CHAPTER 24

CIFS/9000 and Samba

CIFS/9000 Overview

CIFS/9000 Server is an HP product based on Samba that provides file and print services to CIFS clients. The server is an HP 9000 and the clients are usually Windows-based systems, however, the client could also be an HP 9000 or any system running CIFS client software. CIFS/9000 Client is used to access file and print services on a CIFS server. To an HP-UX user running CIFS/9000 Client software, the shares on CIFS/9000 Server look like UNIX filesystems whether they are on a UNIX server or a Windows server. The server could be an HP 9000 running CIFS/9000 Server, a Windows-based system, or any system running CIFS as a server. In this chapter, I'll cover only CIFS/9000 Server being accessed by a Windows-based system since that is the use for which most HP 9000 users will deploy CIFS/9000.

CIFS stands for Common Internet File System, which is a Windows specification for remote file access. CIFS is a remote access protocol based largely on Server Message Block (SMB) that has long been used as the native file-sharing protocol on Windows-based systems. CIFS is a remote file access protocol that sits on top of the host file system(s). As mentioned earlier there is both a CIFS Server, for which I'll use and HP 9000 system in this chapter, as well a CIFS Client, for which I'll use a Windows-based system in this chapter. We'll setup CIFS/9000 server such that an HP-UX file system can be mounted on a Windows-based system.

CIFS/9000 Server is based on Samba open source software. For that reason, the second half of this chapter is from one of my other books covering Samba running on a Linux system. The HP 9000 setup of CIFS/9000 Server covered in the first half of this chapter is very similar to the Samba setup on a Linux system covered in the second half of this chapter.

My goal in the first half of this chapter is to give an overview of the setup of CIFS/9000 and demonstrate a subset of its functionality. We'll focus on the file server functionality only. We'll set up shares on the CIFS/9000 server system that will appear as a drive letters and icons on a Windows system.

CIFS/9000 Server provides this file-sharing functionality using Server Message Block (SMB) protocol as described earlier. SMB runs on top of TCP/IP. In our example in this chapter, both the Windows system and CIFS/9000 Server system are running TCP/IP and SMB. These provide all of the technology that is required to establish file sharing between the two systems.

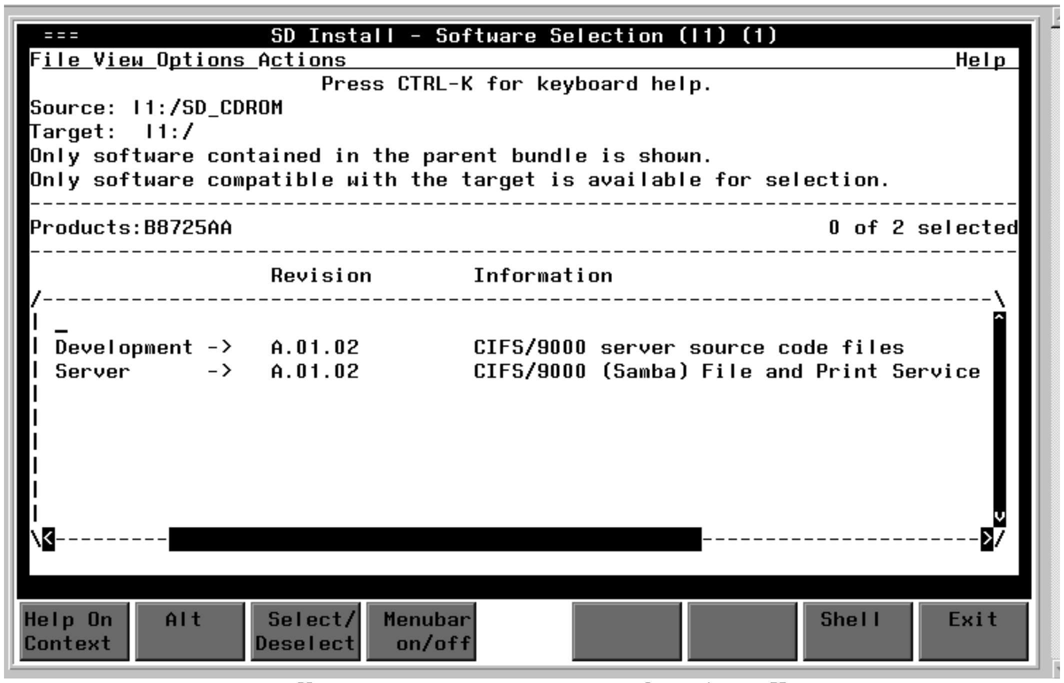
Chapter 26 covers the Network File System (NFS) running on a Windows system and accessing files on a UNIX system. This functionality is similar to that which we'll cover with CIFS/9000 in this chapter. In addition to the file-sharing capability, CIFS/9000 also provides printer sharing and additional user access control capability. We won't focus on these capabilities in this chapter, however, CIFS/9000 does indeed provide some advanced functionality in these areas.

At the time of this writing, CIFS/9000 contains the functinality just mentioned, including: file-sharing; printer sharing; and advanced user access control of files. There are many advancements taking place with Samba that will be incorporated into CIFS/9000 and other software provided under the GNU Public License (GPL) as free software. Because the software is free, many individuals have access to it and spend time enhancing the software. For this reason, you may find that additional functionality is included in CIFS/9000 in the future.

Installing CIFS/9000

man page
"sw" - 2

CIFS/9000 is loaded from CD-ROM like any other HP application. Figure 24-1 shows the **swintall** screen with the two CIFS/9000 components that are part of the parent bundle. As shown in the figure, you receive both the source code and the compiled version of CIFS/9000 Server as one bundled product. The same would be true of the CIFS/9000 Client software. The source code is included because CIFS/9000 Server is based on Samba Open Source software for which source code is freely available.



```
 ===                  SD Install - Software Selection (l1) (1)
File View Options Actions                                            Help
                       Press CTRL-K for keyboard help.
Source: l1:/SD_CDROM
Target:  l1:/
Only software contained in the parent bundle is shown.
Only software compatible with the target is available for selection.
------------------------------------------------------------------------------
Products:B8725AA                                              0 of 2 selected
------------------------------------------------------------------------------
                   Revision        Information

| Development ->   A.01.02         CIFS/9000 server source code files
| Server      ->   A.01.02         CIFS/9000 (Samba) File and Print Service
```

Figure 24-1 Installing CIFS/9000 Server with **swinstall**

After having installed both components of the parent bundle shown in Figure 24-1, we have all of the software present on our system to configure CIFS/9000 Server. Although I don't plan on modifying the source code, I may view it at some point, so I loaded it as well.

Let's next change directory to **/opt** (for optional software) and see what directories have been created in the following listing:

```
# cd /opt
# ls -l
total 10
dr-xr-xr-x   6 bin        bin            96 Jul 11 19:27 audio
dr-xr-xr-x  10 bin        bin          1024 Jul 11 19:27 dce
dr-xr-xr-x   4 bin        bin          1024 Jul 11 19:44 dcelocal
dr-xr-xr-x   3 bin        bin            96 Jul 11 19:25 fc
dr-xr-xr-x   3 bin        bin            96 Jul 11 19:18 fcms
dr-xr-xr-x   4 bin        bin            96 Jul 11 19:44 graphics
drwxr-xr-x   5 root       sys            96 Jul 11 19:29 hparray
dr-xr-xr-x   4 bin        bin            96 Jul 11 19:27 ifor
dr-xr-xr-x  16 bin        bin          1024 Jul 12 13:10 ignite
dr-xr-xr-x   3 bin        bin            96 Jul 11 19:26 image
drwxr-xr-x   2 root       root           96 Jul 11 19:14 lost+found
dr-xr-xr-x   7 bin        bin            96 Jul 11 19:25 nettladm
drwxr-xr-x   2 root       sys          1024 Jul 11 22:57 networkdocs
dr-xr-xr-x   7 bin        bin            96 Jul 11 19:27 pd
drwxr-xr-x   8 root       users        1024 Jul 18 11:53 samba
drwxr-xr-x   3 root       users          96 Jul 18 11:53 samba_src
dr-xr-xr-x   5 bin        bin            96 Jul 11 19:25 upgrade
dr-xr-xr-x   4 bin        bin            96 Jul 11 19:27 video
drwxr-xr-x   3 root       sys            96 Jul 12 13:10 webadmin
#
```

This listing shows that there are both the **samba** and **samba_src** directories that we loaded in **/opt**. The CIFS/9000 product with which we'll be working is loaded in the **samba** directory, so let's change to it and then see what files and directories are present under **/opt/samba** and **/opt/samba/bin**:

```
# ls -l /opt/samba
total 44
-rwxr-xr-x   1 root       users       17982 Jul  7  1999 COPYING
drwxr-xr-x   4 root       users          96 Jul 18 11:53 HA
-rwxr-xr-x   1 root       users         772 Jan 10  2000 README
drwxr-xr-x   2 root       users        1024 Jul 18 11:53 bin
drwxr-xr-x   7 root       users        1024 Jul 18 11:53 docs
drwxr-xr-x   6 root       users          96 Jul 18 11:53 man
drwxr-xr-x   2 root       users        1024 Jul 18 11:53 script
drwxr-xr-x   6 root       users          96 Jul 18 11:53 swat
```

```
# ls -l /opt/samba/bin
total 13588
-rwxr-xr-x   1 root       users      1708 Apr 20 17:38 addtosmbpass
-rwxr-xr-x   1 root       users       446 Apr 20 17:38 convert_smbpasswd
-rwxr-xr-x   1 root       users    292124 Apr 20 17:38 make_printerdef
-rwxr-xr-x   1 root       users    292115 Apr 20 17:38 make_smbcodepage
-rwxr-xr-x   1 root       users    579513 Apr 20 17:38 nmbd
-rwxr-xr-x   1 root       users    402933 Apr 20 17:38 nmblookup
-rwxr-xr-x   1 root       users    752117 Apr 20 17:38 rpcclient
-rwxr-xr-x   1 root       users     12988 Apr 20 17:38 samba_setup
-rwxr-xr-x   1 root       users    476863 Apr 20 17:38 smbclient
-rwxr-xr-x   1 root       users   1302921 Apr 20 17:38 smbd
-rwxr-xr-x   1 root       users    707074 Apr 20 17:38 smbpasswd
-rwxr-xr-x   1 root       users    402929 Apr 20 17:38 smbspool
-rwxr-xr-x   1 root       users    324986 Apr 20 17:38 smbstatus
-rwxr-xr-x   1 root       users      4862 Apr 20 17:38 smbtar
-rwxr-xr-x   1 root       users      1860 Apr 20 17:38 startsmb
-rwxr-xr-x   1 root       users      2340 Apr 20 17:38 stopsmb
-rwxr-xr-x   1 root       users    789251 Apr 20 17:38 swat
-rwxr-xr-x   1 root       users    288012 Apr 20 17:38 testparm
-rwxr-xr-x   1 root       users    312629 Apr 20 17:38 testprns
# export PATH=$PATH:/opt/samba/bin
#
```

There are some files in **/opt/samba/bin** of interest. Let's take a look at one of these used to configure CIFS/9000 in the upcoming section.

Configuring CIFS/9000

Among the executables that appear in the **/opt/samba/bin** directory is **samba_setup**. When you run this program, it performs some of the initial CIFS/9000 setup, including the name of the domain or workgroup and selecting the authentication to be used. Authentication gets a little tricky because you have a few options. The Samba part of this chapter covers some of the authentication issues. Let's now run **samba_setup**:

```
# samba_setup


This is a utility to help you configure your HP-UX system as
a SAMBA server.  This should only be run once after the initial
installation of SAMBA.
Do you wish to continue with the configuration?
(Y or N)?  : y

Proceeding with samba_setup...


You now must choose what level of authentication you
would like this SAMBA server to use:
1) domain
2) server
3) user
4) share
5) CANCEL
#? 3

You have chosen user level authentication
Is this correct?
(Y or N)?  : y

Enter the name you wish the SAMBA server to be known as (the
netbios name), or just hit enter if you want to keep the default
server name (l1):



Enter the name of the domain or workgroup that
you want this SAMBA server to be a part of: users


Configuring SAMBA server with user level authentication...


Your SAMBA server should now be ready to start.


You may wish to put /opt/samba/bin in the PATH for root
so that you don't have to type the fully qualified path
name for the SAMBA utilities and commands.
#
```

The minimal configuration we completed, along with information related to the Samba configuration, is in **/etc/opt/samba/smb.conf**, which is shown in the following listing:

```
# cat /etc/opt/samba/smb.conf
# 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
#
# Any line which starts with a ; (semi-colon) or a # (hash)
# is a comment and is ignored. In this example we will use a #
# for commentry and a ; for parts of the config file that you
# may wish to enable
#
# NOTE: Whenever you modify this file you should run the command
"testparm"
# to check that you have not many any basic syntactic errors.
#
#=======================
Global Settings
=======================
[global]
   netbios name = l1

# workgroup = NT-Domain-Name or Workgroup-Name, eg: REDHAT4
   workgroup = users

# server string is the equivalent of the NT Description field
   server string = Samba Server

# this tells Samba to use a separate log file for each machine
# that connects
   log file = /var/opt/samba/log.%m

# Put a capping on the size of the log files (in Kb).
   max log size = 1000

# Security mode. Most people will want user level security. See
# security_level.txt for details.
   security = user
# Use password server option only with security = server or domain
   password server = *

# You may wish to use password encryption. Please read
# ENCRYPTION.txt, Win95.txt and WinNT.txt in the Samba documenta-
tion.
# Do not enable this option unless you have read those documents
  encrypt passwords = no

# Most people will find that this option gives better performance.
# See speed.txt and the manual pages for details
   socket options = TCP_NODELAY

# Browser Control Options:
# set local master to no if you don't want Samba to become a master
# browser on your network. Otherwise the normal election rules apply
   local master = no

   read only = no
   preserve case = yes
   short preserve case = no
   dos filetime resolution = yes
   syslog = 0

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

# This one is useful for people to share files
[tmp]
   path = /tmp

#
```

You can modify this file directory to make changes for your environment.

In order to run CIFS/9000, we need to start the Samba server. This is done manually with the **startsmb** program as shown in the following listing:

```
# /opt/samba/bin/startsmb
Samba started successfully; process ids: smbd: 25363, nmbd: 25361
# ps -ef | grep samba
    root 25363     1  0 12:10:45 ?         0:00 /opt/samba/bin/smbd -D
    root 25361     1  0 12:10:45 ?         0:00 /opt/samba/bin/nmbd -D
#
```

man page

ps - 13

We have successfully started the Samba server with **startsmb** as confirmed by the Samba daemons shown in the **ps** output. We want to start Samba every time our system boots, so we can modify the **/etc/rc.config.d/samba** file and change the value of *RUN_SAMBA* to *1* as shown in the following listing:

```
# pwd
/etc/rc.config.d
# cat samba
#
# (c) Copyright Hewlett-Packard Company 1999
#
# This program is free software; you can redistribute it and/or mod-
ify
# it under the terms of the GNU General Public License as published
```

```
by
# the Free Software Foundation; either version 2 of the License, or
(at
# your option) any later version.
#
# This program is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See GNU Gen-
eral
# Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
#
# samba configuration: set RUN_SAMBA to a non-zero value to
# initiate the SAMBA server at run level 2.
#
# Installed at /etc/rc.config.d/samba
#

RUN_SAMBA=1
#
```

man page
"sw" - 2

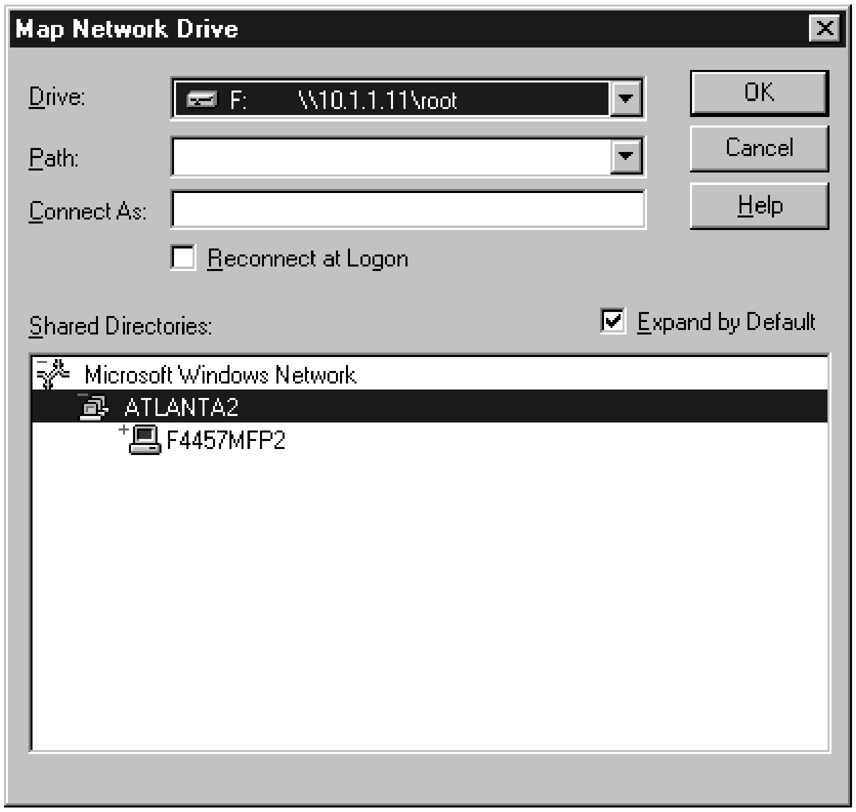
Up to this point, we have loaded CIFS/9000 using **swinstall**, performed some minimal configuration, and started Samba. The entire process has taken only minutes and we're ready to use CIFS/9000.

Now that we have loaded Samba and performed the initial configuration, we'll move to our Windows system and access an HP-UX file system on the Windows system as if it were local.

Map a Network Drive

Although we haven't done much work, we're now ready to map a network drive on our Windows system to an HP-UX file system.

Let's map the root file system on our CIFS/9000 server to **F:** on our Windows system as shown in Figure 24-2:



Drive: F: \\10.1.1.11\root

Path:

Connect As:

☐ Reconnect at Logon

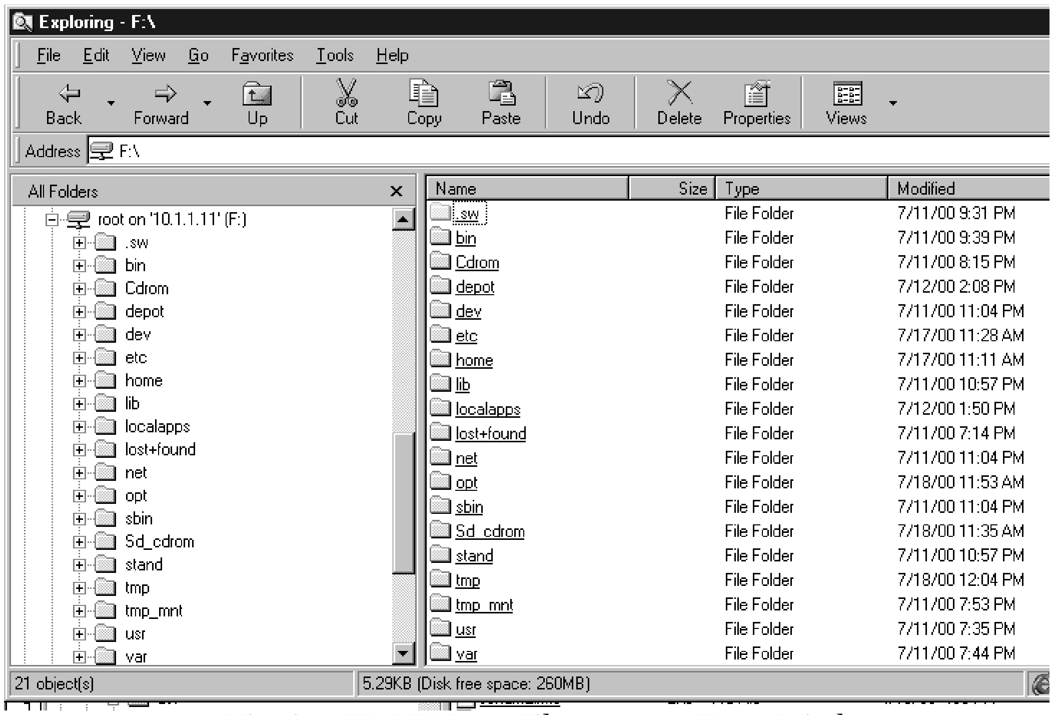
Shared Directories: ☑ Expand by Default

Microsoft Windows Network
ATLANTA2
F4457MFP2

Figure 24-2 Map Root FileSystem on CIFS/9000 Server to **F:**

We now have access to the root filesystem on our CIFS/9000 server (10.1.1.11), as though it were local to our Windows system.

Figure 24-3 shows accessing **F:**, which is root on our CIFS/9000 Server, to view the files and directories:



Name	Size	Type	Modified
.sw		File Folder	7/11/00 9:31 PM
bin		File Folder	7/11/00 9:39 PM
Cdrom		File Folder	7/11/00 8:15 PM
depot		File Folder	7/12/00 2:08 PM
dev		File Folder	7/11/00 11:04 PM
etc		File Folder	7/17/00 11:28 AM
home		File Folder	7/17/00 11:11 AM
lib		File Folder	7/11/00 10:57 PM
localapps		File Folder	7/12/00 1:50 PM
lost+found		File Folder	7/11/00 7:14 PM
net		File Folder	7/11/00 11:04 PM
opt		File Folder	7/18/00 11:53 AM
sbin		File Folder	7/11/00 11:04 PM
Sd_cdrom		File Folder	7/18/00 11:35 AM
stand		File Folder	7/11/00 10:57 PM
tmp		File Folder	7/18/00 12:04 PM
tmp_mnt		File Folder	7/11/00 7:53 PM
usr		File Folder	7/11/00 7:35 PM
var		File Folder	7/11/00 7:44 PM

21 object(s)

5.29KB (Disk free space: 260MB)

Figure 24-3 Viewing HP-UX Root Filesystem as **F:** on Windows

We can also use the **net view** command on our Windows system to see both the root filesystem on our CIFS/9000 server that we have mapped to **F:** as well as **tmp**, which was in **smb.conf** but was not mapped to a drive letter, as shown in the following listing:

```
C:\> net view \\10.1.1.11

Shared resources at \\10.1.1.11

Samba Server
```

```
Share name     Type     Used as     Comment
-------------------------------------------
root           Disk     F:          Home Directories
tmp            Disk

The command completed successfully.

C:\>
```

We have only scratched the surface of CIFS/9000 functionality in this section. The subjects of user authentication, printing, ACLs, and many other topics were not covered. The next section covering Samba on Linux includes some additional sections of interest, such as the Web-based configuration tool for Samba called SWAT, that you may want to review. There are some documents on *docs.hp.com* that contain more detailed information on CIFS/9000 that you'll want to download and print if you plan to use CIFS/9000 in your environment.

Samba Overview

Samba is an application that allows a UNIX host to act as a file server for Windows systems. The Windows systems can access UNIX file-systems and printers using their native Windows networking.

Our goal in this chapter is to give an overview of the setup of Samba and demonstrate a subset of its functionality. We'll focus on the file server functionality only. We'll set up shares on a remote UNIX system that will appear as a drive letters and icons on a Windows system. Because Samba comes with the Red Hat Linux software used throughout this book, we'll set up and run Samba on a Linux system. Samba is available for most UNIX variants.

Samba provides its file-sharing functionality using Server Message Block (SMB) protocol. SMB runs on top of TCP/IP. In our example in this chapter, both the Windows system and UNIX system are running TCP/IP and SMB. These provide all of the technology that is required to establish file sharing between the two systems.

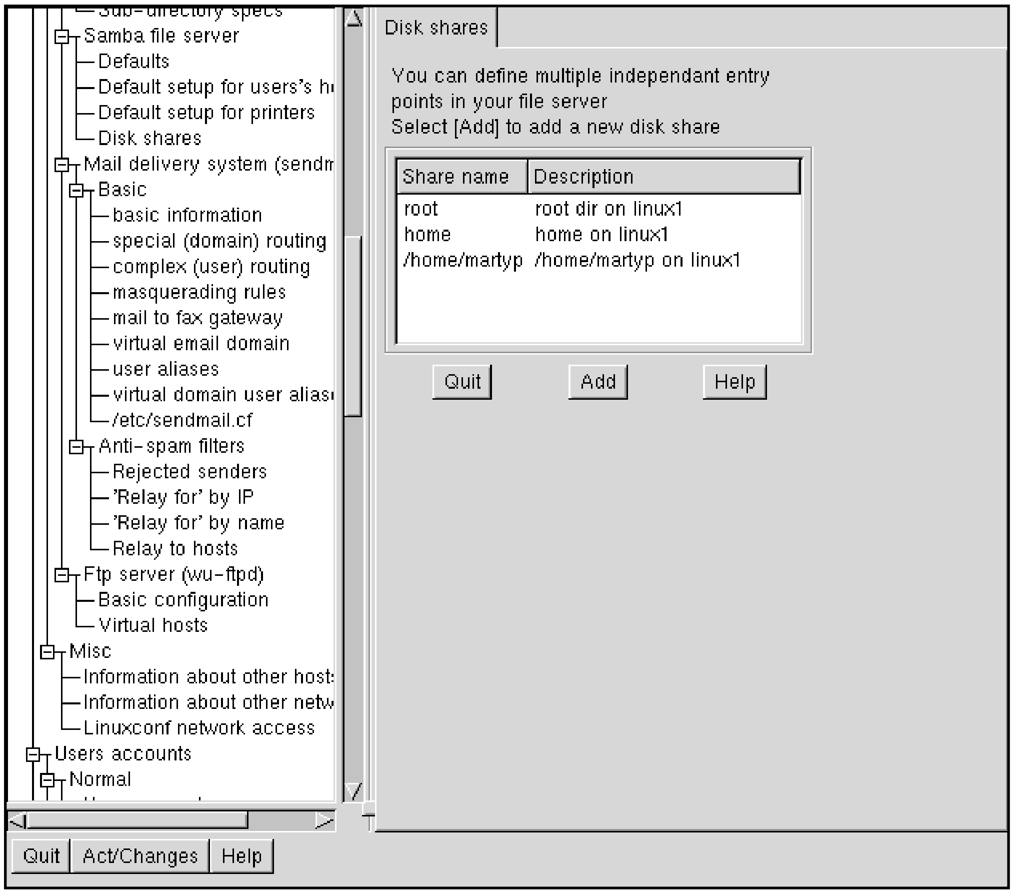
Chapter 26 covered Network File System (NFS) running on a Windows system and accessing files on a UNIX system. This functionality is similar to that which we'll cover with Samba in this section. In addition to the file-sharing capability, Samba also provides printer sharing and addtional user access control capability. We won't focus on these capabilities in this chapter, however, Samba does indeed provide some advanced functionality in these areas.

At the time of this writing Samba contains the functionality just mentioned file-sharing, printer sharing, and advanced user access control of files. There are many advancements taking place with Samba and other software provided under GNU Public License (GPL) as free software. Because the software is free, many individuals have access to it and spend time enhancing the software. For this reason, you may find that additional functionality is included in Samba and other such software. There are many enhancements planned for Samba, including an administration tool and other advancements that were not available at the time this chapter was written. There is more information about obtaining Samba and other free software at the end of this chapter.

Setup

Because Samba is supplied on the Red Hat Linux CD-ROM, we'll walk through a simple Samba setup using Red Hat Linux. When installing Red Hat Linux, you can select the software packages you wish to load, as you can on most all UNIX variants. If you did not load Samba at the time you originally loaded the operating system, you can use the *Gnome RPM* tool or **rpm** from the command line to load Samba or any other software. These tools were briefly discussed in the System Administration chapter.

Using *Linuxconf*, we'll perform some simple tasks to set up Samba. Figure 24-4 shows the *Linuxconf* window *Disk Shares* under *Samba file server* with the three disk shares we'll be using in this example:

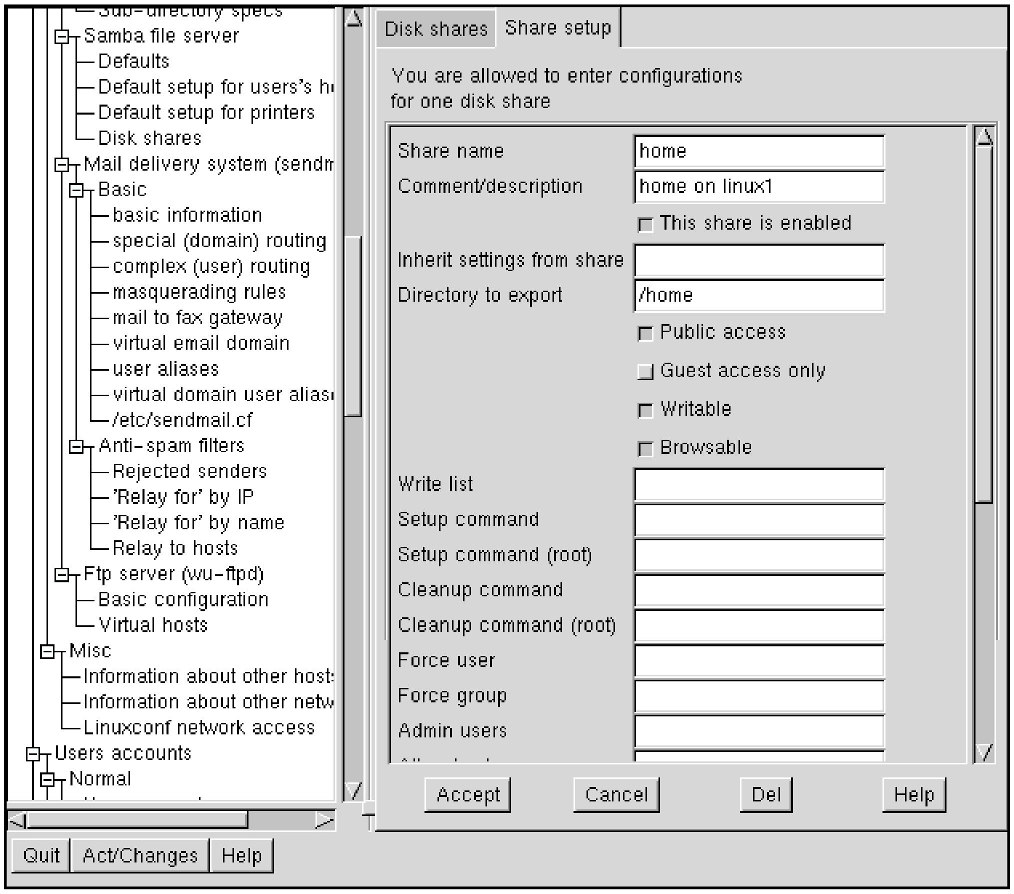


You can define multiple independant entry
points in your file server
Select [Add] to add a new disk share

Share name	Description
root	root dir on linux1
home	home on linux1
/home/martyp	/home/martyp on linux1

Figure 24-4 Share Information in *Linuxconf*

There are three share names on our Linux system that we will make available to other systems running SMB. Figure 24-5 shows more detailed information about the middle share called *home*:



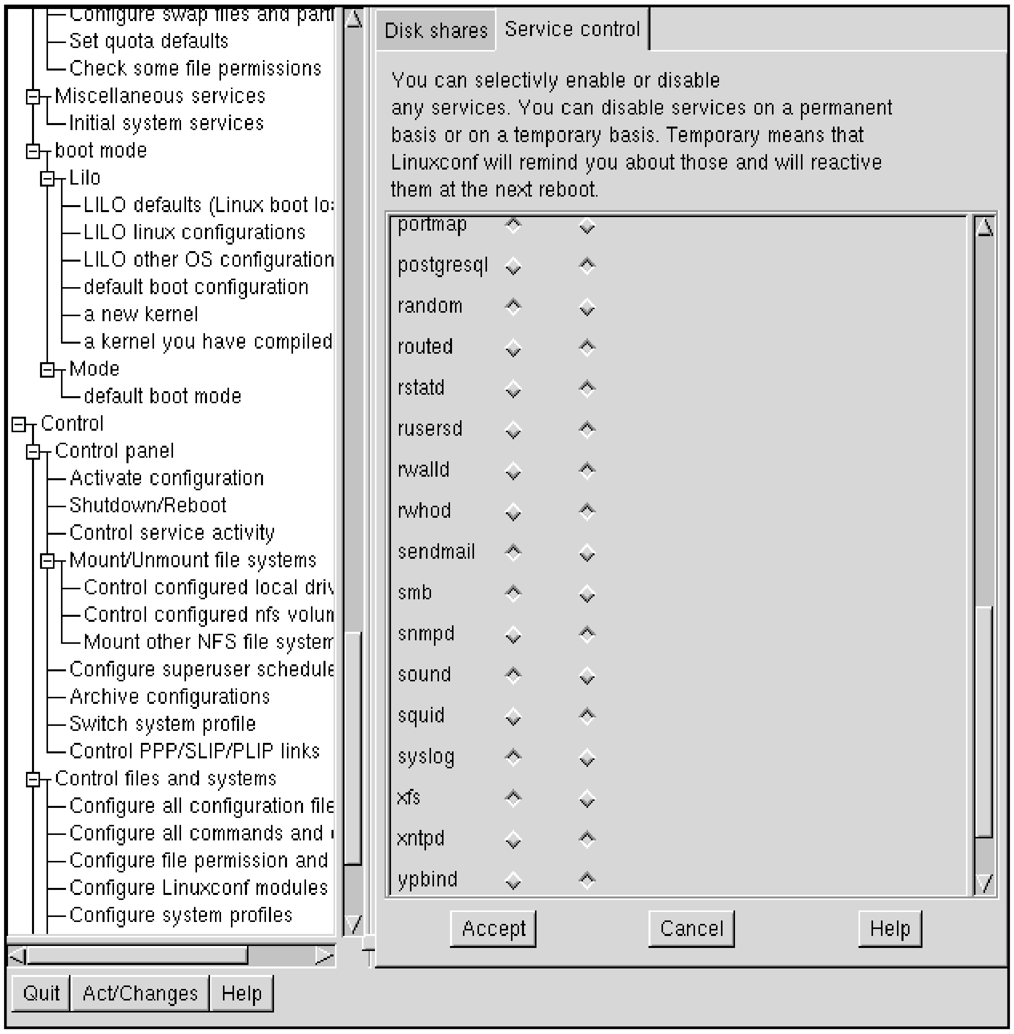
You are allowed to enter configurations
for one disk share

Share name	home
Comment/description	home on linux1
	This share is enabled
Inherit settings from share	
Directory to export	/home
	Public access
	Guest access only
	Writable
	Browsable
Write list	
Setup command	
Setup command (root)	
Cleanup command	
Cleanup command (root)	
Force user	
Force group	
Admin users	

Figure 24-5 **/home** Share Information in *Linuxconf*

This share is for the **/home** directory on *linux1*. Note the four selections related to permissions in the figure. We have granted *Public access*, *Writable*, and *Browsable* rights on this share. We have not restricted it to *Guest only access*. On this share, we have been unrestrictive with respect to the rights granted it. You'll want to consider these rights carefully on your system as you go about assigning these rights. Keep in mind that we're not using any user authentication in our examples, in order to keep them simple. In practice, however, you'll want to make sure that you assign appropriate rights to the shares.

With three shares having been assigned, we'll use *Linuxconf* to start *smb*. We'll do so by enabling *smb* in *Control service activity*, as shown in Figure 24-6:


Configure swap files and part
Set quota defaults
Check some file permissions
Miscellaneous services
Initial system services
boot mode
Lilo
LILO defaults (Linux boot lo
LILO linux configurations
LILO other OS configuration
default boot configuration
a new kernel
a kernel you have compiled
Mode
default boot mode
Control
Control panel
Activate configuration
Shutdown/Reboot
Control service activity
Mount/Unmount file systems
Control configured local driv
Control configured nfs volun
Mount other NFS file system
Configure superuser schedule
Archive configurations
Switch system profile
Control PPP/SLIP/PLIP links
Control files and systems
Configure all configuration file
Configure all commands and
Configure file permission and
Configure Linuxconf modules
Configure system profiles
Quit
Act/Changes
Help
Disk shares
Service control
Accept
Cancel
Help

You can selectivly enable or disable
any services. You can disable services on a permanent
basis or on a temporary basis. Temporary means that
Linuxconf will remind you about those and will reactive
them at the next reboot.

portmap
postgresql
random
routed
rstatd
rusersd
rwalld
rwhod
sendmail
smb
snmpd
sound
squid
syslog
xfs
xntpd
ypbind

Figure 24-6 Start *smb* in *Control service activity* in *Linuxconf*

smb has been enabled so that the service will start when the system boots.

Before we begin using *smb*, let’s perform a couple of quick checks on the work we have performed with *Linuxconf*. We could have accomplished manually everything we have done with *Linuxconf*.

The first check is to view the file **/etc/smb.conf**. This is the file that contains all our SMB configuration information. For now, let’s go right to the "Share Definitions" section to confirm that the three shares we configured in *Linuxconf* have proper entries in **/etc/smb.conf**:

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

# 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


# 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


# 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
[root]
    comment = root dir on linux1
    available = yes
    path = /
    public = yes
    guest only = no
    writable = yes
    browseable = yes
    only user = no
[home]
    comment = home on linux1
    available = yes
    path = /home
    public = yes
    guest only = no
    writable = yes
    browseable = yes
    only user = no
[/home/martyp]
    comment = /home/martyp on linux1
    available = yes
    path = /home/martyp
    public = yes
    guest only = no
    writable = yes
    browseable = yes
    only user = no
```

There are indeed entries in **/etc/smb.conf** for the three shares we configured in *Linuxconf* with the permissions we set up. These are shown near the end of the listing. Next we'll run a Samba utilitiy called **testparm**. This utility will check our **/etc/smb.conf** file for errors. This utility produces a very long output which I won't include here, but you'll want to run this and check for any warnings or errors it produces.

For our **/etc/smb.conf** file, **testparm** produced only one warning that appeared at the very beginning of the file, which is shown in the following listing:

```
# testparm smb.conf

Load smb config files from /etc/smb.conf
Processing section "[homes]"
Processing section "[printers]"
Processing section "[root]"
Processing section "[home]"
Processing section "[/home/martyp]"
```

```
Loaded services file OK.
WARNING: You have some share names that are longer than 8 chars
These may give errors while browsing or may not be accessible
to some older clients
Press enter to see a dump of your service definitions
```

•

•

•

testparm produced the warning for a long share name and also included a list of our three shares, which looked to be in order. We won't address the long share name warning because the Windows system used in the example can handle the long name. If, however, we were on a DOS system, there would be a problem with this name. We'll see shortly how these potential name incompatibilities are addressed by Linux.

The next check we want to perform is to see that the daemon for Samba, called **smbd**, is indeed running, as shown in the following listing:

man page
ps - 13

```
# ps -efl | grep smbd

140 S root     490     1  0  60   0    -   553 do_sel Sep28 ?        00:00:00 [smbd]
140 S root    1493   490  0  60   0    -   896 do_sel Sep29 ?       00:00:00 smbd -D
000 S root   1976  1951  0  70   0   -   288 pipe_r 09:09 pts/1   00:00:00 grep smbd
```

man page
netstat - 12

This **ps** output shows that **smbd** is running. Next let's check that the *netbios-ssn* service is running in the "LISTEN" state as shown in the following example:

man page
grep - 19

```
# netstat -a | grep netbios

tcp        0      0 linux1:netbios-ssn      f4457mfp2:1047          ESTABLISHED
tcp        0      0 *:netbios-ssn           *:*                     LISTEN
udp        0      0 linux1:netbios-dgm      *:*
udp        0      0 linux1:netbios-ns       *:*
udp        0      0 *:netbios-dgm           *:*
udp        0      0 *:netbios-ns            *:*
```

The output of this listing shows that *netbios* is running. There is an "ESTABLISHED" connection shown which you won't see until you have created a connection to your Samba server. I had already established this connection as I prepared the examples in this chapter.

We can also use the Samba client to access files on the Windows system from our Linux system. Although we won't cover this capability, mostly because the Linux system will normally act as a file server and not the other way around, there is a utility called **smbclient** that provides a lot of useful information. Let's now get the overall status of the Samba setup with the **smbclient** utility, as shown in the following listing:

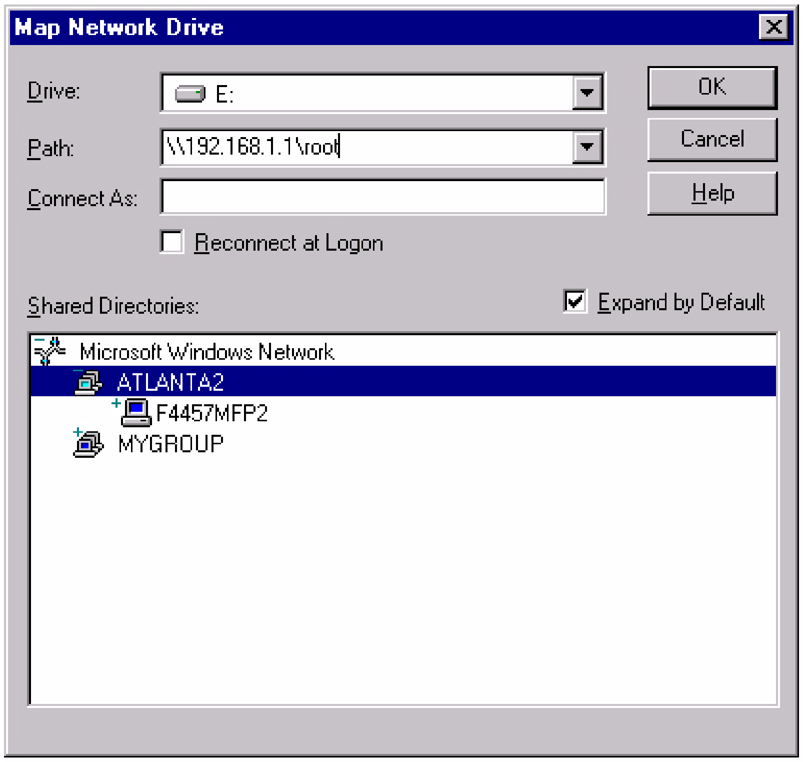
```
# smbclient -L linux1

Added interface ip=192.168.1.1 bcast=192.168.1.255 nmask=255.255.255.0

Domain=[MYGROUP] OS=[Unix] Server=[Samba 2.0.3]

                   Sharename      Type      Comment
                   ---------      ----      -------
                   root           Disk      root dir on linux1
                   home           Disk      home on linux1
                   /home/martyp   Disk      /home/martyp on linux1
                   IPC$           IPC       IPC Service (Samba Server)
                   lp             Printer

                   Server               Comment
                   ---------            -------
                   LINUX1               Samba Server

                   Workgroup            Master
                   ---------            -------
                   ATLANTA2             F4457MFP2
                   MYGROUP              LINUX1
```

This utility produces a useful summary of the Samba setup, including the three shares we set up, the Samba server for our example, and other useful information.

We could continue to test Samba on the server, but the listings we viewed give every indication that Samba is running. Let's now move to the Windows client and use *explorer* to access the shares we made available on our Samba server.

Using Shares

Using *Explorer*, we'll now map a network drive. We'll map **/root** on the Samba server *linux1* to drive **E:** on the Windows system as shown in Figure 24-7:



Drive: E:

Path: \\192.168.1.1\root

Connect As:

☐ Reconnect at Logon

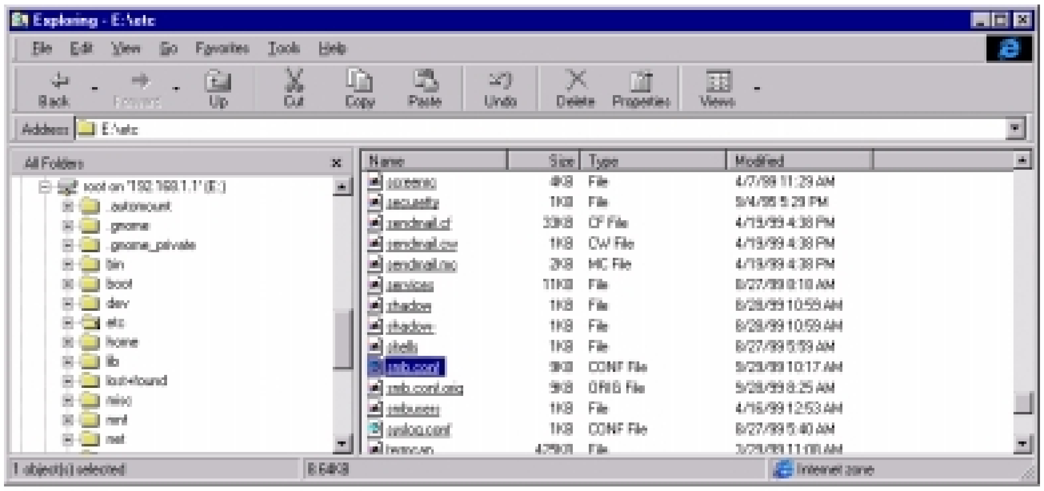
Shared Directories:

☑ Expand by Default

Microsoft Windows Network
ATLANTA2
F4457MFP2
MYGROUP

Figure 24-7 Map Drive **E:** to **/root**

We could have mapped this network drive at the command line on the PC using **net use E:\\192.168.1.1\root**. After having successfully mapped **E:** to **/** on *linux1* using *Explorer,* we can access this directory on our Windows system as shown in Figure 24-8:


Exploring - E:\etc
File Edit View Go Favorites Tools Help
Back Forward Up Cut Copy Paste Undo Delete Properties Views
Address E:\etc
All Folders

root on '192.168.1.1' (E:)
.automount
.gnome
.gnome_private
bin
boot
dev
etc
home
lib
lost+found
misc
mnt
net

Name	Size	Type	Modified
screenrc	4KB	File	4/7/99 11:29 AM
securetty	1KB	File	9/4/98 5:29 PM
sendmail.cf	33KB	CF File	4/19/99 4:38 PM
sendmail.cw	1KB	CW File	4/19/99 4:38 PM
sendmail.mc	2KB	MC File	4/19/99 4:38 PM
services	11KB	File	8/27/99 8:18 AM
shadow	1KB	File	8/28/99 10:59 AM
shadow-	1KB	File	8/28/99 10:59 AM
shells	1KB	File	8/27/99 5:59 AM
smb.conf	9KB	CONF File	9/29/99 10:17 AM
smb.conf.orig	9KB	ORIG File	9/28/99 8:25 AM
smbusers	1KB	File	4/16/99 12:53 AM
syslog.conf	1KB	CONF File	8/27/99 5:40 AM
termcap	479KB	File	3/29/99 11:08 AM

1 object(s) selected   8.64KB   Internet zone

Figure 24-8 View **E:\etc** on Windows System

In Figure 24-8, we have changed to the **/etc** directory on *linux1* and have selected the **smb.conf** file. You can see from this example that **/** on *linux1* is fully accessable on the Windows system. We have access to all three shares on *linux1*, as the following listing shows on the Windows system:

```
c: net view \\192.168.1.1

Shared resources at \\192.168.1.1

Samba Server
```

```
Share name   Type         Used as  Comment

-------------------------------------------------------------
/home/martyp   Disk       G:      /home/martyp on linux1
home          Disk        F:       home on linux1
lp            Print
root           Disk         E:       root dir on linux1
The command completed successfully.
```

This listing shows that all three of the *linux1* shares are now available on the Windows system. We have not configured any printer sharing, which is also included as part of Samba functionality, so there is no reference to any printers.

Additional Samba Topics

Samba Web Configuration Tool (SWAT)

SWAT is a Web-based administration tool for Samba. It is easy to configure and provides a simple interface for most Samba configuration tasks. On our Red Hat Linux system, the following steps had to be performed to get SWAT running. If you have a different UNIX variant, then your steps will be different.

Confirm that the following line exists in **/etc/services**:

```
swat          901/tcp
```

Next, uncomment the following line in **/etc/inetd.conf**. This line was already in the file as part of the Linux operating system load:

man page

ps - 13

```
swat stream tcp nowait.400 root /usr/sbin/swat swat
```

Kill the **inetd** process by first finding its Process ID (PID) and then issuing the **kill** command for that PID:

```
# ps -ef | grep inetd
# kill -1 PID of inetd
```

man page
grep - 19

Now we can run SWAT from a browser interface, in this case Netscape, by specifying the IP address of the Samba server and the port number *901*:

man page
kill - 22

```
# netscape http://192.168.1.1:901
```

The browser requests a user name and password when SWAT is invoked to ensure that only users with sufficient rights can make modifications to the Samba configuration in SWAT.

Figure 24-9 shows the SWAT interface. It includes links for *HOME, GLOBALS, SHARES, PRINTERS, STATUS, VIEW,* and *PASSWORD.*

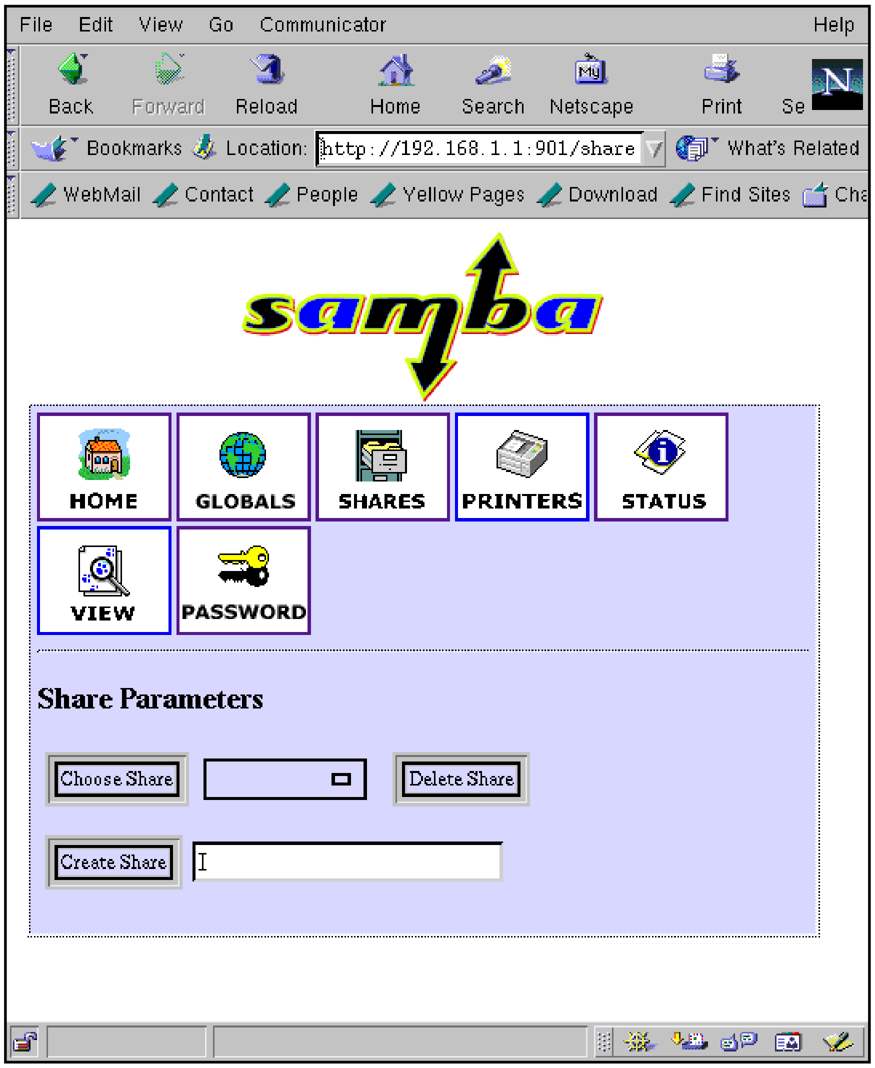


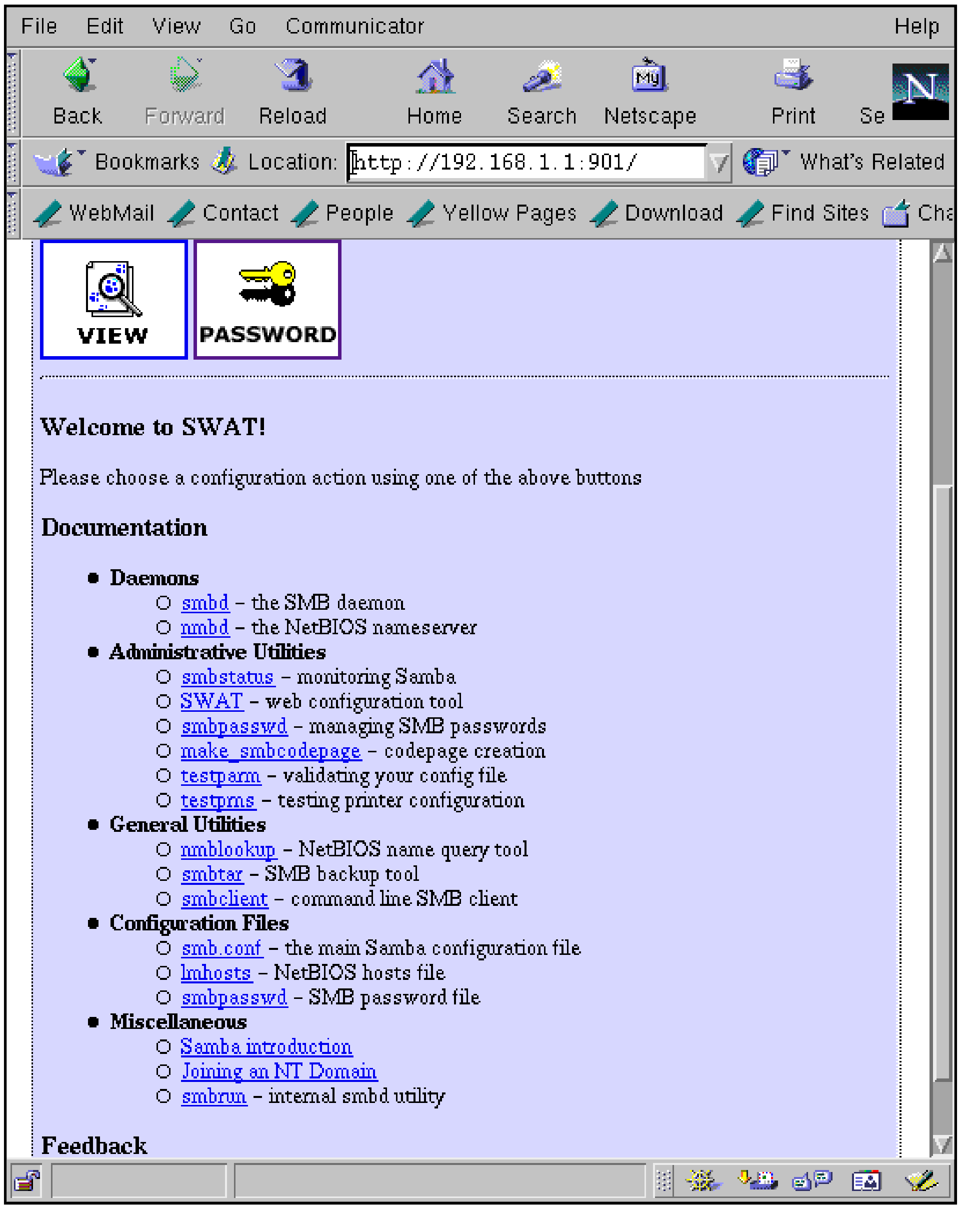
Figure 24-9 SWAT with *SHARES* Selected

The three shares we had earlier conifgured are available in SWAT by selecting *SHARES* from Figure 24-9. SWAT is a good interface for Samba configuration, but there is no substitute for knowing some of the manual processes we experienced earlier in the chapter.

For changes to take effect that are made with SWAT, we had to restart **smbd** with the following command:

```
# /etc/rc.d/init.d/smb restart
```

There is great documentation on the Linux system describing SWAT and all of its capabilities. Figure 24-10 shows the *HOME* page for SWAT with documentation on many of the topics we have covered in this chapter.



Welcome to SWAT!

Please choose a configuration action using one of the above buttons

Documentation

- **Daemons**
  - smbd – the SMB daemon
  - nmbd – the NetBIOS nameserver
- **Administrative Utilities**
  - smbstatus – monitoring Samba
  - SWAT – web configuration tool
  - smbpasswd – managing SMB passwords
  - make_smbcodepage – codepage creation
  - testparm – validating your config file
  - testprns – testing printer configuration
- **General Utilities**
  - nmblookup – NetBIOS name query tool
  - smbtar – SMB backup tool
  - smbclient – command line SMB client
- **Configuration Files**
  - smb.conf – the main Samba configuration file
  - lmhosts – NetBIOS hosts file
  - smbpasswd – SMB password file
- **Miscellaneous**
  - Samba introduction
  - Joining an NT Domain
  - smbrun – internal smbd utility

Feedback

Figure 24-10 SWAT *HOME* Showing Documentation Available

There is also extensive online documentation for all Samba-related software at *www.samba.org*.

Log Files

Like most UNIX applications, Samba provides extensive logging. The **smb.conf** file contains a section that allows you to specify the level of Samba logging you wish to take place. The short section below shows that you can have separate log files for each Windows machine that connects, and you can specify the maximum size of the log file:

```
# this tells Samba to use a separate log file for each machine
# that connects
    log file = /var/log/samba/log.%m

# Put a capping on the size of the log files (in Kb).
    max log size = 50
```

The directory **/var/log/samba** contains a variety of Samba log files, including the log file for the Windows system used in our examples in this chapter, called *f4457mfp2,* as shown in the following listing:

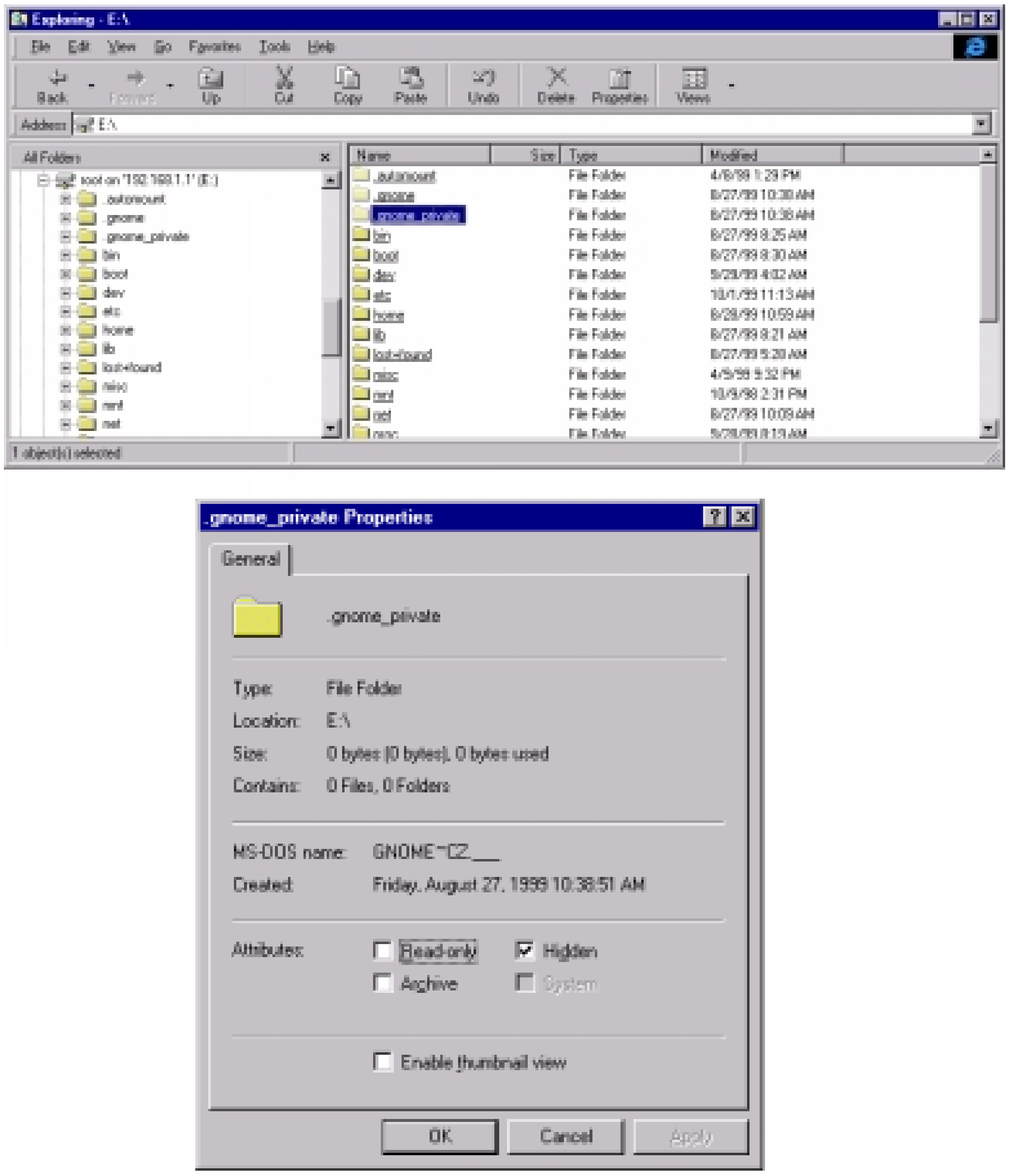
man page

ls - 15

```
# ls -l /var/log/samba

total 14
-rw-r--r--   1 root    root       1915 Sep 30 09:29 log.f4457mfp2
-rw-r--r--   1 root    root        213 Sep 29 10:14 log.linux1
-rw-r--r--   1 root    root          0 Sep 29 04:02 log.nmb
-rw-r--r--   1 root    root       8234 Sep 28 12:25 log.nmb.1
-rw-r--r--   1 root    root       2015 Sep 28 12:19 log.smb
-rw-r--r--   1 root    root          0 Oct  1 11:17 smbconf3.txt
```

File Name Mangling

Among the many Windows and UNIX incompatibilities that exist are file names. Depending on the version of Windows you are using, there may be extensive file name incompatibilities with UNIX. Figure 24-11 is an *Explorer* window with the file **gnome_private** selected, which is on the Linux server. On our Windows NT system, this file name looks fine. However, also in this figure is a *Properties* window showing that the DOS name for this file would change dramatically if we were on a DOS system.



Name	Size	Type	Modified
.automount		File Folder	4/8/99 1:29 PM
.gnome		File Folder	8/27/99 10:38 AM
.gnome_private		File Folder	8/27/99 10:38 AM
bin		File Folder	8/27/99 8:25 AM
boot		File Folder	8/27/99 8:30 AM
dev		File Folder	5/28/99 4:02 AM
etc		File Folder	10/1/99 11:13 AM
home		File Folder	8/28/99 10:59 AM
lib		File Folder	8/27/99 8:21 AM
lost+found		File Folder	8/27/99 5:20 AM
misc		File Folder	4/5/99 5:32 PM
mnt		File Folder	10/9/98 2:31 PM
net		File Folder	8/27/99 10:09 AM
proc		File Folder	5/28/99 8:13 AM

Type:	File Folder
Location:	E:\
Size:	0 bytes (0 bytes), 0 bytes used
Contains:	0 Files, 0 Folders
MS-DOS name:	GNOME~CZ___
Created:	Friday, August 27, 1999 10:38:51 AM

Attributes: Read-only [✓ Hidden] Archive System

Enable thumbnail view

Figure 24-11 **.gnome_private** in *Explorer* (top) and Mangled Name (bottom)

In the case of my Windows system, there is no problem handling the file name **.gnome_private** as it appears on the Linux system. In the case of a DOS system, however, there would be extensive "mangling" of the file name that would have to take place. DOS uses only 8.3 file names, or those with eight characters and a three-character extension.

Samba mangles files that start with a dot, such as the one in this example, by removing the leading dot, printing the first five characters and then a tilde, and then applying a hash algorithm to the original filename to come up with the last two characters. This results in a total of eight characters for the filename. All the characters are uppercase.

If the file does not begin with a dot, then the file name will be generated in the same way as described in the previous paragraph. The extension consists of the first three characters to the right of the dot converted to uppercase. This results in a total of eight characters for the filename and three for the extension.

In our example, the filename **.gnome_private** would be given a DOS name of **GNOME~CZ**. You have some control over mangling in the **smb.conf** file.

User Issues

I have avoided making a serious user-related configuration in this chapter in the interest of keeping the examples simple. You will probably not have this luxury unless you are in an environment where you are the only user on both the Windows and UNIX systems.

Users and groups have always been an important part of every UNIX system. Users and groups were not as important in the Windows world until more recently. This change results in some Windows environments in which there is not a complete user and group policy in place which could be used by Samba.

Samba takes into account both an environment in which you have set up Windows users and groups and one in which you may not have worked out all the issues related to Windows users and groups. User-level authentication in Samba is set up in such a way that a client can use a given service if they supply the correct user name and password. Share-level authentication takes place by granting access based on the rights of the "guest account" on the UNIX system. This is true unless a client used a user name and password in this or a previous session. Needless to say, there is a lot to consider with user authentication.

The **smb.conf** file has an entry in which you can specify the security as "user" or "share," as shown in the following lines:

```
security = user

security = share
```

man page

passwd - 5

Most systems employ user-level security. When this is done users are checked against their names in the **passwd** file and access is granted accordingly.

There are many additional issues related to user authentication that you'll want to investigate if you set up Samba. The documentation supplied with Samba as part of Red Hat Linux is excellent, and the background information on the Web sites listed later in this chapter is informative as well.

Samba Utilities and Programs

We have used several Samba utilities and programs in this chapter. The following list gives a description of the most often used Samba-related commands. There are manual pages for all these, which are part of most Samba installations.

- **smbd** - This is the daemon that provides file and print services to SMB clients, such as the Windows system used in our examples throughout this chapter.
- **nmbd** - This is the daemon that provides NetBIOS name server capability and browsing.

- **smbclient** - A program that gives the server access to remotely mounted SMB shares on other servers.
- **testparm** - A test program for **/etc/smb.conf**.
- **smbstatus** - Program that displays status information about current Samba connections.
- **smbpasswd** - Program used to change a user's SMB password on the local machine.
- **smbrun** - Program that runs shell commands for **smbd**.
- **smbtar** - Program to back up SMB shares directly to a UNIX tape drive.
- **smbmount** - Used to mount an SMB file system.
- **smbumount** - Used to unmount an SMB file system.

The online manual pages for these and other Samba-related commands provide more detail. Even in a simple setup such as the one performed in this chapter, you will want to run some of these programs.

Obtaining Samba

In the examples used throughout this chapter, we set up Samba on a Linux system that had Samba installed on it as part of the Red Hat Linux CD-ROM. If Samba does not come on the CD-ROM provided with your UNIX variant or if you wish to be sure that you're loading the very latest Samba, then you can obtain Samba from the Web.

www.samba.org is the place to start. From this Web site, you can select a "download site" in your country. You can also select "Web sites" on *www.samba.org* that provide a wealth of information on

Samba, including the GNU General Public License mentioned earlier in the chapter.

There is extensive documentation on Samba-related Web sites, including detailed descriptions of the programs that I listed earlier and used in this chapter.

If you decide to download Samba, you'll probably be given an option of loading a precompiled Samba on your system or building and compiling Samba yourself. The choice you make depends on a lot of factors. If you have a good, reliable Samba distribution, as we did in this chapter when working with Red Hat Linux, then working with a precompiled Samba may be best. If you're interested in learning more about how Samba works and is configured, and want the very latest and greatest version, then download the source and compile it yourself.

Even if you have a great prepackaged Samba, as we did in this chapter, it is still worth visiting the Samba-related Web sites to view the extensive documentation available.