CHAPTER 18

Interconnecting Network Operating Systems

The Network+ Certification exam expects you to know how to

- 2.13 Identify the purpose of network services and protocols (for example: NFS—Network File System, AFP—Apple File Protocol, LPD—Line Printer Daemon)
- 2.16 Define the function of the following remote access protocols and services: RDP (Remote Desktop Protocol)
- 3.4 Given a remote connectivity scenario [comprising] a protocol, an authentication scheme, and physical connectivity, configure the connection. Includes connection to the following servers: UNIX/Linux/MAC OS X Server, NetWare, Windows, AppleShare IP (Internet Protocol)
- 4.5 Given a troubleshooting scenario between a client and the following server environments, identify the cause of a stated problem: UNIX/Linux/Mac OS X Server, NetWare, Windows, AppleShare IP

To achieve these goals, you must be able to

- Understand the interconnection issues of Windows 9*x*, NT, 2000, XP, and 2003 with other network operating systems
- Learn the interconnection issues of NetWare relating to other network operating systems
- Explain the interconnection issues of Macintosh with other network operating systems
- Describe the interconnection issues of UNIX/Linux with other network operating systems

Many real-world networks blend systems running many different operating systems. If you came to my office today, for example, you'd find the following systems running with these operating systems:

- One Windows Server 2003 (new file server)
- Two Windows 2000 Servers (domain controllers)
- One Novell NetWare 5 Server (file server)
- One Novell NetWare 6 Server (file server)

- One Fedora Linux (testing system)
- One SuSE Linux (Samba server)
- Ten Windows XP Professional (workstations)
- Three Windows 2000 Professional (workstations)
- Two Windows XP Home (laptops)
- Two Macs running OS X (graphics workstations)
- Five systems running different versions of Windows 9*x* (testing systems)

While Total Seminars may have a more heterogeneous network than most real-world offices, it's more common than not to walk into a network and find many different operating systems all at work. The challenge you face is to make these different operating systems share their resources!

The widespread adoption of TCP/IP makes today's world a world of interoperability. As long as one uses TCP/IP applications such as WWW, FTP, and SMTP/POP e-mail, it doesn't matter what brand name of network operating system (NOS) is running on either the server or the client. I use the Safari web browser on my Macintosh laptop, for example, and have no problem accessing a web page on a Windows 2003 system running the Internet Information Services (IIS) web server program. My old Windows 98 system running the WSFTP_LE FTP client application easily transfers files to and from a Fedora Linux computer running the WU-FTP FTP server program. With TCP/IP applications, the NOS is unimportant and invisible.

Historical/Conceptual

This wasn't always the case with networking. In the days before all these wonderful, universal TCP/IP applications, each NOS had its own, usually proprietary method of enabling clients to access shared resources on servers—in particular shared folders and printers. To connect to a particular NOS server you had to have the same brand of client. Interoperability wasn't something that folks at Novell, Apple, or Microsoft seriously considered or even desired. Microsoft would be thrilled if every server and client system ran only Windows. The Novell folks would love for every server in the world to run NetWare; and the Apple people know that *everyone* should own a Mac.

The odd men out in this "we don't want to work with anyone else" attitude were the makers of the UNIX and later Linux operating systems. All versions of UNIX/Linux use TCP/IP at their core and as a result all versions of UNIX/Linux run TCP/IP applications. Long before the Internet dominated the world, UNIX LANs happily used TCP/IP applications to share folders and printers. Many of these applications (with names such as NFS and LPD) were adopted by Microsoft, Novell, and Apple into their network operating systems and became some of the first methods of interoperability among different network operating systems.

One issue that all the NOS makers learned early on was that interoperability makes their network operating systems more attractive to potential purchasers. No one wants to adopt a new NOS that requires them to replace all of their clients or even get rid of their current servers. By creating methods that enabled one brand of client-access shared resources on another brand of server, NOS makers could enable a new adopter to transition from one NOS to another. As a result, every NOS now has some method to enable their servers or clients to access shared folders and printers on other network operating systems.

This chapter provides an overview of the many ways different network operating systems interconnect to share folders and printers. There's no way you can expect to memorize all the steps necessary to make every type of NOS share every type of possible resource with every type of NOS client that's available. There are just way too many NOS combinations out there! Entire books are written on topics such as "Making NetWare Servers Share Folders with Windows Clients." To make this chapter more manageable, I've selected the network operating systems that we most commonly use to act as servers: Microsoft Windows, Novell NetWare, Apple Macintosh, and UNIX/Linux; and the three most common client systems: Windows, UNIX/Linux, and Macintosh. In each section, I'll discuss how one type of server connects to each type of client. Because this chapter is about interconnections among different network operating systems, I'm not going to include Linux clients when I discuss connecting to Linux servers, Macintosh clients when discussing Macintosh servers, or Windows clients in the Windows servers section. Also, because NetWare is a server-only NOS, I will cover all three client types when we cover NetWare servers.

So, what exactly are the serving systems we want to discuss? Clearly we need to cover Windows NT, 2000/2003, and Windows XP. I'll also discuss connecting to Windows 9*x* systems—don't forget they can share folders and printers, too! Next up: NetWare 3.*x*, followed by NetWare 4.*x*/5.*x*/6.*x* servers. The interconnectivity methods of NetWare 3.*x* are quite different from those of NetWare 4.*x*/5.*x*/6.*x*. Macintosh serving systems are next-to-last, followed, finally, by UNIX/Linux. It's a tall order, but an interesting one, too! Let's get started.

Test Specific

Connecting to Windows

The easiest way to get a non-Windows operating system like Macintosh or UNIX/Linux to connect to a Windows computer is to make that non-Windows system look like another Windows computer to the system you are trying to access. The challenge to this is that Windows systems themselves have evolved over the years so the method used to connect a Macintosh to access a Windows 98 SE system's shared folder is different from the way you connect that same Macintosh to a Windows Server 2003 system running Active Directory. But Windows is the predominant operating system, so the latest versions of both Macintosh and UNIX/Linux systems come with plenty of software enabling them to make most of these connections easily. Sadly, not everyone owns the latest versions of these operating systems, so we'll need to look at a few of the methods that older Macs and UNIX/Linux systems used to connect to Windows.

For a non-Windows computer to act like a Windows computer it needs to run the same network protocol, use the same naming convention, and work with the security functions of the version of Windows it wants to access. A Windows 9*x* system will use NetBEUI or TCP/IP, while most other versions of Windows use TCP/IP. Windows NT and 9*x* use NetBIOS, while later versions use DNS.

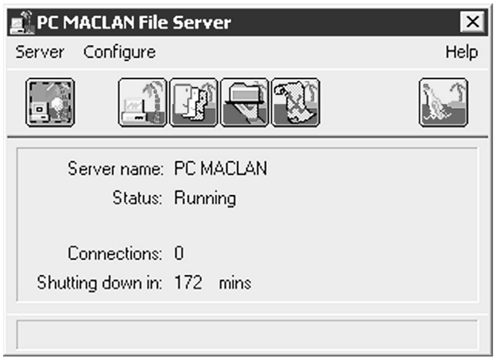
Connecting Macintosh to Windows 9*x* Shared Resources

Connecting a Macintosh to a Windows 9*x* system differs depending on whether you're using a Mac with OS X or if you use an earlier Mac operating system. For many years, Apple relied on third-party programs to do the interconnections between Mac and Windows. Starting with the OS X 10.1 operation system, Macintosh computers use the same tools and methods to connect to Windows that all UNIX/Linux systems currently use: namely Samba, which we'll discuss in the UNIX/Linux section following this one. The OS X operating system is so different from its predecessors that it makes more sense to not even think of it as a Macintosh OS but rather as a UNIX/Linux OS. So let's talk about the pre-OS X days in this section and discuss the OS X Macs under the Connecting UNIX/Linux systems to Windows 9*x* system section next.

The first area where you will get in trouble connecting Macs to any Windows system is in network protocol. Many older Macs use the AppleTalk protocol, while Windows 9*x* systems use NetBEUI or TCP/IP.

TIP Later Macintoshes all use TCP/IP, but the Network+ exam seems not to know this—if the test says Macintosh, assume AppleTalk!

Communication between Macintosh and Windows systems is further complicated by the fact that Macs and PCs use different higher-level protocols, too. Macintosh systems use the AppleTalk protocol for roughly the same jobs NetBIOS handles in Windows systems. So, even when your Macintosh and Windows systems are both using TCP/IP, you still have two different protocols handling the sessions, network naming conventions, and other important jobs. Windows 9*x* systems do not come with the AppleTalk protocol, and Macintoshes don't come with NetBIOS, so you're not going to get a Windows 9*x* client to talk to a Macintosh without some extra software. No problem! Now that all newer Macintosh systems support TCP/IP, a number of third-party vendors sell excellent programs that enable Macintosh computers to access shared folders and printers on Windows 9*x* systems. Figure 18-1 shows the popular interconnectivity program *PC MACLAN* running on a Windows system.



Server name: PC MACLAN
Status: Running

Connections: 0
Shutting down in: 172 mins

Figure 18-1 PC MACLAN running on a Windows system

Another program to connect Mac to Windows is called DAVE. Unlike PCMACLAN, DAVE is a two-way product: it enables Windows computers to access shared resources on Macs and Macs to access shared resources on Windows. We'll see more of DAVE in the "Connecting to Macintosh" section later in this chapter.

Connecting UNIX/Linux Systems to Windows 9x Systems

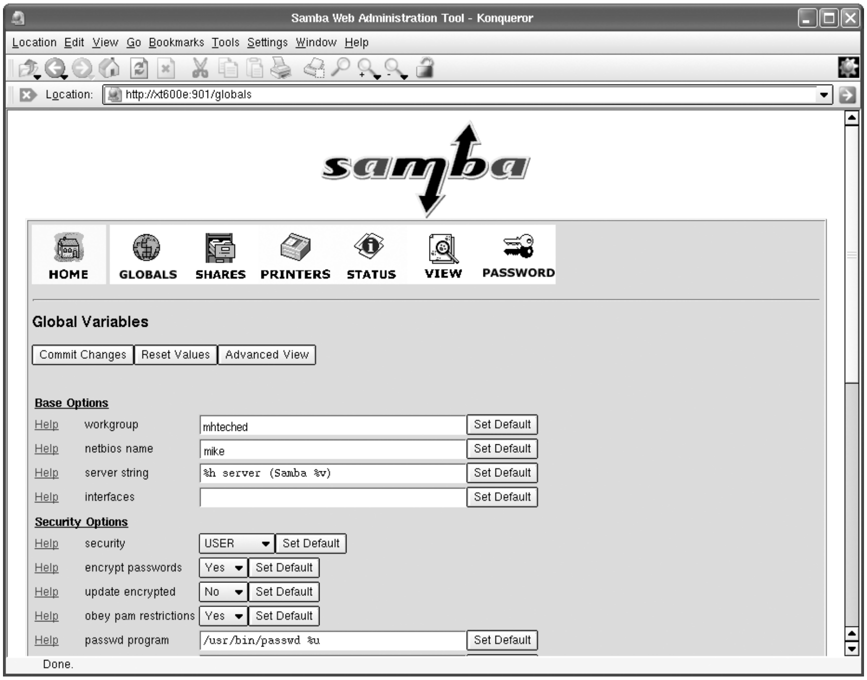
Connecting UNIX/Linux systems to Windows 9*x* systems for native access to shared folders requires you to run a program called Samba on the UNIX/Linux system. *Samba* makes the UNIX/Linux box look like just another Windows 9*x* system to the Windows server. Once you've got Samba running on the UNIX/Linux box, it will have access to the Windows system's shared folders. Samba is also used to connect Windows machines to shared folders and printers on UNIX/Linux and Mac OS X systems, as you'll see later in this chapter.

Like most UNIX/Linux applications, Samba isn't interesting to look at—it runs in the background and doesn't even have an interface. All Samba's configuration is stored in text files on each UNIX/Linux system. The primary configuration file for Samba is the smb.conf file. This file stores basic information such as the NetBIOS name for the UNIX/ Linux system and the workgroup or domain to join. Here's a small bit of a typical smb.conf file:

Learning how to configure Samba is challenging—if you head over to your local technical bookstore, you'll see that the thickest books are the ones on configuring Samba. Fortunately, there are third-party tools that help automate this process, such as the popular SWAT program (Figure 18-2).

Shared Windows folders manifest themselves as drives that must then be mounted in UNIX/Linux. The most common tool is called smbmount. Many UNIX/Linux distributions now have programs similar to My Network Places or Windows Explorer that display mounted Samba drives (Figure 18-3).

Macintosh computers running OS X also use Samba, but Apple does a great job of automating the process. Figure 18-4 shows a Macintosh system accessing shared folders on a Windows system.



http://xt600e:901/globals

Global Variables

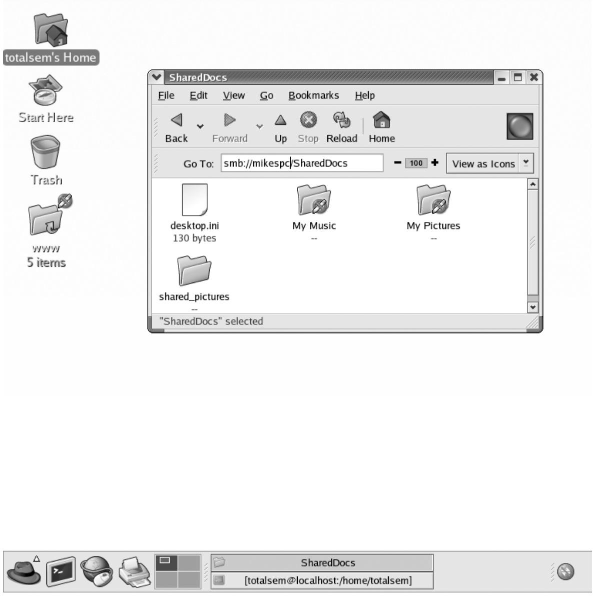
Base Options

workgroup: mhteched
netbios name: mike
server string: %h server (Samba %v)
interfaces:

Security Options

security: USER
encrypt passwords: Yes
update encrypted: No
obey pam restrictions: Yes
passwd program: /usr/bin/passwd %u

Figure 18-2 SWAT running on Linux



Go To: smb://mikespc/SharedDocs

"SharedDocs" selected

Figure 18-3 Shared Windows folders on Linux system

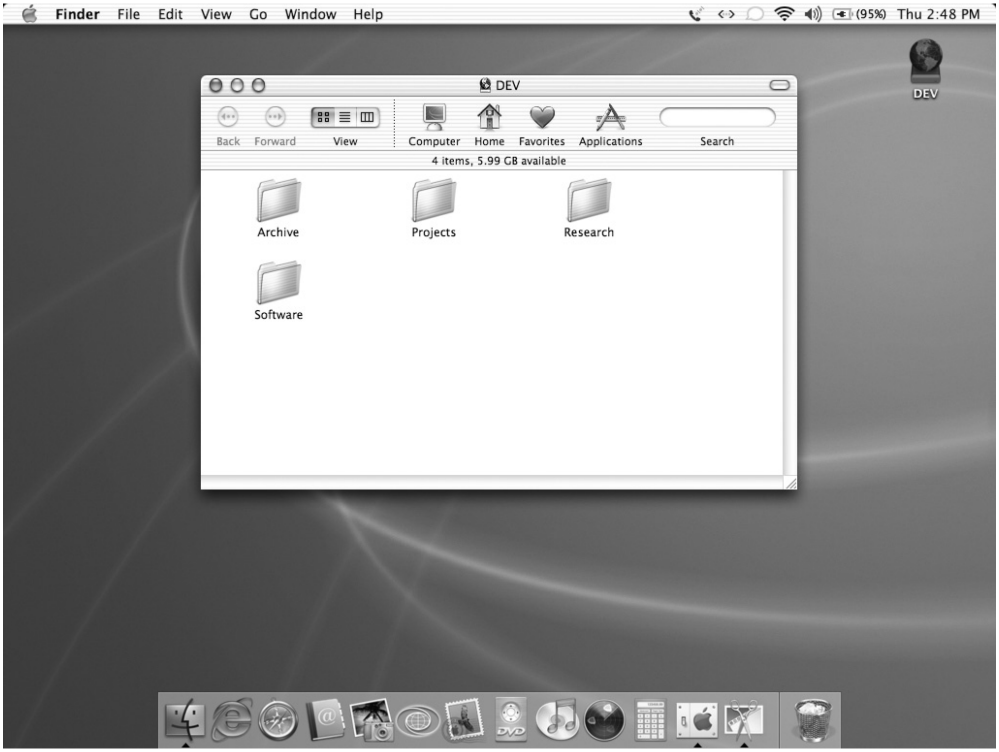


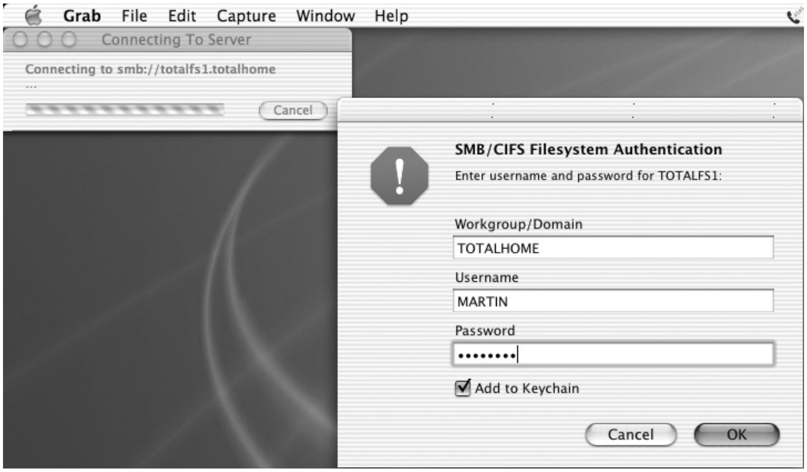
Figure 18-4 Mac OS X connecting to Windows system

Connecting to Windows Workstations (NT/2000/XP)

There's not much difference in how you connect a Macintosh or UNIX/Linux system to a Windows NT Workstation, Windows 2000 Professional, or Windows XP system than to a Windows 9*x* system. The tools just described are still used but unlike Windows 9*x*, you'll need a local account on any Windows system you want to access. Figure 18-5 shows a Macintosh system prompting for a username and password to access a shared folder on Windows XP.

Connecting to Windows Server Systems

The Windows Server platforms—Windows NT Server, Windows 2000 Server, and Windows Server 2003—all support Macintosh and UNIX/Linux systems natively. *Native support* means that the Windows server will act like Macintosh or UNIX/Linux systems—you don't have to make the Macs or UNIX/Linux computers act like Windows machines. This support is manifested via special services that you install into your Windows server. You may also use the Macintosh and UNIX/Linux program mentioned earlier, but you'll run into a big problem if you want those machines to join a domain or an Active Directory.



Connecting to smb://totalfs1.totalhome
...

SMB/CIFS Filesystem Authentication

Enter username and password for TOTALFS1:

Workgroup/Domain: TOTALHOME

Username: MARTIN

Password: ••••••••

☑ Add to Keychain

Figure 18-5 Prompting for username and password

Windows domains and Active Directory cause a bit of a headache for Macintosh and UNIX/Linux systems. The services Microsoft provides to support Macintosh and UNIX/Linux clients create separate folders on the Windows server that are not part of the domain or Active Directory. This effectively treats Macintosh and UNIX/Linux systems as separate entities, and excludes them from accessing the same shared folders as the Windows systems on the domain or Active Directory. The tools that Macintosh and UNIX/Linux use to access Windows shared resources do support domains, but there's no program yet that enables a Macintosh or UNIX/Linux system to join an Active Directory—with the lone exception of DAVE for Macintosh. Given that we've already seen the tools Macs and UNIX/Linux systems use to connect to Windows, let's concentrate on the services provided by Microsoft as we see how to connect to Windows server systems.

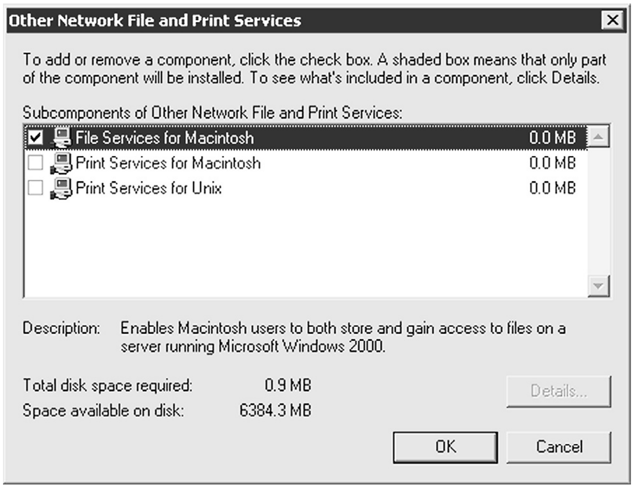
Connecting Macintosh to Windows Server Shared Resources

Windows NT, 2000, and 2003 Server come with both an AppleTalk protocol and two services: *File Services for Macintosh (FSM)* and *Print Services for Macintosh (PSM)*. With AppleTalk protocol support, File Services for Macintosh and Print Services for Macintosh, Windows NT, 2000, and 2003 Server have all the functionality they need to provide seamless interconnectivity for Macintosh computers running pre-Mac OS X operating system versions. Systems running OS X usually don't bother with these Microsoft services because the tools that come with OS X are preinstalled, faster, and easier to use than FSM and PSM.

TIP The Windows NT version of FSM, called File Server for Macintosh service, must run with AppleTalk. Later versions will work with TCP/IP.

FSM and PSM are installed as services from the Windows Components section of the Add/Remove programs applet (see Figure 18-6). There's nothing to configure, you just click the check box. The configuration work comes into play when you create shared folders.

Figure 18-6 Installing File Services for Macintosh on a Windows 2000 system



Other Network File and Print Services

To add or remove a component, click the check box. A shaded box means that only part of the component will be installed. To see what's included in a component, click Details.

Subcomponents of Other Network File and Print Services:

☑ File Services for Macintosh 0.0 MB
☐ Print Services for Macintosh 0.0 MB
☐ Print Services for Unix 0.0 MB

Description: Enables Macintosh users to both store and gain access to files on a server running Microsoft Windows 2000.

Total disk space required: 0.9 MB
Space available on disk: 6384.3 MB

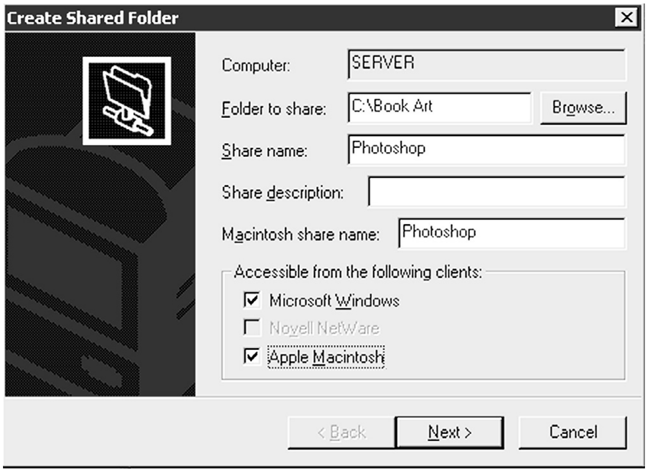
Once you've installed the File Services for Macintosh, you need to create a special volume that will store the files you want the Microsoft clients to access. When you create a share on the Windows server, you must specify that it is to be shared by Macintosh clients (see Figure 18-7). All the Macintosh systems will also need to have valid user accounts on the Windows server.

Using Print Services for Macintosh is even easier than FSM. Just make sure that the system sharing the printer is running AppleTalk to support Macintosh systems. Once the service is running, Macintosh systems will see the server's shared printers in their Chooser program.

Connecting UNIX/Linux Systems to Windows Server Shared Resources

The primary method to get any UNIX/Linux system to access a shared resource on a Windows system is Samba. Samba's inability to have a UNIX/Linux system join an Active

Figure 18-7 Creating a new share for Macintosh systems on a Windows server



Create Shared Folder

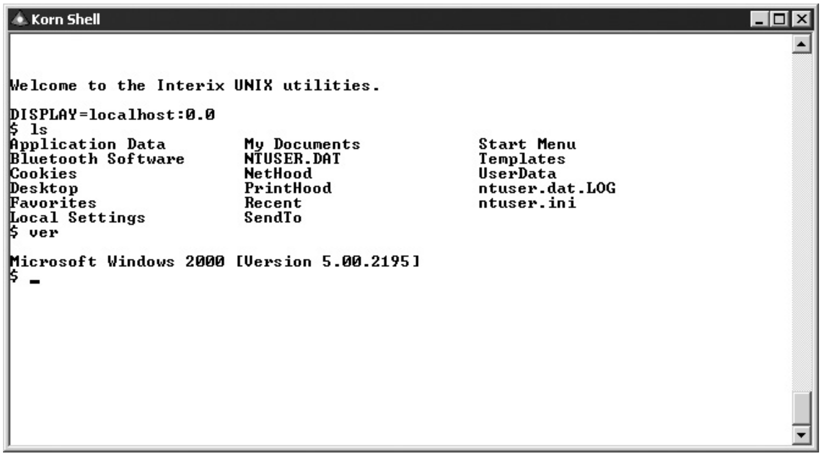
Computer: SERVER
Folder to share: C:\Book Art
Share name: Photoshop
Share description:
Macintosh share name: Photoshop

Accessible from the following clients:
☑ Microsoft Windows
☐ Novell NetWare
☑ Apple Macintosh

Directory isn't too much of a problem for most users. Remember, having a computer join an Active Directory is different than accessing shared resources in an Active Directory environment.

For many years Microsoft has provided a product called *Services for UNIX (SFU)*. SFU is a UNIX subsystem for Windows that manifests itself primarily as a UNIX-style shell, as shown in Figure 18-8. SFU has been around for a number of years, but was only available as an add-on, and at additional cost.

SFU also includes many of the standard TCP/IP applications for file and print sharing such as NFS, FTP, and LPD (Figure 18-9). So while Microsoft doesn't push FSU as an interoperability tool to access shared resources, it's the only option other than running Samba on the UNIX/Linux systems.

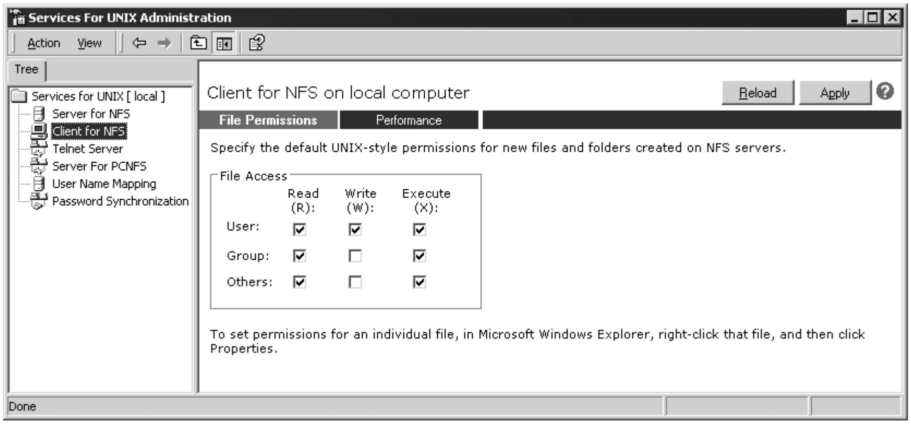


```
Welcome to the Interix UNIX utilities.

DISPLAY=localhost:0.0
$ ls
Application Data       My Documents          Start Menu
Bluetooth Software     NTUSER.DAT            Templates
Cookies                NetHood               UserData
Desktop                PrintHood             ntuser.dat.LOG
Favorites              Recent                ntuser.ini
Local Settings         SendTo
$ ver

Microsoft Windows 2000 [Version 5.00.2195]
$ _
```

Figure 18-8 SFU running in Windows 2000



Client for NFS on local computer

Specify the default UNIX-style permissions for new files and folders created on NFS servers.

File Access

	Read (R):	Write (W):	Execute (X):
User:	☑	☑	☑
Group:	☑	□	☑
Others:	☑	□	☑

To set permissions for an individual file, in Microsoft Windows Explorer, right-click that file, and then click Properties.

Figure 18-9 NFS Client configuration in SFU

Connecting to NetWare

If there's one NOS that makes interconnectivity a snap, it's Novell NetWare—or at least NetWare 4.*x*/5.*x*/6.*x*. Because NetWare is purely a server NOS, it must, by definition, perform interconnectivity every time you use it: there's no such thing as a NetWare client operating system. Windows, Macintosh, and UNIX/Linux systems must all use some form of *NetWare client software* to connect to a NetWare server, and Novell has, by necessity, created excellent client software for all of them. You can download the clients directly from Novell's web site, http://www.novell.com/download.

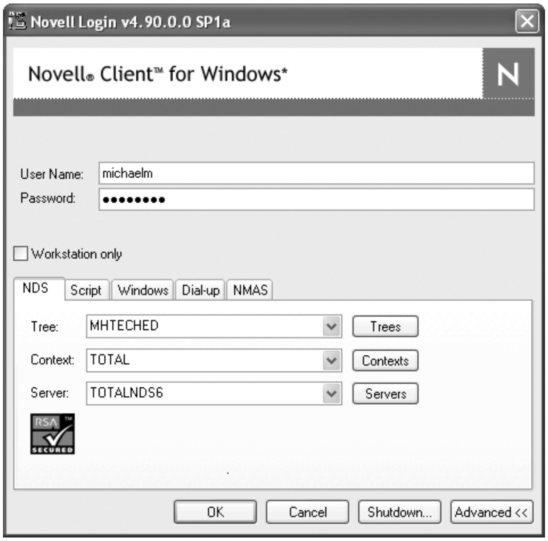
Connecting Windows Systems to NetWare

Novell has made client software for every Microsoft operating system including the ancient DOS of the 1980s. Over time, Novell has continued to evolve the NetWare client software: going graphical with the onset of Windows, supporting mixed environments of NetWare and Windows servers, and constantly updating to use the latest features of NetWare. As of this writing, the latest NetWare clients were version 4.9 for Windows NT/2000/XP and version 3.4 for Windows 9*x*. Figure 18-10 shows the logon screen from a modern NetWare client running on Windows XP.

Novell called the first client written for Windows 95, *Client32*. *Client32* is long obsolete, but the name has stuck as the common term for any Windows NetWare client provided by Novell. The latest versions of Client32 are powerful, providing support for IPX/SPX, the old NetWare 3.*x* binderies, and the more modern NetWare 4.*x*/5.*x*/6.*x* NDS trees. In addition, anyone running Client32 has complete control over the entire NetWare network—assuming of course they have the supervisor/admin rights! Figure 18-11 shows a NetWare client accessing the NDS tree of a small NetWare 6 network.

Microsoft has never been happy with NetWare clients taking over the login and other functions and for many years has supplied its own client with Windows. This client is

Figure 18-10 NetWare client logon



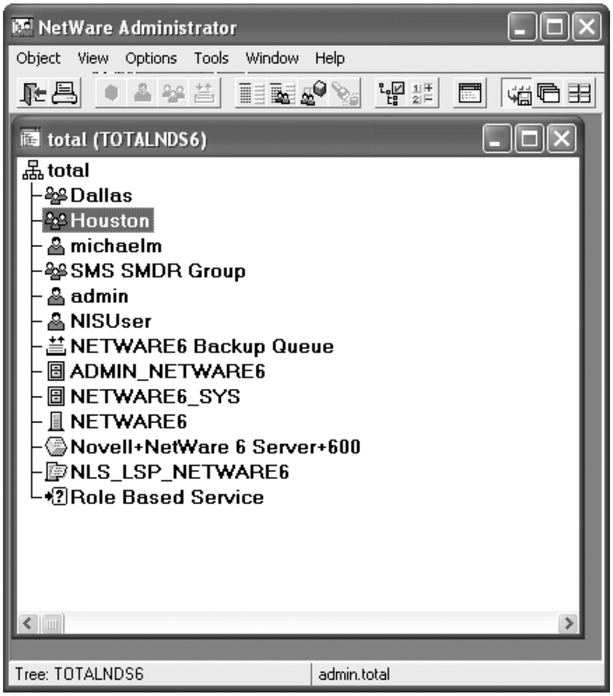
Novell Login v4.90.0.0 SP1a

Novell® Client™ for Windows*

User Name: michaelm
Password: ●●●●●●●●●

□ Workstation only

Tree: MHTECHED
Context: TOTAL
Server: TOTALNDS6



- total
  - Dallas
  - Houston
  - michaelm
  - SMS SMDR Group
  - admin
  - NISUser
  - NETWARE6 Backup Queue
  - ADMIN_NETWARE6
  - NETWARE6_SYS
  - NETWARE6
  - Novell+NetWare 6 Server+600
  - NLS_LSP_NETWARE6
  - Role Based Service

Tree: TOTALNDS6 | admin.total

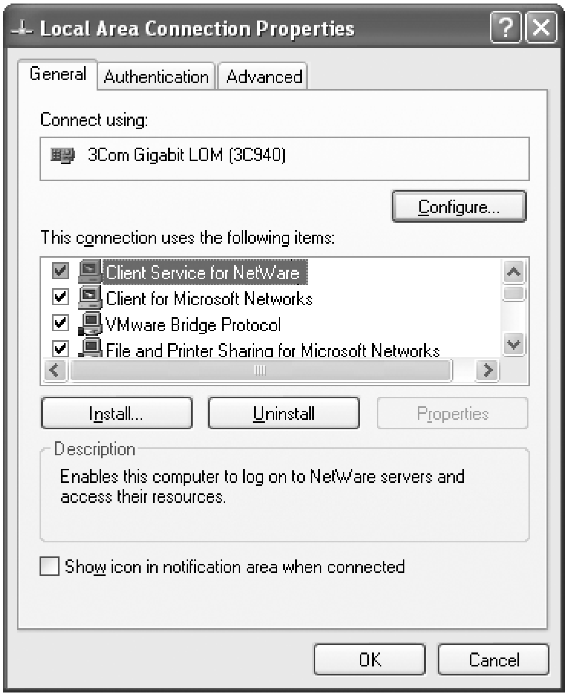
Figure 18-11 NDS tree from NetWare client

called Microsoft *Client for NetWare Networks* in Windows 9*x* and *Client Service for NetWare (CSNW)* in Windows NT, 2000, and XP (Figure 18-12). The Microsoft client comes with Windows but is not installed by default. (Windows 95 is the only exception to this.)

The Microsoft Client for NetWare Networks is weak compared to Novell's Client32. It enables a Windows 9*x* client to connect to resources on a NetWare server but little else. The Microsoft Client for NetWare Networks has two key weaknesses. First, it requires the IPX/SPX-compatible protocol and cannot connect to NetWare servers using TCP/IP. As Novell joins the rest of the networking industry in its headlong stampede toward TCP/IP, Windows 9*x* clients running the Microsoft Client for NetWare Networks get left behind. Second, the Microsoft Client for NetWare Networks does not understand *Novell Directory Services (NDS)*, NetWare's default security, and the directory systems for NetWare 4, 5, and 6.

NOTE Later versions of NetWare use the term *eDirectory* in place of NDS.

Client Service for NetWare is more robust than the Microsoft Client for NetWare Networks in that it supports both NetWare 3 binderies and NDS trees. Figure 18-13 shows the configuration screen for CSNW on a Windows XP computer. Note the two radio buttons



Local Area Connection Properties

Connect using:

3Com Gigabit LOM (3C940)

This connection uses the following items:

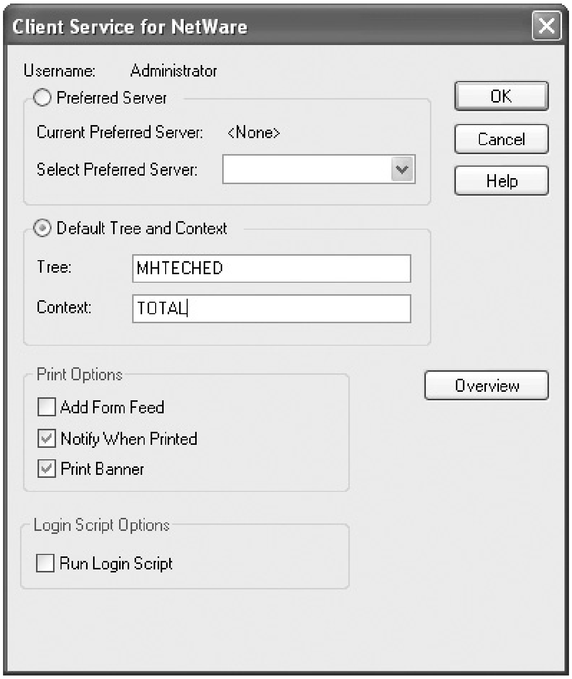
☑ Client Service for NetWare
☑ Client for Microsoft Networks
☑ VMware Bridge Protocol
☑ File and Printer Sharing for Microsoft Networks

Description

Enables this computer to log on to NetWare servers and access their resources.

□ Show icon in notification area when connected

Figure 18-12 Client Service for NetWare



Client Service for NetWare

Username: Administrator

○ Preferred Server

Current Preferred Server: <None>

Select Preferred Server:

● Default Tree and Context

Tree: MHTECHED

Context: TOTAL

Print Options

□ Add Form Feed
☑ Notify When Printed
☑ Print Banner

Login Script Options

□ Run Login Script

Figure 18-13 CSNW configuration

at the top. Choosing the Preferred Server radio button and entering the name of a NetWare server connects you to a NetWare Bindery. Choosing the Default Tree and Context radio button enables the system to connect to a NetWare NDS tree.

CSNW, like the Microsoft Client for NetWare Networks, only supports IPX/SPX. If your NetWare servers only run TCP/IP, you won't use CSNW.

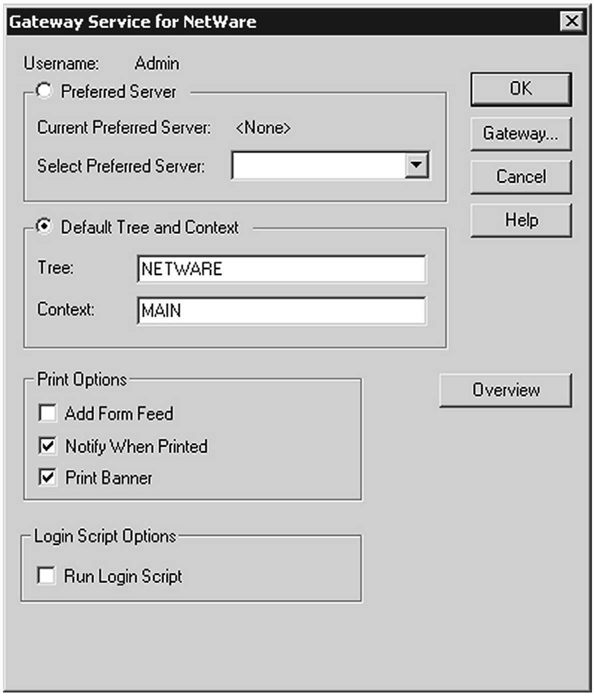
Windows NT/2000/2003 Server systems come with an interesting little program called *Gateway Services for NetWare (GSNW)* (see Figure 18-14). GSNW enables a single Windows Server system to act as a gateway to a NetWare network. The gateway computer logs in to the NetWare network and enables all of the Windows computers to access the NetWare network through the gateway without running a NetWare client. GSNW sounds great, but in reality it is slow and prone to lockups. GSNW is also dependent on IPX/SPX. GSNW is not supplied with Windows Server 2003.

TIP Make sure you understand the difference between CSNW and GSNW!

Connecting Macintosh to NetWare

To connect a Macintosh to a NetWare server, you need to install the *Macintosh Client for NetWare* on the Mac. Figure 18-15 shows the Macintosh Client for NetWare. Once the Macintosh NetWare client is installed, the OS has full access to the shared resources on

Figure 18-14 Configuring GSNW on a Windows 2000 server



Gateway Service for NetWare

Username: Admin

○ Preferred Server

Current Preferred Server: <None>

Select Preferred Server:

● Default Tree and Context

Tree: NETWARE

Context: MAIN

Print Options

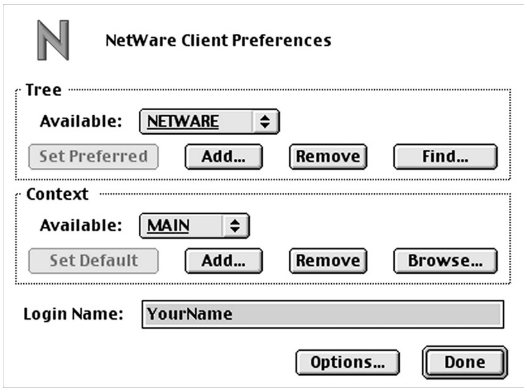
□ Add Form Feed

☑ Notify When Printed

☑ Print Banner

Login Script Options

□ Run Login Script


NetWare Client Preferences

Tree
Available: NETWARE

Context
Available: MAIN

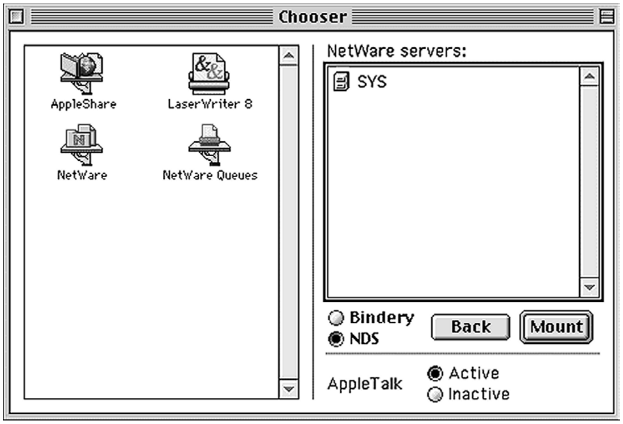
Login Name: YourName

Figure 18-15 Macintosh Client for NetWare

the NetWare server, both folders and printers. Figure 18-16 shows a shared folder on a NetWare server being accessed from a Macintosh client. One interesting aspect unique to the Macintosh client is that it's the only one you have to pay to use because Novell stopped making Macintosh clients in the mid 1990s. A third-party vendor called ProSoft (www.prosoft.com) now makes the Macintosh NetWare client software. Both the Windows (any version) and UNIX/Linux clients are available free from Novell.

NetWare client software (of all sorts) can't do its job unless the client system runs the correct protocol. Modern clients all use TCP/IP, but there are many older clients still in use in the world. Fortunately, Novell provides IPX drivers for all network client systems to support the occasional NetWare network that still uses IPX as its network protocol. But whether your network runs IPX or IP, the NetWare client will work perfectly for Windows, Macintosh, or UNIX/Linux systems.

NOTE To download Novell client software, go to: http://www.novell.com/download.


Chooser

NetWare servers:
SYS

○ Bindery
● NDS

AppleTalk ● Active ○ Inactive

Figure 18-16 Macintosh client accessing shared folder on NetWare server

Native File Access

NetWare versions 5.1 and 6 support a fascinating technology that Novell calls Native File Access. *Native File Access* is a series of programs run on NetWare Servers that make the shared NetWare folders look like the native format for Windows and Macintosh computers. Native File Access removes the need to install a NetWare client on each computer—the NetWare servers and shared folders look like Windows systems to Windows clients and Macintosh systems to Mac clients! Native File Access is gaining in popularity with NetWare users.

Connecting to Macintosh

Even though modern Macintosh operating systems all use IP as their network protocol, Apple still relies on the venerable AppleTalk for higher-level network functions, just as Windows 9*x* still relies on NetBIOS even though most Windows 9*x* systems now use IP. This means that any time you have a Macintosh system talking to any other type of system, you must have some form of software on one end or the other that translates the AppleTalk information (like network names) into something the client NOS can understand.

Once again, the evolution of OS X over its predecessors makes a big difference in how other operating systems connect to Macintosh computers. OS X runs Samba by default so Windows systems automatically see Macs on their network. UNIX/Linux systems can connect to a Mac via Samba or by using TCP/IP applications like NFS or FTP. In the next section, we'll look at pre-OS X Macs.

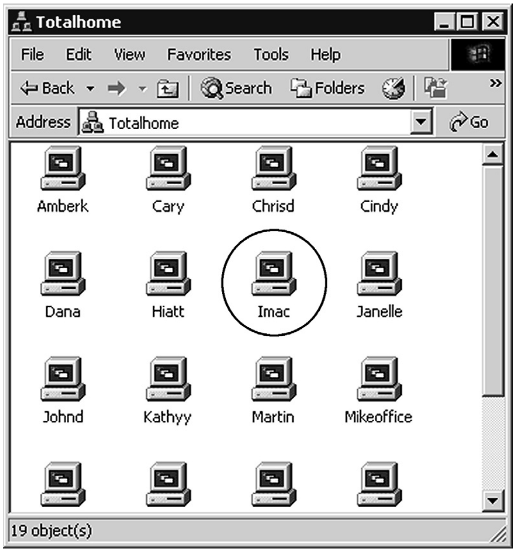
Connecting Windows Systems to Macintosh Serving Systems

The most common way to get a Macintosh system to share its resources with Windows and UNIX/Linux systems is via the AppleShare IP program. *AppleShare IP* has built-in server message block (SMB) support, making it something like a Samba for Macintosh. AppleShare also provides printing support to clients running the TCP/IP LPR program or the more modern CUPS. Conveniently, this is no problem because both Windows and UNIX/Linux clients support LPR.

AppleShare IP is powerful but is an extra cost item and requires fairly serious Macintosh hardware to run. AppleShare IP also includes a web server, an FTP server, and other items that might be more than you need just to connect to a Macintosh or two in a smaller network. For smaller jobs, many networks use the DAVE program from Thursby Software. *DAVE* runs on a Macintosh, enabling any Windows systems to access shared folders and printers on Macintosh systems. DAVE has the added benefit of not requiring the Windows system to run any form of client software—the DAVE software runs only on the Macintosh. Figure 18-17 shows a DAVE-shared Macintosh as seen by a Windows client system.

I should add that DAVE is not just for Mac-to-Windows sharing. It works equally well to enable Mac clients to access shared folders on Windows systems. A number of companies provide Mac-to-Windows and Windows-to-Mac interconnectivity tools. If

Figure 18-17 A DAVE-shared Macintosh named Imac as seen by Windows



you're interested in getting Mac and Windows systems to work together, check out www.macwindows.com.

Connecting UNIX/Linux Systems to Macintosh Sharing Systems

Until the advent of the *Mac OS X* operating system, Macintosh systems lacked a truly handy way for UNIX/Linux systems to connect to shared folders on Macintosh systems. Macintosh systems have used various NFS server programs, which worked moderately well. UNIX/Linux systems can use their built-in NFS-based tools to access NFS shares on Macintosh systems. With the introduction of the UNIX-based OS X, Macintosh systems now share the same UNIX NFS as pure UNIX/Linux systems. UNIX/Linux systems use CUPS as well as good old LPR/LPD to share printers.

TIP Apple has a full-blown, UNIX-based server operating system called Mac OS X Server. It has complete, native support for Macintosh, UNIX/Linux, and Windows clients, including DNS and DHCP, File and Printer Sharing, and more. It has a built-in VPN, NAT, web server, mail server, QuickTime streaming media server, and more. Apple definitely did the Mac OS X Server right! For more information on Mac OS X Server, go straight to the source: www.apple.com.

Connecting to UNIX/Linux

UNIX/Linux systems rely on Samba or NFS to enable non-UNIX/Linux client systems to access their resources. Samba is popular because you only have to configure the UNIX/Linux system for its resources to be visible to all Windows clients. However, you must do the Samba configuration on each UNIX/Linux system. This is fine as long as you only have a few UNIX/Linux servers and are using Windows clients, but if you have lots of UNIX/Linux systems, or if you're using Macintosh clients, you need some other options.

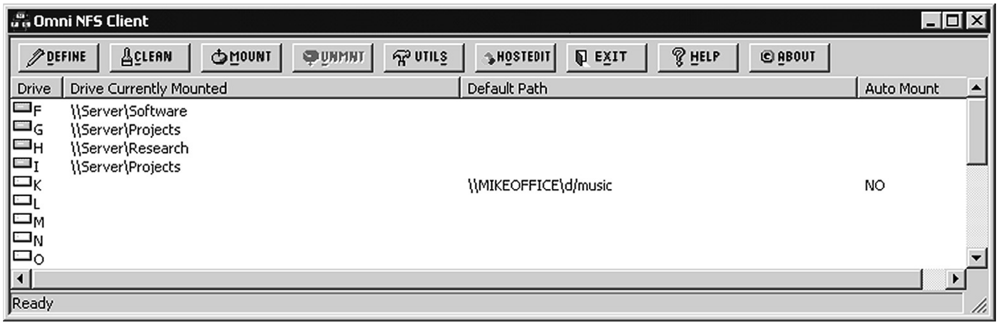
Connecting Windows Systems to UNIX/Linux Sharing Systems

To reduce network congestion and to make configuration easier, Microsoft provides *Microsoft Windows Services for UNIX (MWSU)*. This group of services, really just a toolbox of NFS programs, can access any type of NFS volume, including any from a UNIX/Linux system. MWSU is an add-on product that will run on any Windows NT or Windows 2000 system. To enable a Windows 9*x* system to access a UNIX/Linux folder, you need to find a good third-party NFS client program. Figure 18-18 shows the OMNI NFS client program from Xlink Technology running on a Windows 9*x* system.

One interesting part of MWSU is the *Gateway Services for UNIX*. As with Gateway Services for NetWare, MWSU enables a Windows NT/2000 Server system to act as a gateway connecting a Windows network and UNIX/Linux serving systems. None of the Windows systems need to run an NFS client program, and all of the UNIX/Linux servers appear to the Windows systems as though they're Windows servers.

Connecting Macintosh Systems to UNIX/Linux Sharing Systems

Okay, I've covered almost all of the possible NOS interconnectivity combinations. The last combination I need to cover is connecting Macintosh systems to shared resources on UNIX/Linux systems. You should be able to tell me the answer at this point—can you? Remember that Mac OS X has full NFS support for file sharing and



Drive	Drive Currently Mounted	Default Path	Auto Mount
F	\\Server\Software		
G	\\Server\Projects		
H	\\Server\Research		
I	\\Server\Projects		
K		\\MIKEOFFICE\d/music	NO
L			
M			
N			
O			

Figure 18-18 The OMNI NFS client running on a Windows 9*x* system

LPD/LPR support for accessing shared printers. But I can't end this chapter without at least one more peek at some fun third-party software. Many UNIX/Linux servers that need to provide access to Macintosh systems use a little program called Netatalk. *Netatalk* creates AppleTalk-compliant folder and printer shares on UNIX/Linux systems. While Netatalk itself is a text-based program, many folks use it with a graphical front end like the *AppleTalk Configurator* (see Figure 18-19).

When All Else Fails, Terminal Emulate!

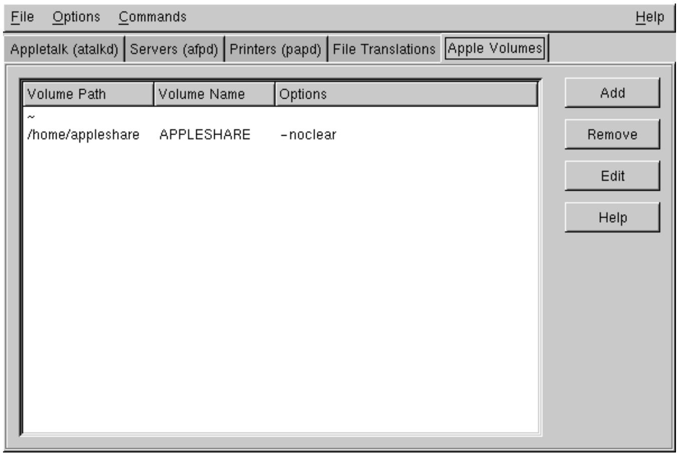
Terminal emulation has been a part of TCP/IP from its earliest days, in the form of good old Telnet. Because it dates from pre-GUI days, Telnet is a text-based utility, and all modern operating systems are graphical. Citrix Corporation made the first (arguably) popular (also arguably) terminal emulation product—the *Winframe/MetaFrame* products (Figure 18-20). Citrix isn't free, but it runs on any operating system and is a mature and dependable product.

In keeping with the GUI world we live in today, many operating systems come equipped with some type of graphical terminal emulator. Some, like Windows and Linux, include built-in emulators like the handy Windows 2000 *Terminal Services* (see Figure 18-21).

All Microsoft remote access products run on the Remote Desktop Protocol (RDP). RDP, developed by Citrix and licensed by Microsoft, provides a protocol for everyone who wants to make a program to connect to a Windows system. RDP typically runs on TCP port 3389.

Unfortunately, Terminal Services only works in the Windows environment; however, a number of third parties make absolutely amazing terminal-emulation programs that

Figure 18-19 The AppleTalk Configurator graphical front end



Volume Path	Volume Name	Options
~		
/home/appleshare	APPLESHARE	-noclear

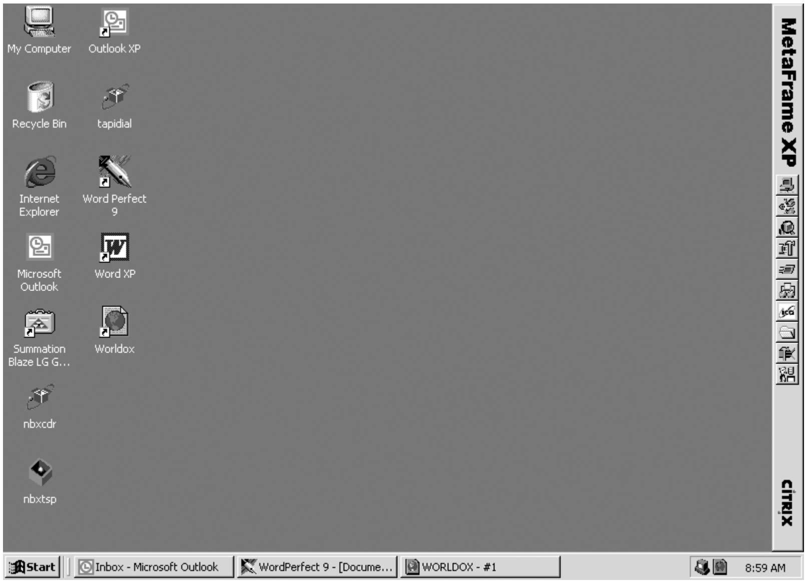


Figure 18-20
Citrix in action

run on any operating system. One of the best of these is *VNC*, which stands for virtual network computing (see Figure 18-22). VNC doesn't let you share folders or printers, because it is only a terminal emulator. But it runs on almost every client NOS, is solid as a rock, and even runs from a web browser. Why bother sharing if you can literally be at the screen? Oh, and did I mention that VNC is free?

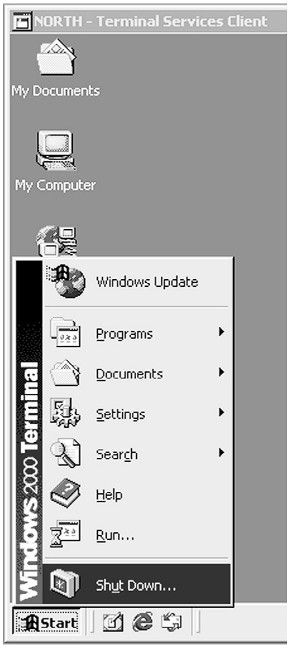
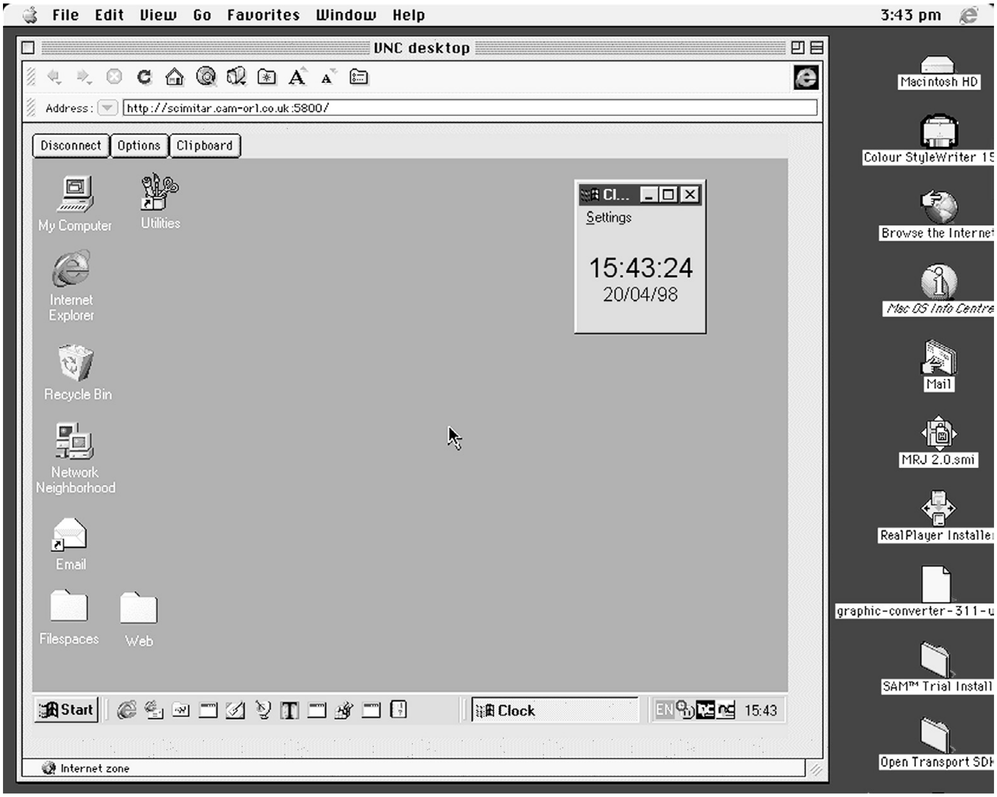


Figure 18-21
Windows 2000 Terminal Services in action



Address: http://scimitar.cam-orl.co.uk:5800/

Figure 18-22 The VNC desktop

Chapter Review

Questions

1. The universal adoption of what protocol suite has made the concept of interconnectivity between network operating systems much easier?

   A. TCP/IP

   B. IPX/SPX

   C. Samba

   D. AppleTalk

2. What protocol suite has Macintosh historically used?

   A. TCP/IP

   B. IPX/SPX

C. Samba

D. AppleTalk

3. What program makes a UNIX/Linux system look like a Windows 9*x* system, and enables it to access a Windows system's shared folders?

A. TCP/IP

B. IPX/SPX

C. Samba

D. AppleTalk

4. Which of the following file system protocols must be installed on a Windows 9*x* system for that system to share its folders like a UNIX/Linux system?

A. NT file system (NTFS)

B. FAT32

C. Samba

D. Network File System (NFS)

5. Which of the following services must be installed on a Windows NT or 2000 server for files and printers to be shared with a Macintosh client?

A. File and Print Sharing

B. File Sharing for Macintosh

C. Services for UNIX

D. Client Services for NetWare

6. Which of the following services must be installed on a Windows NT or 2000 server for files and printers to be shared with a UNIX/Linux client?

A. File and Print Sharing

B. File Sharing for Macintosh

C. Services for UNIX

D. Client Services for NetWare

7. What client software must be installed on a Macintosh client system to enable it to connect to a NetWare server?

A. File and Print Sharing

B. File Sharing for Macintosh

C. Services for UNIX

D. Macintosh Client for NetWare

8. When you connect a Windows or UNIX/Linux system to a Macintosh server, what program is used to share the Mac resources?
   - **A.** Macintosh Client for NetWare
   - **B.** File Sharing for Macintosh
   - **C.** AppleShare IP
   - **D.** Client Services for NetWare
9. What program can you install on a Macintosh serving system in a smaller network, where you don't need all the extras that come with AppleShare IP, to enable a Windows system to access shared resources on that Mac?
   - **A.** TCP/IP
   - **B.** DAVE
   - **C.** WIN2MAC
   - **D.** File Sharing for Macintosh
10. Windows 2000 has a built-in terminal emulator. What is it called?
    - **A.** Windows 2000 Terminal Services
    - **B.** Windows 2000 Terminal Emulator
    - **C.** Windows 2000 Telnet
    - **D.** Windows 2000 Emulation Services

Answers

1. **A.** The universal adoption of TCP/IP has made interconnectivity much easier than it was in the days when different networks ran IPX/SPX, NetBEUI, AppleTalk, and other network protocols.
2. **D.** Many older Macs use the AppleTalk protocol, and most Windows systems use NetBEUI or TCP/IP. Later Macs all use TCP/IP, but for the Network+ exam, assume AppleTalk for all Macintosh questions unless the question specifies otherwise.
3. **C.** Samba makes a UNIX/Linux system look like a Windows 9*x* system, and enables it to access a Windows system's shared folders.
4. **D.** The NFS protocol is the closest thing the TCP world has to Windows' folder sharing functions. Installing an NFS server program on a Windows 9*x* system will enable it to share its folders just like a UNIX/Linux system.
5. **B.** With File Sharing for Macintosh and AppleTalk protocol support, Windows NT Server and Windows 2000 Server have everything they need to provide Macintosh clients with seamless interconnectivity to Windows NT/2000 serving systems.

6. **C.** Microsoft provides a product called Services for UNIX (SFU) to enable NT and 2000 serving systems to share their resources with UNIX/Linux clients, but it comes as an add-on, and at additional cost.
7. **D.** A Macintosh client needs Macintosh Client for NetWare software to connect to a NetWare server.
8. **C.** The AppleShare IP program enables a Macintosh serving system to share its resources with Windows and UNIX/Linux clients. AppleShare IP has built-in SMB support, making it something like a Samba for Macintosh. AppleShare also provides printing support, but only to clients running the TCP/IP LPR program—which isn't a problem because Windows and UNIX/Linux clients both support LPR.
9. **B.** In a smaller network, where you don't need all the extras that come with AppleShare IP, you can install a program called DAVE on a Macintosh serving system, enabling Windows clients to access shared folders and printers on the Mac.
10. **A.** Windows' built-in terminal emulator is called Windows 2000 Terminal Services.