,ch03.26970 Page 50 Friday, November 19, 1999 3:28 PM

3

Configuring Windows Clients

You'll be glad to know that configuring Windows to use your new Samba server is quite simple. SMB is Microsoft's native language for resource sharing on a local area network, so much of the installation and setup on the Windows client side has been taken care of already. The primary issues that we will cover in this chapter involve communication and coordination between Windows and Unix, two completely different operating systems.

Samba uses TCP/IP to talk to its clients on the network. If you aren't already using TCP/IP on your Windows computers, this chapter will show you how to install it. Then you'll need to configure your Windows machines to operate on a TCP/IP network. Once these two requirements have been taken care of, we can show how to access a shared disk on the Samba server.

This chapter is divided into three sections. The first section covers setting up Windows 95/98 computers while the second covers Windows NT 4.0 machines. The final section provides some prerequisite information on how SMB connections are made from Windows clients and servers, which is useful as we move into the later chapters of the book.

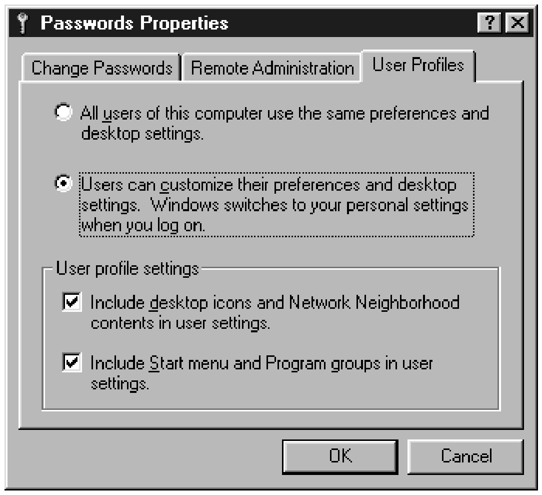
Setting Up Windows 95/98 Computers

Unfortunately, Windows 95/98 wasn't designed for a PC to have more than one user; that concept is more inherent to a Unix operating system or Windows NT. However, Windows 95/98 does have *limited* support for multiple users: if you tell it, the operating system will keep a separate profile (desktop layout) and password file for each user. This is a far cry from true multiuser security. In other words, Windows 95/98 won't try to keep one user from destroying the work of another on the local hard drive like Unix, but profiles are a place to start.

Accounts and Passwords

The first thing we need to do is to tell Windows to keep user profiles separate, and to collect usernames and passwords to authenticate anyone trying to access a Samba share. We do so via the Password settings in the Control Panel. If you are not familiar with the Windows Control Panel, you can access it by choosing the Settings menu item from the pop-up menu of the Start button in the lower-left corner of the screen. Alternatively, you'll find it as a folder under the icon in the upper-left corner that represents your computer and is typically labeled My Computer.

After selecting the Passwords icon in the Control Panel, click on the User Profiles tab on the far right. You should see the dialog box shown in Figure 3-1. Then click the lower of the two radio buttons that starts "Users can customize their preferences...." This causes Windows to store a separate profile for each user, and saves the username and password you provide, which it will use later when it connects to an SMB/CIFS server. Finally, check *both* the options under the User Profile Settings border, as shown in the figure.



( ) All users of this computer use the same preferences and desktop settings.

(●) Users can customize their preferences and desktop settings. Windows switches to your personal settings when you log on.

User profile settings

☑ Include desktop icons and Network Neighborhood contents in user settings.

☑ Include Start menu and Program groups in user settings.

Figure 3-1. The Passwords Properties panel

The next step is to select the Change Passwords tab on the left side of the dialog box. In order for Samba to allow you access to its shares, the username and password you give to Windows must match the account and password on the Samba server. If you don't have this tab in your dialog box, don't worry; it's probably because you haven't given yourself a Windows username and password yet. Simply click the OK button at the bottom and respond Yes when Windows asks to reboot. Then, skip down to the section entitled "Logging in for the first time."

Changing the Windows password

After selecting the Change Passwords tab, the dialog box in Figure 3-2 will appear.

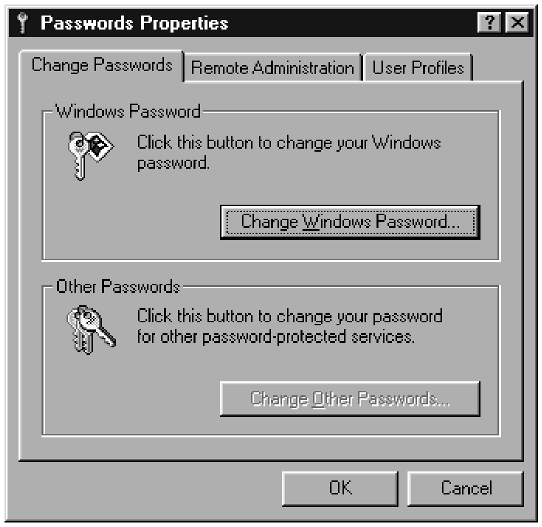

Passwords Properties
Change Passwords
Remote Administration
User Profiles
Windows Password
Click this button to change your Windows password.
Change Windows Password...
Other Passwords
Click this button to change your password for other password-protected services.
Change Other Passwords...
OK
Cancel

Figure 3-2. The Change Passwords tab

Select the Change Windows Password button. The Change Windows Password dialog box should appear, as shown in Figure 3-3. From here, you can change your password to match the password of the account on the Samba server through which you intend to log in.

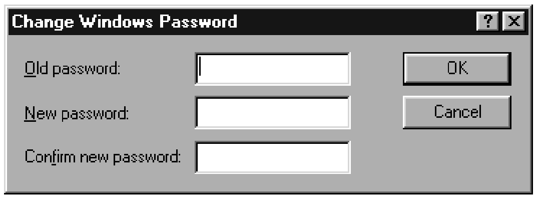
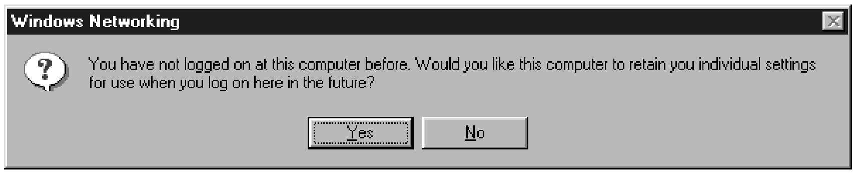


Figure 3-3. The Change Windows Password dialog box

Logging in for the first time

If you didn't have a Change Passwords tab in the Passwords Properties window, then after Windows has finished rebooting, it will ask you to log in with a username and a password. Give yourself the same username and password that you have on the Samba server. After confirming your new username and password, or if you already have one, Windows should ask you if you want to have a profile, using the dialog shown in Figure 3-4.


Windows Networking

You have not logged on at this computer before. Would you like this computer to retain you individual settings for use when you log on here in the future?

Figure 3-4. Windows Networking profiles

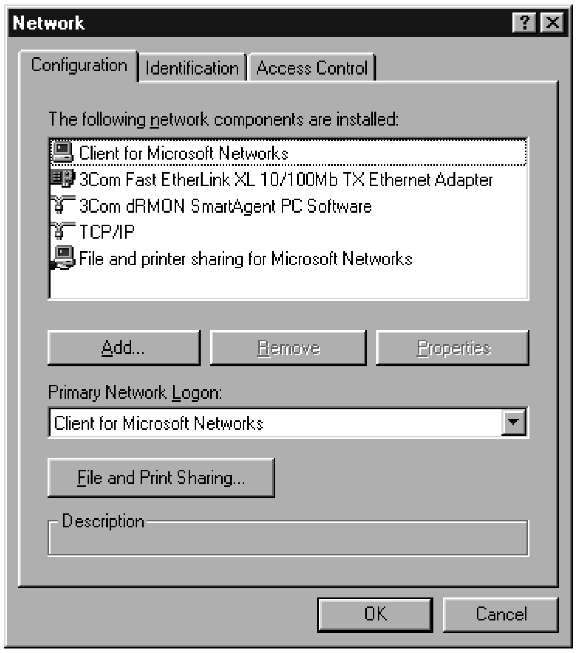
Answer Yes, upon which Windows will create a separate profile and password file for you and save a copy of your password in the file. Now when you connect to Samba, Windows will send its password, which will be used to authenticate you for each share. We won't worry about profiles for the moment; we'll cover them in Chapter 6, *Users, Security, and Domains*. We should point out, however, that there is a small security risk: someone can steal the password file and decrypt the passwords because it's weakly encrypted. Unfortunately, there isn't a solution to this with Windows 95/98. In Windows 2000 (NT 5.0), the password encryption should be replaced with a much better algorithm.

Setting Up the Network

The next thing we need to do is make sure we have the TCP/IP networking protocol set up correctly. To do this, double-click on the Network icon in the Control Panel. You should see the network configuration dialog box, as shown in Figure 3-5.

Microsoft networking works by binding specific protocols, such as IPX or TCP/IP, to a specific hardware device, such as an Ethernet card or a dialup connection. By routing a protocol through a hardware device, the machine can act as a client or server for a particular type of network. For Samba, we are interested in binding the TCP/IP protocol through a networking device, making the machine a client for Microsoft networks. Thus, when the dialog box appears, you should see at least the Client for Microsoft Networks component installed on the machine, and hopefully a networking device (preferably an Ethernet card) bound to the TCP/IP protocol. If there is only one networking hardware device, you'll see the TCP/IP protocol listed below that device. If it appears similar to Figure 3-5, the protocol is bound to the device.

You may also see "File and printer sharing for Microsoft Networks," which is useful. In addition, you might see NetBEUI or Novell Networking, which are standard


Network

Configuration | Identification | Access Control

The following network components are installed:

- Client for Microsoft Networks
- 3Com Fast EtherLink XL 10/100Mb TX Ethernet Adapter
- 3Com dRMON SmartAgent PC Software
- TCP/IP
- File and printer sharing for Microsoft Networks

Primary Network Logon:

Client for Microsoft Networks

Figure 3-5. The Windows 95/98 Network panel

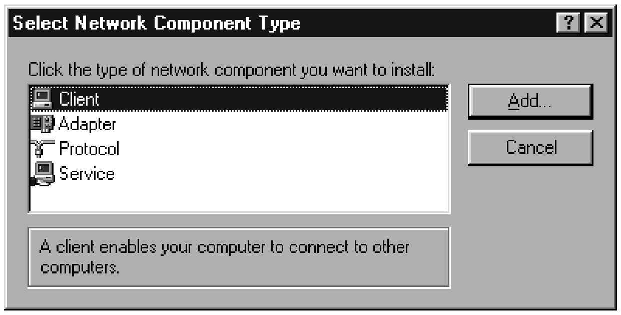
with Windows installations but undesirable when TCP/IP is running. Remove NetBEUI if you possibly can—it's unnecessary and makes debugging Windows browsing difficult. If you don't have any Novell servers on your network, you can remove Novell (IPX/SPX) as well.

Adding TCP/IP

If you don't see TCP/IP listed at all, you'll need to install the protocol. If you already have TCP/IP, skip this section, and continue with the section "Setting Your Name and Workgroup," later in this chapter.

Installing TCP/IP isn't difficult since Microsoft distributes its own version of TCP/IP for free on their installation CD-ROM. You can add the protocol by clicking on the Add button below the component window. Indicate that you wish to add a specific protocol by selecting Protocol and clicking Add... on the following dialog box, which should look similar to Figure 3-6.

After that, select the protocol TCP/IP from manufacturer Microsoft, as shown in Figure 3-7, then click OK. After doing so, you will be returned to the network dialog. Click OK there to close the dialog box, upon which Windows will install the necessary components from disk and reboot the machine.



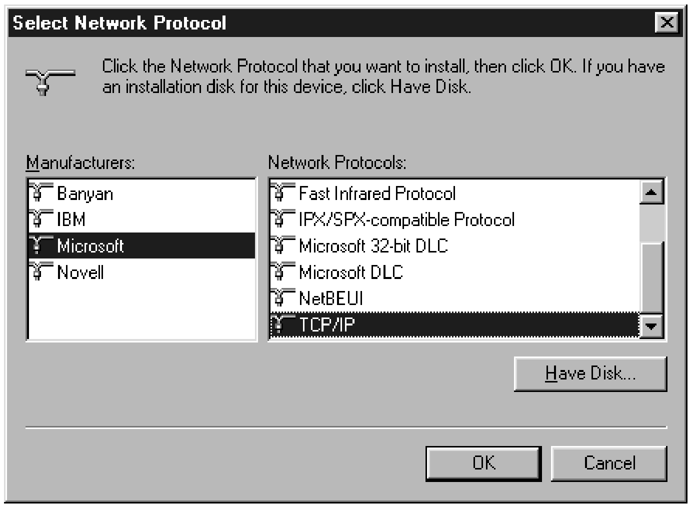
Select Network Component Type

Click the type of network component you want to install:

- Client
- Adapter
- Protocol
- Service

A client enables your computer to connect to other computers.

Figure 3-6. Selecting a protocol to install



Select Network Protocol

Click the Network Protocol that you want to install, then click OK. If you have an installation disk for this device, click Have Disk.

Manufacturers:
- Banyan
- IBM
- Microsoft
- Novell

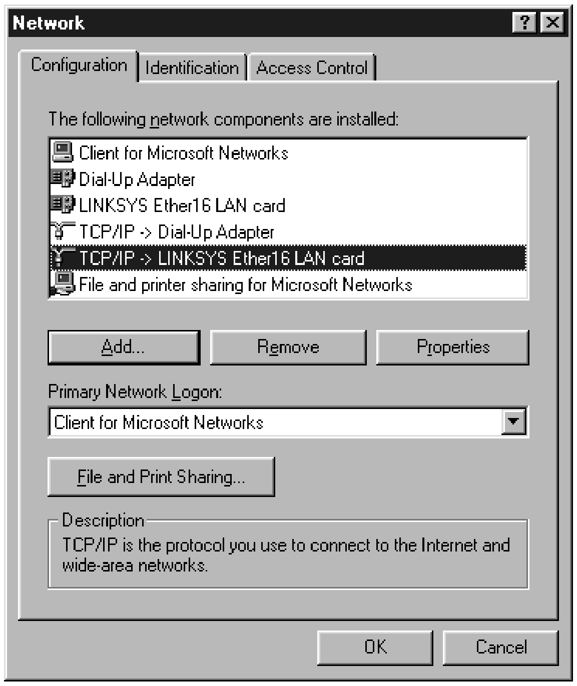
Network Protocols:
- Fast Infrared Protocol
- IPX/SPX-compatible Protocol
- Microsoft 32-bit DLC
- Microsoft DLC
- NetBEUI
- TCP/IP

Figure 3-7. Selecting a protocol to install

Configuring TCP/IP

If you have more than one networking device (for example, both an Ethernet card and a dialup networking modem), each appropriate hardware device should be "linked" to the TCP/IP protocol with an arrow, as shown in Figure 3-8. Select the TCP/IP protocol linked to the networking device that will be accessing the Samba network. When it is highlighted, click the Properties button.

After doing so, the TCP/IP Properties panel for that device is displayed, as shown in Figure 3-9.



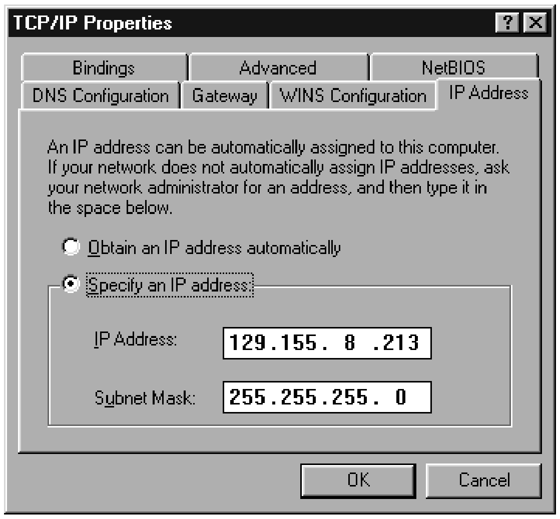
The following network components are installed:

- Client for Microsoft Networks
- Dial-Up Adapter
- LINKSYS Ether16 LAN card
- TCP/IP -> Dial-Up Adapter
- TCP/IP -> LINKSYS Ether16 LAN card
- File and printer sharing for Microsoft Networks

Primary Network Logon: Client for Microsoft Networks

Description
TCP/IP is the protocol you use to connect to the Internet and wide-area networks.

Figure 3-8. Selecting the correct TCP/IP protocol



An IP address can be automatically assigned to this computer. If your network does not automatically assign IP addresses, ask your network administrator for an address, and then type it in the space below.

Obtain an IP address automatically [✓ Specify an IP address:]

IP Address: 129.155. 8 .213

Subnet Mask: 255.255.255. 0

Figure 3-9. STCP/IP Properties panel

There are seven tabs near the top of this panel, and you will need to configure four of them:

- IP address
- DNS configuration
- WINS configuration
- Bindings

IP Address tab

The IP Address tab is shown in Figure 3-9. Press the "Specify an IP address" radio button and enter the client's address and subnet mask in the space provided. You or your network manager should have selected an address for the machine. The values should place the computer on the same subnet as the Samba server. For example, if the server's address is 192.168.236.86, and its network mask 255.255.255.0, you might use address 192.168.236.10 (if it is available) for the Windows 98 computer, along with the same netmask as the server. If you already use DHCP on your network to provide IP addresses to Windows machines, select the "Obtain an IP address automatically" button.

DNS Configuration tab

Domain Name Service (DNS) is responsible for translating Internet computer names such as *hobbes.example.com* into machine-readable IP addresses such as 192.168.236.10. There are two ways to accomplish this on a Windows 98 machine: you can specify a server to do the translation for you or you can keep a local list of name/address pairs to refer to.

Networks that are connected to the Internet typically use a server, since the hosts files required would otherwise be huge. For an unconnected LAN, the list of possible hosts is small and well-known and might be kept on a Unix machine in the */etc/hosts* file. If you are in doubt as to whether a DNS server is being used, or what its address might be, look at the file */etc/resolv.conf* on your Unix servers. Any machine using DNS will have this file, which looks like:

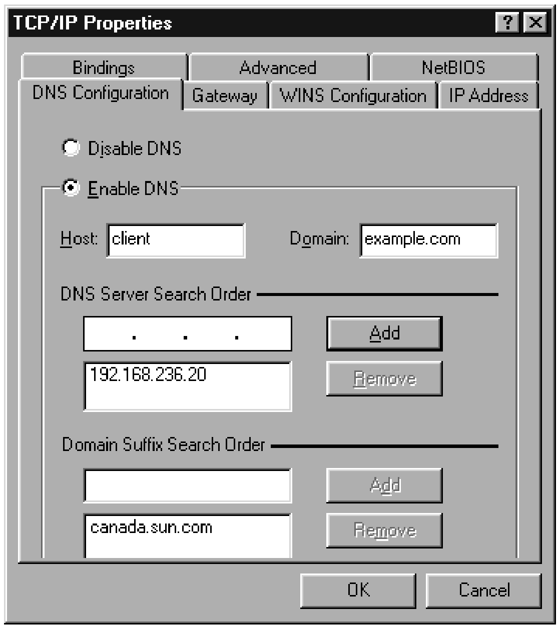
```
#resolv.conf
domain example.com
nameserver 127.0.0.1
nameserver 192.168.236.20
```

In the example shown, the second `nameserver` line in the list contains the IP address of another machine on the local network: 192.168.236.20. It's a good candidate for a DNS server.[*]

[*] We can disqualify the other address because every Unix machine has a localhost address of 127.0.0.1 whether it is connected to a network or not. This address is required for some system tools to operate correctly.

You must type the correct IP address of one or more DNS servers (note that you *cannot* use its Internet name, such as *dns.oreilly.com*) into the appropriate field in Figure 3-10. Be sure not to use 127.0.0.1—that will never be the correct DNS server address!

Try to select addresses on your own network. Any name servers listed in */etc/resolv.conf* should work, but you'll get better performance by using a server nearby. (If you don't find */etc/resolv.conf* files on your Unix machines, just disable DNS until you can find the address of at least one DNS server.) Let's assume you only have one DNS server, and its address is 192.168.236.20. Click the Enable DNS radio button, as shown in Figure 3-10, and add the server's address to the top DNS Server Search Order field.



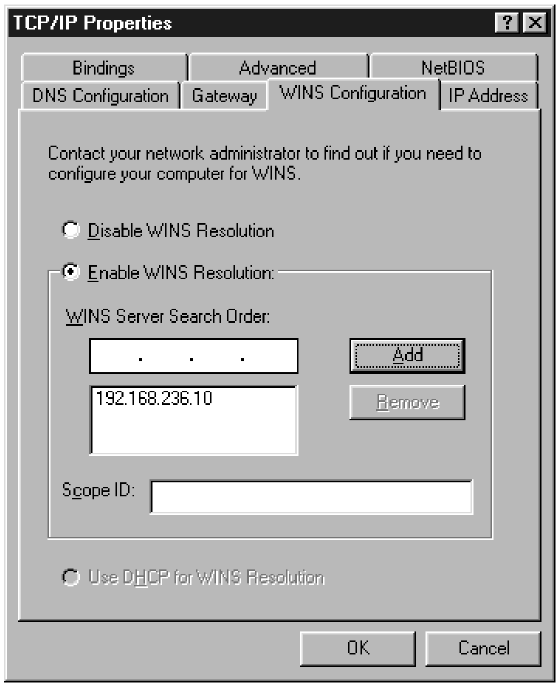
Disable DNS
[✓ Enable DNS]
Host: client
Domain: example.com
DNS Server Search Order
192.168.236.20
Domain Suffix Search Order
canada.sun.com

Figure 3-10. The DNS Configuration tab

Also, provide the name of the Windows 95/98 machine and the Internet domain you're in. You can safely ignore the Domain Suffix Search Order field for anything related to Samba.

WINS Configuration tab

WINS is the Windows Internet Name Service, its version of a NetBIOS name server. If you've enabled WINS on Samba, you must tell Windows the Samba server's address. If you are using WINS servers that are entirely Windows NT, enter each of them here as well. The dialog box shown after selecting the WINS Configuration tab is shown in Figure 3-11.



Contact your network administrator to find out if you need to configure your computer for WINS.

Disable WINS Resolution
Enable WINS Resolution: [✓]
WINS Server Search Order:
192.168.236.10
Scope ID:
Use DHCP for WINS Resolution

Figure 3-11. The WINS Configuration tab

Do *not* mix a Samba WINS server and a Windows NT server as a primary/backup combination in the WINS dialog. Because the two cannot replicate their databases, this will cause name resolution to perform incorrectly.

From here, select Enable WINS Resolution and enter the WINS server's address in the space provided, then press Add. Do not enter anything in the Scope ID field.

Hosts files

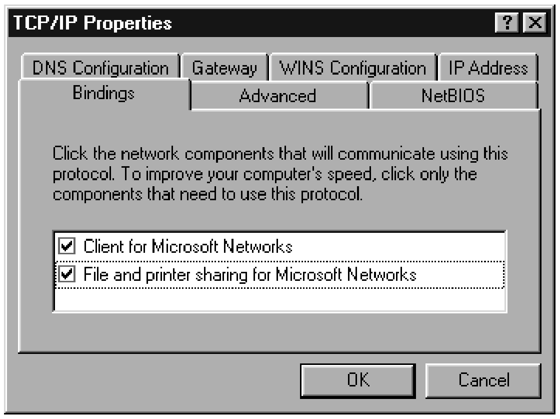
If you do not have either DNS or WINS, and you don't wish to use broadcast resolution, you'll need to provide a table of IP addresses and hostnames, in the standard Unix */etc/hosts* format. On a Windows machine, this goes in \WINDOWS\HOSTS under whichever drive you installed Windows on (typically C:\). A sample host file follows:

```
# 127.0.0.1             localhost
192.168.236.1       escrime.example.com escrime
192.168.236.2      riposte.example.com riposte
192.168.236.3       wizzin.example.com wizzin
192.168.236.4       touche.example.com touche
192.168.236.10     hobbes.example.com hobbes
```

You can copy this file directly from any of your Unix machines' */etc/hosts*; the format is identical. However, *you should only use hosts files in Windows as a last resort for name resolution.*

Check the bindings

The final tab to look at is Bindings, as shown in Figure 3-12.



Click the network components that will communicate using this protocol. To improve your computer's speed, click only the components that need to use this protocol.

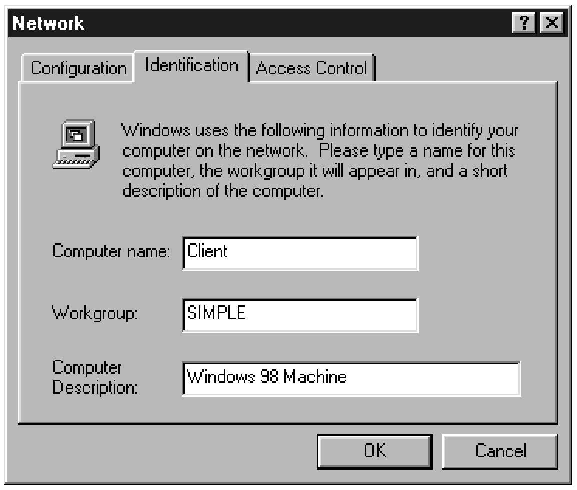
☑ Client for Microsoft Networks
☑ File and printer sharing for Microsoft Networks

Figure 3-12. The Bindings tab

You should have a check beside Client for Microsoft Networks, indicating that it's using TCP/IP. If you have "File and printer sharing for Microsoft Networks" in the dialog, it should also be checked, as shown in the figure.

Setting Your Name and Workgroup

Finally, press the OK button in the TCP/IP configuration panel, and you'll be taken back to the Network Configuration screen. Then select the Identification tab, which will take you to the dialog box shown in Figure 3-13.



Windows uses the following information to identify your computer on the network. Please type a name for this computer, the workgroup it will appear in, and a short description of the computer.

Computer name: Client

Workgroup: SIMPLE

Computer Description: Windows 98 Machine

Figure 3-13. The Identification tab

Here, for the second time, set your machine's name. This time, instead of your DNS hostname and domain, you're setting your NetBIOS name. However, it is best to make this the *same* as your hostname. Try not to make a spelling mistake: it can be very confusing to configure a machine if TCP thinks it's `fred` and SMB thinks its `ferd`!

You also set your workgroup name here. In our case, it's SIMPLE, but if you used a different one in Chapter 2, *Installing Samba on a Unix System*, when creating the Samba configuration file, use that here as well. Try to avoid calling it WORKGROUP or you'll be in the same workgroup as every unconfigured (or ill-configured) machine in the world.

Accessing the Samba Server

Click on the OK button to complete the configuration; you will need to reboot in order for your changes to take effect.

Now for the big moment. Your Samba server is running, and you have set up your Windows 95/98 client to communicate with it. After rebooting, log in and double-click the Network Neighborhood icon on the desktop. You should see your Samba server listed as a member of the workgroup, as shown in Figure 3-14.

Double-clicking the server name will show the resources that the server is offering to the network, as shown in Figure 3-15 (in this case a printer and the *test* directory).

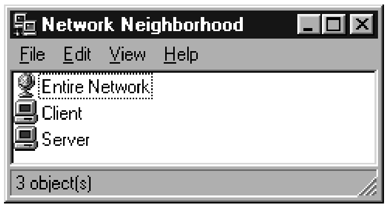


Figure 3-14. Windows Network Neighborhood

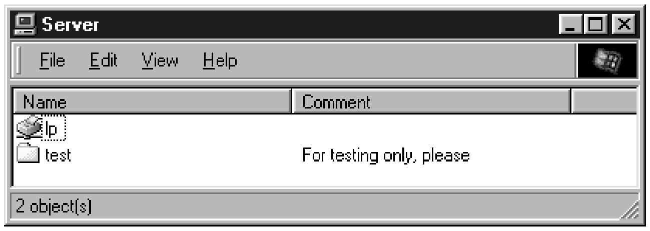


Figure 3-15. Shares on Server

If you are presented with a dialog requesting the password for a user `IPC$`, then Samba did not accept the password that was sent from the client. In this case, the username and the password that were created on the client side *must* match the username/password combination on the Samba server. If you are using Windows 98 or Windows NT Service Pack 3 or above, this is probably because the client is sending encrypted passwords instead of plaintext passwords. You can remedy this situation by performing two steps on the Samba server. First, add the following entry to the `[global]` section of your Samba configuration file: `encrypt password=yes`. Second, find the *smbpasswd* program on the samba server (it is located in */usr/local/samba/bin* by default) and use it to add an entry to Samba's encrypted password database. For example, to add user `steve` to Samba's encrypted password database, type ***`smbpasswd -a steve`***. The first time you enter this password, the program will output an error message indicating that the password database does not exist; it will then create the database, which is typically stored in */usr/local/samba/private/smbpasswd*.

If you don't see the server listed, start Windows Explorer (not Internet Explorer!) and select Map Network Drive from the Tools menu. This will give you a dialog box into which you can type the name of your server and the share `test` in the Windows UNC format: `\\`*`server`*`\test`, like we did in the first chapter. This should attempt to contact the Samba server and its temporary share. If things still aren't right, go to Chapter 9, *Troubleshooting Samba*, for troubleshooting assistance.

Setting Up Windows NT 4.0 Computers

Configuring Windows NT is a little different than configuring Windows 95/98. In order to use Samba with Windows NT, you will need both the Workstation service and the TCP/IP protocol. Both come standard with NT, but we'll work through installing and configuring them because they may not be configured correctly.

There are six basic steps:

1. Assign the machine a name.
2. Install the Workstation service.
3. Install the TCP/IP protocol.
4. Set the machine's name and IP address.
5. Configure the DNS and WINS name services.
6. Bind the protocol and service together.

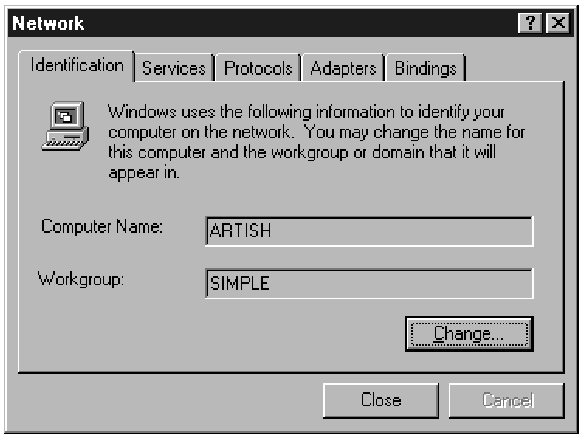
Basic Configuration

This section presents an outline of the steps to follow for getting Windows NT to cooperate with Samba. If you need more details on Windows NT network administration, refer to Craig Hunt and Robert Bruce Thompsom's *Windows NT TCP/IP Network Administration* (O'Reilly), an excellent guide. You should perform these steps as the "Administrator" user.

Name the machine

The first thing you need to do is to give the machine a NetBIOS name. From the Control Panel, double click on the Network icon. This will take you to the Network dialog box for the machine. The first tab in this dialog box should be the Identification tab, as illustrated in Figure 3-16.

Here, you need to identify your machine with a name (we use the name Artish here) and change the default workgroup to the one you specified in the *smb.conf* file of your Samba server. In this case, the workgroup name is SIMPLE. However, you cannot edit either name here (as you could in Windows 95/98), but instead must use the Change button below the two text fields. Pressing this button raises an Identification Changes dialog box, where you can reset the workgroup and the machine name, as shown in Figure 3-17.

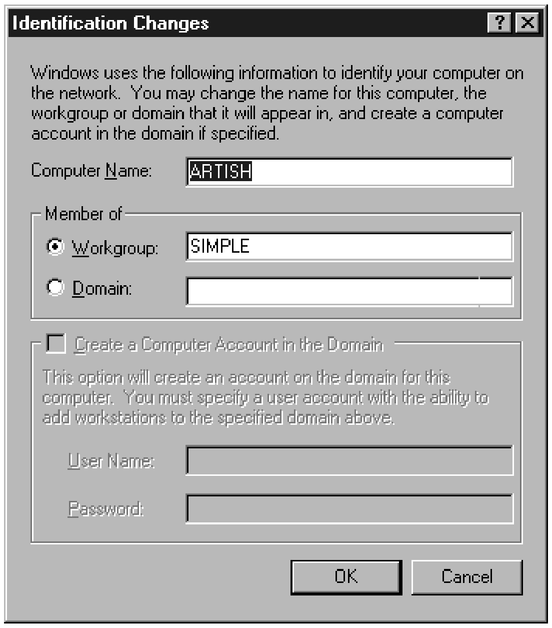


Windows uses the following information to identify your computer on the network. You may change the name for this computer and the workgroup or domain that it will appear in.

Computer Name: ARTISH

Workgroup: SIMPLE

Figure 3-16. Network panel Identification tab



Windows uses the following information to identify your computer on the network. You may change the name for this computer, the workgroup or domain that it will appear in, and create a computer account in the domain if specified.

Computer Name: ARTISH

Member of

[✓ Workgroup:] SIMPLE

Domain:

Create a Computer Account in the Domain

This option will create an account on the domain for this computer. You must specify a user account with the ability to add workstations to the specified domain above.

User Name:

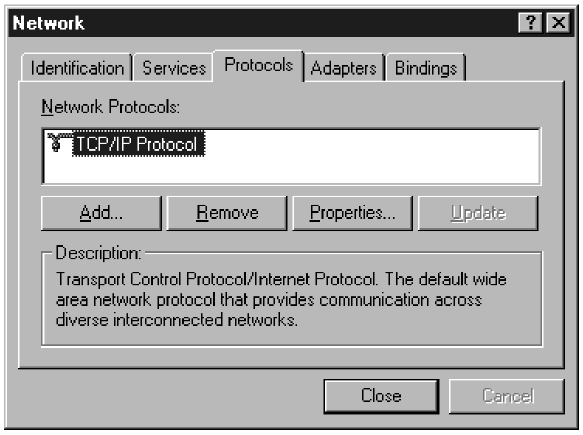
Password:

Figure 3-17. Changing the identification

A word of warning: you will have to set the machine name again later while configuring TCP/IP, so be sure that the two names match. The name you set here is the NetBIOS name. You're allowed to make it different from the TCP/IP hostname, but doing so is usually not a good thing. Don't worry that Windows NT forces the computer name and the workgroup to be all capital letters; it's smart enough to figure out what you mean when it connects to the network.

Installing the TCP/IP protocol

Next, select the Protocols tab in the Network dialog box, and look to see if you have the TCP/IP protocol installed, as shown in Figure 3-18.



Transport Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

Figure 3-18. The Protocols tab

If the protocol is not installed, you need to add it. Press the Add button, which will display the Select Network Protocol dialog box shown in Figure 3-19. Unlike Windows 95/98, you should immediately see the TCP/IP protocol as one of the last protocols listed.

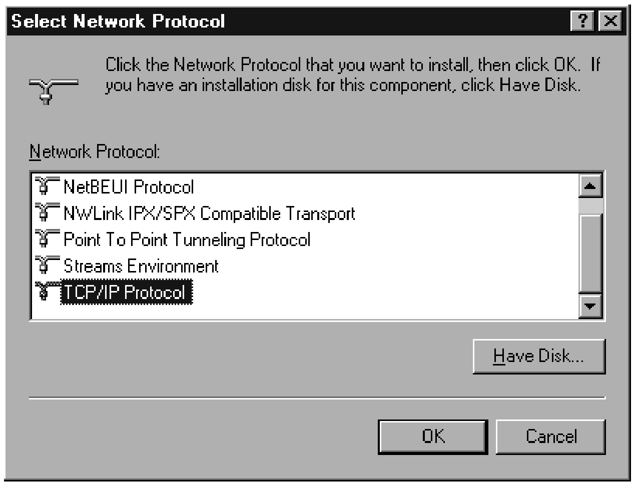
Select TCP/IP as the protocol and confirm it. If possible, install only the TCP/IP protocol. You usually do not want NetBEUI installed because this causes the machine to look for services under two different protocols, only one of which is likely in use.*

Installing the Workstation service

After installing TCP/IP, press the Services tab in the Network panel and check that you have a Workstation service, as shown at the end of the list in Figure 3-20.

This service is actually the Microsoft Networking Client, which allows the machine to access SMB services. The Workstation service is mandatory. The service is installed by default on both Windows NT Workstation 4.0 and Server 4.0. If it's not

* A common occurrence: after looking at the unused protocol for a while, the machine will time out and try the good one. This fruitless searching gives you terrible performance and mysterious delays.

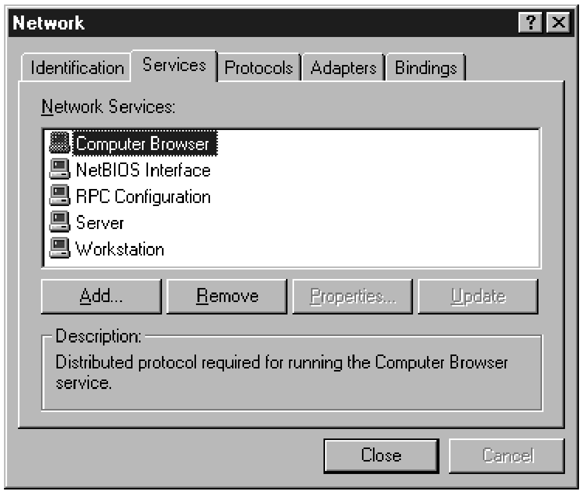

Select Network Protocol

Click the Network Protocol that you want to install, then click OK. If you have an installation disk for this component, click Have Disk.

Network Protocol:

- NetBEUI Protocol
- NWLink IPX/SPX Compatible Transport
- Point To Point Tunneling Protocol
- Streams Environment
- TCP/IP Protocol

Figure 3-19. Select Network Protocol dialog box


Network

Network Services:

- Computer Browser
- NetBIOS Interface
- RPC Configuration
- Server
- Workstation

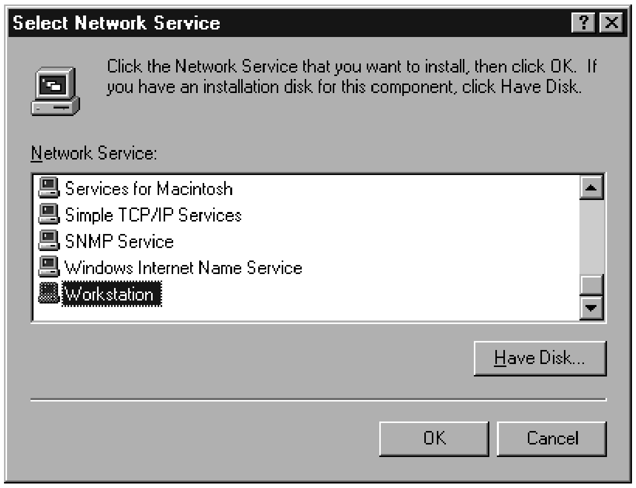
Description:
Distributed protocol required for running the Computer Browser service.

Figure 3-20. Network Services panel dialog box

there, you can install it much like TCP/IP. In this case you need to press the Add button and then select Workstation Service, as shown in Figure 3-21.

Configuring TCP/IP

After you've installed the Workstation service, return to the Protocols tab and select the TCP/IP Protocol entry in the window. Then click the Properties button


Select Network Service

Click the Network Service that you want to install, then click OK. If you have an installation disk for this component, click Have Disk.

Network Service:

Services for Macintosh
Simple TCP/IP Services
SNMP Service
Windows Internet Name Service
Workstation

Figure 3-21. Select Network Service dialog box

below the window. The Microsoft TCP/IP Protocol panel will be displayed. There are five tabs on the Windows NT panel, and (like Windows 95/98) you will need to work on three of them:

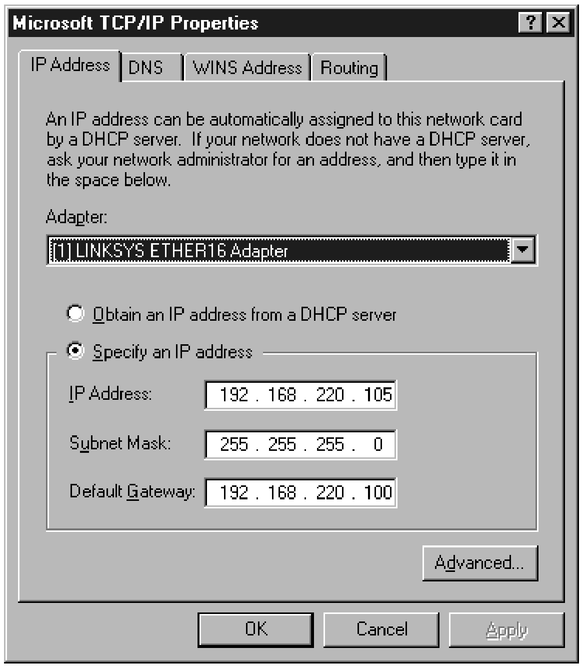
- IP address
- DNS
- WINS address

IP Address tab

The IP Address tab is shown in Figure 3-22.

Select the "Specify an IP address" radio button and enter the computer's address and subnet mask in the space provided for the proper adapter (Ethernet card). You or your network manager should have selected an address for the client on the same subnet (LAN) as the Samba server. For example, if the server's address is 192.168.236.86 and its network mask 255.255.255.0, you might use the address 192.168.236.10, if it is available, for the NT workstation, along with the same netmask. If you use DHCP on your network, select the "Obtain an IP Address from a DHCP server" button.

If you don't have an IP address to use, and you are on a network by yourself, steal ours, as the 192.168.*x.x* subnet is specifically reserved by the Internic for LANs. If you're not by yourself, see your system administrator for some available addresses on your network.



An IP address can be automatically assigned to this network card by a DHCP server. If your network does not have a DHCP server, ask your network administrator for an address, and then type it in the space below.

Adapter: [1] LINKSYS ETHER16 Adapter

○ Obtain an IP address from a DHCP server
[✓ Specify an IP address]

IP Address: 192 . 168 . 220 . 105
Subnet Mask: 255 . 255 . 255 . 0
Default Gateway: 192 . 168 . 220 . 100

Figure 3-22. Microsoft TCP/IP Properties for Windows NT

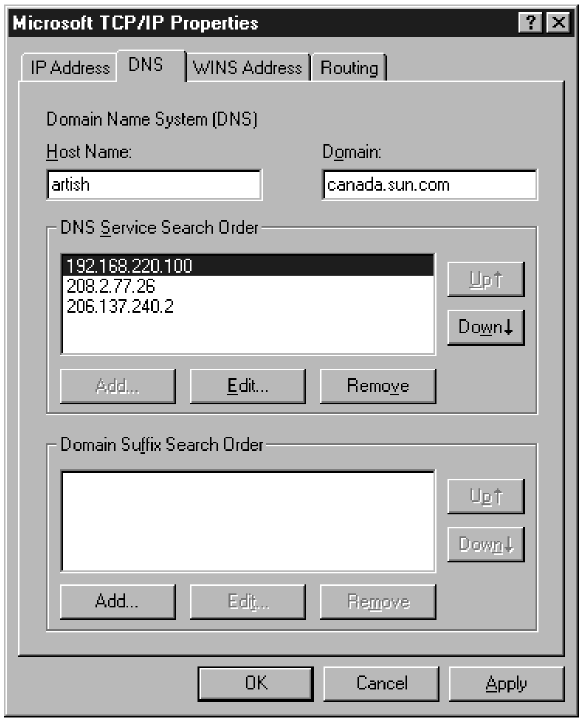
The gateway field refers to a machine typically known as a *router*. If you have routers connecting multiple networks, you should put in the IP address of the one on your subnet.

DNS tab

Next we go to the tab for DNS, as shown in Figure 3-23. This brings up the DNS panel.

The Domain Name System (DNS) is responsible for translating human-readable computer names such as *atrish.example.com* into IP addresses such as 192.168.236.10. There are two ways to accomplish this on a NT machine. First, you can specify a DNS server to do the translation for you, or you can keep a local list of name/address pairs for your workstation to refer to.

For a LAN that's not on the Internet, the list of possible hosts is typically small and well known, and may be kept in a file locally. Networks that are connected to the Internet typically use DNS service since it isn't possible to guess ahead of time what addresses you might be accessing out on the net. If you are in doubt as to whether a DNS server is being used, or what its address might be, look at the file



Host Name: artish
Domain: canada.sun.com

DNS Service Search Order:

```
192.168.220.100
208.2.77.26
206.137.240.2
```

Domain Suffix Search Order:

Figure 3-23. The DNS panel

/etc/resolv.conf on your Samba server: any machine using DNS will have this file. It looks like the following:

```
#resolv.conf
domain example.com
nameserver 127.0.0.1
nameserver 192.168.236.20
```

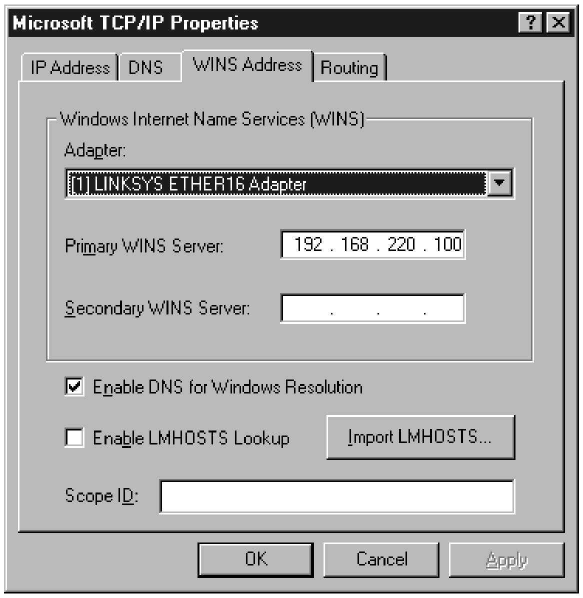
In this example, the first nameserver in the list is 127.0.0.1, which indicates that the Samba server is also a DNS server for this LAN.[*] In that case, you would use its network IP address (not 127.0.0.1, its localhost address) when filling in the DNS Configuration dialog box. Otherwise, use the other addresses you find in the lines beginning with `nameserver`. Try to select ones on your own network. Any name servers listed in */etc/resolv.conf* should work, but you'll get better performance by using a server nearby.

[*] The address 127.0.0.1 is known as the *localhost* address, and always refers to itself. For example, if you type `ping 127.0.0.1` on a Unix server, you should always get a response, as you're pinging the host itself.

Finally, enter the machine name once more, making sure that it's the same one listed in the Identification tab of the Network dialog box (before the NetBIOS name). Also, enter the DNS domain on which this machine resides. For example, if your workstation has a domain name such as *example.com*, enter it here. You can safely ignore the other options.

WINS Address tab

If you are not using a DNS server, you still need a way of translating NetBIOS names to addresses and back again. We recommend that you configure both DNS and WINS; NT has a preference for WINS and WINS can use DNS as a fallback if it cannot resolve any machine address. The WINS Address tab is shown in Figure 3-24.


Microsoft TCP/IP Properties

IP Address | DNS | WINS Address | Routing

Windows Internet Name Services (WINS)

Adapter: [1] LINKSYS ETHER16 Adapter

Primary WINS Server: 192 . 168 . 220 . 100

Secondary WINS Server: . . .

[✓] Enable DNS for Windows Resolution

[ ] Enable LMHOSTS Lookup

Scope ID:

Figure 3-24. The WINS Address tab

If you have a WINS server, enter its address in the space marked Primary WINS Server. If your Samba server is providing WINS service (in other words, you have the line `wins service = yes` in the *smb.conf* file of your Samba server), provide the Samba server's IP address here. Otherwise, provide the address of another WINS server on your network.

You probably noticed that there is a field here for the adaptor; this field must specify the Ethernet adaptor that you're running TCP/IP on so that WINS will provide name service on the correct network. If you have both a LAN and a dialup adaptor, make sure you have the LAN's adaptor here.

Finally, select the "Enable DNS for Windows Resolution" checkbox, so WINS will try DNS as a fallback if it can't find a name. You can safely ignore the other options.

Hosts files

If you don't have either DNS or WINS, and you don't wish to use broadcast name resolution, you'll need to provide a table of IP addresses and hosts names, in standard Unix */etc/hosts* format. We recommend against this because maintenance of this file on any dynamic network is troublesome, but we will explain it just the same. The Windows host file should appear in the *\WINDOWS\HOSTS* directory of whatever local drive Windows is installed on. A sample follows:

```
127.0.0.1       localhost
192.168.236.1    escrime    escrime.example.com
192.168.236.2    riposte    riposte.example.com
192.168.236.3    wizzin     wizzin.example.com
192.168.236.4    touche     touche.example.com
192.168.236.5    gurgi      gurgi.example.com
192.168.236.6    jessiac    jessiac.example.com
192.168.236.7    skyline    skyline.example.com
```

If you wish, you can copy the contents directly from the Samba server's */etc/hosts*. The format is identical. This file will then serve the same purpose as the hosts file on the Unix server. Again, *hosts* files on Windows should only be used as a last resort.

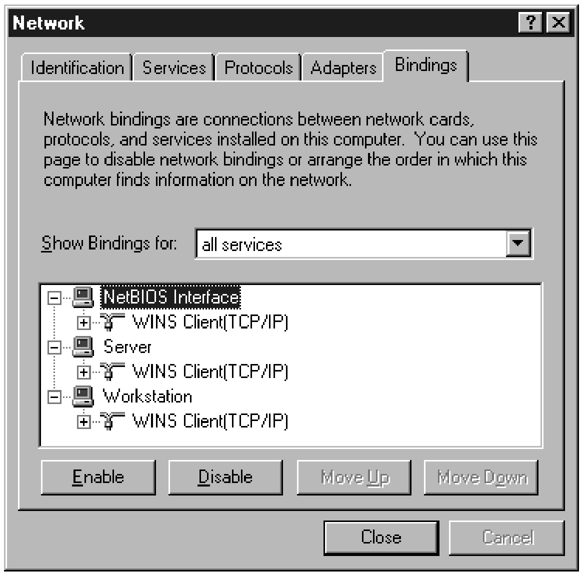
Bindings

The term *bindings* is a way of saying "connected together at configuration time." It means that the TCP/IP protocol will channel through the Ethernet card (instead of, say, a dialup connection), and is actually connected properly. If you return to the Network dialog box and set the Show field to "all services" and click on all the + buttons in the tree, you should see a display similar to Figure 3-25.

This means that the Workstation, Server, and NetBIOS interface services are connected to the WINS client. This is the correct binding for Microsoft TCP/IP.

Connecting to the Samba Server

You can safely leave the default values for the remainder of the tabs in the Network dialog box. Click on the OK button to complete the configuration. Once the proper files are loaded (if any), you will need to reboot in order for your changes to take effect.



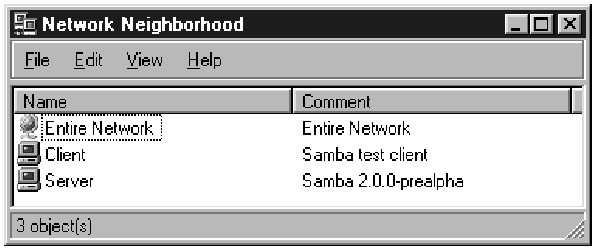
Network bindings are connections between network cards, protocols, and services installed on this computer. You can use this page to disable network bindings or arrange the order in which this computer finds information on the network.

Show Bindings for: all services

NetBIOS Interface
WINS Client(TCP/IP)
Server
WINS Client(TCP/IP)
Workstation
WINS Client(TCP/IP)

Figure 3-25. Service bindings

Now for the big moment. Your Samba server is running and you have set up your NT client to communicate with it. After the machine reboots, login and double-click the Network Neighborhood icon on the desktop, and you should see your Samba server listed as a member of the workgroup, as shown in Figure 3-26.



Name	Comment
Entire Network	Entire Network
Client	Samba test client
Server	Samba 2.0.0-prealpha

3 object(s)

Figure 3-26. Windows NT Network Neighborhood

Double-clicking the server name will show the resources that the server is offering to the network, as shown in Figure 3-27. In this case, the test and the default printer are offered to the Window NT workstation. For more information, see the warning under the "Accessing the Samba Server" section, earlier in this chapter.

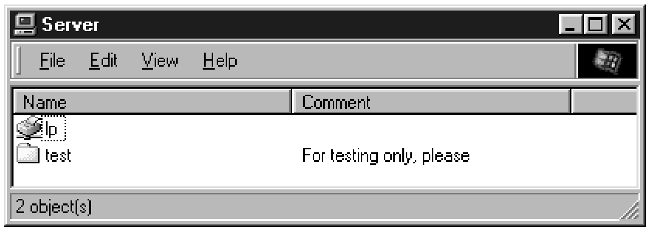


Figure 3-27. Server's shares

If you are presented with a dialog requesting the password for a user `IPC$`, then Samba did not accept the password that was sent from the client. In this case, the username and the password that were created on the client side *must* match the username/password combination on the Samba server. If you are using Windows 98 or Windows NT Service Pack 3 or above, this is probably because the client is sending encrypted passwords instead of plaintext passwords. You can remedy this situation by performing two steps on the Samba server. First, add the following entry to the `[global]` section of your Samba configuration file: `encrypt password=yes`. Second, find the *smbpasswd* program on the samba server (it is located in */usr/local/samba/bin* by default) and use it to add an entry to Samba's encrypted password database. For example, to add user `steve` to Samba's encrypted password database, type *`smbpasswd -a steve`*. The first time you enter this password, the program will output an error message indicating that the password database does not exist; it will then create the database, which is typically stored in */usr/local/samba/private/smbpasswd.*

If you don't see the server listed, don't panic. Start the Windows NT Explorer (not Internet Explorer!) and select Map Network Drive from the Tools menu. A dialog box appears that allows you to type the name of your server and its share directory in Windows format. For example, you would enter `\\`*`server`*`\`*`temp`* if your server happened to be named "server." If things still aren't right, go directly to the section "The Fault Tree" in Chapter 9, to see if you can troubleshoot what is wrong with the network.

If it works, congratulations! Try writing to the server and sending data to the network printer. You will be pleasantly surprised how seamlessly everything works! Now that you've finished setting up the Samba server and its clients, we can starting talking about how Samba works and how to configure it to your liking.

An Introduction to SMB/CIFS

We'll wrap up this chapter with a short tutorial on SMB/CIFS. SMB/CIFS is the protocol that Windows 95/98 and NT machines use to communicate with the Samba server and each other. At a high level, the SMB protocol suite is relatively simple. It includes commands for all of the file and print operations that you might do on a local disk or printer, such as:

- Opening and closing a file
- Creating and deleting files and directories
- Reading and writing a file
- Searching for files
- Queueing and dequeueing files to a print spool

Each of these operations can be encoded into an SMB message and transmitted to and from a server. The original name SMB comes from their data format: these are versions of the standard DOS system-call data structures, or *Server Message Blocks*, redesigned for transmitting to another machine across a network.

SMB Format

Richard Sharpe of the Samba team defines SMB as a "request-response" protocol.[*] In effect, this means that a client sends an SMB request to a server, and the server sends an SMB response back to the client. Rarely does a server send a message that is not in response to a client.

An SMB message is not as complex as you might think. Let's take a closer look at the internal structure of such a message. It can be broken down into two parts: the *header*, which is a fixed size, and the *command string*, whose size can vary dramatically based on the contents of the message.

SMB header format

Table 3-1 shows the format of an SMB header. SMB commands are not required to use all the fields in the SMB header. For example, when a client first attempts to connect to a server, it does not yet have a tree identifier (TID) value—one is assigned after it successfully connects—so a null TID (0xFFFF) is placed in its header field. Other fields may be padded with zeros when not used.

[*] See *http://anu.samba.org/cifs/docs/what-is-smb.html* for Richard's excellent summary of SMB.

The fields of the SMB header are listed in Table 3-1.

Table 3-1. SMB Header Fields

Field	**Size (bytes)**	**Description**
`0xFF 'SMB'`	1	Protocol identifier
`COM`	1	Command code, from 0x00 to 0xFF
`RCLS`	1	Error class
`REH`	1	Reserved
`ERR`	2	Error code
`REB`	1	Reserved
`RES`	14	Reserved
`TID`	2	Tree identifier; a unique ID for a resource in use by client
`PID`	2	Caller process ID
`UID`	2	User identifier
`MID`	2	Multiplex identifier; used to route requests inside a process

SMB command format

Immediately after the header is a variable number of bytes that constitute an SMB command or reply. Each command, such as Open File (COM field identifier: `SMBopen`) or Get Print Queue (`SMBsplretq`), has its own set of parameters and data. Like the SMB header fields, not all of the command fields need to be filled, depending on the specific command. For example, the Get Server Attributes (`SMBdskattr`) command sets the WCT and BCC fields to zero. The fields of the command segment are shown in Table 3-2.

Table 3-2. SMB Command Contents

Field	Size in Bytes	Description
`WCT`	1	Word count
`VWV`	Variable	Parameter words (size given by WCT)
`BCC`	2	Parameter byte count
`DATA`	Variable	Data (size given by BCC)

Don't worry if you don't understand each of these fields; they are not necessary for using Samba at an administrator level. However, they do come in handy when debugging system messages. We will show you some of the more common SMB messages that clients and servers send using a modified version of *tcpdump* later in this section. (If you would like an SMB sniffer with a graphical interface, try "ethereal," which uses the GTK libraries; see the Samba homepage for more information on this tool.)

If you would like more information on each of the commands for the SMB protocol, see the SMB/CIFS documentation at *ftp://ftp.microsoft.com/developr/drg/CIFS/*.

SMB variations

The SMB protocol has been extended with new commands several times since its inception. Each new version is backwards compatible with the previous versions. This makes it quite possible for a LAN to have various clients and servers running different versions of the SMB protocol at once.

Table 3-3 outlines the major versions of the SMB protocol. Within each "dialect" of SMB are many sub-versions that include commands supporting particular releases of major operating systems. The ID string is used by clients and servers to determine what level of the protocol they will speak to each other.

Table 3-3. SMB Protocol Dialects

Protocol Name	ID String	Used By
Core	`PC NETWORK PROGRAM 1.0`	
Core Plus	`MICROSOFT NETWORKS 1.03`	
LAN Manager 1.0	`LANMAN1.0`	
LAN Manager 2.0	`LM1.2X002`	
LAN Manager 2.1	`LANMAN2.1`	
NT LAN Manager 1.0	`NT LM 0.12`	Windows NT 4.0
Samba's NT LM 0.12	`Samba`	Samba
Common Internet File System	`CIFS 1.0`	Windows 2000

Samba implements the `NT LM 0.12` specification for NT LAN Manager 1.0. It is backwards compatible with all of the other SMB variants. The CIFS specification is, in reality, LAN Manager 0.12 with a few specific additions.

SMB Clients and Servers

As mentioned earlier, SMB is a client/server protocol. In the purest sense, this means that a client sends a request to a server, which acts on the request and returns a reply. However, the client/server roles can often be reversed, sometimes within the context of a single SMB session. For example, consider the two Windows 95/98 computers in Figure 3-28. The computer named WIZZIN shares a printer to the network, and the computer named ESCRIME shares a disk directory. WIZZIN is in the client role when accessing ESCRIME's network drive, and in the server role when printing a job for ESCRIME.

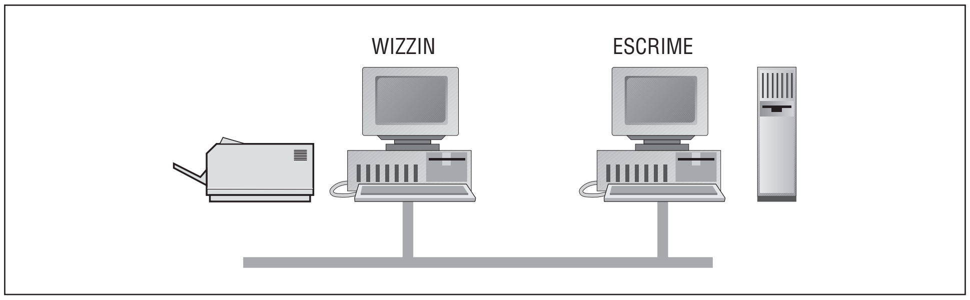


Figure 3-28. Two computers that both have resources to share

This brings out an important point in Samba terminology:

- A *server* is a machine with a resource to share.
- A *client* is a machine that wishes to use that resource.
- A server can be a client (of another computer's resource) at any given time.

Note that there are no implications as to the amount of resources that make up a server, or whether it has a large disk space or fast processor. A server could be an old 486 with a printer attached to it, or it could be an UltraSparc station with a 10 gigabyte disk service.

Microsoft Windows products have both the SMB client and server built in to the operating system. Wndows NT 4.0 uses a newer SMB protocol than Windows for Workgroups, and it offers an enhanced form of network security which will be discussed in Chapter 6. In addition, there are a large number of commercial SMB server products available from companies such as Sun, Compaq, SCO, Hewlett-Packard, Syntax, and IBM. Unfortunately, on the client side there are far fewer offerings, limited mainly to Digital Equipment's Pathworks product, and of course, Samba.

A Simple SMB Connection

Before we close this chapter, let's take a look at a simple SMB connection. This is some pretty technical data—which isn't really necessary to administer Samba—so you can skip over it if you like. We present this information largely as a way to help you get familiar with how the SMB protocol negotiates connections with other computers on the network.

There are four steps that the client and server must complete in order to establish a connection to a resource:

1. Establish a virtual connection.
2. Negotiate the protocol variant to speak.

3. Set session parameters.
4. Make a tree connection to a resource.

We will examine each of these steps through the eyes of a useful tool that we mentioned earlier: the modified *tcpdump* that is available from the Samba web site.

You can download this program at *samba.org* in the *samba/ftp/tcpdump-smb* directory; the latest version as of this writing is 3.4-5. Use this program as you would use the standard *tcpdump* application, but add the `-s 1500` switch to ensure that you get the whole packet and not just the first few bytes.

Establishing a virtual connection

When a user first makes a request to access a network disk or send a print job to a remote printer, NetBIOS takes care of making a connection at the session layer. The result is a bidirectional virtual channel between the client and server. In reality, there are only two messages that the client and server need to establish this connection. This is shown in the following example session request and response, as captured by *tcpdump*:

```
>>> NBT Packet
NBT Session Request
Flags=0x81000044
Destination=ESCRIME      NameType=0x20 (Server)
Source=WIZZIN            NameType=0x00 (Workstation)

>>> NBT Packet
NBT Session Granted
Flags=0x82000000
```

Negotiating the Protocol Variant

At this point, there is an open channel between the client and server. Next, the client sends a message to the server to negotiate an SMB protocol. As mentioned earlier, the client sets its tree identifier (TID) field to zero, since it does not yet know what TID to use. A *tree identifier* is a number that represents a connection to a share on a server.

The command in the message is `SMBnegprot`, a request to negotiate a protocol variant that will be used for the entire session. Note that the client sends to the server a list of all of the variants that it can speak, not vice versa.

The server responds to the `SMBnegprot` request with an index into the list of variants that the client offered, starting with index 0, or with the value 0xFF if none of the protocol variants are acceptable. Continuing this example, the server responds with the value 5, which indicates that the `NT LM 0.12` dialect will be used for the remainder of the session:

```
>>> NBT Packet
NBT Session Packet
Flags=0x0
Length=154

SMB PACKET: SMBnegprot (REQUEST)
SMB Command   =  0x72
Error class   =  0x0
Error code    =  0
Flags1        =  0x0
Flags2        =  0x0
Tree ID       =  0
Proc ID       =  5371
UID           =  0
MID           =  385
Word Count    =  0
Dialect=PC NETWORK PROGRAM 1.0
Dialect=MICROSOFT NETWORKS 3.0
Dialect=DOS LM1.2X002
Dialect=DOS LANMAN2.1
Dialect=Windows for Workgroups 3.1a
Dialect=NT LM 0.12

>>> NBT Packet
NBT Session Packet
Flags=0x0
Length=69

SMB PACKET: SMBnegprot (REPLY)
SMB Command   =  0x72
Error class   =  0x0
Error code    =  0
Flags1        =  0x0
Flags2        =  0x1
Tree ID       =  0
Proc ID       =  5371
UID           =  0
MID           =  385
Word Count    =  02
[000] 05 00
```

Set Session and Login Parameters

The next step is to transmit session and login parameters for the session. This includes the account name and password (if there is one), the workgroup name, the maximum size of data that can be transferred, and the number of pending requests that may be in the queue at any one time.

In the following example, the Session Setup command presented allows for an additional SMB command to be piggybacked onto it. The letter X at the end of the command name indicates this, and the hexadecimal code of the second command is given in the `Com2` field. In this case the command is `0x75`, which is the Tree Connect and X command. The `SMBtconX` message looks for the name of the resource in the *smb_buf* buffer. (This is the last field listed in the following request.) In this example, *smb_buf* contains the string `\\ESCRIME\PUBLIC`, which is the full pathname to a shared directory on node ESCRIME. Using the "and X" commands like this speeds up each transaction, since the server doesn't have to wait on the client to make a second request.

Note that the TID is still zero. The server will provide a TID to the client once the session has been established and a connection has been made to the requested resource. In addition, note that the password is sent in the open. We can change this later using encrypted passwords:

```
>>> NBT Packet
NBT Session Packet
Flags=0x0
Length=139

SMB PACKET: SMBsesssetupX (REQUEST)
SMB Command   =  0x73
Error class   =  0x0
Error code    =  0
Flags1        =  0x10
Flags2        =  0x0
Tree ID       =  0
Proc ID       =  5371
UID           =  1
MID           =  385
Word Count    =  13
Com2=0x75
Res1=0x0
Off2=106
MaxBuffer=2920
MaxMpx=2
VcNumber=0
SessionKey=0x1FF2
CaseInsensitivePasswordLength=1
CaseSensitivePasswordLength=1
Res=0x0
Capabilities=0x1
Pass1&Pass2&Account&Domain&OS&LanMan=
  KRISTIN PARKSTR Windows 4.0 Windows 4.0
```

```
PassLen=2
Passwd&Path&Device=
smb_bcc=22
smb_buf[]=\\ESCRIME\PUBLIC
```

Making Connection to a Resource

For the final step, the server returns a TID to the client, indicating that the user has been authorized access and that the resource is ready to be used. It also sets the *ServiceType* field to "A" to indicate that this is a file service. Available service types are:

- "A" for a disk or file
- "LPT1" for a spooled output
- "COMM" for a direct-connect printer or modem
- "IPC" for a named pipe

The output is:

```
>>> NBT Packet
NBT Session Packet
Flags=0x0
Length=78

SMB PACKET: SMBsesssetupX (REPLY)
SMB Command   =  0x73
Error class   =  0x0
Error code    =  0
Flags1        =  0x80
Flags2        =  0x1
Tree ID       =  121
Proc ID       =  5371
UID           =  1
MID           =  385
Word Count    =  3
Com2=0x75
Off2=68
Action=0x1
[000] Unix Samba 1.9.1
[010] PARKSTR

SMB PACKET: SMBtconX (REPLY) (CHAINED)
smbvwv[]=
Com2=0xFF
Off2=78
smbbuf[]=
ServiceType=A:
```

Now that a TID has been assigned, the client may issue any sort of command that it would use on a local disk drive. It can open files, read and write to them, delete them, create new files, search for filenames, and so on.