192.168.3.25 192.168.5.255
```

If other computers are having trouble finding the browse list, you can tell Samba to share the list with specific computers or subnets:

```
; remote announce = 192.168.1.255 192.168.2.44
```

Unless you specifically designate this computer to be a local master, Samba participates in browser elections like any other Microsoft Windows computer, using the **os level** that you designate. Alternatively, if you don't already have a Domain Controller acting as a browse master, you can give the responsibility to the Samba server. Or if you just want to make it easier for the local computer to win the browser election, activate the **preferred master** command.

```
; local master = no
; os level = 33
; domain master = yes
; preferred master = yes
```

If you have Windows 95 computers on your network, you'll want to activate this command. Remember, many Windows 95 computers can't handle encryption.

```
; domain logons = yes
```

The following commands set up Microsoft command line batch files by computer and user. The command afterwards stores Microsoft user profiles on the local Samba server. That means these commands can't be tested on the Red Hat exams unless you have access to a Microsoft Windows computer. Since I can't tell you what's on the Red Hat exams, it is up to you to determine whether it's practical to have separate Microsoft Windows computers available during those exams.

```
; logon script = %m.bat
; logon script = %U.bat
; logon path = \\%L\Profiles\%U
```

If you have multiple browse methods, you can set the order used by your Samba server to search for other computers. This particular command looks to the WINS (Windows Internet Name Service) server first, followed by any available lmhosts file; if both fail, a broadcast request for names on the network.

```
; name resolve order = wins lmhosts bcst
```

If you activate the following command, Samba activates a WINS server on the local computer:

```
; wins support = yes
```

Alternatively, you can point the local computer to a remote WINS server on the network; of course, you'd have to substitute the IP address for w.x.y.z.

```
; wins server = w.x.y.z
```

If there are non-WINS capable computers on the network, such as Linux computers without the Samba server software, you can activate this command:

```
; wins proxy = yes
```

on the
! o b

Adding Linux to a Microsoft Windows NT/2000/XP network can be made easier by configuring the Samba service to look like another Windows host on the network. You can configure the Samba server to act as a WINS client of the WINS server, share files and printers just like all the other Windows hosts, and participate in the browser service.

If you change this setting to **yes**, name searches can go through available DNS databases.

```
dns proxy = no
```

The remaining commands are fairly self-explanatory; however, I've never changed these settings, as changes to these variables can cause trouble with file transfers.

```
# Case Preservation can be handy - system default is _no_
# NOTE: These can be set on a per share basis
; preserve case = no
; short preserve case = no
# Default case is normally upper case for all DOS files
; default case = lower
# Be very careful with case sensitivity - it can break things!
; case sensitive = no
```

on the
! o b

If you want to set up a Samba server to use a Microsoft Windows database of usernames and passwords, you'll need to activate the winbindd daemon. With the right commands in smb.conf, you can also set up Microsoft users and groups with the UIDs and GIDs of your choice on your Linux system.

Share Settings

Share settings are organized into *stanzas*, which are groups of commands associated with a share name. (With respect to stanzas, some believe that well-constructed configuration code is like good poetry.) The first four lines in this section define the [homes] share, which automatically shares the home directory of the logged in user. Remember, RHEL 3 includes Samba version 3.0. Every user gets access to their own home directory; the **browsable = no** command keeps users away from each other's home directory.

There is no default /homes directory. It's just a label. You don't need to supply a home directory, because Samba will read the user's account record in /etc/passwd and /etc/shadow to determine the directory to be shared.

By default, this does not allow access to unknown users (**guest ok = no**). If you prefer, you can limit the systems that can use this share (**hosts allow = ?**, and **hosts deny = ?**).

```
#===== Share Definitions =====
[homes]
    comment = Home Directories
    browseable = no
    writable = yes
```

exam

Watch

If you forget the meaning of a specific Samba variable, you can find more information each variable with the man smb.conf command.

Those of you who are familiar with Red Hat Linux 9 may note that it supports Samba version 2.2.7. The basic [homes] directory share is the same; however, you may notice that the Samba version 2.2.7 version of this file in Red Hat Linux 9 includes three more variables:

- **valid users = %S** By default, all users with a local account can access their home directory using this share; this command adds the name of the service.
- **create mode = 664** Sets permissions when you create or copy files to your home directory through this share. The default, and thus what happens in the default RHEL 3 smb.conf file, corresponds to a create mode of 744. This gives the file owner rwx permissions, and other users on that computer read-only permissions. I explain the concept of Linux file permissions briefly in Chapter 1.
- **directory mode = 775** Similar to create mode; sets permissions when you create directories through this share. The default for RHEL 3 if you don't specify a **directory mode** is 755.

If you activate the commands in the following stanza, you can set up a netlogon share for Microsoft Windows workstations. As there are no netlogon shares even for Samba-enabled Linux workstations, this section requires a Microsoft Windows computer to verify functionality. If you believe that you'll have access to a Microsoft Windows computer during the Red Hat exams, you'll want to study this section carefully.

```
# Un-comment the following and create the netlogon directory for
# Domain Logons
; [netlogon]
;   comment = Network Logon Service
;   path = /home/netlogon
;   guest ok = yes
;   writable = no
;   share modes = no
```

This next stanza configures profiles for Microsoft Windows workstations. As these profiles become a part of a Microsoft Windows registry when you log on to one of those workstations, you're unlikely to configure this section in a network of Linux-only computers. Make your own judgment on whether you might have to configure this section during the RHCE exam.

```
# Un-comment the following to provide a specific roving profile
# share; the default is to use the user's home directory
;[Profiles]
;    path = /home/profiles
;    browseable = no
;    guest ok = yes
```

The [printers] stanza normally works as is, to allow access by all users with accounts on your computer or domain. If you want to limit access to specific users, see the [fredsprn] stanza later in this default file. While the spool directory (/var/spool/samba) is not browsable, the associated printers are browsable by their NetBIOS names. When you configure a printer later in this chapter, the name you designate automatically becomes the NetBIOS name.

```
# NOTE: If you have a BSD-style print system there is no need to
# specifically define each individual printer
[printers]
    comment = All Printers
    path = /var/spool/samba
    browseable = no
# Set public = yes to allow user 'guest account' to print
    guest ok = no
    writable = no
    printable = yes
```

If you activate the [tmp] share, it's a standard location for people to download and share files; all users get write access to this share.

```
# This one is useful for people to share files
;[tmp]
;    comment = Temporary file space
;    path = /tmp
;    read only = no
;    public = yes
```

The following stanza, as suggested by the comment, configures the /home/samba directory to be shared by the group named staff. You can configure this group in /etc

/group or through the Network Information System (Chapter 10). And you need to configure special ownership and permissions for /home/samba, as described in the User Private Group scheme in Chapter 11.

```
# A publicly accessible directory, but read only, except for people
# in the "staff" group
;[public]
;   comment = Public Stuff
;   path = /home/samba
;   public = yes
;   read only = yes
;   write list = @staff
```

If you activate the following stanza, it would configure a printer exclusively for one user. The default RHEL 3 smb.conf file has an error in this file; it lists the path to Fred's home directory as /homes/fred.

```
# A private printer, usable only by fred. Spool data will be
# placed in fred's home directory. Note that fred must have
# write access to the spool directory,
# wherever it is.
;[fredsprn]
;   comment = Fred's Printer
;   valid users = fred
;   path = /home/fred
;   printer = freds_printer
;   public = no
;   writable = no
;   printable = yes
```

This stanza configures a directory for Fred's exclusive use. I'm not sure why it's included in the Samba configuration file, as the user fred would already have a home directory on this computer. A better location for the **path**, which you need to create, is within the /home directory.

```
# A private directory, usable only by fred. Note that fred
# requires write access to the directory.
;[fredsdir]
;   comment = Fred's Service
;   path = /usr/somewhere/private
;   valid users = fred
;   public = no
;   writable = yes
;   printable = no
```

You can also set up directories by workstation; the `%m` variable is replaced by the name of the computer (machine) that connects to the Samba server. A better location for the path is the `/home/pc/%m` directory.

```
# The %m gets replaced with the machine name that is connecting.
;[pchome]
; comment = PC Directories
; path = /usr/pc/%m
; public = no
; writable = yes
```

The following stanza is slightly different from the `[tmp]` share. Once connected, the only user that connects is a guest. Unless you've configured a guest user, this defaults to the user nobody.

```
# A publicly accessible directory, read/write to all users. Note
# that all files created in the directory by users will be owned
# by the default user, so any user with access can delete any
# other user's files. Obviously this directory must be writable
# by the default user. Another user could of course be specified,
# in which case all files would be owned by that user instead.
;[public]
; path = /usr/somewhere/else/public
; public = yes
; only guest = yes
; writable = yes
; printable = no
```

Finally, this is another variation on the User Private Group scheme, which creates a group directory. Unlike the `[public]` stanza, this share is private.

```
# The following two entries demonstrate how to share a directory so
# that two users can place files there that will be owned by the
# specific users. In this setup, the directory should be writable
# by both users and should have the sticky bit set on it to prevent
# abuse. Obviously this could be extended to as many users as required.
;[myshare]
; comment = Mary's and Fred's stuff
; path = /usr/somewhere/shared
; valid users = mary fred
; public = no
; writable = yes
; printable = no
; create mask = 0765
```

To summarize, the settings for each shared directory start with a section name, such as `[tmp]`. This section name contains the name that will be seen by Microsoft clients only if the service is set to be browseable (**browseable = yes**).



There are a number of variables in `smb.conf` that are not spelled correctly, such as `browseable`. In some cases, the correct spelling (`browsable`) also works. They are still accepted Samba variables, and generally should be spelled per the Samba defaults, not standard written English.

Joining a Domain

If you've configured a Samba server, and it's not the PDC for your network, you'll need to set it to join the domain. Essentially, you're configuring an account on the Domain Controller for the network. As long as there's one domain on this network, it's easy to do with the following command:

```
# net rpc join -U root
```

This assumes that root is the administrative user on the PDC; if you're joining a domain governed by a Microsoft Windows computer, the administrative user is *administrator*. If successful, you're prompted for the root password on the remote PDC. An account for the local computer is added to the PDC's user database in `/etc/passwd`.



If you're using Red Hat Linux 9 with Samba 2.2.7, the `net rpc` command is not available. You'll either have to upgrade or use the `smbpasswd -j domainname -r domaincomputer -U root` command.

Configuring Samba Users

You could set up identical usernames and passwords for your Microsoft Windows and Samba-enabled Linux computers. However, this is not always possible. For example, Microsoft usernames often start with a capital letter, which is not allowed for a Linux username. In that case, you'll want to set up a database of Samba users and passwords which correspond to current Microsoft usernames and passwords on your network.

If you're comfortable with the command line interface, the quickest way to set up Samba users is with the `smbadduser` and `smbpasswd` commands. Remember, you can create a new Samba user only from valid accounts on your Linux computer.

Managing Samba Users

You can set up Samba users on a list independent from your Linux users who have accounts on your Linux system. The Samba development team chose to do this because

- There is no reason to grant Samba access to all Linux users.

- You may wish to manage user access via Windows NT/2000/XP, so Linux wouldn't necessarily even know about your Samba users.
- Samba user authentication may involve clear text passwords (for compatibility with Windows 95 and Windows 3.1). This could potentially compromise your Linux system.

To support these features, you can set up separate user accounts in the `/etc/samba` directory, in the `smbusers` and `smbpasswd` files. Two steps are required to make and enable a new Samba user:

1. Create a Samba user entry by name and add a password for the user. Samba users can be created only from the current users on your Linux system.
2. Enable Samba access for the new user.

If the username that you want does not yet exist, create it with the **useradd** *username* command. Then you can set that user up as a Samba user with the **smbpasswd** command. Use the following command; you're prompted to enter a password. That password can be different from the password used to log in directly to that Linux computer.

```
# smbpasswd -a newUser
New SMB password:
Retype SMB password:
#
```

Next, to enable Samba access for this user, run the following command:

```
# smbpasswd -e newUser
Enabled user newUser
#
```

Changes made by **smbpasswd** are passed to the Samba server to be copied to the system with the username and password database for your network.



The `smbadduser` command is no longer available in Samba 3.0 which is included with RHEL 3.

If you've configured Samba as a PDC for your network, the `/etc/passwd` file should govern the basic usernames and passwords for your system. You can use the **mksmbpasswd.sh** script to add all passwords to the `/etc/samba/smbpasswd` configuration file.

The **smbpasswd** command is powerful in RHEL 3; it includes a number of switches that you should learn, as described in Table 8-1.

TABLE 8-1

Various
smbpasswd
Commands

smbpasswd Switch	Description
-a <i>username</i>	Adds the specified username to /etc/samba/smbpasswd.
-d <i>username</i>	Disables the specified username; thus disables that password from Microsoft networking.
-e <i>username</i>	Enables the specified username; opposite of -d.
-r <i>computername</i>	Allows you to change your Windows or Samba password on a remote computer. Normally goes with -U.
-U <i>username</i>	Normally changes the username on a remote computer, if specified with the -r switch.
-x <i>username</i>	Deletes the specified username to /etc/samba/smbpasswd.

If you need to configure different usernames and passwords for your Linux and Microsoft computers, you'll need to edit them directly into the /etc/samba/smbusers file—or you can use the Samba Server Configuration utility which I describe shortly.

EXERCISE 8-1

Using Home Directories

In this exercise, you'll learn about the basic home directory share. You'll need at least two computers, one of which should be a Samba server. The other can be a Linux or Microsoft Windows workstation. You'll connect to the Samba server from the workstation, and access the files in your home directory on the Samba server.

1. Install and configure Samba to start using the methods described earlier in this chapter.
2. Open the /etc/samba/smb.conf configuration file. Look for the current value of **workgroup**.
3. Make sure that the computers on your network have the same value of **workgroup**. If your computer is on a domain, set **workgroup** to the name of the domain. If you don't already have a WINS server on this network, you'll also want to activate the **wins support = yes** command.
4. Test the syntax of your Samba configuration file with the **testparm** command (I'll describe how this works shortly).

5. Read and address any problems that you might see in the output from the **testparm** command. Fix any syntax problems with your **smb.conf** configuration file.
 6. Set up the root user on the server in the Samba database with the following commands (enter an appropriate password when prompted):


```
# smbpasswd -a root
# smbpasswd -e root
```
 7. Make Samba reread the **smb.conf** file with the following command:


```
# service smb reload
```
 8. Now go to a remote Linux or Microsoft Windows workstation on the same Domain or Workgroup.
 9. If you can browse the list of computers from the Samba server with the following command, browsing and probably WINS is working properly. (Alternatively, from a Microsoft Windows computer, you should see a list of computers in the Network Neighborhood or My Network Places window.)


```
# smbclient -L sambaserver -U root
```
 10. Enter the root username on the remote Samba server.
 11. If you're on a Linux computer, use the **smbmnt** or **mount** command (depending on whether you're regular or a root user) to configure the remote [homes] directory share on an empty local directory. For example, as the root user, you could mount on the local **/mnt/share** directory with the following command:


```
# mount -o username=root "//sambaserver/homes" /mnt/share
```
 12. Test the result. Can you browse your home directory on the remote computer?
-

The Red Hat Samba Server Configuration Utility

RHEL 3 includes Red Hat's graphical configuration tool for Samba, **redhat-config-samba**, which you can install from the RPM of the same name. Before you use this tool to modify your configuration, back up the files in your **/etc/samba** directory.

Also known as the Samba Server Configuration utility, you can use this tool to set basic global parameters and configure shared directories. You can start it from a GUI

command line with the **redhat-config-samba** command, or you can click Main Menu | System Settings | Server Settings | Samba. You've seen the basic tool back in Figure 8-1.

You will find that this tool is straightforward to use. You can configure general Samba settings such as security level and workgroup through the Preferences | Server Settings command. The Add button enables you to set up a new share. The Preferences | Samba Users command allows you to modify the smbusers and smbpasswd files in the /etc/samba directory.

You can also use this tool to configure Samba usernames and passwords. In other words, you can use this tool to configure your smb.conf file as well as Samba usernames and passwords through the smbusers and smbpasswd files. Naturally, these files are stored in the /etc/samba directory.

There are drawbacks to the Samba Server Configuration utility. For example, you can't use it to edit all global parameters or share printers. You can't use it to set a Samba member server to join a domain.

exam

Watch

The Samba Server Configuration utility may not do everything you need. To configure most global settings,

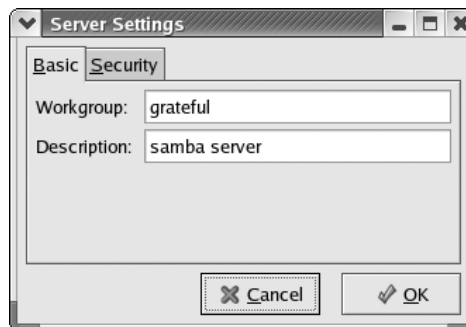
special printer shares, to join a domain, to control samba services, your fastest option is to work from the command line interface.

Global Settings

To see what the Samba Server Configuration utility can do to the global settings in the smb.conf configuration file, click Preferences | Server Settings. As you can probably guess from Figure 8-5, the basic settings set the **workgroup** and **server string** variables.

FIGURE 8-5

Samba Server
basic settings



When you use this utility and assign default variables, it erases the variable from your `smb.conf` file. For example, if you set the **workgroup** name to Workgroup, this utility erases the **workgroup** command line from `smb.conf`. Therefore, it's an excellent idea to back up `smb.conf` before using the Samba Server Configuration utility.

In contrast, the Security tab supports a few more settings, as you can see in Figure 8-6. The entries are fairly straightforward. If you want more information on these variables, refer to the discussion on `smb.conf` earlier in this chapter:

- Authentication Mode sets the **security** value in `/etc/samba/smb.conf`. The default is **user**.
- Authentication Server sets up the location of the **password server**. There is no default.
- Encrypt Passwords is associated with the variable of the same name. The default is **yes**.
- Guest Account is associated with the variable of the same name. The default in RHEL 3 is **nobody**.



If you've selected a default, you may still see the variable in the `smb.conf` file in comments. Alternatively, you may see the variable in an unexpected location relative to the default comments.

Share Settings

Click the Add button. This opens the Create Samba Share window shown in Figure 8-7. The Basic tab helps you define the basic parameters associated with the share:

- Directory defines the directory that you want to share, using the **path** variable. The share name is taken from the last part of the directory name. For example, if you're sharing the `/usr/share/to/path1`, Samba designates `[path1]` as the share name.
- Description allows you to define the comment that users can see in the browse list.
- Basic permissions lets you set **writable** as yes or no; read only (**writable=no**) is the default. (This is another case where two spellings for the same variable are acceptable; `writable` is also an acceptable spelling in `smb.conf`.)

The Access tab is simpler; it allows you to limit access to specific users from the `smbpasswd` configuration file. In other words, you can only limit access to users from

FIGURE 8-6

Samba Server
security settings



the Samba password database. Once you've clicked OK, the Samba Server Configuration tool automatically updates the `smb.conf` configuration file.

Samba Users

The Samba Server Configuration tool also allows you to configure Samba users, based on the users already present in your `/etc/passwd` configuration file. Unfortunately, it can only use local password databases as of this writing. However, that's good enough to configure Microsoft usernames on this computer. To add Samba users from the Samba Server Configuration tool, click Preferences | Samba Users. This opens the Samba Users window shown in Figure 8-8.

As you can see, this window includes a list of currently configured Samba users. Click Add User. This opens the Create New Samba User window shown in Figure 8-9, where you can:

- Select an existing username from `/etc/passwd`.
- Enter the corresponding Microsoft Windows username.
- Set up a password for that Samba user. It can be different from that user's Linux password.

FIGURE 8-7

Basic components
of Create Samba
Share

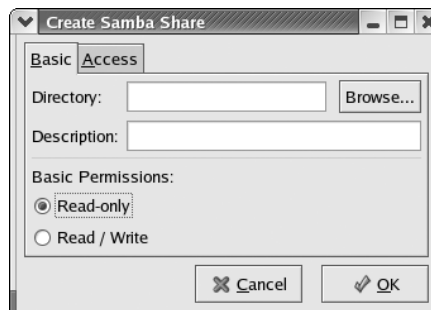
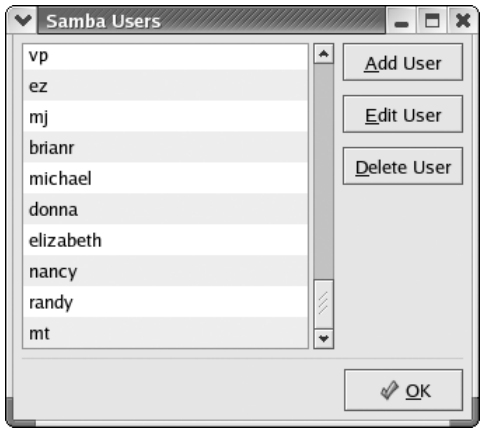


FIGURE 8-8

Current
Samba users



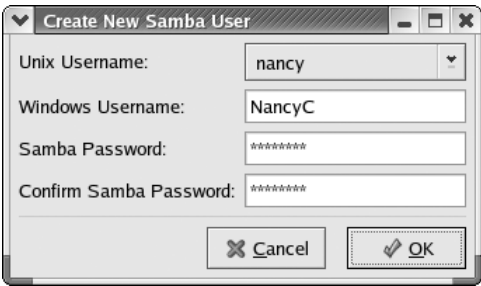
Click OK when you're done. Naturally, you can also change the Windows username and password for each Samba user, or even delete Samba users with the Edit User and Delete User buttons. Click OK to exit from the Samba Users window.

Creating a Public Share

Now let's create a public access share for use with the entire network. First, create the /home/PublicShare directory for this purpose. Click Add to open the Create Samba Share window. Enter the directory that you want to share, /home/PublicShare, in the directory text box. Enter an appropriate description, and select Read/Write access. In the Access tab, select the Allow Access To Everyone option. Click OK, and exit from the Samba Server Configuration tool with the File | Quit command.

FIGURE 8-9

Adding another
Samba user



Now, you'll have to finish the task directly from the text editor. The instructions so far add the following commands in the `/etc/samba/smb.conf` configuration file:

```
[PublicShare]
    comment= Shared Public Directory
    path = /home/PublicShare
    writeable = yes
    guest ok = yes
```

Now you want to modify these commands. The `[PublicShare]` should be accessible to all users with a Linux account on your computer. It should also deny access to guest users and others. You want to provide access to anyone in your domain (`.myCompany.com`), and you want to deny access to everyone in the suspect domain (which we'll call `evil.crackers.com`). Finally, your shares should be browseable to valid users.

You'll want to change the last command in this stanza. As **guest ok = no** is the default, you can just erase the command, or replace it as shown:

```
guest ok = no
```

To provide access to all users in the given domain, I'd add the following command:

```
hosts allow = .myCompany.com
```

If you wanted to deny access to one specific computer on that network, you could add **EXCEPT** `guest.myCompany.com` to the end of this command. Alternatively, if this domain is on the `192.168.99.0` network, you could use one of the following commands:

```
hosts allow = 192.168.99.
hosts allow = 192.168.99.0/255.255.255.0
```

exam

watch

The digit '1' in front of the 777 directory permission string is known as the "sticky bit." By enabling the sticky bit, you are saying that anyone can do anything in the directory (because

of the 777 permission value) but only to files they make! Otherwise, any user could delete or rename any file in your PublicShare, regardless of the file's owner.

You could specifically deny access to computers with a command such as the following:

```
hosts deny = evil.crackers.com
```

Or you could substitute IP addresses in the same format as with the **hosts allow** command. You've defined the share attributes in the Samba `smb.conf` configuration file. But you need to modify the directory associated with the share with the following command:

```
# chmod 1777 /home/PublicShare
```

Testing Changes to `/etc/samba/smb.conf`

After making any changes to `/etc/samba/smb.conf`, it is always a good idea to test your system before putting it into production. You can do a simple syntax check on the Samba configuration file with the **testparm** test utility as shown in Figure 8-10. This does not actually check to see if the service is running or functioning correctly, it checks only basic text syntax and command stanzas.

FIGURE 8-10

Testing `smb.conf`
syntax

```
[root@Enterprise3 root]# testparm | more
Processing section "[homes]"
Processing section "[printers]"
Processing section "[tmp]"
Processing section "[samba]"
Processing section "[PublicShare]"
Load smb config files from /etc/samba/smb.conf
Loaded services file OK.
Server role: ROLE_DOMAIN_MEMBER
Press enter to see a dump of your service definitions
# Global parameters
[global]
    workgroup = GRATEFUL
    server string = samba server
    security = DOMAIN
    password server = 192.168.30.4
    log file = /var/log/samba/%m.log
    max log size = 50
    socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
    dns proxy = No
    printing = cups

[homes]
--More--
```

EXERCISE 8-2**Configuring Samba with Shares**

In this exercise, you'll be configuring Samba to do something useful, sharing a directory and any configured printers. For this purpose, you can't do anything with the Samba Server Configuration tool; therefore, you'll need to edit the `/etc/samba/smb.conf` file directly in a text editor.

1. Install the Samba RPMs or Windows File Server Package Group as described earlier in this chapter.
2. Create a `/home/ftp/public` directory. Change ownership to the `ftp` user and group, with full permissions (770).
3. Open the `/etc/samba/smb.conf` file in a text editor.
4. Configure Samba to share all installed print queues to all users. Normally the default `[printers]` stanza in `smb.conf` should suffice. However, as you're about to create a guest account, you'll want to add the last command shown here:

```
[printers]
    comment = All printers
    path = /var/spool/samba
    browseable = no
    printable = yes
```

guest ok = yes

5. Configure Samba to share as public, in read-only mode, the `/home/ftp/pub` directory tree. In the Share Definitions section, you could add the following commands:

```
[pub]
    comment = shared FTP directory
    path = /home/ftp/pub
```

6. Allow guest access to all public shares. In `smb.conf`, this means adding the following line to the `[pub]` stanza:

```
    guest ok = yes
```

7. Since you're about to create a guest account, you'll need to activate the following command in `smb.conf`:

```
; guest account = pcguest
```

8. Create a guest account for `pcguest`, associate it with an unused UID and GID 600. Set the password to be “anonymous.” While you can do this with the Red Hat User Manager discussed in Chapter 4, the quickest way to do this is with the following commands:

```
# useradd pcguest -u 600
# passwd pcguest
```

9. Create separate log files for each computer host that connects. This is already active by default with the following command:

```
log file = /var/log/samba/%m.log
```

10. If you have a WINS server configured, authorize Samba to participate as a WINS client. Use the IP of that WINS server. While you don’t want to enable the **wins support** command, you do want to point the **wins server** command to the IP address of your WINS server (I’ve added a random IP address):

```
wins server = 10.11.12.13
```

11. Write and save your changes to the `smb.conf` file.
12. You can see if Samba is already running with the **service smb status** command. If it’s stopped, you can start it with the **service smb start** command. If it’s running, you can make Samba reread your configuration file with the following command:

```
# service smb reload
```

This final option allows you to change your Samba configuration without disconnecting users from your Samba server.

CERTIFICATION OBJECTIVE 8.02

Print Services

RHEL 3 comes with one print service, the Common Unix Printing System (CUPS). It’s the successor to the Line Print Daemon (LPD), which is no longer offered with

RHEL 3 or Fedora Linux. However, Red Hat includes software that allows you to use a number of LPD commands to monitor and maintain printers.

You can configure printers directly through the CUPS configuration files in the `/etc/cups` directory. Alternatively, RHEL 3 includes two quality GUI tools that you can use to configure local and remote printers on your network. One is a Web-based interface, and Red Hat is focusing its efforts on its own Printer Configuration utility.

CUPS is the Linux/Unix implementation of the Internet Print Protocol (IPP). I expect IPP to become a fairly universal standard for printer configuration sometime in the future, even for Microsoft and Apple computers.

Installing and Starting CUPS

CUPS and a number of print databases are installed with the Printing Support package group. It includes ten RPM packages which appear unrelated. If you haven't already installed this package group during the RHEL 3 installation process, it's likely most efficient to install it using the Red Hat Package Management utility described in Chapter 4. If you want to learn to install the packages on your own, you can review the packages in the Package Management utility or the `/usr/share/comps/i386/comps.xml` file.

It's easy to start and configure CUPS to start when Linux boots on your computer. The `cups` service script works like most of the other services on RHEL 3. In other words, you can start it with the following command:

```
# service cups start
```

Naturally, by substituting **restart**, **stop**, or **reload** for **start**, you can restart or stop the CUPS service, or reload the CUPS configuration files.

CUPS Configuration Files

There are a number of printer configuration files in the `/etc/cups` directory. Each of these files include a substantial number of commands and comments. I'm going to summarize some key commands in these files, which are listed in Table 8-2.

While I still believe that it's best in general to edit configuration files directly, the commands associated with CUPS can be difficult to learn. Several of the files in `/etc/cups` don't include the same quality of comments as other services. And the Red Hat configuration tool provides the functionality that you need—it's considerably better than the functionality of the Samba Server Configuration tool, for example.

TABLE 8-2 CUPS Configuration Files

CUPS Configuration File	Description
/etc/cups/classes.conf	Specifies different classes of printers. A print <i>class</i> can include several printers; when you print to a print class, CUPS sends your print job to the first available printer in that class, as specified here.
/etc/cups/client.conf	Sets the default CUPS server for this computer; it can be local or another remote print server.
/etc/cups/cupsd.conf	The main CUPS configuration file.
/etc/cups/mime.convs	Includes file format filters, such as images and documents.
/etc/cups/mime.types	Sets file types that can be processed through CUPS printers.
/etc/cups/printers.conf	Documents printers configured by the CUPS Web-based tool.
/etc/cups/pstoraster.convs	Includes a conversion filter which supports PostScript printers.
/etc/printcap	Adds a list of printers for sharing; used by Samba.

The Red Hat Printer Configuration Tool

The main Red Hat Printer Configuration tool works well. You can start it with the **redhat-config-printer** command, or you can click Main Menu | System Settings | Printing. Either option opens the utility shown in Figure 8-11. It is a “front end” that can help you configure the files in the `/etc/cups` directory.

You can use this tool to manage printers. It can configure print queues to local ports or through remote systems:

- Local CUPS ports
- Remote Unix/Linux LPD print services
- Shared Windows printers
- Novell NCP print queues
- HP JetDirect print servers

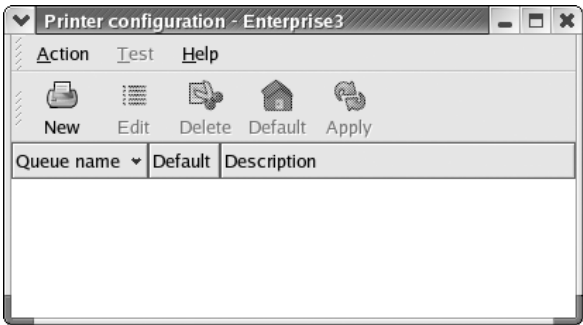


Red Hat has developed a text-mode version of the Printer Configuration tool. You can start it from the text console with the `redhat-config-printer-tui` command.

Previous versions of Red Hat used the **printtool** GUI printer manager and the **printconf-gui** utility. These commands now are “front-ends” which have been

FIGURE 8-11

Red Hat's Printer Configuration utility



reconfigured to start the Printer Configuration tool. You should be able to use this tool to connect to just about any type of printer available on your network.

Now let's try defining a local printer. Click Add. This should automatically start the Add a New Print Queue wizard as shown in Figure 8-12.

Click Forward. In the Queue name screen shown in Figure 8-13, designate a name and comment for the printer. When you share printers over a network, the name and comment can help you identify the printer. The printer name must be one word. Click Next to continue.

If you selected a local printer, the next step is to select a printer port. The normal default is the first printer port, /dev/lp0. This corresponds to LPT1: in the Microsoft world. You might see /dev/usb/lp0 for a printer on a USB port. This is the default option shown in Figure 8-14.

FIGURE 8-12

Starting the printer configuration process

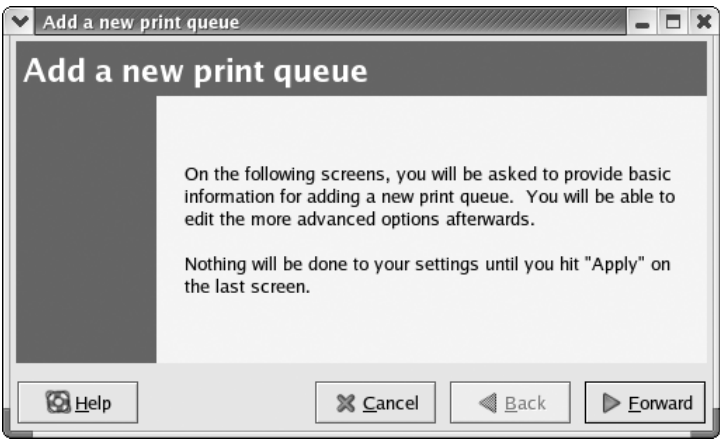
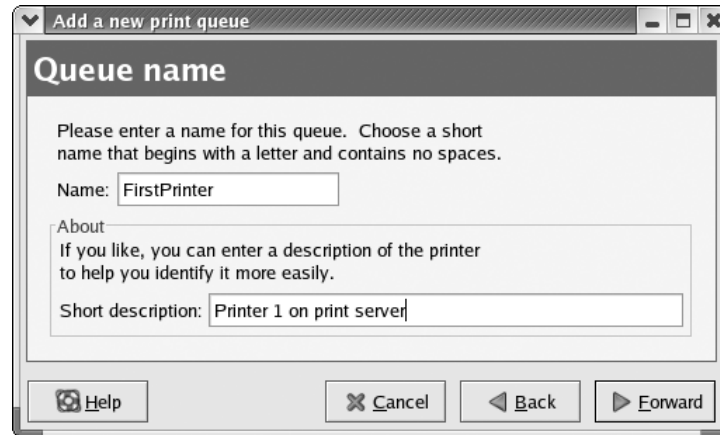


FIGURE 8-13

Naming the printer



Alternatively, if you're configuring a printer over a network, click the Select A Queue Type drop-down text box. You'll get to choose between printers shared on several different types of networks, as described at the start of this section. If you select a network option, the window changes to allow you to enter network information. For example, if you select a Networked CUPS (IPP) printer, the window changes to Figure 8-15.

In this setup, you'll need the hostname or IP address of the print server, the name of the printer, and any required usernames and passwords to authorize access to that network printer. Once you've identified the printer location, click Forward. There

FIGURE 8-14

Selecting the printer connection

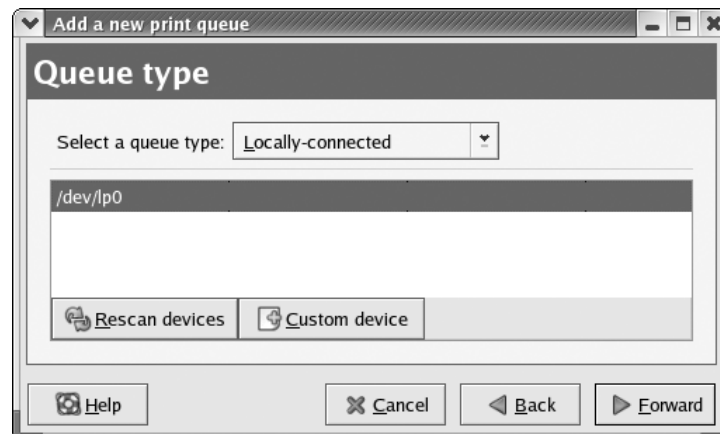
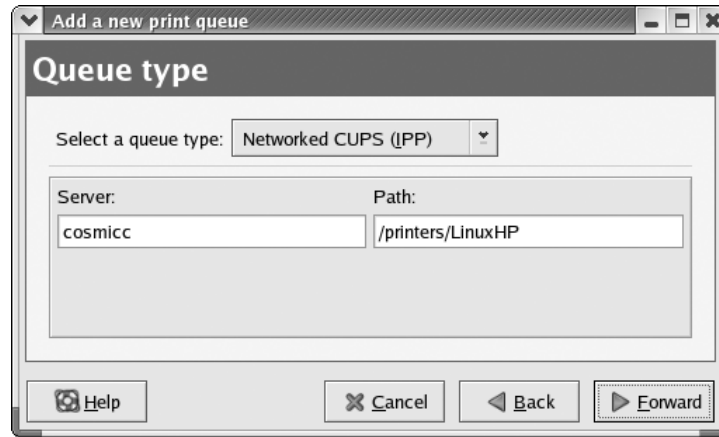


FIGURE 8-15

Configuring a
CUPS network
printer



may be a slight delay as the tool looks through the network. Now you can select a printer manufacturer and driver, as shown in Figure 8-16.

There are two steps here. First, click the drop-down text box where you see Generic (Click To Select Manufacturer). You'll be able to choose between several dozen manufacturers. Once you select a manufacturer, the screen changes to a database of printer models associated with that manufacturer. Select a driver and then click Forward.

Before the Printer Configuration tool creates a print queue, it documents what it's about to do. If you're satisfied with the configuration, click Apply. This returns you to the main Printer Configuration menu shown previously in Figure 8-11, and an offer to print a test page. You have to select Yes to apply changes.

FIGURE 8-16

Selecting a driver

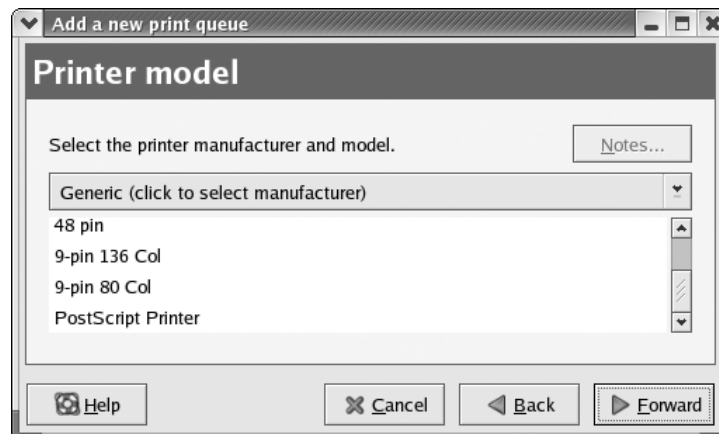
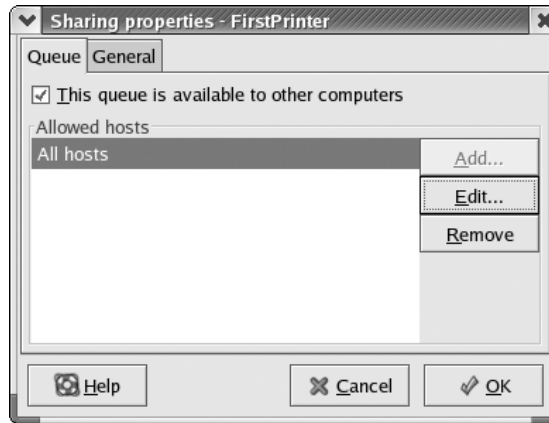


FIGURE 8-17

Sharing a
CUPS printer



The main Printer Configuration menu should now include the name of your printer. To write these changes to the configuration files in the `/etc/cups` directory, as well as `/etc/printcap`, click the Apply button.

If you have trouble here, you may not have shared the CUPS printer at the server. To do so at the server, highlight the name of the printer that you want to share. Right-click on it and select Sharing in the pop-up menu that appears. This opens the Sharing Properties window named for the printer on the server. Under the Queue tab shown in Figure 8-17, activate the “This queue is available to other computers” option. This works with the Printer Configuration tool included with RHEL 3 and Red Hat Linux 9.

The Line Print Daemon Commands

Even though RHEL 3 uses CUPS, you can still control your print system with LPD commands. Three major commands are associated with the `lpd` service: `lpr`, `lpq`, and `lprm`. They are used to add print requests, list queued print requests, and remove print requests, respectively. One more command can help you administer one or more print queues: `lpc`.

lpc: Line Print Control

To view all known queues, run the `lpc status` command; it implements the result shown in Figure 8-18. As you can see, the output helps you easily scan all configured print devices and queues.



The `lpc` command that comes with CUPS does not support starting or stopping of print queues.

FIGURE 8-18

Status of
configured
printers

```
[root@CosmicC root]# lpc status
LinuxHP:
    printer is on device 'parallel' speed -1
    queuing is enabled
    printing is enabled
    1 entries
    daemon present
WinPrint1:
    printer is on device 'parallel' speed -1
    queuing is enabled
    printing is enabled
    4 entries
    daemon present
WinPrint2:
    printer is on device 'parallel' speed -1
    queuing is enabled
    printing is enabled
    no entries
    daemon present
[root@CosmicC root]#
```

lpr: Line Print Request

Any user can use **lpr** to send print requests to any local print queue. You can **lpr** any files to a queue, or you can redirect any output via **lpr**. If you wanted to print to the queue named **color**, you'd use a command such as **lpr -Pcolor filename**. Note there is no space between the **-P** switch and the name of the queue.

lpq: Line Print Query

Note that your printer is no longer printing requests. Now let's queue up a new job. Issue these commands:

```
# lpq
WinPrint1 is ready and printing
Rank      Owner   Job   Files                               Total Size
active    root    373   smbprn.000486.6JkBaq              10240 bytes
1st       root    374   smbprn.000487.6JkBaq              10240 bytes
2nd       root    376   smbprn.000488.6JkBaq              10240 bytes
```

FIGURE 8-19

GNOME Print Manager



Now you can delete the jobs of your choice. It's simple; just use the `lprm` command with the job number:

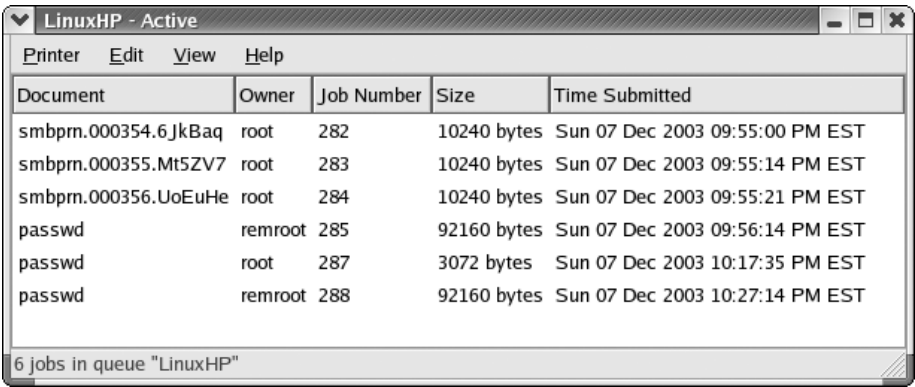
```
# lprm 376
```

GUI Front-Ends

RHEL 3 includes two GUI front-ends on the GNOME desktop. They run one of the commands noted in this section and format it into a graphical interface. To view the printers that you've configured on the GUI, click Main Menu | System Tools | Print Manager. One example is shown in Figure 8-19.

FIGURE 8-20

CUPS Web-based interface



Each of these printers may have a print queue. For example, Figure 8-20 illustrates the result after double-clicking on the LinuxHP print icon. You can review the print jobs waiting in the spool to print on the LinuxHP printer.

The CUPS Web-Based Interface

Another way to configure CUPS is through the Web-based interface. Open up the browser of your choice on the local Linux computer. Direct it to the `http://localhost:631` address to get to the main CUPS configuration menu, as shown in Figure 8-20. There are seven options atop the Administration menu:

- **ESP** Brings you to the Easy Software Products home page at `www.easysw.com`. CUPS is officially one of their products, licensed under the Linux GPL.
- **Administration** A basic interface that allows you to set up classes of printers, manage print jobs, and add or manage printers. You can also administer these features through the Classes, Jobs, or Printers options.
- **Classes** Enables you to set up groups of printers together.
- **Help** Provides extensive documentation.
- **Jobs** Enables you to view and manage active print jobs.
- **Printers** Enables you to manage existing printers and set up new printers.
- **Software** Enables you to review the current version and upgrade if appropriate.

You can manage Classes, Jobs, and Printers from the Administrative menu. You can read the Help section on your own, and software upgrades to CUPS are not covered in this book. The heart of CUPS is in the Administrative menu. If you click on it, you're prompted to enter your administrative root username and password.

One drawback to CUPS is that RHEL 3 does not support the same extensive database of printers as you see in the Red Hat Printer Configuration tool. On the other hand, the Red Hat tool does not support the creation of printer classes.



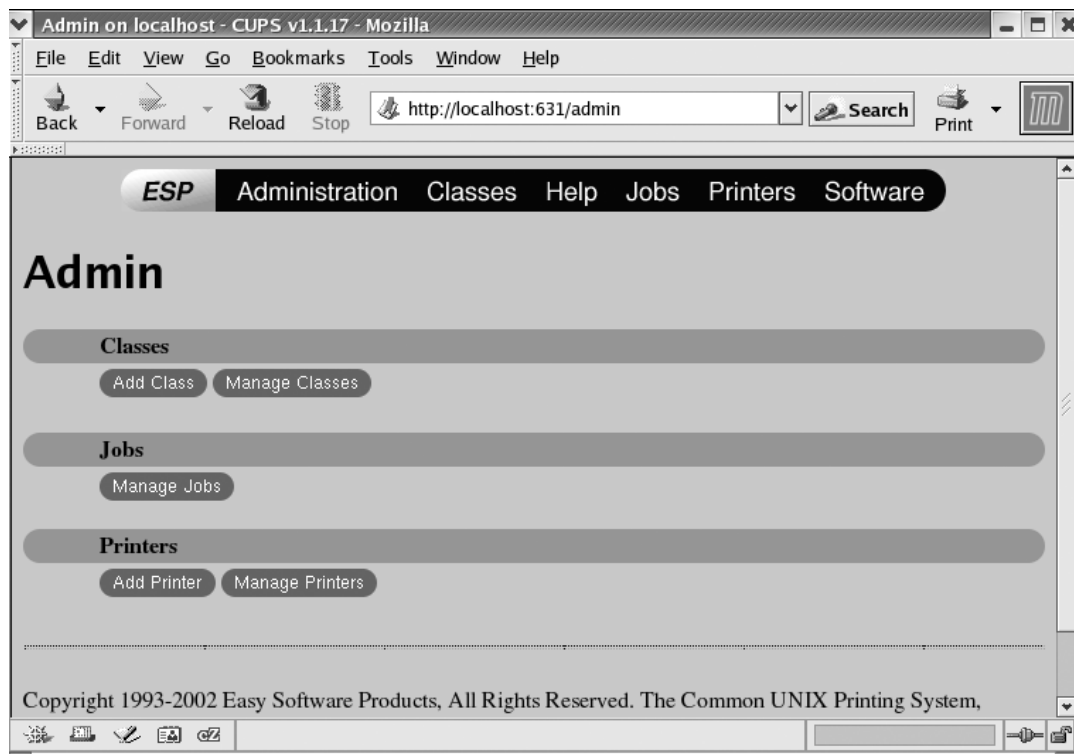
Other Linux distributions don't include Red Hat's Printer Configuration tool. Many do include extensive printer databases in the CUPS Web-based tool.

CUPS Administration

Once you've entered your administrative credentials, you'll see the Admin setup page shown in Figure 8-21. There are three sections here. Classes enable you to configure a group of printers together. When you use a specific class, CUPS directs your print job to the first available printer in this class. Jobs help you manage the print jobs currently in the print queue. Printers allow you to add new printers and manage existing printers.

As the Red Hat Printer Configuration tool is superior, I won't explain the Add Printer menu in this book. However, if you want to create a printer class, the CUPS Web-based tool is the easiest way to go. As described earlier, a CUPS printer class allows you to print to a group of printers. CUPS then selects the first available printer in that class.

FIGURE 8-21 CUPS Admin menu



To proceed, click Add Class. You're taken to a menu where you need to fill in three text boxes, as shown in Figure 8-22:

- **Name** The name of the printer class. If you want to print to this printer group, you'll call or connect to this printer class name.
- **Location** Must be set to the name or IP address of the local print server.
- **Description** Allows you to add the comment of your choice to help identify the printer class.

After you click Continue, you'll get to include one or more configured printers in your printer class. Do so, and click Continue. The CUPS Web-based tool now creates a print class as shown in Figure 8-23.

FIGURE 8-22 Configuring a printer class

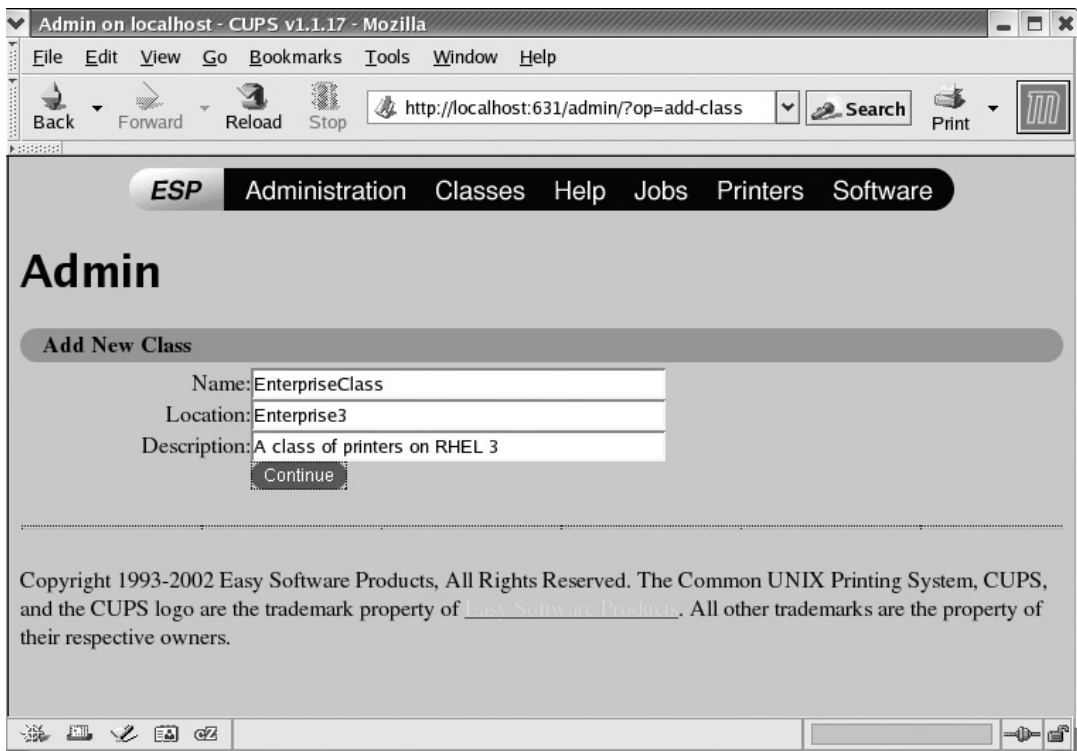
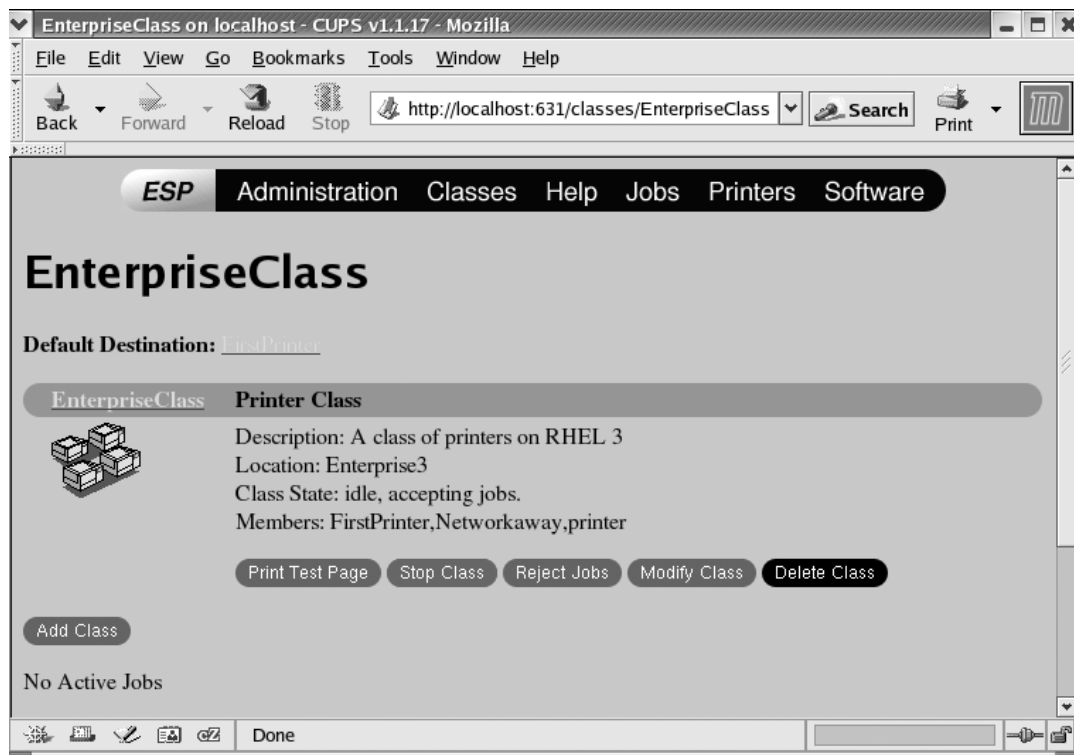


FIGURE 8-23 Your new printer class

Verifying CUPS Sharing

Once you've configured security for a CUPS printer, you can verify it in `/etc/cups/cupsd.conf`. The security commands in this file are fairly straightforward. By default, access to a CUPS printer is limited to the local computer with a stanza such as:

```
<Location /printers/printer>
    Order Deny,Allow
    Deny From All
    Allow From 127.0.0.1
    AuthType None
</Location>
```

If you configure a CUPS printer to be shared with the world, you'll see one more command in the stanza. For example, `printer2` is shared with the world because of the command in bold:

```
<Location /printers/printer>
    Order Deny,Allow
    Deny From All
    Allow From 127.0.0.1
    AuthType None
Allow From All
</Location>
```

Naturally, you can limit access to specific IP address networks. For example, to limit access to the 192.168.30.0 LAN, you'd substitute the following command for **Allow From All**:

```
Allow From 192.168.30.0/255.255.255.0
```

You can also substitute computer or domain names. For example, the following command allows access from the `example.com` domain:

```
Allow From *.example.com
```

You can't configure these limits using the Red Hat Printer Configuration utility (**redhat-config-printer**). If you want to set up host-based security in this way, you'll have to edit the `cupsd.conf` configuration file directly.

CERTIFICATION SUMMARY

A number of important network client services are associated with Red Hat Enterprise Linux 3. Two of them are Samba and CUPS. Samba sets up interoperability between Microsoft Windows and Linux/Unix computers. CUPS allows to set up printers locally or over the network.

Samba allows a Linux computer to appear like any other Microsoft computer on a Microsoft Windows-based network. Samba is based on the Server Message Block protocol, which allows Microsoft computers to communicate on a TCP/IP network. It has evolved as Microsoft has adapted SMB to the Common Internet File System.

The main Samba configuration file, `/etc/samba/smb.conf`, includes separate sections for global settings and share definitions. The Red Hat Samba Server Configuration

tool is a GUI tool that makes it easier to configure `smb.conf`. Changes to `smb.conf` can be easily tested with the **testparm** utility.

Linux includes the Common Unix Printing System (CUPS). With CUPS, the configuration files are stored primarily in the `/etc/cups` directory. However, it also includes a list of printers in `/etc/printcap` to accommodate sharing through Samba. You can edit the CUPS configuration files directly, with Red Hat's Printer Configuration tool or the CUPS Web-based interface.

Because RHCE is a performance-based exam, it is important to practice all the skills discussed in this chapter. You may need to use these skills on the exam!



TWO-MINUTE DRILL

Here are some of the key points from the certification objectives in Chapter 8.

Samba Networking

- ☐ Samba allows Microsoft Windows computers to share files and printers across networks, using the Server Message Block (SMB) protocol on the TCP/IP protocol stack.
- ☐ Samba includes a client and a server. Variations on the **mount -t, smbfs**, or **smbmnt** command allow you to connect to a Microsoft Windows shared directory.
- ☐ The main Samba configuration file is `/etc/samba/smb.conf`. You can configure it in a text editor or a GUI tool such as the Samba Server Configuration tool.
- ☐ Samba allows you to configure your Linux computer as a member of a Microsoft Windows 9x-style Workgroup.
- ☐ Samba allows you to configure your Linux computer as a Microsoft Windows server. It can also provide Microsoft browsing, WINS, and Domain Controller services, even on an Active Directory network.

Print Services

- ☐ The Printer Configuration tool, which you can start with the **redhat-config-printer** command, can be used to configure most popular printers in `/etc/printcap`.
- ☐ CUPS provides a Web-based interface similar to Samba's SWAT. Once enabled, you can get to this interface in your browser by navigating to `http://localhost:631`.

SELF TEST

The following questions will help you measure your understanding of the material presented in this chapter. Read all the choices carefully, as there may be more than one correct answer. Choose all correct answers for each question. Don't focus exclusively on these questions. There are no longer any multiple choice questions on the Red Hat exams. These questions test your understanding of the chapter. Getting results, not memorizing trivia, is what counts on the Red Hat exams.

Samba Networking

1. A group that prefers Microsoft servers has set up a Windows 2000 server to handle their file and print sharing services. This server correctly refers to a WINS server on 192.168.55.3 for name resolution and configures all user logins through the PDC on 192.168.55.8. If you're configuring the PDC, what options do you have to configure in the local Samba configuration file?
 - A. domain master = yes
 - B. wins support = yes
 - C. password server = 192.168.55.3
 - D. security = user
2. Which of the following files or commands is not a component of the Samba file-sharing service?
 - A. /usr/bin/smbd
 - B. /usr/bin/nmbd
 - C. /usr/bin/smbclient
 - D. /etc/samba/smb.conf
3. You made a couple of quick changes to your Samba configuration file and you need to test it quickly for syntax errors. Which command should you run?
 - A. **smbmount**
 - B. **smbclient**
 - C. **smbfs**
 - D. **testparm**
4. You are asked to share the HR downloadable documents to Microsoft Windows users who are not that familiar with FTP and want a shared drive connection. Which of the following commands forces the Samba service to reread the configuration file—without having to disconnect your Microsoft users or restarting the service?
 - A. testparm
 - B. /etc/smb.conf

- C. `/etc/rc.d/init.d/smb reload`
 - D. `service smb restart`
5. Microsoft Windows users are complaining that they cannot see the shared Human Resources directory of documents, `[hrdocs]`, in their Network Neighborhood diagram. Which of the following options should you delete from the `[hrdocs]` stanza in `smb.conf`?
- A. `read only = yes`
 - B. `browseable = yes`
 - C. `guest ok = yes`
 - D. `printable = yes`
6. Which of the following files can you configure with **mount** or **smbmnt** style commands on a user-by-user basis?
- A. `/home/.bashrc`
 - B. `~/.bash_logout`
 - C. `/etc/fstab`
 - D. `/boot/.bashrc`
7. Based on the following code, which of the following things do you need to do to limit access to the following `[tmp]` share in `/etc/samba/smb.conf` to the local network, with a network IP address of 192.168.99.0?

```
# This one is useful for people to share files
[tmp]
    comment = Temporary file space
    path = /tmp
    read only = no
    public = yes
```

- A. A firewall on any connections to outside networks which blocks at least Samba ports.
- B. Add the following command to the stanza:
`local only = yes`
- C. Add the following command to the stanza:
`hosts allow = 192.168.99.`
- D. Make sure the **password server** command points to the PDC on your network

Print Services

8. You want to look at your current printer configuration in a GUI desktop interface. What command might you use to start a GUI printer configuration tool?
 - A. redhat-printer-config
 - B. redhat-config-printer
 - C. printconf-gui
 - D. lprsetup
9. You have a group of printers that you want to configure as a printer class. Which of the following tools can you use?
 - A. Direct editing of `/etc/cups/classes.conf`
 - B. Red Hat's Printer Configuration tool
 - C. The CUPS Web-based tool
 - D. Direct editing of `/etc/printcap`
10. Which of the following commands can you use to remove a print job with a known job number from the print queue?
 - A. `lpr`
 - B. `lpc`
 - C. `lpq`
 - D. `lprm`

LAB QUESTIONS

Lab 1: Configuring Samba

Part 1: Installing and Starting Samba

1. Ensure that all four components of the Samba service are correctly installed. What RPMs did you install and how did you install them?
2. Use one of the service management tools available to you to ensure that the Samba services are configured to start correctly when you boot Linux. What tool did you use?
3. Start Samba services now. You can use either the service management script located directly in `/etc/rc.d/init.d`, or you can use the “service” startup tool. How did you start your Samba service?
4. Verify that Samba services are now running. How did you do this?

Part 2: Configuring Samba's Global Settings

1. You'll use Red Hat's Samba Server Configuration tool to configure your Samba service. Start this tool. If you didn't log in as the root user, did something happen before the tool started?
2. Configure the Samba global settings. You will provide Workgroup services to your users. Set the Workgroup name to something appropriate for your company.
3. Can you limit access to your company's domain name (such as example.com) through this tool? If you have to edit the Samba configuration file directly, what do you have to do?
4. Can you prevent access to evil.cracker.com through this tool? If you have to edit the Samba configuration file directly, what do you have to do?
5. Commit your changes. What do you need to do to make Samba reread the configuration file?

Part 3: Configuring File Shares

1. Open the main Samba configuration file.
2. Navigate to the predefined [homes] share.
3. Ensure that the [homes] share is available only to hosts on your example.com network.
4. Ensure that the share is writable to authenticated users but not available to guest users.
5. Commit your changes.
6. Create a new share called [public].
7. Change the path to the public share to /home/public.
8. Configure the public share so anyone in your domain can access the share.
9. Create the /home/public directory as required. Change the permissions to this directory to 1777.
10. Why do you set permissions to 1777?
11. Commit your changes.

Part 4: Setting Up Printer Shares

1. Your Linux server has many printers defined. You want to offer access to them to your desktop client users. Enable access to the generic printers share now.
2. Again, restrict access to your print shares to members of your example.com domain.
3. Commit your changes.

Part 5: Verifying the smb.conf File

1. You want to verify your changes. Start a terminal window. Run the syntax tester tool on your Samba configuration tool. What program did you use?
2. Review the `/etc/samba/smb.conf` file. Look over each section including the `[globals]` section. Ensure that all updates are correct and reflect the requirements previously stated. Go back and make changes, if necessary. Commit all changes.
3. Again, go back and make revisions if the test program indicates problems with the `smb.conf` file.

Part 6: Starting the Samba Servers

1. Navigate to a command line interface.
2. Start the Samba server. Which daemons does it start?
3. If possible, go to a Microsoft Windows computer on your network. Use a Microsoft browsing tool such as Network Neighborhood or My Network Places in Windows Explorer. See if you can connect to the Samba public share. Alternatively, you can go to another Linux computer, and browse Nautilus by navigating to `smb:///`.
4. Congratulations! You have just configured your Samba server to share files with your local workgroup.

Part 7: Persistency Check It is important for your server (and critical to pass the RHCE exam) that any changes you make to your server should be persistent. This means that changes should be active when you reboot Linux. Perform an orderly reboot of your server now and verify that Samba starts when you boot Linux.

1. How did you make your changes persistent?
2. What command did you use to perform an orderly shutdown?

Lab 2: Creating a Printer

In this lab, you'll want to use the Red Hat Printer Configuration utility to connect a printer to your Linux system. The printer can be local or remotely connected through your LAN. As you'll want to create a printer class, you'll need more than one printer.

If you have only one physical printer, you can set up multiple print queues with different printer names. CUPS sees each print queue as if it were a separate printer.

Once you've created multiple printers, open a GUI Web browser. Navigate to <http://localhost:631>. Click the Class link and then click Add Class. Follow the prompts to create a printer class with the printers that you've configured.

Once you've created a new printer class, inspect the result in the `/etc/cups/classes.conf` configuration file. Check the contents of your `/etc/printcap` file. What are the names of the printers that you see? Are there any surprises in the list?

If you have a Microsoft Windows computer on your network, activate the Samba service if required. Check the printer names as shown in the browse list. Are there any surprises on this list?

SELF TEST ANSWERS

Samba Networking

1. ☒ **D**. If you're configuring a PDC, you'll want the **security = user** command in the local `/etc/samba/smb.conf` file.
☒ **A** is incorrect because being a Browse Master is not a requirement for a PDC. **B** is incorrect because the WINS server is on a different computer. **C** is incorrect because the local computer is the PDC, which keeps the password database.
2. ☒ **C**. This is a tricky question. **smbclient** is a client command; it is not part of the service that provides Samba file shares.
☒ **A**, **B**, and **D** are all components that are either daemons (**smbd** and **nmbd**) that manage the service or are the server's configuration file (`smb.conf`).
3. ☒ **D**. **testparm** is the `smb.conf` configuration file syntax and semantics checker.
☒ **A** is wrong because **mount -t smbfs** is used by the root user to make remote SMB shares available to Linux. It wouldn't test your local configuration. **B** is wrong because **smbclient** is the Linux SMB client program for accessing Samba services on the network. **C** is incorrect because `smbfs` is the filesystem type used when a Samba share is mounted through the **mount** command or via `/etc/fstab`.
4. ☒ **C**. This is one way to tell Samba to reload the configuration file without restarting. This is a little tricky, as I've been using the **service** command instead of the `/etc/rc.d/init.d` directory path.
☒ **A** is incorrect because **testparm** is just a syntax checker for `smb.conf`. **B** is incorrect because `smb.conf` is just the Samba configuration file (albeit in the wrong location; for RHEL 3, it should be `/etc/samba/smb.conf`). Finally, **D** is incorrect because it disconnects users and restarts the service.
5. ☒ **B**. A share must be marked as browseable before it will show up in a Microsoft tool such as Network Neighborhood or My Network Places.
☒ **A**, **C**, and **D** are all incorrect. While you probably should make a share that contains Human Resource documents read-only, it's not required to make the share browseable. Making a share accessible to authorized users is not related to whether guests are allowed to read the share. Making the share printable does not affect whether the share is browseable.
6. ☒ **B**. The `.bash_logout` file in users' home directories (`~`) is a good place to include **umount** and **smbumount** commands.
☒ **A**, **C**, and **D** are all incorrect. There is no `.bashrc` file in the `/home` directory, at least not for individual users. The `/etc/fstab` file is a good place to configure mounted directories for all

users simultaneously. The default `/boot` directory does not include a `.bashrc` file and is an inappropriate location for such.

7. ☒ **A** and **C**. A firewall can effectively block communication to a Samba server from an outside network. And the given **hosts allow** command limits access to the given network; no additional **hosts deny** command is required.
☒ **B** and **D** are both incorrect. There is no **local only** command in `smb.conf`. And the location of the PDC does not affect the security of your Samba server.

Print Services

8. ☒ **B** and **C**. The **redhat-config-printer** and **printconf-gui** commands, when run in the GUI desktop, start the Red Hat Printer Configuration tool.
☒ **A** is wrong because there is no such command as **redhat-printer-config**. **D** is wrong because there is no such file as `lprsetup`.
9. ☒ **A** and **D**; or **C**. If you know the language associated with CUPS configuration files, you can set up a printer class by directly editing the `/etc/cups/classes.conf` file. You'll need to add the name of the printer class to the `/etc/printcap` list. Alternatively, you can configure a printer class through the CUPS Web-based tool.
☒ **B** is incorrect because, as of this writing, Red Hat's Printer Configuration Tool does not support the creation of a print class.
10. ☒ **D**. The **lprm** command can be used to remove a print job with a known job number from the print queue.
☒ **A**, **B**, and **C** are all incorrect. The **lpr** command prints a job. The **lpc** command checks the status of your printers. The **lpq** command checks the status of the default printer.

LAB ANSWERS

Lab I

The first chapter lab on Samba is designed to be easy to follow. However, you'll need explicit Linux knowledge to complete some specific steps. Answers to these steps can be found in the following:

Part I

Step 1: You've installed the Windows File Manager package group, which includes the `samba-client`, `samba`, and `redhat-config-samba` RPMs. These RPMs depend on the `samba-common` RPM, which you'll also need to install.

Step 2: You can use the `chkconfig --level 35 smb on` command or the Service Configuration utility described in Chapter 4 to make sure Samba starts the next time you boot Linux.

Step 3: Use the `service smb start` command to begin the Samba service.

Step 4: One way to verify Samba is to look for the existence of the `smbd` and `nmbd` processes in the process table. Use `ps aux | grep mbd` to see if these processes are present. Another way is with a service command such as `service smb status`.

Part 2

Step 1: To use the Samba Server Configuration tool, you'll need the root password.

Step 2: Many administrators stick with the default Microsoft Windows workgroup name of `WORKGROUP`. You can find it in the output from the `smbclient -L //clientname` command.

Step 3: If you want to limit access to your Samba server, you can't do it through the Samba Server Configuration tool. You'll want to set up the `hosts allow` command in `/etc/samba/smb.conf`.

Step 4: If you want to restrict access from a specific computer to your Samba server, you can't do it through the Samba Server Configuration tool. You'll want to set up the `hosts deny = evil crackers.com` command in `/etc/samba/smb.conf`.

Step 5: When you exit the Samba Server Configurator, or save the `smb.conf` file, you can make Samba read the changes with the `service smb reload` command. But before committing the changes, you should test them with the `testparm` command.

Part 3

Step 1: Open the main Samba configuration file, `/etc/samba/smb.conf`, in a text editor.

Step 2: Navigate to the `[homes]` share in the last half of this file.

Step 3: Unless there is a limitation in the Globals part of this file, you can limit the `[homes]` share with the `hosts allow = example.com`. Commit your changes. Restart or reload `smb` under the Status menu or with the appropriate `service` command.

Step 4: Add a `guest ok = no` to the `[homes]` stanza.

Step 5: Save the changes you've made so far.

Step 6: At the end of the file, start a `[public]` stanza. Add an appropriate comment for the stanza.

Step 7: Set `path = /home/public`. Save your changes to the `smb.conf` file.

Step 8: Make sure to set `hosts allow = example.com`.

Step 9: Set permissions for the public share, with the following commands:

```
# mkdir /home/public
# chmod 1777 /home/public
```

Create a new directory, `/home/public`; configure that share in SWAT and call it `public`. Set the `hosts allow` setting in SWAT under the Shares menu, and list the domain associated with your network. Deny to all others.

Step 10: The 777 aspect of permissions grants read, write, and execute/search permissions to all users (root, root's group, and everyone else). The 1 at the beginning of the permission value sets the sticky bit. This bit, when set on directories, restricts users from deleting or renaming files they don't own.

Step 11: Commit your changes with the **service smb reload** command.

Part 4

Step 1: Open your `/etc/samba/smb.conf` configuration file. Navigate to the `[printers]` stanza. The default version of this stanza should already enable access to all users who connect.

Step 2: In this stanza, use the **hosts allow** command as before.

Step 3: Commit your changes by closing and saving the `smb.conf` file, and then running the **service smb reload** command.

Part 5

Step 1: You can use the Samba syntax checker, **testparm**, to make sure there are no glaring problems to your Samba configuration file.

Step 2: This is more of an exercise; if you don't have any problems, you might want to deliberately add some to your `smb.conf` file, and rerun **testparm** again. It's helpful to be familiar with different kinds of Samba syntax issues. Don't forget to restore a working version of the `smb.conf` file!

Part 6

Step 1: Again, this is more of an exercise than a lab. The steps are generally self-explanatory. If your Samba configuration is successful, you should be able to review browseable shares from a Microsoft Windows computer on the same LAN.

Part 7

Step 1: To complete many Linux configuration changes, you need to make sure that the service will start automatically when you reboot your computer. In general, the key command is **chkconfig**. In this case, the **chkconfig --level 35 smb on** command sets up the **smbd** daemon to become active when you boot Linux into either runlevel 3 or runlevel 5.

Step 2: The command you should always use to perform an orderly shutdown is **shutdown -r now**.

Lab 2

Starting the Printer Configuration utility is easy. One way in the GUI is to click Main Menu | System Settings | Printing. Then you can click the Add button to start what is known in the Microsoft world as a configuration wizard. If you like, you can create two different print queues with the same printer. Just repeat the same process, using a different printer name.

Open the Web browser of your choice. The default is Mozilla, which you can start in the GNOME desktop by clicking the globe icon adjacent to the Main Menu icon. You can then navigate to `http://localhost:631`.

Once the CUPS Web-based tool opens, you can click the Classes or the Manage Printer Classes link. Click Add Class in the Class Web page. Enter the root username and password when prompted.

Enter the single-word name of your choice for the printer class name. Add the name or IP address of the computer in the Location text box. Enter the comment of your choice in the Description text box. Click Continue; select more than one computer as members of the printer class. Click Continue; once your new printer class is confirmed, you can close the browser.

You should now see the printer class and member printers in the `/etc/cups/classes.conf` file. You'll find a list of printers in `/etc/printcap`; you'll find the names of any configured printers *and* printer classes in this file. You should also see the list of printers in `/etc/printcap` in any Microsoft Windows Network Neighborhoods or My Network Places that is connected to the same network. This assumes that you've activated a Samba server on the local print server computer, of course.