

C H A P T E R 1

Deployment Strategy and Details



The deployment process for Windows NT Workstation 4.0 consists of several distinct phases, including the following:

- u Assemble executive and planning teams
- u Review Windows NT Workstation 4.0
- u Identify the preferred network-client configuration
- u Prepare the planning and support teams
- u Perform lab tests of the client configuration
- u Plan the pilot rollout
- u Conduct the pilot rollout
- u Finalize the rollout plan
- u Roll out Windows NT Workstation 4.0

Not all of these tasks may be necessary for your organization.

Acquiring Resources

This phase involves gathering the resources, including equipment, software, and staff, to properly plan for testing and evaluating Windows NT Workstation 4.0. Members of the support team should receive training during this phase.

Assemble the Teams

The steps involved in deploying Windows NT Workstation are performed by the following teams:

Executive Team

The executive team includes the deployment project manager (usually the head of the Information Systems department) and members of the executive committee of the corporation. This team must include one or more individuals with decision-making authority for organization policies and procedures.

Planning Team

The planning team includes the deployment project manager, key installation team members, and a representative from the support and training teams.

Installation Team

The installation team includes technicians and individuals who will be conducting the installation. This team must include a specialist in 32-bit applications who can evaluate the proposed Windows NT Workstation 4.0 configuration for compatibility.

Training Team

The training team includes individuals responsible for user training.

Support Team

The support team includes staff from the help desk or support department, and select individuals from the planning team. This team develops a plan for supporting Windows NT Workstation 4.0 during and after deployment, integrating new methods and processes as needed into the existing support scheme.

In many cases, team members must continue with their regular work, in addition to working with their assigned deployment teams. Make sure you have the support of their managers, and that adjustments can be made to their regular work loads as needed.

The composition of the installation, training, and support teams depends in large part on the early decisions made by the executive and planning teams. Therefore, the first teams to be assembled are the executive and planning teams.

Executive and Planning Teams

The deployment project manager participates on the executive team and leads the planning team. This individual is usually the head of the Information Systems (IS) department; however, the executive committee may find another individual to be more appropriate, depending on the organization.

When setting up the planning team, it is important to include a set of individuals representing the groups involved in the deployment process. This includes people from the corporate support and employee training departments, the corporate standards committee, and key installation team members. Individuals from the finance and accounting groups will need to take part in planning and evaluation later on, but need not be assigned to the team for the full duration of the deployment process.

The executive and planning teams make top-level decisions, including the decision to purchase and deploy Windows NT Workstation. They may also decide which computers should have Windows NT Workstation installed on them, and which computers should run other operating systems, such as Windows 95. Once these decisions are made, these teams can assemble the remaining teams, and continue to oversee the deployment process.

Installation Team

The installation team is busy long before the final deployment. This team sets up the lab, and then uses it to test the deployment process. During this phase, the installation team, together with the planning team, has the opportunity to fine-tune the chosen configurations of Windows NT Workstation.

While testing the deployment process in the lab, the installation team creates a detailed checklist that can be used during the pilot and final deployments. Because the installation team is likely to increase in size for the final rollout, the checklist should include enough information to be useful to those who did not participate in the testing phase.

The Installation team should include an applications expert who can evaluate 32-bit applications running with Windows NT Workstation 4.0.

Training Team

For the deployment to be successful, the users need to know the advantages of the new operating system, and how to use its features to make their jobs easier. The training team is responsible for providing this knowledge to the users.

You can either hire a training vendor to act as the training team, or assemble the team from employees in your organization. Let this team know what configuration of Windows NT Workstation you have chosen, and keep them informed of any refinements you make to that configuration during the lab test phase.

Your Training team can use the materials provided by Microsoft Authorized Technical Education Centers (ATECs) as a basis for their user training course. In addition to developing and presenting the training class, the training team also sets up the lab or classroom with computers for training.

Support Team

The support team develops a plan for supporting Windows NT Workstation 4.0 during and after deployment. The Support team should become familiar with the Windows NT Workstation product documentation and the *Windows NT Workstation 4.0 Resource Kit* before Windows NT Workstation is actually deployed. Team members can receive instruction at a Microsoft Authorized Technical Education Center and participate the Certified Professional program. Call (800) SOLPROV (or (800) 765-7768) for information about authorized training offered for Windows NT Workstation 4.0 and the Certified Professional program, and for referral to a local Microsoft Solution Provider Authorized Technical Education Center (ATEC).

If Microsoft Systems Management Server is available, the support team should also be familiar with the Help Desk and Diagnostics features of Systems Management Server. Systems Management Server is a separate product available from Microsoft.

Conduct a Sample Inventory

You will need to survey a representative sample of your organization to identify the hardware and software typically used on client and server computers. By doing this sample inventory of your organization's active equipment, you can accurately simulate the organizational environment in the lab. Such a simulation helps you make broad decisions about your organization's computing infrastructure, such as which network protocols to use, or which desktop configuration best fits the applications used in your organization. Simulating the network environment also helps you configure client computers for use in a mixed environment of operating systems, servers, and network protocols.

When you are ready for final deployment, you will need a more complete inventory of the computers in your organization. The longer the time between inventory and deployment, the more chance there is for components to be replaced, or disk space filled, in the interim. These changes could cause the installation of Windows NT Workstation to fail. For best results, a complete and accurate inventory should be performed just before deployment.

Software management tools are available to help you ascertain the hardware and software configurations of computers on your network. For detailed information about a large number of computers on a network, use a system management program such as the Microsoft Systems Management Server to conduct the inventory.

Test Lab

The test lab is used to test the deployment process. It can also be used to test the configurations you have chosen for various groups of computers in your organization, including any applications.

Set aside a physical space for this lab, and acquire a mix of computers that accurately reflects your organization's computing environment. Be sure there are computers to represent every kind of hardware used in the computers to which you will deploy Windows NT Workstation – for example, if your organization uses five different models of computer, from two different manufacturers, include all five models in the lab. If you are currently running Windows for Workgroups on some computers and another operating system on others, make sure that both operating systems are represented in the lab. Likewise, make sure that the peripherals used with the computers to which you will deploy Windows NT Workstation are present in the lab. In addition, try to duplicate variables such as free disk space.

If your organization uses portable computers that dial in from remote locations, or if you use additional servers or mainframe computers for business data, give the lab computers full access to the network and an analog phone line.

You'll also want to make sure you have enough disk space on a test server to implement the rollout.

Test Windows NT Workstation 4.0 features in the lab with all of the applications used in your organization, before performing the pilot installation. Maintain a complete list of any problems found in the testing lab.

Decide on the Preferred Client Configuration

With the planning and installation teams assembled and educated about Windows NT Workstation 4.0 capabilities, determine the preferred network configuration and preferred feature set for these client computers. The teams will use this configuration for evaluation and testing, prior to full implementation of Windows NT Workstation 4.0 in your organization.

Starting with the ideal configuration, which uses the most functional and best-performing client software, evaluate each feature against your organization's needs and environment to determine whether the feature is appropriate and compatible.

You may want different configurations for different groups. For example, the marketing group might need different features and applications than the human resources. If you are considering different configuration alternatives, repeat the evaluation for each configuration.

The following sections describe feature options and decisions to evaluate in specifying the network client configuration.

User Interface

Windows NT Workstation 4.0 has the same interface as Windows 95.

Network Protocols

Networking protocols and services in Windows NT Workstation are all 32-bit. These include TCP/IP, support for the Internet, Dial-Up networking, NetBEUI, peer-to-peer networking, and Client Services for NetWare (with support for NetWare NDS networks). The networking support feature set includes:

- u Integration with existing network systems.
- u Data I/O across the network
- u More than one redirector can be run at one time, thereby enabling access to servers for multiple networks without having to reload the operating system for a new network client. For example, a Microsoft Windows NT Workstation client can access both NetWare servers and existing mainframe servers at the same time.
- u Seamless networking in the Windows NT Workstation 4.0 user interface; users can browse the server for multiple networks in Network Neighborhood, all within the same name space—users don't need to know which type of network they are browsing.

TCP/IP

To provide the best network interoperability over WANs and network routers, include TCP/IP in your ideal configuration. The 32-bit TCP/IP stack included with Windows NT Workstation incorporates both SLIP and PPP. Microsoft TCP/IP, in combination with Windows NT Workstation, provides a scalable solution for enterprise networks that include a mix of system types.

When TCP/IP is used as the enterprise networking protocol, an IP addressing scheme is needed for your organization. If your servers run Windows NT Server, you can use Dynamic Host Control Protocol (DHCP) and Windows Internet Naming Service (WINS) for easy TCP/IP address management. With DHCP, administrators can centrally define global and subnet TCP/IP parameters for interconnected networks. The DHCP service dynamically configures the IP address

and subnet mask of each workstation. For name resolution on TCP/IP internetworks, use WINS Servers or LMHosts files.

Windows NT Workstation allows users to run with more than a single IP address at the same time when using Remote Access Services.

When TCP/IP is used as a transport protocol with Windows NT Workstation, computers can communicate with other kinds of systems such as UNIX workstations and servers, or an IP configured printer, without additional networking software.

NetBEUI (NetBIOS Extended User Interface)

NetBEUI is a small, efficient, and fast protocol tuned for small LANs. It is designed to support department-sized LANs consisting of 20 to 200 workstations. NetBEUI does not support traffic across routers, but provides for both connectionless and connection-oriented traffic on a single network segment. NetBEUI is self-configuring and self-tuning. It can be installed and bound to a network card automatically when Windows NT is installed.

NWLink IPX/SPX

Windows NT includes the NWLink protocol, which is an implementation of the internetworking packet exchange (IPX) and sequenced packet exchange (SPX) transport protocols used by NetWare. NWLink NetBIOS is a Microsoft-enhanced implementation of Novell NetBIOS, and transmits Novell NetBIOS packets between a NetWare server running Novell NetBIOS and a Windows NT computer, or between two Windows NT computers.

The Microsoft implementations of the IPX, SPX, and Novell NetBIOS protocols can coexist with other protocols on the same network adapter card.

For instructions on configuring NWLink, see Help.

DLC (Data Link Control)

The DLC protocol provided with Windows NT is used primarily to access IBM mainframe computers, rather than for general networking on Windows NT. For example, Microsoft SNA Server for Windows NT uses the DLC protocol device driver when communicating with mainframes on the token ring interface. DLC is also used to connect to printers that are connected directly to a LAN, rather than to a specific computer.

Windows NT DLC allows Windows NT computers to connect to IBM mainframes using 3270 emulators. You can also connect to IBM AS/400 computers using 5250 emulators. DLC works with either token ring or ethernet media access control (MAC) drivers.

The DLC protocol works with Windows NT-based programs and with MS-DOS-based and 16-bit Windows-based programs.

AppleTalk Protocol

The AppleTalk protocol is used to deliver data to a network destination when a Windows NT Server computer configured with Windows NT Services for Macintosh is available on the network. This makes it possible for PCs and Apple Macintosh workstations to share files and printers.

The AppleTalk protocol is also used by software application developers who are creating cross-platform applications for Windows NT and the Macintosh. When used for transferring files across Ethernet or for remote debugging in this way, Windows NT Server Services for Macintosh is not required on the network.

Other 32-Bit Network Protocols

If you are using another type of network, contact your network vendor regarding the availability of 32-bit, protected-mode network client software. Solutions are available for Banyan Vines networks, and several vendors distribute NFS (network file system) solutions for better integration with UNIX-based systems.

Remote Access Service (RAS)

Windows NT Remote Access Service (RAS) client software gives the user access to server-based dial-in packages such as Windows NT Server Remote Access Service, Novell NetWare Connect, and Shiva NetModem. Once the connection is made from the RAS client to the server-based dial-in package, users at remote sites can use the network as if their computers were directly connected to the network. RAS can also be used to connect to remote client computers that have inbound modems.

With some additional configuration of protocols and software on the client computer, RAS provides additional security for remote dial-up connections. The Point-to-Point Protocol (PPP) and Serial Line Internet Protocol (SLIP) are supported in RAS.

Remote Access Service contains the following components:

- u Remote Access client, which allows a computer to dial in to a remote access server and use the resource on a LAN. This component can be installed on a Windows NT Workstation computer.
- u Remote dial-in functionality, which allows one inbound remote connection at a time. This component can be installed on a computer running Windows NT Workstation 4.0 that has a modem configured for inbound calls.
- u PPP, which is a set of industry-standard framing and authentication protocols. Computers running Windows NT Workstation 4.0 can dial into remote networks through any server that complies with the PPP standard.
- u SLIP, which is an older protocol that does not support authentication as part of the protocol. SLIP connections typically rely on text-based logon sessions. Encryption and automatic network parameter negotiations are not supported.

Mobile Computing Features

Windows NT Workstation 4.0 has features that support mobile computing and also allow users to switch between portable and docking-station configurations. The features you choose will depend on the particular hardware and working needs of your mobile-computing users. Some of these features are not installed by default, but can be specified in the installation process during Network Setup. Or, they can be configured later, using control panel network settings.

The following mobile computing features might be useful in your organization:

- u Hardware Profiles
- u Remote Access Service client software for dial-up connection to the organization's network
- u Windows NT Workstation 4.0 Briefcase for synchronizing files between computers
- u Direct Cable Connection for directly linking two computers
- u Remote mail and deferred printing, for working away from the main office
- u Broader support for PCMCIA cards
- u User profiles to provide a custom desktop for each user, no matter where users log on to the network

Network Monitor Agent

The Network Monitor Agent on a Windows NT computer collects and displays statistics about activity detected by the network card in the computer. Users can then view these statistics at a computer running Network Monitor Agent. Administrators can use SMS and Network Monitor to collect statistics from computers that are running Network Monitor Agent. This data helps administrators perform routine troubleshooting tasks, such as locating a server that is down or that is receiving a disproportionate number of work requests.

Recommended Features for Network Clients

The following optional features define how Windows NT Workstation 4.0 will be installed and administered in your organization. These features are recommended for your preferred configuration.

- u User-Level Security
- u User Manager
- u User Profile Editor
- u Remote Administration
- u Peer Resource Sharing Services
- u Windows Messaging
- u Separate Memory Spaces For 16-Bit Applications

User-Level Security

User-level security is based on user account lists stored on servers running Windows NT Server or Novell NetWare. The user accounts specify which users have access rights on the network. Windows NT Workstation 4.0 passes on a user's request for access to the servers for validation. Pass-through user-level security protects shared network resources by requiring that a security provider authenticate a user's request to access resources.

Users and groups have access to local shared resources (including the Registry). User-level security is required for remote administration of the Registry and for network access to full user profiles. User-level security includes the following features:

- u Users can specify access rights for individuals and groups to shared resources.
- u User access is validated based on user accounts on a Windows NT domain, or on a Novell NetWare server, via the bindery.
- u User-level security is required for remote administration of the Registry and for network access to full user profiles.

- u Optionally, share-level security can be used to protect files on Windows NT networks or Windows NT Workstation 4.0 peer networks to protect an individual's data on a computer that is used by more than one person.
- u Protection for line-of-business applications is provided to keep users from accidentally — or purposely — installing their own applications, or modifying existing ones.

User Manager

User Manager allows you to edit and control individual user accounts and policies from a central location. A user with Administrative privileges can use the User Rights Policy Editor to define policies for the local workstation, such as which user accounts can be used to access the local workstation from the network.

Use the Account Policy Editor to set password restrictions and account lockouts.

User Profile Editor

User profiles allow multiple users sharing a single computer to customize their desktops and have those custom settings loaded at logon. Conversely, a single user can move between computers using the same profile if the administrator stores that profile on the server.

The administrator can control whether the users can change profile settings (that is, the appearance of the desktop, automatic network connections, etc.). An administrator can also use profiles to require that a mandatory desktop configuration be loaded each time a user logs on.

A local profile is created by default when a workstation user account is created. User profiles are not needed when only one person uses the computer, or when a custom desktop adds no value. If user profiles are not enabled, the logon process is shortened slightly, because the system does not need to locate and load the profile.

To use the User Profile Editor, a Windows NT Server must be available.

Remote Administration

To administer a computer's Registry from a remote computer, you must join a Windows NT Domain and log on with an account that is part of a Domain Admins group. Remote administration capabilities allow you to conduct a variety of tasks remotely over the network. These include administering the file system, sharing or restricting directories, and querying and making changes to the Registry.

Only members of the Domain Admins Group can use the Remote Registry service of Windows NT Workstation 4.0.

Peer Resource Sharing Services

Peer resource sharing services allow a client computer to share files and resources such as printers and CD-ROM drives with other computers. Peer resource sharing can reduce the traffic and disk space required on central servers by leveraging the power of individual computers. If users are allowed to share local resources on their computers, then peer resource sharing can save network traffic and hard disk space on the server. Remember, however, that Windows NT Workstation has a limit of 10 possible inbound connections to other client computers.

Whether to use peer resource sharing services depends on your site's security needs. For central control, or to prevent users from turning on this feature, use the User Rights Policy Editor.

Security for peer resource sharing services takes the form of user-level security based on the user accounts on a Windows NT Server or NetWare network. If you don't have servers to provide security validation or don't want to use user-level security, you can use share-level security, with each user implementing security and a password scheme on the local computer. Share-level security is set on a directory-by-directory basis.

A Microsoft Windows NT Client Access License is required if the computer will be connecting to servers running Windows NT Server. For information on client access licenses, contact your Microsoft reseller.

If you want, you can maintain centralized control. You can prevent users from turning on peer-to-peer networking. To do so, use the following procedure to disable peer resource sharing.

To disable peer resource sharing

1. On the Windows NT Server, start the User Rights Policy Editor in the User Manager for Domains Utility.
2. Select the Show Advanced User Rights checkbox.
3. Select Create Permanent Shared Objects.
4. Remove any groups from the Grant To box.
5. Click OK.

Windows Messaging

The Windows Messaging feature in Windows NT Workstation 4.0 manages all messaging information in one place, with a single inbox for electronic mail and other messages. In addition, Windows NT Workstation 4.0 comes with a complete small-business mail system—that is, a mail client and a post office—which allows users to exchange electronic mail through a single post office. This mail client integrates well with Microsoft Mail servers, and the post office can be upgraded to provide an enterprise-wide mail system.

You can also use a variety of other mail or messaging systems through Windows Messaging, as long as they use a MAPI 1.0 driver. If you have an existing mail system that doesn't use a MAPI 1.0 driver, you can continue to use that mail system without running the Windows Messaging capability.

Separate Memory Spaces for 16-Bit Applications

Windows NT Workstation allows you to run all 16-bit applications in Multiple Virtual DOS Machines (MVDMs). This ensures that an error in one application will not affect other applications, or bring down the entire operating system.

For compatibility issues, some older applications may need to run in shared memory spaces. Test your older line of business applications against MVDMs to decide whether to keep this feature in your ideal client configuration.

Set Up the Test Lab

Using the preferred client configuration specified in the previous phase, proceed with installing the configuration in the lab for testing and evaluation. Because only the client-computer configuration is being installed, this test only determines whether the preferred configuration performs as expected. If you are upgrading an existing network, determine whether the preferred configuration is compatible with your current line-of-business applications and processes.

Depending on how the test installation proceeds, it may be necessary to modify the configuration, by either adding or removing selected features. If more than one configuration is being considered, side-by-side evaluations of different configurations can be performed to help determine which one works best.

This phase in the deployment process involves four significant efforts:

- u Preparing the test lab
- u Installing Windows NT Workstation on a test computer
- u Testing the preferred client configuration
- u Restoring the system to test the restoration process

Prepare the Test Site and Equipment

In the space allotted for the test lab, the installation team should perform the following tasks:

1. Make sure the test computer meets your organization's standards and the Windows NT Workstation 4.0 minimum standards for operation: a 486 or higher processor with 12 MB RAM and 90 MB of free disk space. If not, perform the hardware upgrades now.
2. Defragment the hard disk and scan it for viruses.
3. Back up and verify key data and configuration files, such as INI, Autoexec.bat, and Config.sys files. Also back up the Windows and DOS directories, and all files in the root directory. Make a system startup disk containing Command.com, Sys.com, and Fdisk.exe.

Install Windows NT Workstation on Test Computers

Once the test computers are in place and have been prepared for the installation of Windows NT Workstation, the installation team should proceed with the following steps:

1. Make sure that the current network client software and connections are functioning properly. Working from a checklist of inventoried applications, make sure that all important applications operate correctly on the test computers before installing Windows NT Workstation.
2. Install Windows NT Workstation on one or more test computers, following the directions in the product documentation and using the preferred client configuration identified in the previous phase.

Test the Preferred Configuration

After you've set up your test computers with Windows NT Workstation 4.0, you'll need to run a variety of tests to make sure that the operating system runs correctly on your network and that you can still perform all of your usual tasks. The planning and installation teams should both be involved in this step.

Use your own testing methodology or test the following to verify correct system operation:

- u Can you connect to and browse the network?
- u Can you print both locally and across the network?
- u Can you perform the core operations of each line of business application locally and on the network (including opening, closing, and printing)? Test all mission-critical applications for proper function.
- u Can you shut down successfully?

If you have several test computers, compare your old client configuration from your previous operating system and your new preferred configuration running Windows NT Workstation 4.0. Compare the two in terms of the following:

- u Functionality for administering the computer
- u Performance for local disk/network/application actions
- u Ease of use for performing common tasks
- u Stability of the computer under stress
- u Compatibility with applications and hardware

If the specified client configuration did not work as expected, modify and document the differences until a working preferred client configuration is installed. Try removing related features from the proposed configuration as a solution. Document any changes made to the original configuration.

If the preferred client configuration works as expected, you may also want to conduct additional testing of the optional software features and components in Windows NT Workstation 4.0. This can help you determine whether you are running Windows NT Workstation 4.0 optimally. For this kind of testing, conduct side-by-side evaluations on two computers, changing individual features on each one, to determine the following:

- u Performance in terms of responsiveness and throughput
- u Ease of use
- u Stability
- u Compatibility
- u Functionality

Ensure that you have identified a configuration that meets the needs of all the different hardware configurations used in your organization. Note which options you want to predefine as entries for an answer file. The answer file can then be used to automate the installation process.

Test the Restoration Process

After thorough testing of the preferred client configuration, completely restore all test computers to the previous client configuration and document the process. The degree to which you need to test and restore the computer depends on the tools available.

1. Perform a complete restoration of operating system files and system capabilities for your old client configuration on the computer running Windows NT Workstation 4.0. If you need to automate the restoration, consider using a commercial backup program, instead of using File Manager or MS-DOS commands to copy files.
2. Evaluate the restoration process for problems. Document the process and the modifications made.
3. Have all members of the installation team participate in installing the preferred configuration on a variety of hardware.

Plan the Pilot Rollout

Once you have identified ideal client configurations for Windows NT Workstation, you will need to plan your pilot rollout. This will include planning and lab-testing work. This phase involves three major efforts:

- u Deciding how to automate the installation
- u Determining pilot rollout logistics
- u Preparing the user training plan

Automating Installation

Automating the installation is a key step in reducing the cost of migration. The method of setup automation will depend upon the existing network infrastructure, the number and variety of computers to be upgraded, and the schedule for deployment, among other factors. Based on these criteria, the following options can help you smoothly deploy Windows NT Workstation:

- u Use Setup Manager to create unattended answer files.

By creating an unattended answer file with predetermined answers for installation questions, the installation process can run from start to finish with limited or no user intervention.

- u Use management software such as Microsoft's Systems Management Server.

Management software allows you to "push" the installation from the server, so that you can install Windows NT Workstation 4.0 on an individual personal computer without ever touching the computer. This allows a small Installation team to deploy Windows NT Workstation to a large number of computers in a short period of time. In addition, this method requires little or no work on the part of the end users, allowing them to stay focused on productive work for the organization. Relieving the end users of the installation task also eliminates a potential source of installation errors.

This method is only possible with a system management program such as Microsoft Systems Management Server. If you plan to use system management software in automating the installation, make sure this has been acquired and tested.

Since you can use Systems Management Server to gather the detailed information about the resources already in use on your network, you might want to begin using it during the earliest stages of the deployment process. You can then use it later to simplify the pilot and final rollouts.

- u Modify login scripts, or send embedded setup scripts in e-mail.

You can automate the installation process by editing the login script for the user, or sending a link in electronic mail to a batch file that runs Windows NT Workstation 4.0 Setup. In these cases, the user only needs to log on or double-click an icon to start the installation.

- u Use the Sysdiff utility.

The Sysdiff utility allows you to record the variations between a standard retail installation of Windows NT Workstation and a customized version. If the difference file created by Sysdiff is available during setup, the Setup program applies the differences automatically.

Use Sysdiff to include applications in your customized installation of Windows NT Workstation. The Sysdiff utility is discussed in Chapter 2, “Customizing Setup.”

- u Use third-party solutions (including OEM pre-installation).

Your organization may choose a third party solution for deploying Windows NT Workstation. These include custom solutions made available through a Microsoft Solution Provider, or pre-installation of the operating system by an Original Equipment Manufacturer.

Unattended Answer Files for Windows NT Workstation 4.0 Setup

Unattended answer files allow you to predefine responses to prompts that appear during Windows NT Workstation 4.0 Setup. If you need to conduct a similar installation more than five times, this is a good option. Begin planning for unattended installations while you are specifying the preferred client configuration. Make sure that you document each feature needed, so that you can automate the selection of these features. The chapters that follow provide detailed information on using unattended mode setup to deploy Windows NT Workstation, with or without an existing client-server network. Answer files are discussed in Chapter 2, “Customizing Setup.”

“Push” Installations for Windows NT Workstation 4.0 Setup

You need to understand and plan in advance how the push installation process will work for a given computer. There are several alternatives for remotely initiating the installation, ranging from the use of a system management program to sending a setup script as an embedded link in electronic mail. You will want to consider how to push the installation for each computer and make sure that the client computers are configured to support this process.

For organizations with 50 or more computers, being physically present to install each client computer is not cost effective. An administrative software solution such as Microsoft's Systems Management Server can save you money and increase productivity. When using administrative software tools, additional client-side software may be needed. Be sure to include this software in the installation plan.

For more information, see Chapter 3, "Deploying Windows NT Workstation on an Existing Client-Server Network."

Test the Rollout

Test the deployment method you choose in the lab before performing the pilot rollout. This will help you compose a complete checklist to use during rollout and will help you identify potential problems. To perform the test, the planning and installation teams will need to perform the following steps:

1. Install Windows NT Workstation 4.0 source files on a distribution server. This can be a dedicated server in the lab, or one of the servers used in daily operations in your organization. Make setup choices based on your preferred client configuration tested in the lab.
2. Create and test an automated installation by creating an unattended answer file, using Setup Manager, to predefine answers for Setup. Document the key parts of the setup file that vary depending on which computers are receiving the installation.
3. Determine and test how you will deploy Windows NT Workstation from the server, without having to touch the client computers. Your options include the following:
 - u Use management software such as Microsoft Systems Management Server.
 - u For smaller LANs, use Unattended Setup in conjunction with Setup Manager.
 - u Modify the login scripts on the server.
 - u Send a setup script (batch file) that runs Windows NT Workstation 4.0.
 - u Send Setup as an embedded link in an electronic mail message.
4. Install Windows NT Workstation on one or more computers in the lab, using the files on the distribution server and your chosen method of deployment.
5. Document the process for the rest of the installation team.

Determine Pilot Rollout Logistics

Although it is a test, the first pilot rollout sets the tone and presents an example of the final rollout. You need to be completely prepared for all aspects of the rollout. This requires that you determine the time it will take for installation, the personnel and tools needed to facilitate the process, and the overall schedule.

The installation and planning teams should create a checklist for the pilot rollout. The checklist should cover, in detail, the following general topics:

- u Who will perform the deployment?

In addition to the installation team members, be sure to assign a system administrator with full rights on the server, including the right to administer mail or database server passwords.

- u Who are the pilot users?

Choose a pilot user group or department that is willing and able to accommodate the rollout. This group, ranging from 15 to 50 persons, should be representative of your overall user base. Try not to select a department that is attempting to meet a schedule deadline during the rollout, or a group that is traditionally slow in adopting new technology.

- u What installation process will be used? Does the deployment methodology meet automation goals?
- u What new software or tools must be purchased, and on what schedule?
- u Has a verified backup been performed for each of the target computers?
- u Have passwords been reset for CMOS, the network, and applications?
- u Have virus checking and disk defragmentation been performed?
- u What is the schedule for pilot installations?

When determining the installation time for the pilot rollout, base the projections on how long it takes for installation of an individual computer; remember to schedule the downtime for each user.

- u How many systems will be installed per day?

Start with a conservative estimate and then increase or decrease the number, based on your experiences with the initial installations.

- u At what time of day should the installations occur?

You may want to schedule installations to occur on weekdays after normal business hours or on weekends.

As you develop the checklist of logistics, consider your goals for the pilot rollout and the factors that define its success. For example, you might set a percentage for successful upgrades or for automated installations that, if achieved, would indicate that the rollout had been successful. Or you might set a threshold of end-user downtime. If downtime stays below this threshold, another indicator of success has been met. Document these goals and criteria, so that teams can monitor performance against them during the rollout.

Develop User Training

The steps in developing a training plan include acquiring a training lab, setting up computers in the lab, and appointing a team member as instructor. (If in-house resources are not available, use a vendor to develop and conduct the training.) The instructor will be responsible for creating and testing the training program.

There are a number of training approaches and a variety of tools you can use. The materials provided by Microsoft Authorized Technical Education Centers (ATECs) are a good place to start.

A recommended approach is to divide the training into sessions corresponding to three distinct topics: The Basics, Corporate-Specific Applications, and Customization.

The “Basics” session includes the top ten functions any user needs to know to accomplish daily work, such as the following:

Function	To do the function, use this
Run programs, load documents, find a file	Start button
Change settings	Control Panel
Get help on a specific topic	F1 or Help command
Switch between applications	Taskbar
Minimize, maximize, and close windows	Window buttons
Browse your hard disk	My Computer and Windows Explorer
Connect to a network drive	Network Neighborhood
Print a document	Point and Print

The Windows NT Workstation 4.0 online Help files and the System Guide included with the product provide the information you need to train users in the basics. Schedule training sessions of no more than 30 minutes each; in each session, users receive information that is *sufficient to be productive* using Windows NT Workstation 4.0.

The “Corporate-Specific Applications” session varies according to your network environment and the types of applications that are run on your network. This session should focus on the top five to ten functions that will change because of the upgrade to Windows NT Workstation 4.0.

The “Customization” session is intended for more experienced users. The purpose of this session is to provide information and guidance that will help these users learn on their own after the training, and teach them how to work more productively with Windows NT Workstation 4.0. Some of these topics could include:

- u Adding items to the Start button
- u Adding items to the desktop (move, copy, shortcut)
- u Using options controlled by the right mouse button
- u Adding a new device (for example, a printer)
- u Changing the desktop (for example, screen saver settings)

After creating and testing the program, schedule training sessions to occur immediately before the rollout so that the users will retain most of what they learn by putting it to use right away.

Develop the Support Plan

The support plan must be ready the first day you begin performing Windows NT Workstation 4.0 installations. Because the quality of support available during the pilot rollout will be seen as an indicator of the quality of the rollout as a whole, it is important that you plan carefully to make sure effective support is available.

Staff the support team for your pilot rollout with some of your best technicians, and dedicate their time solely to the pilot group for the first few weeks. The assigned technicians should carry pagers or be available by phone at all times, to give immediate assistance to users. Make sure the users have the names and phone numbers of persons to contact for assistance, as well as a short list of the top questions and answers, and troubleshooting tips.

To help your users help themselves, edit Windows NT Workstation 4.0 Help with organization-specific information on applications or features they will be using with Windows NT Workstation. Doing so requires placing an *Oem.cnt* file and your custom help file in the user’s Windows directory. For information about *Oem.cnt* and the format of Windows NT Workstation 4.0 help files, see the *Win32 Software Development Kit for Windows 95 and Windows NT*.

Notify Users of the Rollout

Another step at this stage is informing users about the pilot rollout plan. You can use a videotape presentation, an interoffice memo, informal lunch seminars, or an organization-wide meeting as the means for communicating with users about the

rollout. Regardless of the form used, the message must explain to users the benefits of moving to Windows NT Workstation 4.0, and describe the overall plan and process by which each group or department will make the move. This makes it easier for your users to plan for and accept the migration to Windows NT Workstation 4.0 as part of their schedules.

Conduct the Pilot Rollout

The schedule for the pilot rollout should simulate—on a smaller scale—the schedule of the final rollout. In this phase you test the capabilities and performance of the system and gather feedback from the users affected by the pilot deployment.

As you conduct the pilot rollout, you may find that certain tasks take more or less time than expected, that some tasks need to be added, or that some tasks can be left out. Be prepared to modify the pilot rollout schedule to account for such changes, and use the pilot schedule for projecting the final rollout timetable.

The following is a summary of the tasks involved in conducting the pilot rollout:

1. Select a pilot user group that is willing and able (particularly in terms of their workload) to handle the installation process. Typically, “power users” provide the ideal target groups for the pilot rollout.
2. Train the users.
3. Back up the Windows and DOS directories and the files on the root directory of the target computers.
4. Following the logistics checklist prepared in the previous phase, perform the installation in the same manner that you expect to install Windows NT Workstation 4.0 throughout the organization. Compare your results against goals and evaluation criteria (developed in the previous task) for this process.
5. Have your technicians on-site for the initial installations, to document the process and problems and to support the users. Have other technicians monitor time and all measurable factors in the installation process. The measurable factors might include the number of attempted installations, the number of successful installations, the elapsed time for each installation (user downtime), and the number of support calls regarding installation. Record these measurements for later evaluation.
6. Make sure that all computers are “up and running” as expected. Make note of possible improvements to the installation, training, or support, where appropriate.
7. Survey members of the pilot user group about their satisfaction with the installation process and take feedback on what could have been done better.
8. Continue to monitor the pilot installation for a week to make sure that everything continues to run smoothly.

9. Prepare a checklist of issues to resolve for the final rollout. Include in this checklist the areas identified in Step 6 as needing improvement, along with comments from the user survey, and the results of comparing your rollout goals and evaluation criteria against actual performance.
10. If the pilot program did not run smoothly or user feedback was poor, conduct additional pilot installations until the process works well.

Test Performance and Capabilities

In addition to the technicians responsible for conducting the pilot installation, extra technicians should be assigned to measure, observe, and test the installation. As this information is gathered and analyzed, areas for improvement or automation become apparent. Consequently, these extra technicians help ensure the success of both the pilot and final rollouts by making the installation more efficient. In addition, after Windows NT Workstation 4.0 is installed, these technicians can test system capabilities, such as remote administration, for proper operation.

Make sure that all computers are “up and running” as expected. Make note of possible improvements to the installation, training, or support, where appropriate. Monitor the client computers for performance, stability, and functionality. Make sure the deployed installations match the lab configurations.

Continue to monitor the pilot installation for a week to make sure that everything continues to run smoothly.

Survey Users for Feedback

The final part of the pilot rollout involves surveying the users to gauge their satisfaction with the new installation, and to evaluate the level of training and support provided.

Survey members of the pilot user group about their satisfaction with the installation process and ask for their suggestions on what could have been done better. Test the users’ proficiency by having them perform a few common tasks or ask them to use several of the features in Windows NT Workstation 4.0—for example, have them register their survey results on the server.

When collected, combine the survey results with the ideas for improvements identified by the planning and installation teams during the pilot rollout. Use this information to prepare a checklist of open issues that must be resolved prior to the final rollout. Then assign to individual team members the actions for solving problems or making improvements. Indicate on the checklist how and when each item was resolved, adjusting the deployment plan if appropriate.

If the pilot program did not run smoothly or if user feedback was poor, conduct additional pilot installations until the process works well.

Finalize the Rollout Plan

The results of the pilot installation provide the basis for developing a final plan for rollout. Using the actual time and resource requirements from the smaller-scale pilot rollout, the installation, training, and support teams can make projections for time and resources, scaled to the organization-wide scope of the final rollout. If additional resources are required, identify these and acquire them at this time. Also, create a template for a central database in which the configuration and uses of each computer on the network can be tracked.

This is a good time to update organization policies and standards regarding computer and network use, to accommodate Windows NT Workstation 4.0.

Complete the Rollout Logistics and Budget

The executive and planning teams now have the information to schedule the final rollout. The planning team can then budget the resources, in terms of personnel and tools, required to meet these goals.

To do this, estimate the number of computers involved in the final deployment, and the time it will require. List the resources needed to complete the process within the stated timeframe. You may need to hire and train additional Installation team members, or purchase additional software or tools.

The executive team might propose a formal budget for the organization-wide implementation, and present it to management for approval. The budget should include the costs for personnel and resources such as system management software.

After obtaining any necessary approval, purchase the resources required for deployment. If you need additional staff, be sure to hire experienced and qualified individuals for the team, and train them extensively before getting started.

Update the Policies and Practices Guidelines

Prior to final rollout, update organization policies regarding the use of the network and computers by employees. Include items such as password length and expiration requirements, and the level of approval needed to obtain remote dial-up privileges.

Update the corporate standards lists for hardware and software usage, and bring all computers into compliance with the updated list prior to deployment. Use the Windows NT 4.0 Hardware Compatibility List (HCL) to update your organization's hardware requirements. The HCL is updated regularly, and is available on the Microsoft web site at <http://www.microsoft.com>.

Tell the users the new standards. Make sure they know that these standards will be enforced prior to the installation, and that they must bring their computers into compliance.

Create a Template for the Rollout Database

Use a central database to track the progress of each computer involved in the deployment, and to document any areas requiring further action.

If you are using Systems Management Server for the deployment, the status of each computer involved in the deployment is automatically maintained in an SQL database. Information from this database can be incorporated into reports or other databases as needed.

If you are not using Systems Management Server for the deployment, create a template for the database before the full deployment. Populate the database with configuration information for every computer and user in the organization, and place the template on the server.

During final deployment, the installation team compiles the data for each computer and user, indicating whether any additional work is needed. The team can then use the template to track open items following the rollout, and to measure actual progress against original objectives.

Roll Out Windows NT Workstation 4.0

After the extensive research, planning, testing, and analysis performed in the previous phases, the deployment teams arrive at the final phase—rolling out the Windows NT Workstation 4.0 installation to the entire organization. At this phase, weeks of preparation pay off in a smooth migration of all your users to an operating system that is more powerful, more robust, and easier to use.

The steps involved in the final deployment are essentially the same as those for the pilot deployment. Briefly, they are as follows:

1. Set up the distribution servers. (Or use the distribution servers from the pilot rollout.)
2. Customize the server installation. (Or use the files from the pilot rollout.)
3. Notify the users of the upcoming installation.
4. Train the users on Windows NT Workstation 4.0.
5. If needed, upgrade the hardware on the client computers, and remove any software not complying with the organization policy.
6. If needed, back up critical data and configuration files on the client computers.
7. If needed, defragment the client hard disks.
8. Optionally, temporarily reset the user password and ID for each computer. This allows technicians easy access to the client computer so they can make sure that the login scripts and environment operate correctly.
9. Make sure that the client computers are fully operational and the network is running.
10. Prepare for the installation process. Depending on your deployment method, this would involve one or more of the following:
 - u Edit the login scripts.
 - u Run the queries and send the package files in the management software.
 - u Send the setup script, by electronic mail, to the user.
11. Initiate the deployment by the means appropriate to your method of deployment. For example, have the users log on, or leave their computers on, at the end of the business day for an overnight installation.

Troubleshooting Problems With Logon Scripts

Use this list to troubleshoot the most common problems with logon scripts:

- u Make sure the logon script is in the directory specified in the Server option of Control Panel. When Windows NT is installed, the logon script directory is as follows:

```
systemroot\system32\repl\import\scripts
```

The only valid path option is a subdirectory of the default logon script directory. If the path is any other directory or it uses the environment variable **%homepath%**, the logon script fails.

- u If the logon script is on an NTFS partition, make sure the user has Read permission for the logon script directory. If no permissions have been explicitly assigned, the logon script might fail without providing an error message.
- u Make sure the logon script has a filename extension of either .cmd or .bat. The .exe extension is also supported, but only for genuine executable programs. If you use a nondefault file extension for your processor, be sure to specify it with the filename of the logon script.

Attempting to use the .exe extension for a script file results in the following error message:

```
NTVDM CPU has encountered an illegal instruction.
```

If this error message appears, close the window in which the logon script is running.

- u If the logon script is to run on a Windows for Workgroups computer, make sure the Windows NT domain name is specified as a startup option in the Network option of Control Panel.
- u Make sure any new or modified logon scripts have been replicated to all domain controllers. Replication of logon scripts happens periodically, not immediately. To manually force replication, use Server Manager. See the Server Manager chapter of the *Windows NT Server System Guide* for detailed information.

BLANK PAGE

IMPORTANT: This text will appear on screen, but will not print on a PostScript printer.

This page should be the last one in this file; it was inserted by running the InsertBlankPage macro.

Do not type any additional text on this page!

CHAPTER 2

Customizing Setup



You can customize the installation of Windows NT Workstation that you distribute to your users. Customization can be as simple as specifying which features in the retail product you want to use. Or, you can include applications or additional files, such as corporate-specific help files, along with the installation of Windows NT Workstation. You can even tailor the customized setup so that different computers get different configurations. And with Unattended Setup, you can supply answers to all the prompts the user would normally need to respond to during setup. Even the computername and username for each computer can be supplied via Unattended Setup.

This chapter is loosely organized according to the features you will most likely need to implement, and the order in which you would perform the tasks. It does not parallel the order of the setup process.

Overview of the Setup Process

The key to deploying Windows NT Workstation is the **winnt** or **winnt32** command. These commands use a network connection to access the installation files, which are in one central location. This central location is called the distribution sharepoint. The **winnt** command is used to install Windows NT Workstation on a computer currently running a 16-bit Windows operating system, such as Windows 3.1 or Windows for Workgroups 3.1, MS-DOS, or Windows 95. The **winnt32** command is used to upgrade from an earlier version of Windows NT Workstation. Either of these commands can be issued directly from the keyboard, or as a line in a .bat file, or via Systems Management Server.

When you use the **winnt** or **winnt32** command, all the files needed to complete the installation are copied over the network to a temporary directory. Then Setup continues as it would if you were performing the installation from a local drive, going through first text mode setup, and then GUI mode setup.

In text mode setup, all the files required for installation are copied from the temporary directory into the installation directory on the hard disk of the target computer. After text mode setup is complete, GUI mode begins. In GUI mode, the user is presented with a Graphical User Interface (GUI), and is prompted for information used to customize the setup. For example, the user may be given the opportunity to choose components to install. During GUI mode Setup, computer-specific information such as the computername and username are supplied.

The winnt and winnt32 Commands

Before using the **winnt** or **winnt32** command, there must be a network connection between the computer being upgraded (the destination computer) and the distribution sharepoint. If the destination computer is already using networking software, the connection can be made in the usual way. If the computer is not yet running networking software, you can use a Network Installation Startup Disk to start the computer and make the network connection. See “Network Installation Startup Disks,” later in this chapter, for information on creating and using a Network Installation Startup Disk.

The **winnt** or **winnt32** command is run from the destination computer. The command can be issued in any of the following ways:

- u Typed from the keyboard
- u Run from a batch file
- u Included in the Network Installation Startup Disk
- u Issued via Systems Management Server

The syntax of the **winnt** command is as follows:

```
winnt [/s:sourcepath] [/i:inf_file] [/t:drive_letter] [/x] [/b] [/o[x]]  
[/u:answer_file] [/udf:id, [UDF_file]]
```

The syntax of the **winnt32** command is as follows:

```
winnt32 [/s:sourcepath] [/i:inf_file] [/t:drive_letter] [/x] [/b]  
[/o[x]] [/u:answer_file] [/udf:id, [UDF_file]]
```

where:

/s:sourcepath

Specifies the location of the Windows NT files.

/i:inf_file

Specifies the filename (no path) of the setup information file. The default is DOSNET.INF.

/t:drive_letter

Forces Setup to place temporary files on the specified drive.

/x

Prevents Setup from creating Setup boot floppies. Use this when you already have Setup boot floppies (from your administrator, for example).

/b

Causes the boot files to be loaded on the system's hard drive rather than on floppy disks, so that floppy disks do not need to be loaded or removed by the user.

/o

Specifies that Setup only create boot floppies.

/ox

Specifies that Setup create boot floppies for CD-ROM or floppy-based installation.

/u:answer_file

Specifies the location of an answer file that provides answers the user would otherwise be prompted for during Setup.

/udf:id [,UDF_file]

Specifies the identifier that is to be used by the Setup program to apply sections of the UDF_file in place of the same section in the answer file. If no UDF is specified, the Setup program will prompt the user to insert a disk that contains a file called \$UNIQUE\$.UDF. If a UDF is specified, Setup will look for the identifier in that file.

Computers running an earlier version of Windows NT Workstation can upgrade to Windows NT Workstation 4.0 using the **winnt32** command. These computers do not need a Network Installation Startup Disk, because they can connect to the network using the built-in networking features of Windows NT Workstation. The options for **winnt32** are the same as those for **winnt**.

Note If you are using **winnt** to install Windows NT Workstation, and you want convert to the NTFS file system, you must do the conversion as a separate step, after installing Windows NT Workstation.

After Windows NT Workstation is installed, restart the computer and use the command **convert /fs:ntfs** to change the file system.

Options for Customizing Setup

The tools for customizing setup include the following:

- u Answer files (Unattend.txt)
- u Uniqueness Database Files (UDFs)
- u The \$OEM\$ directory and its subdirectories
- u The \$OEM\$\$ directory, added to your distribution sharepoint and referenced from the \$OEM\$\Cmdlines.txt file
- u The **sysdiff** utility

Answer Files (Unattend.txt)

Answer files are text files that provide the answers to some or all of the prompts that the end user would otherwise need to respond to during Setup. The answer file is specified with the **/u** option to the **winnt** or **winnt32** command.

The simplest way to use this feature is to specify in the answer file those features of the Windows NT Workstation retail product that you want to include in the installation. However, you can also use the answer file to display company information during GUI-mode setup, to specify a keyboard layout, to join a domain or workgroup, and to install device drivers, protocols, or network adapters.

A sample answer file, Unattend.txt, is included on the Windows NT Workstation 4.0 product CD. Using this file as a template, you can create one or more answer files to customize your installation of Windows NT Workstation. For a general discussion of answer files see “Answer Files (Unattend.txt),” later in this chapter. See Appendix A, “Answer Files and UDFs,” for a discussion of the sections and syntax you can use in your answer files.

Note The options available in Unattend.txt have changed since earlier versions of Windows NT Workstation.

Uniqueness Database Files (UDFs)

Uniqueness Database Files (UDFs) are an extension of the answer file functionality. A UDF has a structure similar to that of an answer file. However, the UDF begins with an additional section in which uniqueness IDs are specified and mapped to sections in the UDF. If a uniqueness ID is specified in the **winnt** or **winnt32** command, any values specified in the sections associated with that uniqueness ID override the values in the sections in the answer file that have the same section names.

One UDF can have numerous uniqueness IDs, and each uniqueness ID can specify a slightly different configuration. For example, you can have a uniqueness ID for each computer in your organization, with a different computername for each uniqueness ID—all in one UDF.

The \$OEM\$ Directory

To install components, files, or applications that are not included with the Windows NT Workstation retail product, the required files must be added to subdirectories of the \$OEM\$ directory on the distribution sharepoint.

Place hardware-dependent files that are loaded during text mode setup in subdirectories of \$OEM\$\Textmode. Create one directory for each device.

Place files that replace or supplement the system files included with the retail product in subdirectories of \$OEM\$\\$.

Place files for network components, such as protocols, adapters, or network services, in subdirectories of \$OEM\$\Net. Create one directory for each component.

Place files for applications that support a scripted (silent) installation, and any other files you want to copy to the destination computers, in subdirectories of \$OEM\$*drive_letter*, where *drive_letter* is the drive on the destination computer in which the application is to be installed. For example, if you want Microsoft Office to be installed on drive C of the destination computers, you would put the installation files in \$OEM\$\C\MSOffice.

To include in your installation of Windows NT Workstation applications that must be installed interactively, use **sysdiff**.

About Sysdiff

To pre-install applications that do not support a scripted installation, you must use **sysdiff**. You can also use **sysdiff** to install other applications.

To use **sysdiff**, first create a snapshot of a reference system. Then install the applications you want to distribute. After you've installed the applications, create a difference file. The information in the difference file includes all the binary files for the applications, as well as the initialization file settings and registry settings for the applications.

A command in the \$OEM\$\Cmdlines.txt file is used to apply the difference file to new installations of Windows NT Workstation. Or, the difference file can be applied to a computer that is already running Windows NT Workstation 4.0 by issuing the same command from the command line.

Because the difference file includes all the files and all the initialization and registry settings for the applications, it can be a very large package, depending on the number and complexity of the applications you have added. Applying such a large package can increase the time required for setup considerably. As an alternative, you can create an information file (INF) from the difference file, which contains only registry and initialization file directives. The command that creates the INF also creates a directory tree within the \$OEM\$ directory structure, that contains all the files contained in the difference file package. These files are then copied along with other files required by setup during the early phases of setup, and are in place when the INF is invoked from the \$OEM\$\Cmdlines.txt file.

Adding Applications

You can install any number of applications along with Windows NT Workstation, using the procedure that follows.

To use unattended setup to distribute applications with the Windows NT Workstation installation

1. If any of the applications you want to preinstall require an interactive installation, prepare the snapshot and difference files, and optionally an INF file generated from the difference file, as described under “Using Sysdiff,” later in this chapter. Install the applications after creating the snapshot file and before creating the difference file. Place the difference file (or, if you prefer, the INF file and directory structure generated from the difference file) on the distribution sharepoint.
2. For each application that has not been installed in step 1, create a subdirectory of the \$OEM\$\\$\$ directory on your distribution sharepoint. Copy all the installation files (including any directory structure) to this directory you have created.
3. Edit Unattend.txt, adding the line **OemPreinstall = Yes** to the **[Unattended]** section. The edited file is the answer file.
4. Add the installation commands to the \$OEM\$\cmdlines.txt file.
5. If you created a difference file as described in step one, add the **sysdiff /apply** command to the \$OEM\$\Cmdlines.txt file. This command is described under “Using Sysdiff,” later in this chapter. Or, if you generated an INF from the difference file, add the command to apply the INF to the \$OEM\$\Cmdlines.txt file.
6. Optionally, create a UDF to further customize the sections of the answer file for individuals or groups. The use and format of the UDF is discussed in the section, “Using Uniqueness Database Files” later in this chapter.
7. With all the required files in place on the distribution share, use the **winnt** or **winnt32** command at the destination computer to perform the installation. The **winnt** and **winnt32** commands are discussed earlier in this chapter, under “The winnt and winnt32 Commands.”

Using Sysdiff

Sysdiff is used in several steps.

1. First, run **sysdiff /snap** to take a snapshot of the Windows NT retail version of the system after it has been installed on a reference computer. The reference computer needs to be of the same general hardware type (x86, MIPS, Alpha, or PowerPC) as the destination computers. The %systemroot% (for example, C:\Winnt) must be the same on the reference and destination computers.
2. Then, after the system has been customized by installing applications, run **sysdiff /diff** again to create the difference file.
3. Finally, run **sysdiff /apply** as part of an Unattended Setup to apply the difference file to a new installation. The **sysdiff /apply** command can also be run at any time after installation is complete.

Alternately, run **sysdiff /inf** to create an INF and add the files included in the difference file to the \$OEM\$ directory tree. Then invoke the INF with a command in the \$OEM\$\Cmdlines.txt file, or at any time after installation is complete.

4. Optionally, you can run **sysdiff /dump** to dump the difference file information in a form that you can read. Use the **sysdiff /dump** command to review the contents of the difference file.

If you want to distribute computers or hard disks that are ready for GUI-mode setup, use the **winnt /u** or **winnt32 /u** command to install your customized version of Windows NT Workstation, including the difference file — but stop the process after text mode Setup is complete by turning off the computer. Then duplicate the hard disk, using the **xcopy** command. The duplicated disks will include all the information copied during text mode, including the contents of the \$OEM\$ directory and its subdirectories. These directories will be deleted, along with other temporary directories used by Setup, after GUI mode setup is completed. GUI mode setup proceeds the first time the computer is turned on.

The sysdiff process is shown in the following illustrations (Figures 2.1 and 2.2).

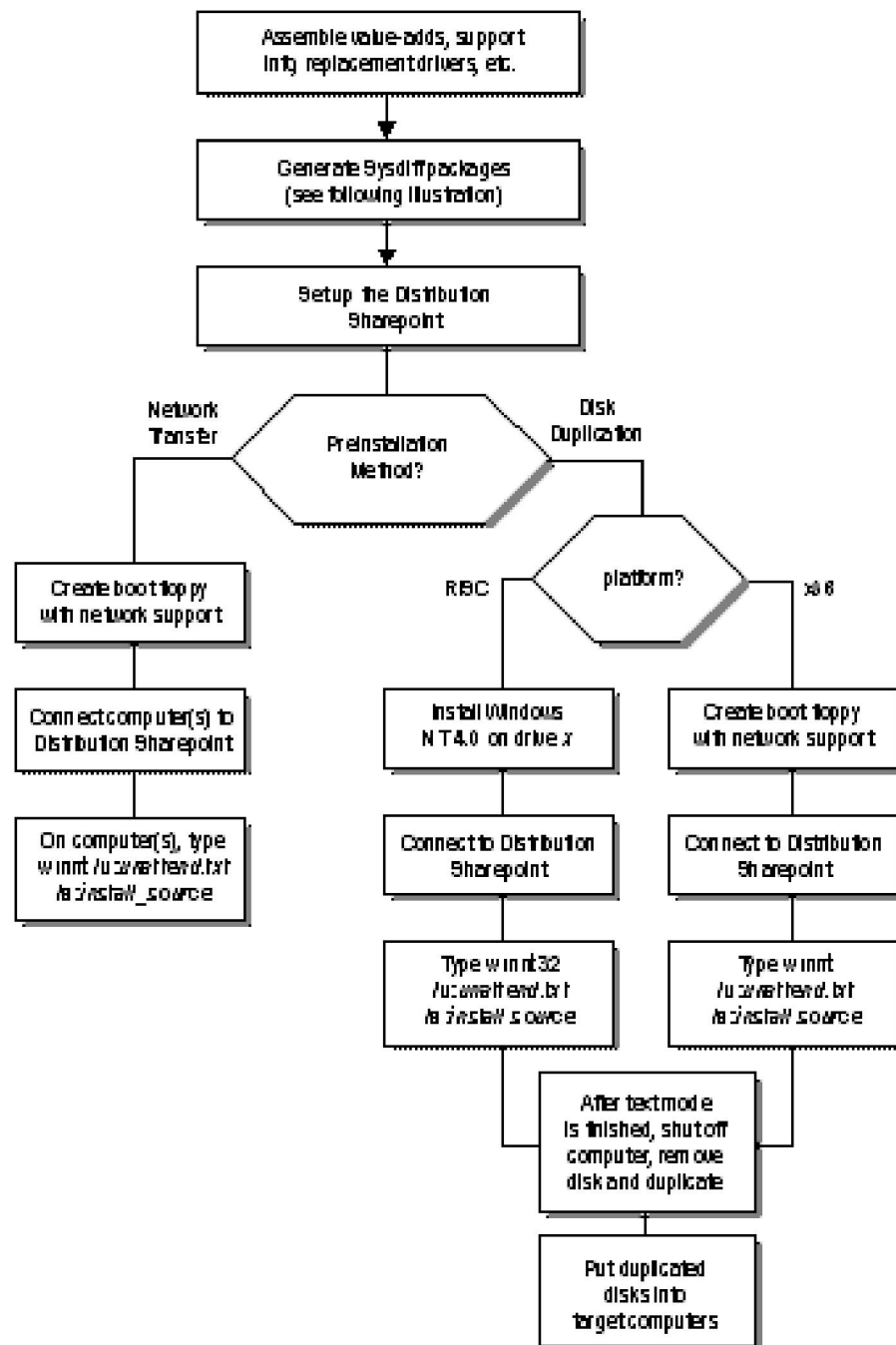


Figure 2.1 Pre-installation using sysdiff

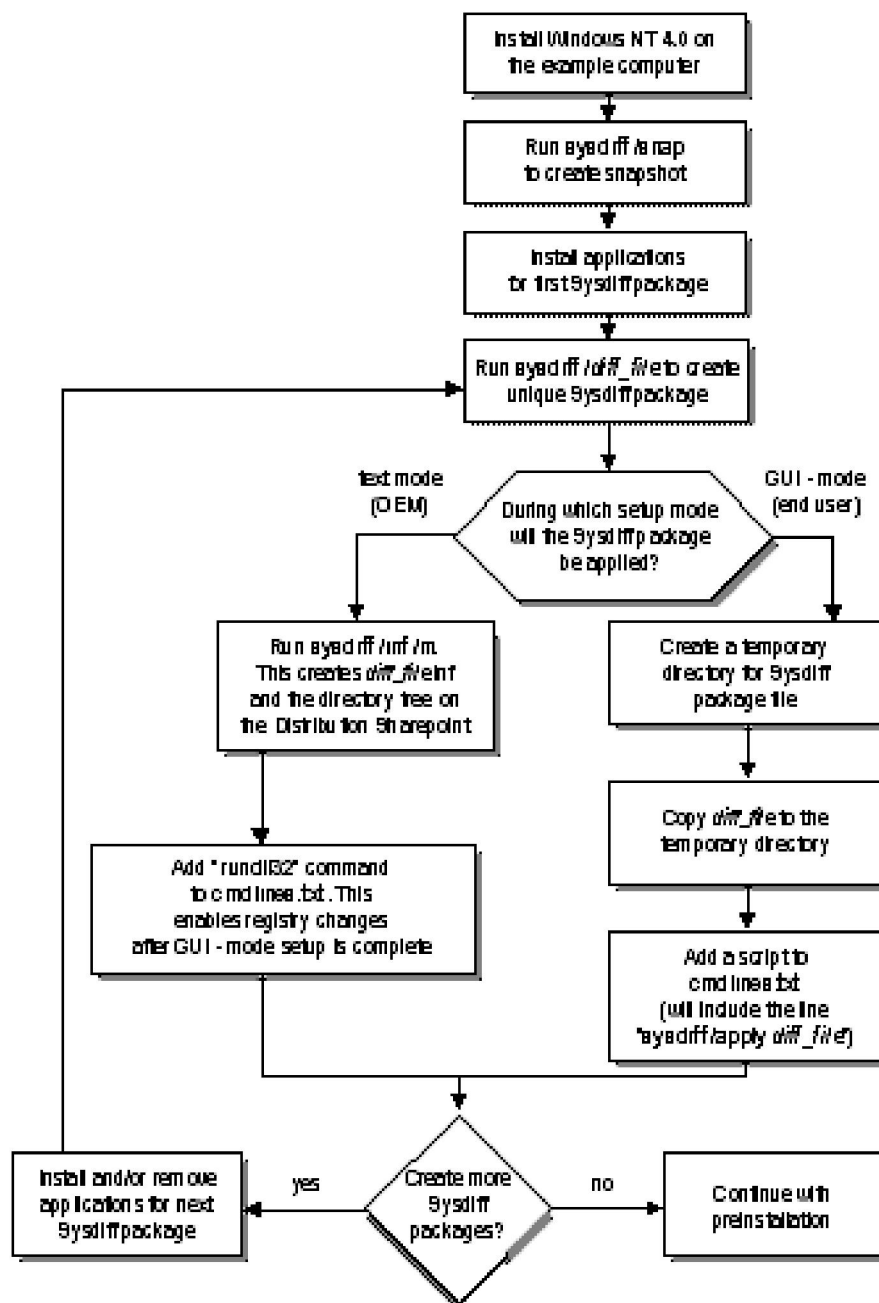


Figure 2.2 Creating and applying a sysdiff package

To create the snapshot file

1. Install the retail version of Windows NT Workstation or Windows NT Server on a computer of the same general type (x86, Alpha, MIPS, or PowerPC) as the destination computers.
2. At the command prompt, type the following command:

```
sysdiff /snap [/log:log_file] snapshot_file
```

where:

log_file

is the name of an optional log file created by **sysdiff**.

snapshot_file

is the name of the file containing the snapshot of the system. Specify any valid Win32-based filename.

To create the difference file

1. After the snapshot file has been created, install the applications that you want to include in the distributed version of the system.
2. At the command prompt, type the following command:

```
sysdiff /diff [/c:title] [/log:log_file] snapshot_file  
difference_file
```

where:

/c:title

specifies a title for the difference package

log_file

is the name of an optional log file created by **sysdiff**.

snapshot_file

is the name of the file containing the snapshot of the system. This file must be created from the same installation as you modified in step one. If you use a file created on another computer or from another installation, **sysdiff** will fail.

difference_file

is the name of the file containing the differences between the system as it existed when the snapshot file was created and the system as it currently exists. Specify any valid Win32®-based filename.

To apply the difference file

1. Place the difference file in a subdirectory of the \$OEM\$\\$\$ directory.
2. Add the following command to the \$OEM\$\Cmdlines.txt file:

```
sysdiff /apply /m [/log:log_file ] difference_file
```

where:

/m

causes the file changes in the per-user profile structure to be mapped to the Default User profile structure, rather than to the profile structure for the user logged on when the difference file was created.

For example, if you were logged on as Administrator when you installed the applications to be included in the difference file, then the .lnk files and other user profile files would be created in the %WinDir%\Profiles\Administrator directory (where %WinDir% is the value of the %USERPROFILE% environment variable). But this directory does not yet exist when the files are copied during GUI-mode setup. For a preinstallation, these files should be placed into %WinDir%\Profiles\Default User instead of %WinDir%\Profiles\Administrator. By using the */m* switch to place the files in the Default User profile structure, you ensure that the applications are available to all users when they first log in.

log_file

is the name of a log file to which **sysdiff** writes information describing its actions. This parameter is optional.

difference_file

specifies a file generated by an earlier invocation of **sysdiff /diff**. The %SystemRoot% must be the same as it was on the system that was used to generate the original *difference_file*. For example, if a difference file was generated on a Windows NT Workstation installation in C:\Winnt, then that difference file can be applied on other computers only if those computers are running Windows NT installed in C:\Winnt.

Note You can also run the **sysdiff /apply** command from the command line. For example, you can use this command to apply a difference file that contains applications to an existing installation of Windows NT Workstation 4.0.

To create an INF

Run the following command:

```
SYSDIFF /inf /m [/u] sysdiff_file oem_root
```

where:

/m

causes the file changes in the per-user profile structure to be mapped to the Default User profile structure, rather than to the profile structure for the user logged on when the difference file was created.

For example, if you were logged on as Administrator when you installed the applications to be included in the difference file, then the .lnk files and other

user profile files would be created in the %WinDir%\Profiles\Administrator directory (where %WinDir% is the value of the %USERPROFILE% environment variable). But this directory does not yet exist when the files are copied during GUI-mode setup. For a preinstallation, these files should be placed into %WinDir%\Profiles\Default User instead of %WinDir%\Profiles\Administrator. By using the **/m** switch to place the files in the Default User profile structure, you ensure that the applications are available to all users when they first log in.

/u

specifies that the INF file be generated as a Unicode text file. By default, the file is generated using the system ANSI codepage.

sysdiff_file

is a Win32 path to a file that was created by sysdiff's difference mode.

oem_root

is the Win32 path of a directory. The \$OEM\$ structure will be created in this directory, and the INF will be placed there with the name Instapp.inf.

The output of this command is a file with the name *dif_file.inf*, where *dif_file* is the name of the difference file (without the three character filename extension). If the difference filename is more than eight characters long, it will be truncated to eight characters. For example, if *sysdiff_file* was named Custom.dif, the INF created by the **sysdiff /inf** command will be Custom.inf. If *sysdiff_file* was named Marketing.dif, the INF created by the **sysdiff /inf** command will be Custom_m.inf.

Because the INF file is copied during text mode Setup, it should be placed in the \$OEM\$\Textmode directory and listed in the Txtsetup.oem file in that directory. On x86 machines, Txtsetup.oem and all files listed in it must also be listed in the [OEMBootFiles] section of the answer file.

To invoke the INF

Add a line to \$OEM\$\Cmdlines.txt to invoke the INF you created from the sysdiff difference file. The command is of the same form as you would use to invoke any Windows 95-style INF. The format is as follows:

```
"RUNDLL32 syssetup,SetupInfObjectInstallAction section 128 inf"
```

where:

Section

specifies the name of the section in the INF file.

Inf

specifies the name of the INF file. This should be specified as a relative path to avoid invoking Setup's default INF rules, which look for an unqualified filename in the system inf directory instead of the current directory. For example, specify **..\newtools.inf**, not just **newtools.inf**.

The command is always enclosed in double quotation marks.

To dump the contents of a difference file

- Use the command,

```
sysdiff /dump difference_file dump_file
```

This creates a text file with the name specified in the *dump_file* parameter that contains information regarding the contents of the difference file.

Sysdiff.inf

When sysdiff is run, it looks for a file called Sysdiff.inf in the directory containing Sysdiff.exe. This file contains information used by sysdiff to exclude certain files and registry entries from snapshots or difference files. The Sysdiff.inf file can be customized, but you should use at least a basic Sysdiff.inf when performing the **sysdiff /snap** and **sysdiff /diff** commands. Otherwise, **sysdiff** will attempt to include such files as Pagefile.sys, which will almost certainly cause **sysdiff** to fail.

Example Sysdiff.inf File

The following is an example of a typical Sysdiff.inf file. It is heavily commented to serve as a guide if you choose to customize the file.

```
[Version]
;
; This section is required, as it identifies the file as
; a Win4-style INF. Just leave this section as-is.
;
Signature = $chicago$

;
; General notes for file/dir exclusion sections:
;
; *: refers to all drives.
; ?: refers to the drive with the system on it.
; :: is substituted with %systemroot%
;
; Lines that are not in valid format (such as those that
; don't start with *:\, ?:\, ::, or <x>:\) are ignored.
;

[ExcludeDrives]
;
; By default, all valid local hard drives are scanned from the root
during
; snapshots and diffs. This section can be used to exclude entire
drives.
; The first character on each line is the drive letter of a hard drive
to exclude.
```

```

;

[ExcludeDirectoryTrees]
;
; By default, all directories on a drive are scanned during
snapshots/diffs.
; This section allows entire directory trees to be excluded.
; Each line is a fully-qualified path of a tree to be excluded -- the
directory
; and all of its subtrees are excluded from the snapshot or diff.
;
*:\recycled
*:\recycler

[ExcludeSingleDirectories]
;
; Each line is a fully-qualified path of a directory to be
; excluded. The directory's subdirs are NOT excluded.
;
::\system32\config

[ExcludeFiles]
;
; By default, all files in all directories are included in
snapshots/diffs.
; This section allows exclusion of individual files.
; Each line is a fully-qualified path of a file to be excluded.
; If it does not start with x:\ then we assume it's a filename part
; for a file to be excluded wherever it is found.
;
*:\pagefile.sys
ntuser.dat
ntuser.dat.log

[IncludeFilesInDir]
;
; Each line in here is a fully qualified path of a directory
; whose files are all to be included in a diff (marked as
; added/changed). Use this if you want to include files in the diff
; that might not have actually been changed.
;

[ExcludeRegistryTrees]
;
; By default, all registry keys in HKEY_LOCAL_MACHINE\System,
; HKEY_LOCAL_MACHINE\Software, and HKEY_CURRENT_USER are scanned during
; snapshots and diffs. This section allows exclusion of entire registry
subkeys.
; Each line indicates a registry key and subkeys to be excluded.
; The first field is one of HKLM or HKCU

```

```
; The second field is the subkey, which must NOT start with a \.
;
HKLM,SYSTEM\ControlSet001
HKLM,SYSTEM\ControlSet002
HKLM,SYSTEM\ControlSet003
HKLM,SYSTEM\ControlSet004
HKLM,SYSTEM\ControlSet005
HKLM,SYSTEM\ControlSet006
HKLM,SYSTEM\ControlSet007
HKLM,SYSTEM\ControlSet008
HKLM,SYSTEM\ControlSet009

[ExcludeRegistryKeys]
;
; Each line indicates a single registry key to be excluded.
; Subkeys of this key are not excluded.
;
; The first field is one of HKLM or HKCU
; The second field is the subkey, which must NOT start with a \.
;
HKCU,Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU

[ExcludeRegistryValues]
;
; Each line indicates a registry value entry to be excluded.
;
; The first field is one of HKLM or HKCU.
; The second field is the subkey, which must NOT start with \.
; The third field is the value entry name.
;
```

Answer Files (Unattend.txt)

You can make unattended answer files for the various setup configurations used in your organization, and then customize them with UDFs as needed. For example, you can make one answer file for each geographic location, or for each division. To tailor the information to individual users, create a UDF with a section for each user or computer that specifies the user name and computer name. Each user can then receive a tailored installation by specifying both the answer file and the UDF in the **winnt** or **winnt32** command.

Answer files and UDFs are text files with specific section names and keys. For a description of these sections and keys, and the customizations you can perform through these files, see Appendix A, “Answer Files and UDFs.”

If you are using Microsoft Systems Management Server (SMS) to manage computer resources in your organization, you can use package definition files (PDFs) to

install or upgrade Windows NT. See the Microsoft Systems Management Server documentation for information on using PDFs to install operating systems.

Creating an Unattended Answer File

You can create one or more unattended answer files for your organization by editing a copy of the Unattend.txt included with the Windows NT Workstation 4.0 Resource Kit. Use any text editor to modify the file, and save the modified copy as a text file. The modified Unattend.txt can be given any legal filename.

The options available in answer files and UDFs are documented Appendix A, “Answer Files and UDFs.”

Using Uniqueness Database Files

When you are deploying a complete, unattended installation of Windows NT Workstation to numerous computers, some of the information you need to supply via answer files (such as the computername) must be unique to each computer. With Windows NT Workstation 4.0, you can use a single answer file for the information that applies to all users, and one or more uniqueness database files (UDFs) to supply information that is specific to a single computer or a small group of computers.

Uniqueness database files (UDFs) are used to provide replacements for sections of the answer file, or supply additional sections. The replacement sections are specified in a text file similar to the answer file. This file is indexed via strings called uniqueness IDs.

The uniqueness database file is used to specify a set of sections that should be merged into the answer file at the start of GUI Setup. This merging takes place before any affected components actually read the internal representation of the answer file. Thus this entire mechanism is totally transparent.

For a discussion of the sections and keys used in answer files and UDFs, see Appendix A, “Answer Files and UDFs.”

Creating the UDF

The first section of the UDF is the [UniqueIds] section. This section lists all uniqueness ids that are supported by this database. The left hand side is a uniqueness ID. The uniqueness ID can be any string but must not contain the star (*), space (), comma (,), or equals (=) character. The right hand side is a list of sections, each of which should match the name of a section in the answer file. The format is as follows:

```
[UniqueIds]
id1 = section1,section2
```

```
id2 = section1,section2
id3 = section1,section3,section4
```

For example, if you were using computernames for the uniqueness IDs, this section might look like this:

```
[UniqueIds]
janeayer = UserData,Unattended
emilybr = UserData,Unattended
jconrad = UserData,KeyboardDrivers, PointingDeviceDrivers
```

Following the [UniqueIds] section are the sections referenced in [UniqueIds]. These section names can take either of two forms: They can be exactly like the section name in the answer file (for example, [Unattended]); or, the section name can be preceded with the uniqueness ID and a colon (for example, [janeayer:UserData]). This allows you to create specialized replacement sections for each computer name.

If both a general section (such as [Unattended]) and an ID-specific section (such as [jconrad:Unattended]) are available, Setup will use the section for the uniqueness ID specified in the **winnt** or **winnt32** command line.

The sections in the UDF can contain any keys and values that the same-named sections could contain in the answer file. During setup, each key specified in a referenced section overrides the value for the same key in the answer file. Values are substituted as follows.

- u If a key is specified in the answer file but not in the UDF section referenced by the uniqueness ID, the value specified in the answer file is used.
- u If a key is specified in the UDF section referenced by the uniqueness ID, the value specified in the UDF is used.
- u If a key is specified in the answer file, and it appears in the UDF section referenced by the uniqueness ID with no value to the right of the equals sign, the default value will be used. This is equivalent to commenting out the key in the answer file.
- u If a key is specified in the UDF, but the value is left blank, no value will be used for that key. This might result in the user being prompted for the information.
- u If a section or key is used in the UDF, but there is no section by that name in the answer file, the section will be created and used by Setup.
- u If a section is referenced in the [UniqueIds] section but does not exist, the user will be prompted to insert a floppy with a valid uniqueness database on it.

Specifying a UDF During Installation

To specify a uniqueness ID during setup, the user must run the **winnt** or **winnt32** command with the following parameter:

```
/UDF:ID[,database_filename]
```

where:

ID

is the uniqueness ID to use while installing Windows NT Workstation on this computer.

database_filename

is the filename, including the full path, of the UDF.

If both the uniqueness ID and the filename of the UDF are specified, the UDF is copied to the local drive during text-mode Setup, and is used during GUI Setup without user intervention. The UDF can have any legal filename.

If only the uniqueness ID is specified on the **winnt** or **winnt32** command line, Setup requires a floppy disk with a UDF file named \$Unique\$.udf. This disk must be prepared by the administrator and made available to the user in advance. The user is prompted for this disk during GUI mode setup.

In either case, if the supplied UDF is corrupt or if Setup cannot locate the specified uniqueness ID in it, the user is prompted either to insert a floppy with the fixed UDF, or Cancel. If the user chooses Cancel, the values in the answer file will be used. These values might not be appropriate for the computer.

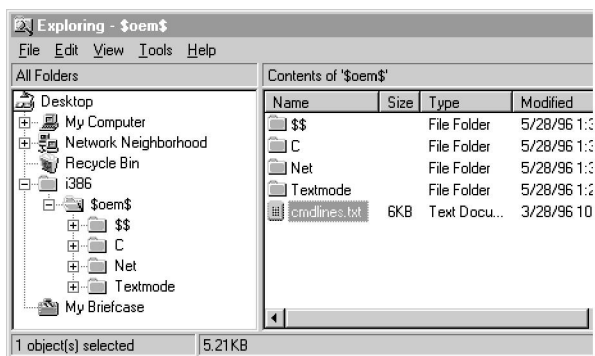
The \$OEM\$ Directory and its Subdirectories

To install components, files, or applications that are not included with the Windows NT Workstation retail product, the required files must be added to subdirectories of the \$OEM\$ directory on the distribution sharepoint.

The \$OEM\$ directory includes the following subdirectories:

- u \$OEM\$
- u \$OEM\$\Textmode
- u \$OEM\$\$
- u \$OEM\$\NET
- u \$OEM\$\drive_letter
- u \$OEM\$\drive_letter

The following sections discuss these directories and their uses.



The \$OEM\$\Cmdlines.txt File

To install files located in the subdirectories of \$OEM\$, list the installation commands in a text file named Cmdlines.txt. Place the Cmdlines.txt file in the \$OEM\$ directory of the distribution sharepoint. The syntax is as follows:

```
[Commands]
"command 1"
"command 2"
"command 3"
.
.
.
```

Enter the entire command line, in double quotation marks. The following is an example of a Cmdlines.txt file with a single command:

```
[Commands]
"Rundll32 setupapi,InstallHinfSection sectionname 128 ..\newtool.inf"
```

The \$OEM\$\Textmode Directory

The \$OEM\$\Textmode directory contains all of the files that the Setup Loader needs to load, and that Text Mode Setup needs to copy to the target computer, so that the system can boot into GUI setup.

If you are installing SCSI, keyboard, video, or pointer device drivers, or HALs, that are not included with the Windows NT Workstation 4.0 retail version, this directory should also contain a standard Txtsetup.oem file. This file contains pointers to all the files required by the Setup Loader and Text Mode Setup to load and install these components. On x86 machines, Txtsetup.oem and all files listed in it (HALs and drivers), must also be listed in the [OEMBootFiles] section of the answer file.

The \$OEM\$\\$\$ Directory

This directory contains system files (either new files or replacements to files included in the retail product) that need to be copied to the various subdirectories in the Windows NT system directory. Maintain the directory structure used in the retail product. In each subdirectory, place the files that need to be copied to the corresponding system directory on the destination computer. If some of the files use long filenames, add the file \$OEM\$\\$\$\\$\$Rename.txt. This file lists all files that have long names, and their corresponding short names.

The \$OEM\$\Net Directory

This directory contains only subdirectories. In each subdirectory place the files for a particular network component, such as network cards, network services, and network protocols. Files in this directory will be used by the network setup module.

The \$OEM\$\drive_letter Directories

The \$OEM\$\drive_letter directories are used to hold files that are to be copied by the Setup program to corresponding drives on the destination computers. For example, files in the \$OEM\$\C directory will be copied to drive C on the destination computer during text mode setup.

Place files for applications that you want to install along with Windows NT Workstation in subdirectories of the \$OEM\$\drive_letter directories on the distribution sharepoint. The files in these subdirectories must have short filenames. To rename them with long filenames after they have been copied to the destination computer, list them in a file named \$\$rename.txt, in the same directory.

Applications included in the \$OEM\$\drive_letter directories must support a scripted (silent) installation; to include applications that require interactive installation, use the **sysdiff** utility.

Setup uses commands specified in the \$OEM\$\Cmdlines.txt file to install applications or copy files that are available in subdirectories of \$OEM\$\drive_letter on the distribution sharepoint. You can use these commands to invoke a Win95-style INF file, run the **sysdiff** command, or perform other actions.

For example, to install MS Office on the C drive, in the \MSOffice directory, place the files in \$OEM\$\C\MSOffice. Then specify the command to install the application in the \$OEM\$\Cmdlines.txt file.

You can also place files that you only want copied (but not “installed”) in subdirectories of \$OEM\$\drive_letter. For example, if you have written help files that cover policies and procedures used in your organization, you can place them in a subdirectory of \$OEM\$\C. Then use a command in the \$OEM\$\Cmdlines.txt file to copy them to the destination computers.

The \$\$Rename.txt Files

To map short filenames used in subdirectories of \$OEM\$ to long filenames that should be assigned to those files after they are copied to the destination computers, list them in a \$\$Rename.txt file. Each subdirectory of \$OEM\$ requires its own \$\$Rename.txt file, if the files in that directory need to be assigned long filenames on the destination computer. For example, if some of the system files in \$OEM\$\$ use long filenames, they must be mapped in the file \$OEM\$\$\$\$\$Rename.txt. If some of the application files in \$OEM\$\C use long filenames, they must be mapped in the file \$OEM\$\C\$\$\$\$Rename.txt. The format for the \$\$Rename.txt file is as follows:

```
[section name]
short name 1 = "long name 1"
short name 2 = "long name 2"
```

where:

section name

is the path to the directory that contains the files. To indicate that the section contains the names of files or subdirectories that are on the root of the drive, use a backslash as the section name, as follows:

[\\]

short name

is the name of the file or subdirectory in this directory, to be renamed.

"long name"

is the new name of the file or subdirectory. This name must be in double quotes.

Network Installation Startup Disks

To deploy Windows NT Workstation to computers that are not currently running networking software, you need to provide some way for them to connect to the distribution sharepoint and begin the installation process. This is done with a Network Installation Startup Disk.

A Network Installation Startup Disk is a bootable MS-DOS system disk containing just enough network client software to connect to the distribution server. You can then use the **winnt** command to install Windows NT Workstation over the network. You can include the **winnt** command line in the Autoexec.bat file on the disk, so that installation begins automatically when the computer is started with the Network Installation Startup Disk in the drive.

This disk can be used to do additional tasks, by adding lines to the Autoexec.bat file on the disk. For example, a series of Network Installation Startup Disks can be produced for an entire division or organization, with each disk specifying a different uniqueness ID in a common UDF. The end users can simply insert the disk into their primary floppy drive, turn on the computer, and walk away, returning to a completed custom installation of Windows NT Workstation.

Creating a Network Installation Startup Disk

Before you create a Network Installation Startup Disk, you need to determine the following:

- u The name to be assigned to the computer.
- u The username to be used. The username identifies a member of the workgroup or domain. Choose a unique name within the workgroup or domain.
- u The name of the user's workgroup and/or domain.
- u The manufacturer and model of the network adapter.
- u The network protocol used on this network.

To create a Network Installation Startup Disk

1. Make sure that the disk has been formatted using the MS-DOS operating system. Use a high-density system disk that fits the destination computer's primary floppy drive. (On most computers this will be drive A. If your system is configured differently, substitute the drive name which acts as your primary floppy drive.) Insert this disk in drive A of a computer running the MS-DOS operating system.
2. At the MS-DOS command prompt, type **sys a:** and then press ENTER to copy the hidden MS-DOS system files (Io.sys and Msdos.sys) and the MS-DOS command interpreter (Command.com) to the network installation startup disk.

3. On the Windows NT Server computer, double-click on the Network Client Administrator Utility in the Network Administration program group. Follow the directions displayed on the screen to create a Network Installation Startup Disk.
4. When prompted to insert a disk, insert the disk created in step 2 in the floppy drive.
5. Continue following the directions displayed on the screen. You now have a Network Installation Startup Disk.
6. The Network Client Administrator utility configures the Network Installation Startup Disk with default settings for the network adapter. Verify that the default settings are correct for your network adapter and modify them if necessary. (The settings are in the A:\Net\protocol.ini file.)

Using the Network Installation Startup Disk

Once the answer files, along with any UDFs or other customization files, are available on the distribution sharepoint, your technicians or end users can use the Network Installation Startup Disk to install Windows NT Workstation over the network.

To use the Network Installation Startup Disk

1. Insert the Network Installation Startup Disk in drive A of the destination computer.
2. Turn on the destination computer. The MS-DOS Network Client installation program is run automatically if the Network Installation Startup Disk is in drive A.
3. When prompted for a username and password, supply the username and password for an account with permission to connect to the directory on the Windows NT Server computer where Windows NT Workstation Setup files are stored. When the computer displays a message about creating a password-list file, type **n** and then press **ENTER**.

4. The client computer must now make a connection to the shared directory on the Windows NT Server computer. To do so, type the following at the command prompt:

```
net use x: \\corpnet\ver4.0  
cd x:
```

where:

x:

is the letter assigned to the network drive, \\corpnet is the name of the distribution server, and ver4.0 is the shared directory containing the Windows NT installation files.

5. Use the command **winnt /u:answer_file** to run the setup program.

For more information, see the guidelines for Computers Running MS-DOS Network Client in Chapter 3, “Deploying Windows NT Workstation on an Existing Client-Server Network.”

Troubleshooting

If the computer displays an error message saying “the specified shared directory cannot be found,” make sure that the directory containing the Windows NT installation files is indeed shared on the Windows NT Server computer.

The NetBEUI protocol can only be used in a LAN or a setup staging area because NetBEUI does not support traffic across routers. Both IPX and TCP/IP can be used in a WAN with Setup Manager. However, not all companies route IPX, so you may need to create boot floppies with TCP/IP for use in a WAN environment.

If the computer displays an error message about lack of memory, modify the Config.sys file on the network installation startup disk to use extended memory. For example, Emm386.exe and Himem.sys provide extended memory for MS-DOS 5.0 and later. If you do not have extended memory, use the NetBEUI or IPX protocols instead of TCP/IP.

Creating .inf Files

Device information (INF) files provide information used by the Windows NT Workstation Setup program to install software that supports a given hardware device. INF files may also be created to permit scripted installations of applications.

Some Windows NT Workstation 4.0 .inf files use the same format as Windows 95 .inf files. Other .inf files use the format used in earlier versions of Windows NT Workstation.

Using Windows 95-style .inf Files

You can use a Windows 95-style .inf file to install applications and other components into Windows NT. For complete details about the syntax and use of statements in Windows 95 INF files, see the *Win32 Software Development Kit for Windows 95 and Windows NT*.

The command line to invoke a Win95-style INF in Windows NT 4.0 is:

```
RUNDLL32 syssetup,SetupInfObjectInstallAction section 128 inf
```

where:

section

specifies the name of the section in the INF.

inf

specifies the name of the INF file. This should be specified as a relative path to avoid invoking Setup's default INF rules, which look for an unqualified filename in the system inf directory instead of the current directory. For example, specify **..\newtools.inf**, not just **newtools.inf**.

For example if you want to install MyGame application into Windows NT, create a Mygame.inf file, such as the following, and place it in the \$OEM\$ directory.

```
This inf is a sample win95 style INF
mygame.inf
installs a game program into Windows NT
```

```
[version]
```

```
signature="$Windows NT$"
```

```
ClassGUID={00000000-0000-0000-0000-000000000000}
```

```
Destination Directories for CopyFiles Sections
```

```
[DestinationDirs]
```

```
BaseCopyProgramFiles           = 24,%INSTALL_DIR%
```

```
MygameDeleteFiles              = 24,%INSTALL_DIR%
```

```
MygameCopyFilesHelp            = 18 LDID_HELP
```

```
[BaseWinOptions]
BaseSection

[Optional Components]
Mygame

[BaseSection]
AddReg                      = BaseAddReg

[Mygame]
OptionDesc                  = %MYGAME_Desc%
Tip                         = %MYGAME_TIP%
IconIndex                   = 64                Windows mini-icon for dialogs
Parent                      = Games
InstallType                 = 0                Manual only.
CopyFiles                   = BaseCopyProgramFiles, MygameCopyFilesHelp
AddReg                      = MYGAME.AddReg
UpdateInis                  = MYGAME.Inis
Uninstall                   = MYGAME.Remove
Upgrade                     = MYGAME.Upgrade
Detect                      = %24%\%INSTALL_DIR%\mygame.exe

[MYGAME.Remove]
AddReg                      = MYGAME.DelReg
DelReg                      = MYGAME.DelPath
DelFiles                    = BaseCopyProgramFiles, MygameCopyFilesHelp,
MygameDeleteFiles
UpdateInis                  = MYGAME.Rem.Inis
Reboot=0

[MYGAME.Upgrade]
CopyFiles                   = BaseCopyProgramFiles, MygameCopyFilesHelp
DelFiles                    = MygameDeleteFiles
AddReg                      = MYGAME.AddReg

[MYGAME.Inis]
setup.ini, progman.groups,, "group15=%GAMES_DESC%"          creates
folder
setup.ini, group15,,
""""%MYGAME_Desc%""", """"%24%\%INSTALL_DIR%\mygame.exe""""",,, """"%24%\%
INSTALL_DIR%""""

[MYGAME.Rem.Inis]
setup.ini, progman.groups,, "group15=%GAMES_DESC%"
folder
setup.ini, group15,, """"%MYGAME_Desc%""""
deletes link

[MYGAME.AddReg]
HKLM,%KEY_OPTIONAL%\Mygame, Installed,, "1"
```

```

HKLM, "%KEY_APP_PATH%\mygame.exe",, , "%24%\%INSTALL_DIR%\mygame.exe"

[MYGAME.DelPath]
HKLM, "%KEY_APP_PATH%\mygame.exe"

[MYGAME.DelReg]
HKLM, %KEY_OPTIONAL%\Mygame, Installed, , "0"

[BaseAddReg]
Create entries for Maint Mode Setup, set all initially to uninstalled:
HKLM, %KEY_OPTIONAL%, "Mygame", , "Mygame"
HKLM, %KEY_OPTIONAL%\Mygame, INF, , "mygame.inf"
HKLM, %KEY_OPTIONAL%\Mygame, Section, , "Mygame"
HKLM, %KEY_OPTIONAL%\Mygame, Installed, , "0"

[BaseCopyProgramFiles]
FONT.DAT
MYGAME.DAT
MYGAME.EXE
MYGAME.MID
MYGAME2.MID
SOUND1.WAV
SOUND2.WAV
SOUND3.WAV

table.bmp
wavemix.inf

[MygameDeleteFiles]
mygame.cnt
mygame.hlp
mygame.inf

[MygameCopyFilesHelp]
mygame.cnt
mygame.hlp

[Strings]
KEY_APP_PATH= "SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths"
KEY_OPTIONAL    =
"SOFTWARE\Microsoft\Windows\CurrentVersion\Setup\OptionalComponents"
GAMES_DESC      = "Accessories\Games"
MYGAME_Desc     = "Mygame"
MYGAME_TIP      = "Company X 3-D mygame game"
INSTALL_DIR     = "Program Files\Windows NT\Mygame"

```

To install MyGame, you would include the following command line in the \$OEM\$\Cmdlines.txt file on your distribution sharepoint.

```
"RUNDLL32 syssetup,SetupInfObjectInstallAction Version 128  
..\mygame.inf"
```

Note All the files needed for this application should be placed in the distribution share point.

Special Issues: Dual Boot Computers

In the deployment lab, you might want to set up some computers to run either Windows NT Workstation 3.51 or Windows NT Workstation 4.0, with the choice of operating system to be made during system startup. This is referred to as dual booting. The following considerations apply in this case:

- u If the configurations using Windows NT Workstation 3.51 and Windows NT Workstation 4.0 are members of the same domain, they must have different computernames.
- u The installation of Windows NT Workstation 4.0 must be done interactively, without using the **winnt /u** or **winnt32** command.

CHAPTER 3

Deploying Windows NT Workstation on an Existing Client-Server Network



There are several methods for upgrading your existing network clients to Windows NT Workstation 4.0.

- u Microsoft Systems Management Server (SMS) provides the most control, speed, security, and flexibility.
- u On smaller LANs, users or technicians can run unattended setup from individual computers, using an answer file. Control of the process is shared with the users or technicians who initiate the unattended setup.
- u Administrators can modify logon scripts so that unattended setup runs automatically when a user logs on to the network.
- u Administrators can create a batch (.bat) file to run unattended setup from individual desktops. They can then send the batch file as an embedded link in e-mail, and the users can run the batch file at their convenience.

“Pull” installations are those in which the user performs the upgrade, using tools provided by the administrator. “Push” installations are those in which the upgrade is performed for the user by the administrator or technicians, either directly or through software such as Systems Management Server. The choice of which type to use depends on several factors, including the corporate culture, the willingness of users to perform the upgrade, and scheduling. For example, if you need all the computers in a department to be upgraded on a certain day, a push installation might be required, since some users might be out of the office that day, or simply might not have a chance to perform the upgrade that day. Push installations can often be done outside regular business hours to minimize the disruption of the users’ workday. In many cases, you can ask users to perform an upgrade within a certain time period, after which the upgrade will be mandatory and will be performed for them. This approach is especially easy with Systems Management Server. This chapter discusses how you can use Systems Management Server to deploy Windows NT Workstation 4.0 throughout an existing client-server network.

Using Systems Management Server for Deployment

Systems Management Server helps you deploy Windows NT Workstation, or any software package, in the following ways.

- u Systems Management Server inventories the computers on the network as the computers log on, at the specific time interval you set. The software and hardware detected on each computer is then listed in an SQL database. This means you can have an up-to-date inventory of the computers in your network at all times, without the expense and disruptions of a manual inventory.
- u With Systems Management Server, you create SMS packages for the configurations that you want to install. Each package contains all the files and information required to install one or more configurations. Packages can be edited, stored, and re-used as often as needed. This means you can be sure that the same package you perfect in the lab and in pilot tests is used to deploy Windows NT Workstation throughout your production network.
- u Query functions in Systems Management Server let you determine which computers meet the criteria for upgrade. You can also use any application that uses SQL to query the database that Systems Management Server creates.
- u You can use queries to identify the computers that are not ready for Windows NT Workstation, and perform necessary hardware upgrades (or delete files to free required disk space) before the scheduled upgrade.
- u Systems Management Server *jobs* deliver the package you created to the computers you specify. You can specify that Systems Management Server perform the installation on only the computers that meet the requirements of a query. Or, perform the query in a separate step, and create a Machine Group from the results of the query. Then send the job to the computers in that Machine Group.
- u After deployment, you can quickly identify and correct any problems, using information from the job statistics produced by Systems Management Server , and the updated inventory.
- u Systems Management Server help desk functions let your support staff provide on-the-spot assistance to your end users, without leaving their own desks.

Systems Management Server gives administrators of large networks an inventory of the computers on the network, which they can query to compile a list of the computers that are capable of running Windows NT Workstation 4.0. Systems Management Server also allows remote administration of utilities to prepare each computer for the installation or upgrade. Systems Management Server can then be used to create a package of installation procedures and a job to execute the package on each workstation with the ability to monitor and evaluate the process while it's carried out.

At this point, it is assumed you have defined your preferred client configurations and set up a LAN in the lab that simulates your production LAN. You should have at least one computer in the lab for each platform (x86, MIPS, Alpha, or PowerPC) you are using in your organization. If you have not already installed Systems Management Server, do so at this time, before proceeding with the lab tests.

Systems Management Server requires an NTFS partition and at least 100 MB of free disk space. You should allow at least 1 GB of disk space for the deployment of Windows NT Workstation. If you did not choose to have Setup convert a partition to NTFS, convert it before beginning the Systems Management Server Setup program. For more information on setting up Systems Management Server, refer to the *Systems Management Server Administrator's Guide*.

Use Systems Management Server to perform a test deployment in the lab. This involves the following steps, which are discussed in greater detail in the remainder of this chapter:

1. Copy the Windows NT Workstation 4.0 setup files, including any customization files, to a shared directory on the distribution server. This will be the distribution sharepoint.
2. Use Systems Management Server queries to determine which computers on the network can support Windows NT Workstation 4.0.
3. Create Machine Groups of these computers.
4. Create a Systems Management Server package to install Windows NT Workstation on the computers in the Machine Groups.
5. Create a Systems Management Server job to execute the package.
6. Monitor the Systems Management Server job status.
7. Evaluate distribution results.

Copy the Setup Files

The setup files for each platform (x86, MIPS, Alpha, or PowerPC) are located in their corresponding directory on the Windows NT Workstation and Windows NT Server Installation CDs. Copy each directory needed for the platforms running on your production LAN to the distribution sharepoint. For convenience, keep the names of the directories on the distribution sharepoint the same for each platform (for example, \i386 copied to \i386). Add any customization files to the distribution sharepoint, as discussed in Chapter 2, “Customizing Setup.”

Be sure the directory on the distribution sharepoint is a shared directory.

Inventory the Test Lab

After Systems Management Server has been installed in your lab or on your production network, it can begin adding computers to its inventory database as the individual computers log on. You can then use the information in the database to determine which computers are ready for the upgrade to Windows NT Workstation 4.0, and which ones need hardware upgrades. Systems Management Server provides an interface in which you can construct queries to the database. The queries you create can be saved, edited, and copied to other sites as needed. They can then be re-used to perform the pilot and final deployments.

Create a “Windows NT Workstation Capable” Query

To identify the computers that have the hardware requirements to run the Windows NT Workstation operating system, create a query from within Systems Management Server. After you run the query, you can create a Machine Group with the results. Then create jobs to run virus detection software and to install Windows NT Workstation on the computers in that Machine Group. Later, you can run queries against the computers in the Machine Group to make sure that all the computers in the group were upgraded as planned.

To create a query

1. Open the Query window in Systems Management Server Administrator.
2. From the File menu, choose New. The Query Properties dialog appears.
3. In the Query Name text box, type the name for the query. The Query Name must be unique within the site.
4. Optionally, in the Comment box, type a comment for the query.
5. In the Architecture box, select Personal Computer, which is the default.
6. Define the criteria for the query by adding expressions to the Find All Personal Computer Items Where box.
7. Click OK to close the Query Properties dialog box.

Once you have created a query to suit your needs, you are ready to run it. The query is run against the contents of the Systems Management Server database that has been populated by Systems Management Server as individual computers logged on.

To run the query

1. From the File menu of the Systems Management Server Administrator window choose Execute Query. The Queries dialog box appears.
2. Drag the query to the site (in the Sites window) on which you want it to run.

By running the query and dragging or pasting the query results into a Machine Group window, you can create a group of all the computers that you want to upgrade to Windows NT Workstation. This group can then be used in subsequent steps.

You might want to create several different machine groups to deploy different configurations. Use variations of the query to create different groups. All the queries should be tested before proceeding further, to make sure you are selecting only those computers you really want to target.

See Chapter 7, “Queries,” in the *Systems Management Server Administrator’s Guide* for more information on creating and executing queries. Also see Chapter 14, “Machine Groups and Site Groups,” in the *Systems Management Server Administrator’s Guide* for more information on machine groups.

Create a Package to Install Windows NT Workstation

Windows NT Workstation allows the use of special information files (.inf files), that can specify all possible custom settings. The format and parameters for these files are described in Chapter 2, “Customizing Setup.”

Systems Management Server can use these information files for automated installations of the software. The vehicle for this is the Package Definition File, or PDF. A PDF is an ASCII file that specifies setup programs, installation options, and execution command lines for the software you will install. If the PDF contains a reference to an .inf file, then the .inf file is made available to the destination computers when the package is delivered through a Systems Management Server job. In fact, several .inf files can be included in a single package.

Since the PDF has options for installing Windows NT Workstation on computers currently running MS-DOS, Windows 3.1, or Windows for Workgroups 3.11, the queries need to target each platform individually.

In order to install Windows NT Workstation on the target computers, for MS-DOS-, Windows 3.1-, and Windows for Workgroups 3.11-based systems, the **winnt** command must be used to install the operating system. For computers that are running an earlier version of Windows NT Workstation, use the **winnt32** command to install the operating system. Both the **winnt** and the **winnt32** commands include a complete set of command-line switches that can further help automate the setup process. For more information on this, see Chapter 2, “Customizing Setup.” Systems Management Server Distribution Packages can be used to run these commands locally on targeted workstations.

Each package is completely self-contained: It has all the files needed for the task or tasks it is designed to do. Furthermore, a single package can contain several different sets of helper files, EXEs, and INFs. The command you want is chosen when you create a job using the package.

To create a package to deploy Windows NT Workstation

1. In the Packages window of Systems Management Server, choose New from the File menu. The Package Properties dialog box appears.
2. Choose Import, and then select the file `\\servername\import.src\enu\nt40.pdf`, where *servername* is the name of the server to which you copied the PDF.

After you select the file, you are returned to the Package Properties dialog box.

3. Choose Windows NT Workstation. The Setup Package for Windows NT Workstation dialog box appears.

Enter the full UNC name of your distribution sharepoint in the Source Directory text box. If you choose to browse for the files, edit the resulting path so that it shows the UNC name (which would begin with `\\servername`) rather than the relative name (which would begin with the drive letter you have assigned to the server during the current session). Be sure to choose the “Automated Setup for Windows NT Workstation” command line.

4. If you want, you can examine the command line for the package by choosing Properties in the Setup Package for Windows NT Workstations dialog box.

You might also choose to view the inventory properties for the package by choosing Inventory in the Setup Package for Windows NT Workstations dialog box.

5. When you are satisfied with the package, click OK until the Package Properties dialog box is closed.

Once the package has been created and appears in the Packages window, you can include it in a job.

For more information on creating packages, see Chapter 10, “Packages,” in the *Systems Management Server Administrator’s Guide*.

Create a Job to Execute the Package

In order for the **winnt** or **winnt32** command in the “setup” package to be run on the destination computers, it must be made available in a Run Command on Workstation job. The job is created by dragging the package to the machine group on the site and filling in the Job Details dialog that appears as a result. You might also choose to take advantage of the job scheduling and configuration options.

After you have set up the job, it appears in the Jobs window as a pending job.

Monitor the Systems Management Server Job Status

Monitor the status of the job by selecting that job in the Job Properties dialog box, clicking on the Status button, and viewing the Sending and Working columns in the Job Status dialog box.

After the job has had a chance to propagate, you can check on details of the job. In the small network you have set up in the lab, this could take as little as 30 minutes. In your production network, you’ll need to allow more time. To view the details of the job, select Details from the Job Status dialog box. The Job Status Details dialog box appears.

When the Status column changes to Complete, the job has been distributed. The total duration of the job depends on a number of factors, including the parameters set for the job and the behavior of your users (who must log on to accept the Systems Management Server Package).

For more information on jobs, see Chapter 11, “Jobs,” in the *Systems Management Server Administrator’s Guide*.

Evaluate Distribution Results

A “complete” job status does not necessarily indicate successful deployment of the operating system. You must perform additional analysis to determine results.

First, make sure the inventory has been updated for all computers in the target group of workstations. Then run queries against the Machine Group to which you sent the Windows NT Workstation installation job. For example, you might create “Successful Windows NT Workstation” and “Unsuccessful Windows NT Workstation” queries.

Examine the computers that were not successfully upgraded on a case-by-case basis. Check, for example, whether there were changes in available disk space or in the availability of some other resource between the time the computer was added to the target group and the time the upgrade was performed.

The computers that were successfully upgraded should be tested to make sure the configuration you specified works as expected.

Job Events

You can check details of job events to troubleshoot problems. For example, if a job has failed, perhaps because the .inf file was not available where it was expected, this would be reported in the job events.

To view the job events

1. From the Systems Management Server Administrator window, choose Jobs. The Jobs window appears.
2. Select the job.
3. From the File menu, choose Properties. The Job Properties dialog box appears.
4. Choose Status. The Job Status dialog box appears.
5. Choose Details.

Viewing job events is further described in the section, “Viewing Job Events and Status” in Chapter 11, “Jobs,” of the *Systems Management Server Administrator’s Guide*.

Job Logs

A log is created as each job progresses; each task is logged to a log file as it is performed. This log file can be examined if the job failed or produced unexpected results, to help you pinpoint the source of the problem.

Use Trace.exe, which you copied to your Systems Management Server server while setting up the lab, to view log files of jobs that have been run. When you start this program, the Systems Management Server Tracer window appears.

You can open a log file by choosing Open from the File menu, and view the contents of the log file.

Additional Help

If you need to contact Microsoft Product Support for help with the deployment, be sure you have copies of the following files for the test or deployment attempt for which you need help:

- u Registry dump
- u Trace logs, including:
 - u Root\System Management Server\logs*.log
 - u Root\Scanman.log
- u All the files, including the PDF, used in your deployment of Windows NT Workstation

Modifying Logon Scripts

Logon scripts for existing network clients can be modified to include a mandatory upgrade. (Logon scripts point to a batch file that runs automatically every time a user logs on.) One example of a logon script might include the command **net time \servername** to set the time on all workstations in the domain to be the same as the server time.

Existing logon scripts can be modified to include the **winnt** or **winnt32** command, providing a forced upgrade when a user logs on. A prompt could be included in the logon script to give the user the option to postpone the upgrade. The user will be prompted for information unless you specify an answer file with the **/u** option and possibly a UDF with the **/udf** option. (For example, if the computername for this particular computer is not specified in an answer file or the UDF that you specify on the command line, the user will be prompted for the computername.) For information on these options to the **winnt** and **winnt32** commands, see Chapter 2, “Customizing Setup.”

Note This method is only viable for smaller LANs. If undue stress is placed on the network from hundreds of concurrent upgrades, the entire network can fail. However, it is a useful alternative in some situations, as when Systems Management Server is not available, staffing resources for deployment are limited, or only a few users typically log on at any one time.

Remember that any time the **winnt** or **winnt32** command is used, the setup files must be placed on a distribution sharepoint. If necessary, answer files may be included and, if used, must be included in the logon script syntax.

Sending a Batch File as an Embedded Link in an e-mail Message

This option is only useful in a small LAN with users who have sufficient expertise to comfortably initiate the upgrade. Users who are given responsibility for their upgrades can introduce variables (such as when each upgrade takes place or what options are included) that can disrupt your planned deployment. To avoid unnecessary stress on the network, send only about 25 setup batch files at one time. Monitor the users who have been given the option to upgrade to keep track of which workstations are upgraded. The files must be kept in place until all users have performed the upgrade. In some cases, you might need to remind users to perform the upgrade so that you can go on to the next group of users, or remove the installation files from the distribution sharepoint.

BLANK PAGE

IMPORTANT: This text will appear on screen, but will not print on a PostScript printer.

This page should be the last one in this file; it was inserted by running the InsertBlankPage macro.

Do not type any additional text on this page!

C H A P T E R 4

Planning for a Mixed Environment



This chapter includes information on the following:

- u Deploying a mixed environment of Windows NT Workstation and Windows 95.
- u Administrative tools to help manage a mixed environment.
- u Networking differences between Windows NT Workstation and Windows 95.

Determining Where to Deploy Windows 95 and Windows NT Workstation

The most important decision an organization needs to make is where to deploy Windows NT Workstation and Windows 95 in its computing environment. The primary criteria for this decision are hardware requirements, device driver support, and software compatibility.

Hardware Requirements

When choosing where to deploy Windows 95 and Windows NT Workstation, you must make sure that destination computers meet the hardware requirements for the operating system they will be using. Also, make sure that the operating system provides support for all of the hardware devices (such as video cards, network adapters, etc.) being used on the destination computer. In many cases, your existing hardware inventory will determine where each operating system is deployed.

Windows 95 is designed to have less demanding hardware requirements, and will work very well with existing hardware. You can run Windows 95 on a computer with the following minimum hardware configuration:

- u 386DX or higher processor
- u 4 MB of memory (8 MB recommended)
- u 35–40 MB of free hard disk space
- u VGA or higher resolution graphics card

Windows NT Workstation 4.0 has higher hardware requirements for Intel-based platforms:

- u 486/33 or higher processor
- u 12 MB of memory (16 MB recommended)
- u 120 MB of free disk space
- u CD-ROM drive (or access to CD-ROM drive over the network)
- u VGA or higher resolution graphics card

Windows NT Workstation also supports symmetric multiprocessing (SMP) configurations and RISC-based platforms such as MIPS, Alpha, and the PowerPC.

Windows 95 provides a broader set of device driver support. Check the Windows NT Hardware Compatibility List (HCL) to ensure that Windows NT drivers exist for all of your devices.

Software Compatibility

Windows NT Workstation provides a standard, comprehensive set of protected environment subsystems that include the following:

- u MS-DOS
- u 16-bit Windows-based applications
- u 32-bit Windows-based applications
- u OS/2 applications¹
- u POSIX

These environment subsystems support most applications available today.

Windows NT Workstation supports most legacy applications, except where the methods used by those applications for accessing system resources would compromise security or reliability. This includes applications that require direct access to system hardware, a virtual device driver (VXD), or a Terminate and Stay Resident program (TSR).

Windows 95 was designed to provide the broadest range of software compatibility, and is often the best choice for older legacy applications. You should test your applications on both Windows 95 and Windows NT Workstation.

¹ OS/2 support is compatible with OS/2 version 1.3 for 16-bit applications. As a separate product available at a nominal cost, Microsoft also offers an additional 16-bit OS/2 Presentation Manager subsystem for Intel-based systems.

Installation Considerations for Windows NT Workstation and Windows 95

After you determine where to deploy both operating systems, review the installation options for Windows NT Workstation and Windows 95. When determining the ideal client configurations for computers in a network that will use both operating systems, be aware of the following installation issues:

- u Server-based setup
- u Upgrading from Windows 95 to Windows NT Workstation and dual-boot scenarios
- u Networking and security options

For detailed information on how to roll-out Windows NT Workstation, see Chapter 1, “Deployment Strategy and Details,” and Chapter 2, “Customizing Setup,” in this deployment guide. For detailed information on how to roll out Windows 95, see Chapters 1 through 6 in the *Windows 95 Resource Kit*.

Server-Based Setup

The term server-based setup refers to running the operating system from the server, rather than locally, on the client computer’s hard disk. When server-based setup is used, the client computer starts with only the files needed to connect to the network and access the operating system files on the server.

Windows NT Workstation does not support server-based setup, due to reliability and security implementations of the operating system. However, Windows 95 can be installed on a server to run as a shared copy on client computers. The benefits of a shared installation include the following:

- u Little or no hard disk space is required on the local computer.
- u Updating drivers for multiple computers is easier.
- u The workstation is more secure and operation is safer for novice users because network logon is required and access to system files is limited. (However, you can also use system policies to enforce the same restrictions for local installations)

There are also some disadvantages to running operating systems from the server:

- u A shared configuration means more network traffic.
- u Larger servers are required.
- u Performance is better when the operating system is installed on the local hard disk.
- u The server must be available for the operating system to work.

Upgrading from Windows 95 to Windows NT Workstation

Differences in the system registries and hardware device support preclude a software upgrade path from Windows 95 to Windows NT Workstation 4.0. Microsoft plans to provide a Windows 95-to-Windows NT Workstation upgrade path in the release following Windows NT Workstation 4.0.

To install Windows NT Workstation on a computer that is currently running Windows 95, you must perform a fresh installation, rather than upgrading Windows 95 with system settings intact. Applications will also need to be re-installed.

Customers planning to upgrade Windows 95 must use the following manual upgrade path:

1. Check to see if all Windows 95 devices and applications are supported by Windows NT Workstation 4.0.
2. Install Windows NT Workstation 4.0 into a separate directory on your hard drive. This will create a “dual boot” system.
3. Re-install all applications.
4. (Optional) Manually delete the Windows 95 directory.

You can upgrade to Windows NT Workstation version 4.0 from any of the following operating systems, keeping applications and system settings as you do so:

- u Any previous version of Windows NT (versions 3.1, 3.5, and 3.51)
- u MS-DOS
- u Windows 3.1
- u Windows for Workgroups 3.11

Dual Booting Between Windows NT Workstation and Windows 95

It is not recommended to set up a computer to dual-boot (giving the user a choice between Windows NT Workstation and Windows 95 when the computer is started). To avoid problems with hardware settings and application installation, choose one operating system for each computer.

If you require a dual-boot configuration, install applications while running Windows NT Workstation and again while running Windows 95, to ensure that the applications are included in the Registries of both operating systems.

Networking Windows NT Workstation and Windows 95

When determining ideal client configurations for a mixed environment of Windows NT Workstation and Windows 95, consider what types of networks these systems will need to support. Both Windows NT Workstation and Windows 95 provide network and protocol support for the following:

- u Client services for NetWare
- u Remote networking
- u TCP/IP, including PPP, SLIP, DHCP, and WINS
- u NetBEUI Protocol
- u NWLink IPX/SPX Compatible Transport
- u Windows Sockets Standard used by popular Internet applications
- u OSF, DCE, and RPC available for advanced client-server applications and used extensively by Windows NT
- u Peer-to-peer networking
- u Internet connectivity

The networking implementation for Windows 95 is slightly different than that for Windows NT Workstation. In order to support all Microsoft networking products that use the Server Message Block (SMB) protocol, install the Client for Microsoft Networks redirector (Vredir.vxd). This allows Windows 95 computers to connect to computers running any of the following networking software:

- u Windows 95
- u Windows NT
- u Windows for Workgroups
- u LAN Manager
- u Workgroup Add-on for MS-DOS

Refer to, “Decide on the Preferred Client Configuration” in Chapter 1 for more information on network protocol support and configuration.

Network Connections

In Windows NT Workstation, a user can always connect to any network resource he or she has permission to access. If the current logon session is for a different account, the user can fill in his or her domain name and username in the Connect As text box in the Connect Network Drive dialog box.

In Windows 95, this “connect as” feature is not available. To connect to a network resource from Windows 95, the user must log on with an account that has access to that resource.

In either operating system, the **net view** command can also be used to map a network drive. For syntax and usage, use the **net help** command.

Persistent Connections

In Windows NT Workstation, persistent network connections are enabled by default. To disable persistent connections, clear the “Reconnect at Logon” checkbox in the Connect Network Drive dialog box.

In Windows 95, Quick Logon is enabled by default. This network logon option restores the mapping of drive letters to network resources, without actually establishing a session for each persistent network connection. If you want to establish sessions for persistent connections during logon, you can enable Logon and Restore Network Connections. For information on how to do so, see chapter 8, “Windows 95 on Microsoft Networks,” of the *Windows 95 Resource Kit*.

For persistent connections from Windows 95, select the “Reconnect at logon” checkbox.

Resource Sharing

Windows NT operating systems allow users to set security permissions on directories, individual files, printers, and other resources. Windows 95 does not support security on individual files.

Before a user can share a resource on a computer running Windows 95, the computer must be configured for share-level or user-level security. Also, File and Printer Sharing services must be installed, using the Network option in Control Panel. The Printers dialog box is used to set access rights to printers in a similar manner.

For computers to share resources such as printers, volumes, CD-ROM drives, and directories, all the computers must be running a common protocol.

Security and Data Protection

Windows NT Workstation was designed to provide the highest levels of system security. Administrators should be aware of the differences in security implementations between Windows NT Workstation and Windows 95 when setting up both peer-to-peer networks and client-server networks.

One of the elements at the core of the Windows NT architecture is integrated security. In its simplest form, Windows NT Workstation includes a secure logon sequence (using the CTRL+ALT+DEL key sequence). This sequence prevents rogue applications from trapping the username and password sequence. In addition, the account lockout feature lets you specify the maximum number of logon attempts. If the correct password is not supplied within this number of attempts, the account cannot be used until an administrator unlocks the account, or a specified period of time has passed. This deters attempts to break into an account by guessing a password.

Windows NT Workstation ensures data and system protection through its ability to define the level of discretionary access control that users can have to the system. The Windows NT security model² allows users to apply security to networking, and to all system objects. Administrators can “lock-down” Windows NT Workstation systems to ensure that end users do not damage key system files or change system configurations. The native Windows NT file system (NTFS) provides this security down to the file level.

Windows NT Workstation also supports multi-user capabilities while retaining a high level of security. Several users can share a single computer system while still maintaining total access control over their personal files. Further, multi-user capabilities allow multiple users to have unique desktops, program groups, and other capabilities.

Windows 95 was not designed to meet such high levels of security and data protection. When specifying a Windows 95 client configuration, administrators need to ensure that user-level security is provided by either Windows NT Server or NetWare, in order to provide pass-through authentication for users accessing resources on remote computers.

² Windows NT was designed to meet C-2 level security specifications, as defined by the United States National Security Agency.

Remote Administration

Several tools are available to help you manage networks using Windows NT Workstation and Windows 95 together. These include the following:

- u System policies and the System Policy Editor
- u User Profiles
- u Systems Management Server or other systems manager software
- u Logon scripts
- u Performance Monitor

For detailed information on system policies, user profiles, and logon scripts, see Chapter 3, “Managing User Work Environments,” in *Windows NT 4.0 Concepts and Planning*. In addition to providing extensive information on the implementation of these features in Windows NT Workstation and Windows NT Server, this chapter includes information on using Windows 95 user profiles on Windows NT Server networks.

System Policies

Like Windows 95, Windows NT Workstation 4.0 now includes system policies, and the System Policy Editor. These are powerful tools for managing a mixed environment network.

System policies allow you to override local registry values for user or computer settings. When a user logs on, system policy settings overwrite the current settings in the user’s registry. This allows administrators to control individual desktop and registry settings.

In Windows 95, however, the following apply:

- u System policies are only stored on the logon server rather than the local computer.
- u Group policies, if used, must be enabled on each computer running Windows 95. You can do so when you install Windows 95, using a custom setup script. Or, use the Add/Remove Programs option in Control Panel after Windows 95 is installed.

System Policy Editor

To centrally manage both Windows NT Workstation and Windows 95 computers, use the System Policy Editor. This tool lets you manage groups of computers by configuring the registry settings of those computers. The alternative is to edit the registry on each computer, using Regedt.exe (for Windows 95) or Regedt32.exe (for Windows NT Workstation). System Policy Editor is easier to use, since it does not require a understanding of the registry structure and syntax.

When you use the System Policy Editor on a computer running Windows NT Workstation or Windows NT Server, a Ntconfig.pol file is created. This file is read and interpreted by the Windows NT Workstation client software, overriding any conflicting information in that computer's registry. When you use the System Policy Editor on a computer running Windows 95, a Config.pol file is created. This file is similar to the Ntconfig.pol file, but has a different file format and reflects differences in the registries of the two operating systems. Place these files on the logon share.

To apply system policies to a network that uses both Windows 95 and Windows NT Workstation, run the System Policy Editor once from each platform, to produce the two different files.

The System Policy Editor cannot be used to set binary values. Hardware component information is often stored as binary data. If you look at a Windows NT registry using RegEdit32.exe, you will see binary value entries marked with data type REG_BINARY. If you look at a Windows 95 registry using RegEdit.exe, you will see binary value entries marked with an icon showing 0s and 1s, rather than the letters ab. These values cannot be set using System Policy Editor. Use the Registry Editor, or make changes on a per-computer basis via the user interface.

You can use the System Policy Editor to edit the HKeyLocalMachine and HKeyCurrentUser portions of the interface. If you need to access a different part of the registry, such as HKeyDynData, use the Registry Editor.

Registry Editor

The alternative to using the System Policy Editor is using the Registry Editor. Both Windows 95 and Windows NT Workstation allow administrators to view and edit the registry on individual remote systems.

You can edit the registry on each computer by using RegEdit.exe (for Windows 95) or RegEdt32.exe (for Windows NT Workstation). For information on the registry and the Registry Editor, see Part 5, “Windows NT Registry.”

User Profiles

User profiles are useful for configuring or managing custom desktops on a Microsoft network (that is, one that uses Microsoft products for networking functionality). A user who logs on from different computers at different times can see the same desktop at every logon. Any changes the user makes to the profile appear the next time the user logs on.

In Windows NT Workstation, a user’s computing environment is determined primarily by the user profile. Windows NT security requires a user profile for each account that has access to the system. User profiles are created by default when a Windows NT user logs on for the first time. You can also create and modify user profiles on a computer running Windows NT Server. To do so, select System from the Control Panel, and select the User Profile tab.

In Windows 95, user profiles are not created by default; they must be enabled by an administrator. For information on setting up User Profiles for Windows 95 computers on a Windows NT Network, see Chapter 15, “User Profiles and System Policies,” in the *Windows 95 Resource Kit*. Also, to take advantage of user profiles from a computer running Windows 95, you must specify Client for Microsoft Networks as the Primary Network Logon client.

Profiles for computers running Windows NT Workstation or Windows NT Server are stored in the “Profile Path” directory on a Windows NT server. The “Profile Path” directory can be found in the user’s account in User Manager for Domains. Windows 95 does not use this directory. Windows 95 profiles are stored in the home directory.

If a user works at a computer running Windows NT Workstation part of the time and at a computer running Windows 95 other times, that user can have two different profiles, one for each operating system. Changes to the settings for one operating system will not affect the settings for the other operating system. (Also, the profiles used by Windows NT Workstation 3.51 and Windows NT Workstation 4.0 differ, reflecting the differences in the interfaces. Users who move between versions of Windows NT Workstation will have different profiles for the two versions of the operating system.)

Because these operating systems use different profiles, some elements of the user environment are easier to control in a mixed environment by creating logon scripts.

Using Logon Scripts

Logon scripts are batch files or executable files that run automatically when a user logs on. They can be used when logging on to a computer running any of the following operating systems:

- u Windows NT Workstation
- u Windows NT Server
- u Windows 95
- u MS-DOS

Logon scripts can configure users' working environments by making network connections and starting applications. You might want to use logon scripts to manage part of the user environment (such as network connections) without managing or dictating the entire environment. Or, you can use logon scripts to create common network connections for multiple users. If you are using LAN Manager 2.x logon scripts, you can continue using them after upgrading to Windows NT Workstation or Windows 95.

Note It is important that logon scripts are in place for every user if you are planning to use a system management software such as Microsoft's Systems Management Server. Logon scripts are modified by the management software in order to take inventory of all the computers on the network.

You can assign a different logon script for each user, or create logon scripts for use by multiple users. To assign a user a logon script, designate the path name of the logon script file in the user's account information on the domain controller. Then, whenever that user logs on, the logon script is downloaded and run.

A logon script is always downloaded from the server that validates a user's logon request. To ensure that logon scripts always work for users, logon scripts for all user accounts in a domain should be maintained on every primary domain controller (PDC) and backup domain controller (BDC) in the domain.

Systems Management Server

The best overall method of remote administration is a systems management program such as Microsoft's Systems Management Server (SMS). With SMS you can manage Windows NT Workstation and Windows 95 clients in much the same manner. Each computer on the network must be configured as an SMS client in order to enable remote administration. Logon scripts must be in place on each computer to be automatically included in the SMS database—otherwise you will have to configure each workstation manually. Windows NT diagnostics and Server Manager functionality are included in SMS for remote administration of Windows NT Workstations and Servers. For more information, see your Systems Management Server documentation.

The alternatives for remote administration are to manage Windows 95 clients from computers running Windows 95, and manage Windows NT Workstation clients from computer running Windows NT Server. For Windows 95 clients, see Chapter 16, "Remote Administration," in *The Windows 95 Resource Kit*.

Server Manager, included with Windows NT Server, is automatically configured for all computers on the network running Windows NT Workstation. Server Manager cannot view or modify computer properties, shared directories, or services on a Windows 95 client. However, it can list the Windows 95 clients on the network. For information on using the Server Manager application, see *Windows NT Server Concepts and Planning*.

Remote Control Agent and Help Desk Options for SMS must be configured on each Windows 95 Client to allow remote administration.

Performance Monitor

Performance Monitor, included with Windows NT Workstation and Windows NT Server, is a graphical tool for measuring the performance of Windows NT computers on the network.

Windows 95 includes a similar tool, System Monitor. These tools are used by administrators to identify bottlenecks or potential problems.

You can use Performance Monitor to monitor the activity on any computer on the network that is running either Windows NT Workstation or Windows NT Server. However, Performance Monitor cannot view the activity on a computer running Windows 95. Likewise, System Monitor can be used to monitor the activity on any computer on the network that is running Windows 95, but it will not view the activity on a computer running Windows NT Workstation or Windows NT Server. A network administrator should have access to a computer running Windows 95, in order to use System Monitor, as well as the computer running Windows NT Workstation or Windows NT Server, in order to use Performance Monitor.

Logon Process

In networks in which the servers run Windows NT Server, user account information can be stored in one of two places: either in a private local user accounts database, or in a domain user accounts database that is shared by all the Windows NT Server computers in the domain.

This section addresses some of the issues involved in logging on to a Microsoft network from computers running Windows NT Workstation and from computers running Windows 95.

Windows NT Workstation

To prepare for account logons from a Windows NT Workstation, use User Manager on the computer running Windows NT Workstation to set up local access to the Windows NT Workstation operating system. Then use the User Manager for Domains on the domain controller to set up access to the specific domain.

When a user logs on to a workgroup computer, that user's logon information is compared with the local user accounts database. When a user logs on to a computer that participates in a domain, that user can choose whether to log on locally, or log on to the domain. (If the domain trusts another domain, the user can alternately choose to log on to the trusted domain.)

If a user's local password doesn't match the password for the domain account, and that user tries to browse the domain or connect to a resource in the domain, access is denied. While tools such as Windows NT Explorer prompt for a valid password, the command-line interface and some applications simply deny access. It is always a better idea to have one set of credentials that apply everywhere in a trusted enterprise.

For a complete discussion of logon scenarios on a Windows NT Network, see the Chapter 2, "Network Security and Domain Planning," in the *Windows NT Server Networking Guide*.

Windows 95

If you want Windows 95 to validate user logons by checking the domain database, logon validation must be enabled on each computer running Windows 95.

To enable logon validation

1. In the Network option in Control Panel on the computer running Windows 95, double-click Client for Microsoft Network in the list of network components.
2. In General Properties, check the Log On to Windows NT Domain option if you want to log on to a Windows NT or LAN Manager domain automatically when starting Windows 95. Otherwise, make sure this option is cleared.
3. When you select logon validation, you must also specify the domain to be used for validation. To do so, type or select a name in the Windows NT Domain box.

Although Windows NT networks allow multiple domains, a computer running Windows 95 can specify only one domain for user-level security. For information on using trust relationships to access multiple domains, see *Windows NT Server 4.0 Concepts and Planning*, which is part of the Windows NT Server documentation set.

For more information, see Chapter 8, “Windows 95 on Microsoft Networks” in the *Windows 95 Resource Kit*.

Password Caching

By default, password caching and unified user logon are enabled in Windows 95. These features work as follows:

When the user supplies a password in order to connect to a resource, that password is saved in a password list file. The next time the user accesses that resource, the password is supplied from the password list. The user only needs to remember one password, the one to log on to the user account.

Password caching and unified user logon are useful when the user needs to log on to multiple networks (for example Windows NT and Novell NetWare networks). To use unified logon, a user account must be available on the network and must contain user account information for the user.

BLANK PAGE

IMPORTANT: This text will appear on screen, but will not print on a PostScript printer.

This page should be the last one in this file; it was inserted by running the InsertBlankPage macro.

Do not type any additional text on this page!

A P P P E N D I X A

Answer Files and UDFs



For information on editing and using answer files and UDFs, see Chapter 2, “Customizing Setup,” of the *Windows NT Workstation 4.0 Resource Guide*. This Appendix describes the sections and keys used in these files. Additional entries may be listed in the sample Unattend.txt file on the product compact disc.

Sections in Answer Files and UDFs

Information in the answer file is divided into sections. Each section begins with the section name in square brackets, on a line by itself. Each section may contain one or more keys, with associated values. Each key is a character string, recognizable by the Setup program, that controls an element of the configuration to be installed. For example, there is a key to specify the computername, a key to specify whether the primary partition is converted to NTFS, and a key to specify the time zone for the computer.

The same sections and keys are used in the UDFs, provided they are used after text-mode setup. Information that the Setup program uses during text-mode setup can only be specified in the answer file, not in the UDF.

The sections are:

[Unattended]

The keys in this section define the behavior of the Setup program during text mode setup. This section can only be specified in the answer file, not in the UDF.

[OEMBootFiles]

To install onto x86 computers, \$OEM\$\OEMFILES\TXTSETUP.OEM and all files listed in it (HALs and drivers), must be listed in this section. This section can only be specified in the answer file, not in the UDF.

[MassStorageDrivers]

This section is used to specify SCSI drivers. This section can only be specified in the answer file, not in the UDF.

[KeyboardDrivers]

This section is used to specify keyboard drivers. This section can only be specified in the answer file, not in the UDF.

[PointingDeviceDrivers]

This section is used to specify pointing device drivers. This section can only be specified in the answer file, not in the UDF.

[OEM_Ads]

This section describes a banner, logo, and background bitmap to be displayed during GUI-mode setup.

[DisplayDrivers]

This section is used to specify display drivers.

[Display]

This section is used to specify settings for video display.

[DetectedMassStorage]

This section is used to specify mass storage devices that Setup is to recognize, even if they are not physically present on the system at the time you perform the installation. This section can only be specified in the answer file, not in the UDF.

[GuiUnattended]

The keys in this section define the behavior of the Setup program during GUI mode setup.

[UserData]

The keys in this section specify information associated with each user or computer.

[LicenseFilePrintData]

This section pertains only to installations of Windows NT Server. It is used to supply information regarding the license agreement.

[Network]

This section is used to specify settings used for networking, such as protocols and adapters. If this section is not present, support for networking will not be installed.

[Modem]

This section header is used to identify whether a modem should be installed or not. It is used by RAS to install a modem if **DeviceType = Modem** is in the list of RAS parameters.

Some of the keys in these sections point to additional sections, which you must create and name. For example, to install protocols other than the defaults, you need to use the **InstallProtocols** key in the **[Network]** section to identify the name of the section in which the protocols are listed. You must also create and populate the section pointed to in the **Install Protocols** key.

Adding Specific Functionality to Your Customized Setup

The keys you can add to the answer file or UDF are listed in the remainder of this Appendix, grouped according to the kind of effect they produce.

Preferred Options in the Retail Product

These options affect the choices a user can make while installing the retail version of Windows NT Workstation.

Preserving Settings of Existing Windows NT Workstation Systems

By default, Setup installs Windows NT Workstation 4.0 with the retail default settings, except where different settings are specified in the answer file and the UDFs. As an option, you can have Setup upgrade existing installations of Windows NT Workstation, with the system settings intact.

To preserve existing settings when upgrading from an earlier version of Windows NT Workstation, add the following line to the [Unattended] section:

```
NtUpgrade = manual | yes | no | single
```

Where:

manual

indicates that the user is to be prompted as to whether the existing installation is to be completely replaced, or upgraded with system settings intact. If the user chooses to replace the existing version, he or she will also be prompted to specify a directory in which to install Windows NT Workstation 4.0. If the user chooses to retain existing settings, the remainder of the settings in the answer file and in the UDF will be ignored.

yes

indicates that the existing installation of Windows NT Workstation is to be upgraded with system settings intact. If **NtUpgrade = yes**, the remainder of the settings in the answer file and in the UDF will be ignored. If multiple installations are detected, the settings in first installation detected are used. On x86-based systems, the system that is listed first in the Boot.ini file is the first installation detected. On Alpha AXP-based, MIPS-based, and PowerPC-based systems, the first installation is the first one listed in the Startup Environment.

no

indicates that the existing installation of Windows NT Workstation is to be completely replaced. The retail default settings will be used, except where keys in the answer file or UDF override them. This is the default option.

single

indicates that if only one existing installation of Windows NT Workstation is detected, it is to be upgraded with system settings intact. If multiple previous installations are detected, the user must indicate which settings are to be retained.

This key can only be specified in the answer file, not in the UDF.

Preserving Settings of Existing Windows for Workgroups 3.1 or Windows 3.1 Systems

By default, Setup installs Windows NT Workstation 4.0 with the retail default settings, except where different settings are specified in the answer file and the UDFs. As an option, you can have Setup upgrade existing installations of Windows for Workgroups 3.1 or Windows 3.1, with the system settings intact.

To preserve existing settings when upgrading from either of these systems, add the following line to the [Unattended] section:

```
Win31Upgrade = yes | no
```

Where:

yes

indicates that if an existing installation of Windows for Workgroups 3.1 or Windows 3.1 is detected, the existing settings are to be preserved.

no

indicates that the existing installation of Windows NT Workstation is to be completely replaced. The retail default settings will be used, except where keys in the answer file or UDF override them. This is the default option.

If any other values are supplied for the Win31Upgrade key, the user will be prompted to specify whether the settings should be retained.

This key can only be specified in the answer file, not in the UDF.

Specifying the Directory in Which to Install Windows NT Workstation

You can specify the directory on the destination computer in which the files for Windows NT Workstation 4.0 are to reside. To do so, add the following line to the [Unattended] section:

```
TargetPath = manual | * | path
```

Where:

manual

indicates that the user is to be prompted for a directory.

*

indicates that the Setup program is to generate a unique directory name, and place the Windows NT Workstation 4.0 files in that directory. This is the default option.

path

indicates the name of the directory in which the Windows NT Workstation 4.0 files are to be placed. If the directory does not exist, it will be created.

This key can only be specified in the answer file, not in the UDF.

Replacing OEM Files with Windows NT Workstation Product Files

You can choose whether to replace older files that Setup finds on the system with files from the retail product that have the same name. By default, these files are overwritten.

To specify whether to keep the older versions of such files, add the following line to the [Unattended] section:

```
OverwriteOemFilesOnUpgrade = yes | no
```

This key can only be specified in the answer file, not in the UDF.

Converting to NTFS

In general, partitions greater than 512 MB should be converted to NTFS. To specify whether to convert the primary partition to NTFS during installation, add the following line to the [Unattended] section:

```
FileSystem = ConvertNTFS | LeaveAlone
```

Specifying When to Configure Graphics Devices

By default, graphics devices are configured during Setup. However, you can specify that the graphics devices be configured the first time someone logs on to the new system. To specify when the graphics devices are to be configured, add the following line to the [Display] section:

```
ConfigureAtLogon = 0 | 1
```

If **ConfigureAtLogon = 0**, graphics configuration will be done during setup. If this option is selected, the following other preconfiguration values may be set, also in the [Display] section:

```
BitsPerPel = n
XResolution = number_of_pixels
YResolution = number_of_pixels
VRefresh = n
Flags = n
```

If an entry has a value of 0, or the entry is not present, the defaults will be applied.

Table A.1 Default Values for Video Settings

Key	Default Value
BitsPerPel	8 If this is not a valid value, the lowest valid value is used.
XResolution	640 If this is not a valid value, the lowest valid value is used.
YResolution	480 If this is not a valid value, the lowest valid value is used.
VRefresh	60 Hz If this is not a valid value, the lowest valid value is used.
Flags	Any that are available.

For example, if all of these keys are set to = 0, the video settings would be 640x480, 256 colors, 60 Hz (as long as that mode exists).

As another example, if BitsPerPel = 16 and all other values are set to 0, then the card will be configured in 640x480, 64 K colors, and 60 Hz (again, as long as that mode exists).

Normally, the user is prompted to test the video settings. For an unattended installation, you might want the settings to be stored without testing. To specify whether the settings should be stored without testing, add the following line to the [Display] section:

```
AutoConfirm = 0 | 1
```

If **AutoConfirm** = **0**, or if it is not specified, the video settings specified elsewhere in the [Display] section are used, and the user must test the selected video mode in order to save the setting and continue with setup.

If **AutoConfirm** = **1**, the settings are stored with no user intervention.

Additional Files and Applications

This section describes keys you can add to the answer file or UDF to add your own files or information to the retail version of Windows NT Workstation.

Displaying Your Company Information During Setup

To specify your own advertising or information during GUI Mode Setup, the following section must exist:

```
[OEM_Ads]
Banner = text
Logo = file_name{, n}
Bitmap = file_name{, n}
```

The value of the **Banner** key is the text string to be displayed as a banner in the upper left of the screen during GUI Mode Setup. The text must contain the substring “Windows NT Workstation” or “Windows NT Server” in order to be displayed. To create a multi-line banner, insert an asterisk (*) where a line break is to appear. If the **Banner** key is empty or does not contain the “Windows NT Workstation” or “Windows NT Server” substring, the Windows NT retail banner will be displayed.

The value of the **Logo** key is the filename of a bitmap to be displayed in the upper right corner of the screen during GUI Mode Setup. If your bitmap is referenced in a dynamic link library (DLL), you can specify the name of the DLL followed by the resource ID. The DLL must be in \$OEM\$\Oemfiles, and the resource ID must be supplied in base-10 format. If only a file name is specified, it must be the name of a .bmp file in \$OEM\$\Oemfiles that contains the bitmap.

If the **Logo** key is not specified, or its value is empty, or it specifies a non-existent bitmap, then the Windows NT retail bitmap will be displayed.

The value of the **Bitmap** key is the name of a bitmap that is to be tiled as a background during end-user setup.

If your bitmap is referenced in a dynamic link library (DLL), you can specify the name of the DLL followed by the resource ID. The DLL must be in \$OEM\$\Oemfiles, and the resource ID must be supplied in base-10 format. If only a file name is specified, it must be the name of a .bmp file in \$OEM\$\Oemfiles that contains the bitmap.

If the **Bitmap** key is not specified, or its value is empty, or it specifies a non-existent bitmap, then the Windows NT retail bitmap will be displayed.

For best results on VGA-based displays this bitmap should be exactly 640 x 480 pixels. On ARC computers (which do not necessarily use VGA mode during setup) other optimal sizes may be possible. If the bitmap is monochrome, it will be tinted blue when displayed (like the retail background bitmap, which is itself monochrome).

Running a Program Concurrently with Setup

To run a custom program concurrently with Setup, add the following line to the [GuiUnattended] section:

```
DetachedProgram = path\filename
```

where *path* and *filename* describe the location of the custom program. For example:

```
DetachedProgram = c:\mt32\mtrun.exe
```

To supply arguments to the custom program, add the following line to the [GuiUnattended] section:

```
Arguments = path\filename/argument
```

where *path\filename* indicates the path to the program and *argument* indicates the arguments you want to supply. For example:

```
Arguments = c:\mt32\scriptl.pcd/H
```

Using Files You Supply in the Subdirectories of \$OEM\$

To include files you have added to customize the application, the following line must be present in the [Unattended] section:

```
OemPreinstall = Yes
```

A **Yes** value indicates that Windows NT Setup might need to install some OEM-supplied files located in subdirectories of \$OEM\$.

The default value is **No**. The default is used when the **OemPreinstall** key is missing. The **No** value implies that Setup will perform a regular unattended installation.

This key can only be specified in the answer file, not in the UDF. The keys described in the following sections are used only if **OemPreinstall = Yes**.

Installing Display Drivers

To install display drivers, include the following section:

```
[DisplayDrivers]
driver_description = Retail | OEM
.
```

This section is optional. If the [DisplayDrivers] section is empty or absent, Setup will install all the Windows NT retail display drivers during Text Mode Setup. During GUI Mode Setup, any drivers that fail to initialize will be disabled.

The *driver_description* value identifies the driver to be installed. This value must match one of the strings defined in the right-hand side of the [Display] section of Txtsetup.sif (except for the string "**Standard VGA (640x480, 16 colors)**") or Txtsetup.oem (for an OEM-supplied driver).

This key can only be specified in the answer file, not in the UDF. This section is valid only if **OemPreinstall = Yes**.

Installing SCSI Drivers

To install SCSI drivers, include the following section:

```
[MassStorageDrivers]
driver_description = Retail | OEM
.
```

This section is optional. If the [MassStorageDrivers] section is empty or absent, Setup will attempt to detect and install all known miniport drivers.

The *driver_description* value identifies the driver to be installed. This value must match one of the strings defined in the [SCSI] section of Txtsetup.sif or Txtsetup.oem (for an OEM-supplied driver).

This key can only be specified in the answer file, not in the UDF. This section is valid only if **OemPreinstall = Yes**.

Installing Keyboard Drivers

To specify keyboard drivers to be installed, include the following section:

```
[KeyboardDrivers]
driver_description = Retail | OEM
.
```

This section is optional. If the [KeyboardDrivers] section is absent or empty, Setup will detect and install a keyboard driver from the retail Windows NT product.

The *driver_description* value identifies the driver to be installed. This string must match one of the strings defined in the [Keyboard] section of the Txtsetup.sif or Txtsetup.oem (for an OEM-supplied driver).

This key can only be specified in the answer file, not in the UDF. This section is valid only if **OemPreinstall = Yes**.

Installing Pointing Device Drivers

To install pointing device drivers, include the following section

```
[PointingDeviceDrivers]
driver_description = Retail | OEM
:
```

This section is optional. If the [PointingDeviceDrivers] section is absent or empty, Setup will detect and install a pointing device driver from the retail Windows NT product.

The *driver_description* value identifies the driver to be installed. This string must match one of the strings defined in the [Mouse] section of Txtsetup.sif or Txtsetup.oem (for an OEM-supplied driver).

This key can only be specified in the answer file, not in the UDF. This section is valid only if **OemPreinstall = Yes**.

Networking Options

If networking is to be installed, you must include a [Network] section. If this section header is missing, network installation will be skipped. If the section is present but empty, the user will be presented with various error messages.

Specifying Manual Installation of Network Components

If you want the network components to be installed interactively during an otherwise unattended installation, include the following line in the [Network] section:

```
Attend = Yes
```

For the installation of network components to proceed without user intervention, this key must be absent.

Detecting Network Adapters

To detect network adapter cards installed on a computer, the following must be specified under the [Network] section:

```
DetectAdapters = adapters_section[,adapters_section]. . .
```

The value for the **DetectAdapters** key is a user-defined section name. For example, you could name a section [DetectAdaptersData] and specify it in this key. Create this section, and add to it the following keys:

```
DetectCount = number_of_cards_to_detect
LimitTo = inf_option [, inf_option] . . .
inf_option = parameter_section
```

The **DetectCount** key specifies the number of detection attempts Setup should make. The default is 1. If more cards than the number specified in **DetectCount** are present, the additional cards will not be detected.

The **LimitTo** key specifies a list of cards; detection will be limited to cards in that list. If this key is not present, any detectable adapters will be installed. The values can be obtained from the network-adapter INF files.

Each *inf_option* listed in the **LimitTo** key can also be listed as a key. The value of these keys will be the parameter normally associated with a network adapter. The values listed here will override detected values.

To detect and install only the first network adapter card, use following line under the [Network] section:

```
DetectAdapters = ""
```

Note that there is no space between the beginning and end quotation marks.

If neither the DetectAdapters key nor the InstallAdapters key (described under “Installing Network Adapter” later in this appendix) is present, user participation will be required. If DetectAdapters is present but contains no value, then the first adapter is detected and installed. The DetectAdapters and InstallAdapters keys can both be present.

Example

```
[Network]
    InstallProtocols = SelectedProtocolsList

[SelectedProtocolsList]
    TC = TCIPParameters
    ATALK = AppleTalkParameters
```

Installing Network Adapters

To install network adapters, the following key must be included in the [Network] section:

```
InstallAdapters = adapters_list_section[, adapters_list_section]. . .
```

The value for the **InstallAdapters** key is a user-defined section name. For example, you could name a section [InstallAdaptersList] and specify it in this key. All network adapters listed in the [InstallAdaptersList] section would then be installed. Create this section, and add to it a line for each adapter to be installed, using the following format:

```
inf_option = parameter_section[, oem_path]
```

If the **InstallAdapters** key and the **DetectAdapters** key are not present, user participation will be required.

Each *inf_option* key in this section defines the *inf_option* to be installed. The answers for the *inf_option* key are adapter-specific. If the adapter is an OEM-supplied card, the path can be specified.

Example

```
[Network]
    InstallAdapters = SelectedAdaptersList

[SelectedAdaptersList]
    NE2000 = NovelParameters
    ELNKII = 3ComParameters
    MSNETULTRA = MacrosoftNetUltraParameters, A:\
```

Note For network adapters, you must know and specify the necessary parameters for the cards you want to preinstall. To determine whether a card supports unattended setup, search for **stf_unattended** in the INFs for that card.

Installing Network Services

To install network services, the following keys should be specified under the [Network] section:

```
InstallServices = services_list_section[, services_list_section] . . .
```

All services listed will be installed. If the **InstallServices** key is absent, no extra services will be installed.

The value for the **InstallServices** key is a user-defined section name. Create the section named by the **InstallServices** key, and in it list the service to be installed, using the following syntax:

```
inf_option = parameter_section[, oem_path]
```

Each *inf_option* key in this section defines the options to be installed. The value for this key is a user-defined section name and the OEM source path, if necessary. Create the section named by the *inf_option* key, and in it list the parameters to be used.

Example

```
[Network]
    InstallServices = SelectedServicesList

[SelectedServicesList]
    SFM = ServicesForMacParameters
    WINS = WINServiceParameters
```

Installing Network Protocols

To detect network adapter cards installed on a computer, the following must be specified under the [Network] section:

```
InstallProtocols = protocol_list_section[, protocol_list_section] . . .
```

The value for the **InstallProtocols** key is a user-defined section name. For example, you could name a section [InstallProtocolsList] and specify it in this key. Create this section, and add to it a line for each protocol to be installed, using the following format:

```
inf_option = parameter_section[, oem_path]
```

The value on the left of the equals sign, *inf_option*, is the notation for the protocol, as it appears in the [Options] section of the INF file for that protocol. Protocol INFs have the filename Oemxpxx.inf, where xx refers to a particular protocol. The value on the right of the equals sign is a user-defined section name and the OEM source path, if necessary. Create the section named by the *inf_option* key, and in it list the parameters to be used for that protocol.

All protocols listed will be installed. If this key is not present, the default protocols will be installed and user participation might be required for Setup to continue.

Specifying the TCP/IP protocol is one case within the general topic of installing protocols, and is discussed in the section, “Installing TCP/IP.”

Example

```
[Network]
    InstallProtocols = SelectedProtocolsList

[SelectedProtocolsList]
    TC = TCPIPParameters
    ATALK = AppleTalkParameters
```

Installing NetBEUI

To install NetBEUI, it must be specified in the “*protocol_list_section*” section specified by the **InstallProtocols** key described in the [Network] section. This is described under “Installing Network Protocols” earlier in this Appendix. In the *protocol_list_section*, include the following line:

```
NBF = NetBEUI_parameters_section
```

The section you specify as the value for NBF must exist, but is empty since NetBEUI does not require any parameters. For example, your answer file might include the following lines:

```
[Network]
    InstallProtocols = SelectedProtocolsList

[SelectedProtocolsList]
    NBF = NetBEUI_params
    TC = TCPIPParameters

[NetBEUI_params]

[TCPIPParameters]
    .
    .
    .
```

Installing TCP/IP

To install TCP/IP, it must be specified in the section specified by the **InstallProtocols** key described in the [Network] section. This is described under “Installing Network Protocols” earlier in this Appendix. Create a separate section for the parameters, and refer to it in the section specified by the **InstallProtocols** key

The supported TCP/IP parameters are as follows:

```
DHCP = Yes | No
ScopeID = scope_ID
```

If you have set DHCP = No, you must also provide the following parameters. For a discussion of these parameters see Chapter 7, “Using Microsoft DHCP Servers,” in the *Windows NT Server Networking Guide*.

```
IPAddress = ddd.ddd.ddd.ddd
Subnet = ddd.ddd.ddd.ddd
Gateway = ddd.ddd.ddd.ddd
DNSServer = IPAddress, IPAddress, IPAddress
           ;Up to 3 IP addresses can be specified
WINSPrimary = IPAddress
WINSSecondary = IPAddress
DNSName = DNS_Domain_Name
```

Both TCP/IP and SNMP allow you to configure computers on a *site basis* and/or on a *computer-specific basis*. A site-based configuration configures each computer with identical parameters, whereas a computer-specific configuration allows a computer to have unique parameters. These two configurations can be mixed together. The generic or site-specific parameters are processed first. If computer-specific parameters exist, they are overlaid on the site-specific ones.

Example: TCP/IP without DHCP

```
[Network]
    InstallProtocols = SelectedProtocolsList

[SelectedProtocolsList]
    TC = TCPIPParameters

[TCPIPParameters]

DHCP          = NO
IPAddress     = 192.9.1.1
Subnet        = 255.0.0.0
Gateway       = 111.1.1.1
DNSServer     = 192.9.1.7, 192.9.1.8, 192.9.1.9
WINSPrimary   = 111.2.2.2
WINSSecondary = 111.3.3.3
DNSName       = microsoft.com
ScopeID       = This_is_the_scope_id
```

Example: TCP/IP with DHCP, and adding DNS and WINS parameters

```
[Network]
    InstallProtocols = SelectedProtocolsList

[SelectedProtocolsList]
    TC = TCIPParameters

[TCIPParameters]

DHCP          = YES
DNSServer     = 192.9.1.7, 192.9.1.8, 192.9.1.9
WINSPrimary   = 111.2.2.2
DNSName       = microsoft.com
ScopeID       = This_is_the_scope_id
```

Installing the SNMP Service

To install the Simple Network Management Protocol (SNMP) service, it must be specified in the section that you create and that you specify in the **InstallServices** key. The **InstallServices** key is described under “Installing Network Services” earlier in this Appendix. On the line that specifies the protocol, you name the section (also created by you) that contains the parameters for the protocol. This is illustrated in the example at the end of this section.

The supported parameters are as follows:

```
Accept_CommunityName = Name1, Name2, Name3 ;maximum of 3 names
Send_Authentication = Yes | No
Any_Host = Yes | No
Limit_Host = host1, host2, host3 ;limit to 3 hosts
Community_Name = name
Traps = IPaddress | IPXaddress ;maximum of 3 addresses
Contact_Name = name ;limit to 1 name
Location = location_name ;limit to 1 name
Service = Physical, Applications, Datalink, Internet, EndToEnd
; Any combination is valid
```

For a discussion of these parameters, see Chapter 11, “Using SNMP for Network Management,” in the *Windows NT Server Networking Guide*.

If a parameter is an IP address, all four octets must be specified in dotted decimal notation only.

Both TCP/IP and SNMP allow you to configure computers on a site basis and/or on a computer-specific basis. A site-based configuration configures each computer with identical parameters, whereas a computer-specific configuration allows a computer to have unique parameters. These two configurations can be mixed together. The generic or site-specific parameters are processed first. If computer-specific parameters exist, they are overlaid on the site-specific ones.

Example

```
[Network]
    InstallServices = SelectedServicesList

[SelectedServicesList]
SNMP = SNMPParameters

[SNMPParameters]
    Accept_CommunityName = MicrosoftCommunity
    Send_Authentication  = YES
    Any_Host              = NO
    Limit_Host            = HostSystem

    Community_Name       = MyCommunity
    Traps                 = 1234567890AB, 192.001.001, 3456123456

    Contact_Name         = NT_Administration
    Location              = Building_26
    Service               = Physical, Datalink, Internet
```

Installing NWLNKIPX

To install NWLNKIPX, it must be specified in the “*protocol_list_section*” section specified by the **InstallProtocols** key described in the [Network] section. This is described under “Installing Network Protocols” earlier in this Appendix. In the *protocol_list_section*, include the following line:

```
NWLNKIPX = NetBEUI_parameters_section
```

The section you specify as the value for NWLNKIPX must exist, but is empty since NWLNKIPX does not require any parameters. For example, your answer file might include the following lines:

```
[Network]
    InstallProtocols = SelectedProtocolsList

[SelectedProtocolsList]
    NWLNKIPX = NWLNKIPX _params
    TC       = TCPIPParameters

[NWLNKIPX _params]

[TCPIPParameters]
    .
    .
    .
```

Installing Client Service for NetWare (NWWKSTA Option)

To install Client Service for NetWare, it must be specified in the section specified by the **InstallServices** key (the **InstallServices** key is included in the [Network] section). To the section specified by the **InstallServices** key, add the following line:

```
NWWKSTA = NetWareClientParametersSection
```

You can assign any name for the *NetWareClientParametersSection*. Create a section with that name and specify the NetWare client parameters that you want to use in that section. The recognized parameters are as follows:

```
DefaultLocation = server_location  
DefaultScriptOptions = 0 | 1 | 3
```

DefaultLocation identifies the default logon server for the NetWare client. The DefaultScriptOptions key specifies the default action to take with scripts, and can be assigned any of the following values:

0	Do not run a script
1	Run NW 3.x level scripts only
3	Run NW 3.x or NW 4.x scripts

Example

```
[Network]  
InstallServices = SelectedServicesList
```

```
[SelectedServicesList]  
NWWKSTA = ClientNetwareParameters
```

```
[ClientNetwareParameters]  
DefaultLocation = NWServer  
DefaultScriptOptions = 0
```

Installing Remote Access Service (RAS)

To install Remote Access Service (RAS), include the following line in the section you defined with the InstallServices key in the [Network] section:

```
RAS = RAS_parameters_section
```

where *RAS_parameters_section* is a section that you define and use to specify the following parameters:

```
PortSections          = MYCOM1, MYCOM2, MYCOM3-25  
DialoutProtocols= TCP/IP | IPX | NETBEUI | ALL>  
DialinProtocols      = TCP/IP | IPX | NETBEUI | ALL  
NetBEUIClientAccess = Network | ThisComputer
```

```
TcpIpClientAccess    = Network | ThisComputer
UseDHCP              = YES | NO
StaticAddressBegin   = IP_address
StaticAddressEnd     = IP_address
ExcludeAddress       = IP_address1-IP_address2
ClientCanRequestIPAddress = YES | NO
IpxClientAccess      = Network | ThisComputer
AutomaticNetworkNumbers = YES | NO
NetworkNumberFrom    = IPX_net_number
AssignSameNetworkNumber = YES | NO
ClientsCanRequestIpxNodeNumber YES | NO
```

Defaults and restrictions are as follows:

PortSections	Any number of port section names
DialoutProtocols	ALL implies all installed protocols
DialinProtocols	ALL implies all installed protocols
NetBEUIClientAccess	Default is Network
TcpIpClientAccess	Default is Network
UseDHCP	Default is YES
StaticAddressBegin	Required if UseDHCP = NO
StaticAddressEnd	Required if UseDHCP = NO
ExcludeAddress	One or more ranges, comma separated. StaticAddressBegin and StaticAddressEnd should be specified.
ClientCanRequestIPAddress	Default is NO
IpxClientAccess	Default is Network
AutomaticNetworkNumbers	Default is YES
NetworkNumberFrom	1 to 0xFFFFFFFF; required if AutomaticNetworkNumbers is NO
AssignSameNetworkNumber	Default is YES
ClientsCanRequestIpxNodeNumber	Default is NO
DeviceType	Modem

If you have specified a “*PortSection*” you must create the section, which can have the following keys:

```
PortName = COM1 | COM2 | COM3-COM25
DeviceType = Modem
DeviceName = device_name
PortUsage = DialOut | DialIn | DialInOut
```

Where:

PortName

indicates the names of the ports to be configured in a particular port section.

DeviceType

indicates the type of device RAS should install. For the current release of Windows NT Workstation, the only available device type is a modem. If **DeviceType = Modem**, the **InstallModem** key must be included in the [Modem] section.

DeviceName

specifies the name of the device to be installed.

PortUsage

defines the dialing properties for the ports being configured.

For a discussion of RAS features and parameters, see Chapter 5, "Understanding Remote Access Service," in the *Networking Supplement* included with the product documentation.

Example: RAS Using One Dialout Port

```
[Network]
    InstallServices = SelectedServicesList

[SelectedServicesList]
    RAS = RemoteAccessParameters

[RemoteAccessParameters]
    PortSections = MYCOM1
    DialoutProtocols = TCP/IP

[MYCOM1]
    PortName      = COM1
    DeviceType    = Modem
    DeviceName    = "Intel Fax Modem 144e"
    PortUsage     = DialOut
```

Example: RAS Using One Dialout, One Dialin and Four Dialin and Dialout Ports, With All Installed Protocols

This configuration uses default values for all protocols:

```
[Network]
    InstallServices = SelectedServicesList

[SelectedServicesList]
    RAS = RemoteAccessParameters
```

```
[RemoteAccessParameters]
  PortSections = Dialin, Dialout, Dialinout
  DialoutProtocols = ALL
  DialinProtocols = ALL
```

```
[Dialin]
  PortName      = COM1
  DeviceType    = Modem
  DeviceName    = "USR Sportster V.34"
  PortUsage     = DialOut
```

```
[Dialout]
  PortName      = COM2
  DeviceType    = Modem
  DeviceName    = "USR Sportster V.34"
  PortUsage     = DialOut
```

```
[Dialinout]
  PortName      = COM3-COM6
  DeviceType    = Modem
  DeviceName    = "USR Sportster V.34"
  PortUsage     = DialInOut
```

Example: RAS 16 Dialin Ports, With TCP/IP and DHCP Enabled

```
[Network]
  InstallServices = SelectedServicesList
```

```
[SelectedServicesList]
  RAS = RemoteAccessParameters
```

```
[RemoteAccessParameters]
  PortSections      = InternetPorts
  DialoutProtocols  = TCP/IP
  DialinProtocols   = TCP/IP
  TcpIpClientAccess = Network
  UseDHCP           = YES
  ClientCanRequestIPAddress = NO
```

```
[InternetPorts]
  PortName      = COM3-COM17
  DeviceType    = Modem
  DeviceName    = "USR Sportster V.34"
  PortUsage     = DialIn
```

Example: RAS 16 Dialin Ports, With TCP/IP and Static IP Address Pool

```
[Network]
    InstallServices = SelectedServicesList

[SelectedServicesList]
    RAS = RemoteAccessParameters

[RemoteAccessParameters]
    PortSections          = InternetPorts
    DialoutProtocols      = TCP/IP
    DialinProtocols       = TCP/IP
    TcpIpClientAccess     = Network
    UseDHCP               = NO
    StaticAddressBegin    = 101.16.1.1
    StaticAddressEnd      = 101.16.1.16
    ClientCanRequestIPAddress = NO

[InternetPorts]
    PortName              = COM3-COM17
    DeviceType            = Modem
    DeviceName            = "USR Sportster V.34"
    PortUsage             = DialIn
```

Example: RAS 16 Dialin Ports, With IPX and Static IPX Address Pool

```
[Network]
    InstallServices = SelectedServicesList

[SelectedServicesList]
    RAS = RemoteAccessParameters

[RemoteAccessParameters]
    PortSections          = InternetPorts
    DialoutProtocols      = IPX
    DialinProtocols       = IPX
    IpxClientAccess       = Network
    AutomaticNetworkNumbers = NO
    NetworkNumberFrom     = 0xa100
    AssignSameNetworkNumber = NO
    ClientsCanRequestIpxNodeNumber = NO

[InternetPorts]
    PortName              = COM3-COM17
    DeviceType            = Modem
    DeviceName            = "USR Sportster V.34"
    PortUsage             = DialIn
```

Hardware Support

This section describes keys you can add to specify what hardware is detected during setup, and what hardware support is installed.

Installing a Modem Driver for Use with RAS

If the RAS parameters include the line `DeviceType = Modem`, the following line must be included in the `[Modem]` section:

```
InstallModem = Modem _Parameter_Section
```

where *Modem_Parameter_Section* is a section that you create and give any name you choose. This section contains the parameters required to install the modem on a specified COM port. If the section is empty, RAS will detect modems on its pre-configured ports and install any modem it finds.

The `[Modem_Parameter_Section]` contains one line for each COM port on which a modem is to be installed, using the following syntax:

```
[Modem_Parameter_Section]  
COMx = "description" [,Manufacturer, Provider]
```

Where:

COMx

identifies the COM port on which the modem is to be installed (for example COM1, COM2...)

description

must match a modem description in a `Mdmxxxxx.inf` file that corresponds to the modem to be installed. This string must be enclosed in quotes.

Manufacturer

is an optional field that identifies the manufacturer of a particular modem when *description* is not unique to a particular manufacturer.

Provider

is an optional field that identifies the provider of a particular modem when *description* is not unique to a particular manufacturer.

Using Hardware Detected by Setup

You can have the user confirm the hardware and mass storage devices detected by Setup, or you can specify that Setup proceed with the installation without seeking confirmation. Mass storage devices include any devices controlled by a SCSI mini port driver. For example, all CD-ROM drives, SCSI adapters, and hard drive controllers (except ATdisk, abiosdsk, and standard floppy disks) are mass storage devices.

To specify whether the user is to confirm the hardware detected, add the following line to the [Unattended] section:

```
ConfirmHardware = yes | no
```

Yes indicates that the user must confirm hardware and mass storage devices detected by the Setup program. **No** indicates that the Setup program should use hardware detected by the Setup program.

This key can only be specified in the answer file, not in the UDF.

Note If OEMPreinstall = yes, then the ConfirmHardware key is ignored.

Specifying Mass Storage Devices

When installing onto an x86-based computer, you can have Setup prepare the system for mass storage devices you specify, even if they are not physically present on the system at the time you perform the installation. To do so, add the section [DetectedMassStorage]. Copy the entries for the devices you want to specify from the [SCSI] section of the Txtsetup.sif file. This file is on the Setup boot floppy disks.

This section can only be specified in the answer file, not in the UDF.

If this section is present and contains entries, support for the specified devices will be included in the installation, as if the devices had actually been detected by Setup.

If this section is present and has no entries, the result is the same as if Setup detected no mass storage devices.

If this section is not present, Setup will attempt to detect the mass storage devices on the computer by loading all known miniport drivers.

Specifying a Keyboard Layout

To specify a keyboard layout, add the following line to the [Unattended] section:

```
KeyboardLayout = layout_description
```

This key is optional. If the **KeyboardLayout** key is not specified or is absent, Setup will detect and install a keyboard layout from the retail Windows NT Workstation product.

The *layout_description* value must match one of the strings in the [KeyboardLayout] section of Txtsetup.sif.

This key can only be specified in the answer file, not in the UDF.

Per-user Settings

For a completely unattended installation, you must specify user settings in either the answer file or the UDF.

Specifying User Data

If the user's full name, the organization name, the computer name, and a product ID for the copy of Windows NT Workstation are not provided to the Setup program, the user will be prompted for them. This information can be included in an answer file, but it is generally more effective to provide it in a UDF. Indeed, the UDF functionality is designed to allow you specify a variant of the basic configuration for each user, rather than create separate answer files, one for each user, with most of the information copied in each answer file.

The section and keys are the same, whether you specify the data in an answer file or UDF. In the [UserData] section, add the following lines:

```
FullName = "User's full name"
OrgName = "Organization name"
ComputerName = computername
ProductId = "123-456789012345"
```

The **ProductId** key specifies the Microsoft product identification number. This must be an authentic number that will be requested by Microsoft Product Support Services if you call for technical support. Note that you must include the hyphen in the product identification number. Note also that for the FullName, OrgName, and ProductId keys, the value must be enclosed in quotation marks.

Specifying a Time Zone

If no time zone is specified, the user will be prompted to select one. To specify a time zone, add the following line to the [GuiUnattended] section:

```
TimeZone = "time_zone"
```

Where *time_zone* is one of the following:

- (GMT) Greenwich Mean Time; Dublin, Edinburgh, London
- (GMT+01:00) Lisbon, Warsaw
- (GMT+01:00) Paris, Madrid
- (GMT+01:00) Berlin, Stockholm, Rome, Bern, Brussels, Vienna
- (GMT+02:00) Eastern Europe
- (GMT+01:00) Prague
- (GMT+02:00) Athens, Helsinki, Istanbul

(GMT-03:00) Rio de Janeiro
(GMT-04:00) Atlantic Time (Canada)
(GMT-05:00) Eastern Time (US & Canada)
(GMT-06:00) Central Time (US & Canada)
(GMT-07:00) Mountain Time (US & Canada)
(GMT-08:00) Pacific Time (US & Canada); Tijuana
(GMT-09:00) Alaska
(GMT-10:00) Hawaii
(GMT-11:00) Midway Island, Samoa
(GMT+12:00) Wellington
(GMT+10:00) Brisbane, Melbourne, Sydney
(GMT+09:30) Adelaide
(GMT+09:00) Tokyo, Osaka, Sapporo, Seoul, Yakutsk
(GMT+08:00) Hong Kong, Perth, Singapore, Taipei
(GMT+07:00) Bangkok, Jakarta, Hanoi
(GMT+05:30) Bombay, Calcutta, Madras, New Delhi, Colombo
(GMT+04:00) Abu Dhabi, Muscat, Tbilisi, Kazan, Volgograd
(GMT+03:30) Tehran
(GMT+03:00) Baghdad, Kuwait, Nairobi, Riyadh
(GMT+02:00) Israel
(GMT-03:30) Newfoundland
(GMT-01:00) Azores, Cape Verde Is.
(GMT-02:00) Mid-Atlantic
(GMT) Monrovia, Casablanca
(GMT-03:00) Buenos Aires, Georgetown
(GMT-04:00) Caracas, La Paz
(GMT-05:00) Indiana (East)
(GMT-05:00) Bogota, Lima
(GMT-06:00) Saskatchewan
(GMT-06:00) Mexico City, Tegucigalpa
(GMT-07:00) Arizona
(GMT-12:00) Enewetak, Kwajalein
(GMT+12:00) Fiji, Kamchatka, Marshall Is.
(GMT+11:00) Magadan, Solomon Is., New Caledonia

(GMT+10:00) Hobart
(GMT+10:00) Guam, Port Moresby, Vladivostok
(GMT+09:30) Darwin
(GMT+08:00) Beijing, Chongqing, Urumqi
(GMT+06:00) Alma Ata, Dhaka
(GMT+05:00) Islamabad, Karachi, Sverdlovsk, Tashkent
(GMT+04:30) Kabul
(GMT+02:00) Cairo
(GMT+02:00) Harare, Pretoria
(GMT+03:00) Moscow, St. Petersburg

For example:

```
TimeZone = "(GMT-08:00) Pacific Time (US & Canada); Tijuana"
```

Joining a Workgroup or Domain

The computer can be part of either a workgroup or a domain. If the computer is a server, a third option, domain controller, is available. The key for establishing the computer as a domain controller is described under “Windows NT Server Settings,” later in this appendix.

To join a workgroup, the following key must be specified under the [Network] section:

```
JoinWorkgroup = workgroup_name
```

The *workgroup_name* value specifies the workgroup the computer will participate in.

To join an existing domain, specify the following under the [Network] section:

```
JoinDomain = domain_name
```

The *domain_name* value specifies the domain to join.

In the lab, you might also want to define user credentials (for example, Administrator) and password required to create a computer account on the domain specified. In this case you would add the following under the [Network] section:

```
JoinDomain = domain_name  
CreateComputerAccount = [user_name_with_creation_rights  
    [, user_password]
```

The **CreateComputerAccount** key supplies the credentials and password to create the computer account. The user account specified in the **CreateComputerAccount** key must already exist in the specified domain.

The [Network] section must contain one of the keys **JoinWorkgroup**, **JoinDomain**, or **InstallDC**. The **InstallDC** key applies only to Windows NT Server, and is described under “Installing a Domain Controller.”

Windows NT Server Settings

The following options apply only to installations of Windows NT Server:

Setting License Options

When installing Windows NT Server, you must specify whether the server is licensed in “per seat” or “per server” mode. If this is not specified in the answer file, Setup will prompt for the information.

To specify the license mode, add the following key to the [LicenseFilePrintData] section:

```
AutoMode = PerSeat | PerServer
```

If the AutoMode key is empty or missing, Setup prompts for the license mode.

PerSeat indicates that a client access license has been purchased for each computer that accesses the server.

PerServer indicates that client access licenses have been purchased for the server to allow a certain number of concurrent connections to the server. If AutoMode is set to PerServer, AutoUsers must also be specified, as follows:

```
AutoUsers = n
```

where n is the number of concurrent client connections that the server is licensed to support. This will be a number between 0 and 999999.

If **AutoMode = PerServer** and **AutoUsers** is missing or empty, Setup will prompt for the number of concurrent client connections that the server is licensed to support.

For information on the licensing modes, see Chapter 12, “Licensing and License Manager,” in the *Concepts and Planning Guide* included with the Windows NT Server retail product. The chapter includes a worksheet to help you decide which licensing mode is best for your organization.

Installing a Domain Controller

When installing Windows NT Server, you can use the `JoinWorkgroup` or `JoinDomain` key in the `[Network]` section (as described under “Joining a Workgroup or Domain” earlier in this Appendix), or you can set the computer to be a domain controller. To establish the computer as a domain controller, add the following line to the `[Network]` section:

```
InstallDC = domain_name
```

If the specified domain does not exist, it will be created. This key is only valid if the server role has been set as a primary or backup domain controller and the **AdvServerType** key has been set accordingly.

If you are creating a backup domain controller (BDC), add the following line to the `[Network]` section to define the user credentials and password required to create a computer account on the domain specified:

```
CreateComputerAccount = [user_name_with_creation_rights  
    [, user_password]
```

The `[Network]` section must contain one of the keys **JoinWorkgroup**, **JoinDomain**, or **InstallDC**.

Example: Creating a Backup Domain Controller for the Marketing Domain

```
[Network]  
InstallDC = Marketing  
CreateComputerAccount = Administrator, FishScale
```

A computer account is also created for this computer, using the Administrator account and supplying the password (FishScale) for the Administrator account.

Setting the Server Role

A computer running Windows NT Server can be a standalone server, a primary domain controller (PDC), or a backup domain controller (BDC). To specify which role the computer is to have in the network, add the following line to the `[GuiUnattended]` section:

```
AdvServerType = SERVERNT | LANMANNT | LANSECNT
```

Where:

SERVERNT

indicates that the computer will be a standalone server.

LANMANNT

indicates that the computer will serve as a primary domain controller (PDC).

LANSECNT

indicates that the computer will be a backup domain controller (BDC).

Installing the Internet Information Server (NT Server Only)

To install Internet Server on a Windows NT Server, add the following line to the [Network] section. This key is only valid for Windows NT Server, and will be ignored when installing Windows NT Workstation.

```
InstallInternetServer = Internet_Server_Parameter_Section  
[Internet_Server_Parameter_Section]
```

where *Internet_Server_Parameter_Section* is a section that you create, and that contains parameters for installing Internet Server.

Example

```
[Network]
```

```
InstallInternetServer = InternetServerParameters  
[InternetServerParameters]
```

Create a section called [INETSTP], and add some or all of the following keys, depending on the features you want to install, to that section:

```
InstallINETSTP= 1 | 0  
InstallFTP=1 | 0  
InstallWWW=1 | 0  
InstallGOPHER=1 | 0  
InstallADMIN=1 | 0  
InstallMOSAIC=1 | 0  
InstallGATEWAY=1 | 0  
InstallDNS=1 | 0  
InstallHELP=1 | 0  
InstallSMALLPROX=1 | 0  
InstallCLIENTADMIN=1 | 0  
WWWRoot=c:\wwwroot  
FTPRoot=c:\ftproot  
GopherRoot=c:\gopheroot  
InstallDir=c:\inetsrv  
EmailName=email_address  
UseGateway=1 | 0  
GatewaysList=\\gateway1 \\gateway2
```

Where the possible values for a key are 1 or 0; 1 means “yes” and 0 means “no.” The effects of these keys are shown in the following table:

InstallINETSTP=	Specifies whether Internet Services will be installed.
InstallFTP=	Specifies whether the FTP Service will be installed.
InstallWWW=	Specifies whether the WWW Service will be installed.
InstallGOPHER=	Specifies whether the Gopher Service will be installed.
InstallADMIN=	Specifies whether the Internet Administrator Manager will be installed.
InstallMOSAIC=	Specifies whether the Internet Explorer Browser will be installed.
InstallGATEWAY=	Specifies whether the Gateway Service will be installed.
InstallDNS=	Specifies whether the DNS Service will be installed.
InstallHELP=	Specifies whether the HELP Files will be installed. (Applies to Windows NT Workstation only.)
InstallSMALLPROX=	Specifies whether the Access Gateway Proxy will be installed. (Applies to Windows NT Workstation only.)
InstallCLIENTADMIN=	Specifies whether the client administrator tools will be installed. (Applies to Windows NT Workstation only.)
WWWRoot=c:\wwwroot	Specifies the virtual root for the WWW service.
FTPRoot=c:\ftproot	Specifies the virtual root for the FTP service.
GopherRoot=c:\gopheroot	Specifies the virtual root for the Gopher service.
InstallDir=c:\inetsrv	Specifies the installation directory for all components of Internet Services.
EmailName=email_name	Specifies the Internet e-mail name of the user.
UseGateway=	Specifies whether a gateway is to be used.
GatewaysList=\\gateway1 \\gateway2	Lists the gateways to be used (if UseGateway=1).

Using an Extended Partition for Text Mode Files

If you are installing Windows NT Workstation on a disk larger than 2 GB, you can cause Setup to extend the partition on which the temporary files are located into any available unpartitioned space that physically follows it on the disk. These temporary files must be installed on a primary partition. The space used must be limited to 1024 cylinders. Writing beyond cylinder 1024 will cause the installation to fail.

To control whether Setup extends the partition on which the temporary files are located, add the following line to the [Unattended] section:

```
ExtendOemPartition = 0 | 1
```

Where:

0

specifies that the partition not be extended.

1

specifies that the partition is to be extended. When the value is **1**, you must set **FileSystem = ConvertNTFS**.

Stopping Setup Before GUI Mode Begins

If **OEMPreinstall = Yes**, the Setup program stops at the end of text mode by default, rather than continuing into GUI mode. This gives you the ability to duplicate disks with the operating system and applications pre-installed. The first time a computer is started with the duplicated disk in place, the system boots into GUI mode. The user can then be prompted for information such as username, computername, and the Workgroup or Domain that the computer is to join.

You can specify whether Setup continues into GUI mode at the end of text mode by adding the following line to the [Unattended] section:

```
NoWaitAfterTextMode = 0 | 1
```

Where:

0

specifies that Setup should stop at the end of text mode.

1

specifies that Setup should automatically reboot into GUI mode at the end of text mode.

Sample Unattend.txt File

```
[Unattended]
    Method = express
    TargetPath = srv1231

[GuiUnattended]
    AdvServerType = SERVERNT
;   AdvServerType = LANMANNT (PDC) | LANSECNT (BDC) | SERVERNT
(STANDALONE)
    TimeZone = "(GMT-08:00) Pacific Time (US & Canada); Tijuana"

[UserData]
    FullName = "John L. Smith"
    OrgName = "Microsoft"
    ComputerName = JohnSm_1
    ProductId = "123456789012345"

[LicenseFilePrintData]
    AutoMode = PerSeat

[Network]
    DetectAdapters = DetectParams
;   InstallAdapters = AdaptersList
    InstallProtocols = ProtocolsList
;   InstallServices = ServicesList

;   JoinWorkgroup = OurWorkGroup

    JoinDomain = NTKWSTA
    CreateComputerAccount = JohnSm, FriedFrogs

; for a PDC install
;   InstallDC = Our_DOMAIN

; for a BDC install
;   InstallDC = Our_DOMAIN
;   CreateComputerAccount = Administrator, FishScale

[DetectParams]
    DetectCount = 1
    LimitTo = IEEPRO, ELNKII

[AdaptersList]
    IEEPRO = IntelEEProParams
    ELNKII = 3ComELIIParams
```

```
[ProtocolsList]
    TC = TCPIPParams
;    NBF = NetBEUIParams
;    NWLNKIPX = NWLinkIPXParams

[ServicesList]
    RAS = RemoteAccessParameters

[IntelEEProParams]
    Interrupt = 5
    IoChannelReady = 4
    Transceiver = 4
    IoAddress = 784

; irq = 5
; i/o = 0x310
; io channel = auto
; transceiver = auto

[3ComELIIParams]
    InterruptNumber = 2
    IOBaseAddress = 512
    Transceiver = 2
    MemoryMapped = 0

;    where Transceiver = 1 - External; 2 - On Board
;    MemoryMapped = 1 - ON; 0 - OFF

; Optional:
;    !AutoNetInterfaceType = 0 | 1 | 2 ...
;    !AutoNetBusNumber = 0 | 1 | ...

[RemoteAccessParameters]
PortSections= MYCOM1
    DialoutProtocols= TCP/IP

[MYCOM1]
    PortName      = COM1
    DeviceType    = Modem
    DeviceName    = "Intel Fax Modem 144e"
    PortUsage     = DialOut
```

BLANK PAGE

IMPORTANT: This text will appear on screen, but will not print on a PostScript printer.

This page should be the last one in this file; it was inserted by running the InsertBlankPage macro.

Do not type any additional text on this page!