

# SECRETS FOR RUNNING WINDOWS ON NETWORKS

WINDOWS AND NETWORKS ARE NATURAL PARTNERS. FIRST OF ALL, BOTH EXPLODED in popularity recently, although both had been hanging around the periphery of the PC business for years. They also share the characteristic of trying to mask extreme system complexity so that users can access great power without having to wade through confusing layers of plumbing. Finally, and most importantly, they complement each other's requirements: Windows needs vast resources and excels at integrating disparate elements; these are what networks provide. Networks cry out for a user environment that allows comprehensible visual metaphors to substitute for hard-to-understand network operations; Windows provides this. So, the two have heightened each other's functionality and increased each other's popularity.

Still, for the user and the network manager, the marriage between Windows and networking is not entirely smooth. Windows is a complicated piece of work, as is any network operating system. Since Windows has to be able to work with all common networks, this complexity increases further, and results in inevitable friction along the borders between Windows and any networking system. When the borders between systems are long and intricate, as with Windows and networks, a weak link somewhere is inevitable.

This chapter delves into how you can make the union of Windows and networks as smooth as possible, how to shore up those weak links and avoid any unexpected obstacles along the path of network integration.

## Setup Tips for Network Administrators

Why is Windows worth all the sweat and toil that it brings network administrators? Two simple reasons: cost and inevitability.

Since Windows is exploding through corporate computing, any wise systems professional will want to deploy it in the most cost-effective way possible. Buying workstation licenses for Windows as a network environment is cheaper than buying a retail copy for each PC. If a network is already in place, using it to distribute Windows makes inescapable economic sense.

Where a network doesn't yet exist, Windows provides the easiest, most robust way to begin networking. In fact, Windows is so popular that implementing a network that required leaving Windows behind would probably spark a user riot. So, the coordinated growth of Windows and networks is an inevitable by-product of the increased use of and interest in desktop systems in corporations today.

What this comes down to, in the end, is a bit of advice to network administrators: Sure, implementing Windows on a network is a bother, but don't stand in the way of progress, get the network licenses you need and get to work. You'll achieve real benefits if you do, and you'll save yourself a wave of user complaints and problems down the line.

## Optimizing Windows on a Network

Making a network Windows installation as smooth and powerful as possible requires something there is precious little of in PCdom: planning. Don't install first and try to make it work second. When it comes to networks, plan, plan, plan. From preinstallation through server tinkering, you should have a solid plan before you take a step. Here, we walk through the process of implementing Windows networks, from the very beginning.

**Decisions to Make before Installation** Options for installing Windows on a network vary greatly. Before you jump into this task, there are a few things you'll need to decide based upon the network's capacity and limitations, as well as your own judgment as to what makes the most sense for your organization. Before you jump into the task, ask yourself the following questions to ensure that you install Windows in the best way possible for your situation:

How much disk space is available on the server and network workstations? If you want to install a separate copy of Windows onto each workstation, you'll need to allow approximately 10Mb of disk space per workstation for it. Can you afford that precious real estate, or would you rather install the majority of the Windows files on the server? The latter requires around 16Mb of space and then uses only a minimal amount of workstation space for individual user files.

Do you want to run Windows Setup from the server or from the actual Windows disks? Whether you install the Windows files on the server or each workstation, you still need to run the Setup program to configure Windows for use on each specific workstation. You can do this as you would for a stand-alone PC, using the Windows disks, or you can run the program from the file server where you've copied the necessary Windows files.

How much say do you want each user to have in setting up Windows on his or her workstation? Do you want to allow the user to choose options, such as printers, fonts, display drivers, and where Windows files are located, or do you want to automate the entire process?

Does Windows out of the box meet your needs? Windows' default configuration may suit your network setup just fine, but in other cases it's just a starting point. You'll install custom applications, add restrictions to groups, and maybe even introduce standard wallpaper with your company logo on it. If you know that you'll be making such changes to the Windows environment of each and every workstation, you can save yourself lots of time by creating a set of instructions for customizing Windows before you install it anywhere.

**Ensure a Problem-Free Network Installation** Once you've made the preceding decisions, you're ready for installation, right? Wrong. Make sure that the Windows Setup on your network is error-free by turning off any network messaging services temporarily that allow messages to pop up on your computer screen. Other TSRs that can automatically appear should also be disabled. If one of these programs happens to pop up during Windows Setup, there's a good chance the installation will fail.

## Using Setup's Command-Line Options

Setup offers a number of command-line choices that can prove helpful for network installations. Use these to tailor each installation to your particular requirements.

**/A Option** The /A switch is for the Administrative Setup option. Use it to copy all the Windows files from the setup disks to the network server. Issuing the command `SETUP /A` does not install a working copy of Windows on the server. It simply expands the files and marks them as read-only (so that they can be used by more than one person at a time), preparing them to be used for installation on workstations. When the files are placed on the server, tiny files with disk names such as `DISK1` are created and placed in the network directory. Don't delete these files. They are needed for the network installation process.

**/N: Network Setup** The Network Setup option, /N, is used from the workstation itself when you want to prepare it to run a shared copy of Windows from the network. Only the few Windows files that contain information about that specific workstation and its user's preferences are copied to the workstation's hard disk. These files include `WINDOWS.INI` and `SYSTEM.INI`; they are stored in a separate Windows directory on the workstation's hard disk or in the user's private network directory.

**/H: Batch Mode Setup** The /H switch for Batch Mode Setup is the option that enables you to set up Windows on many workstations, just the way you want it, with the least amount of work for you and your network users. It works in conjunction with a system settings file that you've predefined. Batch Mode Setup eliminates the need for users to make any choices about the hardware they're using with Windows, and it allows you to install hardware that's not included as a standard Windows option.

Once you've created the system settings file, it is placed in a directory where it can be accessed by users, or it can be copied directly to the workstation itself. With the custom settings file in place, either you or the workstation user can run `SETUP /H` from the workstation.

Even if you aren't setting up Windows on a network, the system settings file that you define can come in handy. Use it to custom-install Windows on stand-alone PCs in your organization. The predefined settings can be called up from a regular setup routine run directly from the Windows disks, rather than from a network file server.

The proper syntax for using the `/H` option to run Setup from a network server is

```
SETUP /H:[drive:\path]filename /N
```

The *filename* points to the system settings file that you've created. If you've placed the system settings file in the same network directory as the Windows files, there's no need to include the drive and path.

For example, you might create a system settings file named `PRINTERS.SHH` that contains the correct settings for the network printers in your organization. The correct Windows setup command would be

```
SETUP /H:PRINTERS.SHH/N
```

## **Creating a System Settings File to Use with Batch Mode Setup**

The system settings file that you will use with Batch Mode Setup can be named anything you'd like, as long as it fits the eight-character limitation that DOS imposes. Regardless of the name, the settings file must have the extension `.SHH` for Windows Setup to recognize it.

A template for a system settings file is included on Windows Disk 1: `SETUP.SHH`. If you have previously used the `SETUP /A` option to copy all the Windows files to a file server, you will find `SETUP.SHH` there. Since this file is a simple text file, you can modify it with any text editor that can save ASCII files.

The system settings file can contain any of the sections listed here:

**[sysinfo]** tells Setup whether to include the System Configuration screen, which allows the user to confirm or change the machine type, display, mouse driver, and network installed.

**[configuration]** sets hardware choices for those components appearing in the System Configuration screen, as well as keyboard type and layout, and the language used.

**[windir]** tells Setup where to install the Windows files for that workstation.

**[userinfo]** tells Setup what to specify as the user name and company name.

**[dontinstall]** tells Windows which of the optional components (README files, accessories, games, screen savers, and wallpaper) you don't want installed on the user's system.

**[options]** tells Windows whether to offer the user the options of setting up applications and starting the Windows tutorial.

**[printers]** tells Windows which printers to install.

**[endinstall]** tells Windows what to do when it is successfully installed, including modifying the `CONFIG.SYS` and `AUTOEXEC.BAT` files on the user's system, returning to DOS, and restarting Windows.

**Preparing to Create a System Settings File** In order to set many of the options in the `.SHH` file, you need to refer to other system settings located in Windows'

SETUP.INF, CONTROL.INF, SYSTEM.INI, and WIN.INI files. You are already familiar with the two .INI files, especially if you've read Chapter 2's line-by-line explanations of how they work. CONTROL.INF and SETUP.INF, however, may be new to you.

These files are used by Windows Setup and contain important information that tells how to proceed given the type of system you have and the choices you specify. Think of SETUP.INF and CONTROL.INF as all of the possible lists of instructions to be used every time Windows is installed. Obviously, looking to these two files for all possible contingencies isn't realistic. While they contain settings for most types of hardware and user preferences, they can't cover everything. But that's actually good news for you because it means you can edit these files yourself to tailor them to the exact needs of your organization.

SETUP.INF and CONTROL.INF are text files, just like WIN.INI and SYSTEM.INI. If you'll be creating a system settings file and you don't know the proper terms to use, it's a good idea to print these two files for reference.

You'll find SETUP.INF on Windows Disk 1. CONTROL.INF and SETUP.INF are created when Windows is installed. If you've used the SETUP /A option to copy the contents of the Windows disks onto the network server, you'll find them there, along with the master files for WIN.INI and SYSTEM.INI, which are WIN.SRC and SYSTEM.SRC.

**[sysinfo]** The [sysinfo] section of the system settings file specifies whether the System Information screen (pictured in Figure 10.1) is displayed during the DOS portion of Windows setup. The screen lists the current settings for the machine type, display type, mouse driver, and network installed. The user either confirms this information or makes the changes necessary to match his or her workstation's system configuration.

Disable this screen if you've already selected the correct settings by way of the other entries in the .SHH file, and you don't want users to think they may need to change this information.

To suppress the screen, add the line `showsysinfo=no`, or leave the section blank. (The default setting is no.) To display this screen, add the line `showsysinfo=yes` beneath the [sysinfo] heading in your system settings file.

**[configuration]** The [configuration] section of the system settings file allows you to tell Windows about the hardware setup for each user's system. You can specify the machine type, display type, mouse driver, network installed, keyboard type and layout, and the language used.

If you only specify some of the available [configuration] entries in the system settings file, the detected or default settings will be used for those items not listed. The following entries are valid for the configuration section:

**machine=** specifies the machine type by using one of the correct profile strings that are listed in the SETUP.INF file's [machine] section. For example, to specify a Hewlett-Packard workstation as the machine type, open the SETUP.INF file in a text editor that can handle such a large file (at over 59K it's too big for Notepad to handle); Write is a convenient choice. Locate the [machine] section in this file. You can do so quickly by using Write's Find command. Now locate the correct profile string from the list, as shown in Figure 10.2. Note that the string is the text that is listed in the left-hand column. The explanation, in this case "Hewlett-Packard: all machines" is in quotation marks. Scan the quoted text to find the proper machine type; then note the profile string to its left. Copy the string, in this case `hewlett_packard`, and paste it back into the system settings file, making the new entry read `machine= hewlett_packard`.

**display=** sets the type of display by using the correct profile string from the [display] section of the SETUP.INF file. You'll need to locate the correct item in the SETUP.INF file (as described in the preceding paragraph). Then, either copy and paste it into your .SHH file (which ensures that you get it exactly right) or note the proper syntax and type it in the .SHH file yourself. For example, to set the display for use with an XGA adapter at 640x480 resolution, you would enter `display=xgalo`.

**mouse=** sets the type of pointing device that will be used on the system. Once again, you'll have to refer to the SETUP.INF file and find this information in the [pointing.device] section. For example, to specify a Mouse Systems serial bus mouse in the system settings file, you would enter mouse=msmouse2.

**network=** sets the kind of network you are using (type and version number) in the system settings file by using the correct string or strings from the [network] section of the SETUP.INF file. For example, to tell Windows that you are using Banyan VINES version 4.1 as your network operating system, you would first locate the string banyan. Now scan down the file to see if there is a section listing entries for the different versions of VINES. In this example, the section is clearly labeled [banyan.versions]. The correct profile string for version 4.1 is listed as xx041000. The version number will follow the first string, separated by a forward slash (/). The complete Banyan VINES entry for the system settings file would be network=banyan/041000.

**keyboard=** sets the type of keyboard being used in the system settings file. Locate the correct entry in the [keyboard.types] section of the SETUP.INF file and enter it in the .SHH file. For example, the entry specifying that the workstation has an Enhanced 101 keyboard would be keyboard=t4s0enha.

**language=** sets the language that will be used by specifying the correct entry from the [language] section of the SETUP.INF file. For example, the entry in the system settings file to set the language to Spanish would read language=esp.

**kblayout=** sets the keyboard layout for the workstation by specifying the correct entry from the [keyboard.tables] section of the SETUP.INF file. For example, the line to specify Spanish as the language would read kblayout=spadll.

For each of the individual entries used in the preceding examples, the complete listing for the [configuration] section of the system settings file would look like this:

```
[configuration]
machine=hewlett_packard
display=xgalo
mouse=msmouse2
network=banyan/041000
keyboard=t4s0enha
language=esp
kblayout=spadll
```

These entries tell Windows Setup that this is a Hewlett-Packard workstation with an XGA display adapter, Mouse Systems bus mouse, Banyan VINES 4.1, 101 Enhanced keyboard with Spanish layout, and that the user will be using the Spanish language.

**[windir]** The [windir] section sets the directory where the Windows files for this user's installation will be stored. If you don't specify a directory in this section, or if your entry is invalid, the user will be prompted for the Windows directory during setup. The SETUP.INF file also contains an entry called defdir= for setting the Windows directory in the [data] section, but the setting you specify here in the system settings file will override it. For example, the line in the [windir] section to install Windows in the WIN31 directory on drive C would be C:\WIN31.

**[userinfo]** Set the user name and company name that Windows prompts for during setup by entering that information in the [userinfo] section. Each entry can be up to 30 characters long. If there are spaces in the name-for example, separating the first and last name-the name must be in quotation marks.

If Windows is set up as a shared copy running on a network, you won't need to specify a user name because you already did so in using the `SETUP /A` option to copy the files to the network server. For individual workstations that are setting up Windows, a user name is required; you'll be prompted during setup if one isn't specified. In both cases, the company name is optional. For example, to tell Windows that Jane Doe, an employee of Allied, will be using this workstation, you would use this setting:

"Jane Doe"  
Allied

**[dontinstall]** Use the `[dontinstall]` section of the system settings file to tell Windows Setup which optional components you'd like to forgo. Whether you can't spare the disk space, you don't want the distraction, or you've got another utility that does the job, Setup can install only some of the items or none at all. However, within a particular optional category—for example, README files—you have to install all the items or none of them. For instance, you can't just install the `NETWORKS.WRI` file; if you want any one file, you're stuck with them all.

If there isn't a `[dontinstall]` entry in the system settings file, Windows will install all of the README files, accessories, games, screen savers, and wallpaper. The following entries are valid for the `[dontinstall]` section:

**readmes** tells Windows not to install these README files: `NETWORKS.WRI`, `PRINTERS.WRI`, `README.WRI`, `SYSINI.WRI`, and `WININI.WRI`.

**accessories** tells Windows not to install these accessories and their associated help files: Calculator, Calendar, Cardfile, Character Map, Clipboard Viewer, Clock, Dr. Watson, Media Player, Notepad, Object Packager, Paintbrush, Recorder, Sound Recorder, Terminal, and the Windows Tutorial.

**games** tells Windows not to install Solitaire and Minesweeper and their associated help files.

**screensavers** tells Windows not to install its included screen savers: blank (default), Flying Windows, Marquee, Mystify, and Starfield Simulation.

**bitmaps** tells Windows not to install the following bitmap files (which can be used as desktop wallpaper): 256 colors, Arcade, Arches, Argyle, Cars, Castle, Chintz, Egypt, Flock, Honey, Leaves, Marble, Red Brick, Rivets, Squares, Tartan, Thatch, Windows Logo, and Zigzag.

**[options]** The `[options]` section lets you tell Windows if you want users to be able to set up applications during Windows Setup, if you want to let them pick what's installed, and whether to start the Windows Tutorial after a successful installation. The following entries are valid for the `[options]` section of the system settings file. Note that if no entry appears in the `[options]` section and you are using a system settings file to automate setup, Windows will not provide users with any of these options.

**setupapps** tells Windows to let users set up the applications that they choose during setup. If this entry is found in the system settings file, users have the option of telling Windows which applications, including both DOS and Windows apps, they want to set up on their system for use with Windows.

**autosetupapps** tells Windows to automatically set up all of the applications that it finds on the user's hard disk, without giving the user a choice in what he or she wants installed.

**tutorial** tells Windows to automatically start the Windows Tutorial upon completion of setup.

**[printers]** The [printers] section in a system settings file lets you specify which printers to install during setup. As in the hardware entries discussed previously, you need to locate the string used to identify your printer. This time, however, you need to refer to the CONTROL.INF file, rather than the SETUP.INF file. Look for the [io.device] section and locate the printers that users will have access to from Windows. As with the name and company entries in the [userinfo] section, the printer name will be enclosed in quotation marks if it contains any blank spaces.

You also need to tell Windows which ports the printer will be using. You can't just specify the correct port, as in LPT1; you must use the same syntax as WIN.INI's [port] section, which adds a colon to the end of the port name. The valid WIN.INI entries include LPT1:, LPT2:, COM1:, and COM2:.

The printer name and its port will be separated by a comma in the system settings file. For example, to tell Windows that your users will be printing to a HP LaserJet II on LPT1 and an Epson LQ-1170 on COM2, the entries in [printers] section would be these:

```
"HP LaserJet Series II", LPT1:  
"Epson LQ-1170 ESC/P 2", COM2:
```

If you omit this section from your system settings file, no printers will be installed.

**[endinstall]** The [endinstall] section tells Windows what should happen to the system once Windows is installed. Entries in this section tell Windows whether to modify the user's CONFIG.SYS and AUTOEXEC.BAT files or save the proposed changes in a separate file and whether to exit to DOS, restart Windows, or reboot the system. The following are valid entries for the [endinstall] section:

**configfiles=** tells Windows Setup to make the necessary changes to the AUTOEXEC.BAT and CONFIG.SYS files, such as including Windows in the PATH statement or adding a command to run SMARTdrive, if you set the value of the statement to read configfiles=modify. In this case, the old AUTOEXEC.BAT and CONFIG.SYS files are renamed AUTOEXEC.OLD and CONFIG.OLD, respectively. If files named AUTOEXEC.OLD and CONFIG.OLD are already present, these older files will be renamed with the file extension .000, as in AUTOEXEC.000. To prevent Setup from changing the CONFIG.SYS and AUTOEXEC.BAT, set this entry to configfiles=save. The original files will remain unchanged, but the proposed changes will be incorporated into new copies of the CONFIG.SYS and AUTOEXEC.BAT files named CONFIG.WIN and AUTOEXEC.WIN, respectively. This enables you to review the changes later and make them yourself.

**endopt=** tells Windows what action the system should take once installation is complete. To return to DOS, make the line read endopt=exit. To restart Windows after Setup, make the line read endopt=restart. To reboot the system after Setup, change the line to read endopt=reboot. However, if the reboot option is specified in the system settings file, but Windows was set up to run from a shared directory on the server, the machine will be returned to DOS instead of rebooting.

## **Automating Setup with the SETUP.INF File**

The Setup Information file (SETUP.INF), like WIN.INI and SYSTEM.INI, is composed of sections that are identified by brackets, for example [setup]. You saw that these files are useful references for creating system settings files (.SHH) to automate the setup process on a network. Now you'll see how you can use these files directly, to customize Setup on your network: for example, to add an installation choice for a printer driver that doesn't ship with Windows; to remove unnecessary choices from Setup options, such as mouse drivers or display drivers that your organization doesn't use; or to add a custom application to the list of those that Windows recognizes and will automatically set up on the workstation during setup.

As always, here's a word of caution before tinkering with such powerful Windows files: Remember to back up the SETUP.INF file before you attempt to customize it. In case you do

forget to save the previous version of SETUP.INF, keep the Windows disks handy; SETUP.INF is located on Disk 1 in uncompressed format.

Before you attempt any additions or changes to SETUP.INF, it's a good idea to familiarize yourself with its contents. The file is commented throughout, so you'll be able to figure out what each section does and how the corresponding settings work. Here's a brief look at the file's contents:

**[setup]** tells Windows where to find on-line help for the Setup program.

**[run]** tells Windows which programs, if any, should be run after Setup is complete.

**[dialog]** tells Windows what text should appear in the Setup dialog boxes.

**[data]** provides the default settings and requirements for installing Windows, such as minimum disk space and memory required, and the default directory, keyboard, and language used.

**[winexec]** tells the DOS part of Windows Setup which kernel file to use for the Windows portion of Setup.

**[disks]** tells Windows which disks to prompt the user for during installation.

**[oemdisks]** tells Windows what additional disks to ask for during installation.

**[user]** tells Windows where the temp file containing the user's name and company is located.

**[windows]** tells the DOS portion of Setup which files to copy into the WINDOWS directory.

**[windows.system]** contains information about which files to copy into the SYSTEM subdirectory.

**[windows.system.386]** contains both information about which files should be copied to the SYSTEM subdirectory for 386 PCs and additional instructions for use if the machine uses the Qualitas expanded memory managers Bluemax or 386Max.

**[shell]** tells Windows what to use as the program shell. The default is Program Manager.

**[display]** contains information for display drivers used by Windows.

**[.3gr]** contains a listing of the font files used with specific 386 grabbers.

**[keyboard.drivers]** contains entries that match short names for the keyboard driver filenames to the actual filenames.

**[keyboard.types]** contains entries that define short names to be used in place of the keyboard descriptions that appear in Setup dialog boxes.

**[keyboard.tables]** contains entries that define short names to be used in place of the DLL filenames, and tells Windows where they are located.

**[codepages]** defines code pages used for international support.

**[pointing.device]** lists drivers for pointing devices supported by Windows.

**[dos.mouse.driver]** tells Windows which DOS mouse driver to use with which Windows mouse driver.

**[Imouse]** contains settings for the Logitech DOS mouse drivers.



**[network]** lists networks supported by Windows and any corresponding files, such as help files, as well as the names of sections that need to be added or modified to .INI files.

**[Network\_Version]** provides Windows with information for specific versions of a network operating system.

**[Network\_Specific]** lists entries for specific networks describing changes to be made to the .INI files.

**[sysfonts]** lists system font files used by Windows.

**[fixedfonts]** lists fixed-pitch font files used by Windows.

**[oemfonts]** lists OEM system font files used by Windows.

**[win.copy]** lists files that Windows Setup needs to copy to the WINDOWS directory or its SYSTEM subdirectory for 286 PCs.

**[win.copy.net]** lists files that Windows Setup needs to copy to the WINDOWS directory or its SYSTEM subdirectory for network workstations not including 386 or higher PCs.

**[win.copy.net.386]** lists files that Windows Setup needs to copy to the WINDOWS directory or its SYSTEM subdirectory for 386 or higher PCs on a network.

**[win.copy.386]** lists files that Windows Setup needs to copy to the WINDOWS directory or its SYSTEM subdirectory for unnetworked 386 or higher PCs.

**[DelFiles]** lists files to delete when upgrading to Windows 3.1.

**[RenFiles]** lists files to rename when upgrading to Windows 3.1.

**[win\_copyFiles]** lists files that need to be copied when you choose to install specific components, games, screen savers, and accessories.

**[new.groups]** tells Setup what Program Manager group changes to make for an upgrade from version 3.0 to 3.1.

**[progman.groups]** tells Setup which Program Manager groups to create for a new installation of Windows 3.1.

**[group#]** tells Windows what items should be in each Program Manager group.

**[fonts]** lists vector and raster screen font files used by Windows.

**[tffonts]** lists TrueType font files used by Windows.

**[compatibility]** lists the filenames of drivers whose references Windows will remove from the CONFIG.SYS because they may cause problems with Windows.

**[incompTSR1]** lists the filenames of TSRs and drivers that have been known to cause problems during Windows Setup.

**[incompTSR2]** lists the filenames of TSRs and drivers that have been known to cause problems when running with Windows.

**[block\_devices]** lists block devices known to cause problems with Windows.

**[installable.drivers]** contains information for multimedia drivers, such as those for sound boards, used by Windows.

**[translate]** lists the Windows 3.0 OEMSETUP.INF filenames and their equivalents in Windows 3.1.

**[update.files]** lists filenames of drivers that should be updated to the version on the Windows 3.1 disks.

**[ini.upd.patches]** lists the temporary filenames of .INI file sections that are updated when upgrading from Windows 3.0 to 3.1.

**[blowaway]** tells Windows that this is the end of Setup information that it needs in SETUP.INF.

**[ini.upd.31]** lists the lines of .INI files that will be updated in Windows 3.1.

**[system]** contains entries that define short names to be used by Windows in place of actual driver names.

**[machine]** contains information for installing the appropriate files for each type of system.

**[special\_adapter]** lists information for using special adapters with Windows.

**[ebios]** lists the filenames needed for extended BIOS support on different systems.

**[language]** lists the language DLLs used by Windows for systems that need language support other than English.

**Add Custom Choices to System Information Screens** If the workstations on your network use a device that isn't supported by Windows, such as a trackball or off-brand mouse, you can add a menu choice for it to the Setup routine. Of course, you will need to get the driver that allows the device to work with Windows from the manufacturer.

First, copy the driver into the same shared directory as the rest of the Windows files. Next, open the SETUP.INF file in a text editor and locate the [pointing.device] section. You'll see a comment that shows the appropriate syntax for the section:

```
;profile = mouse driver, mouse description, VMD, optional work section
```

Below that you'll see the first entry:

```
hpmouse = 2:hpmouse.drv, "HP Mouse (HP-HIL)", x:*vmd
```

The semicolon identifies that what comes after it is a remark and not file information. Now, here's what the rest of the entry means:

*profile* is a text string that's used in place of the longer filename to identify the driver. In the example, it's hpmouse.

*mouse driver* is the actual filename, which ends in .DRV and is preceded by the number of the disk that contains it. Because the custom driver will not be found on a disk, just use 1; Windows will ignore the number because Setup is being run from the file server. In the example, the setting is 2:hpmouse.drv.

*mouse description* is the text that will appear as the choice in Windows Setup. In the example, it's "HP Mouse (HP-HIL)".

VMD stands for virtual mouse device and specifies the driver that allows the pointing device to be used with DOS applications in 386 Enhanced mode. This file ends with the extension .386. Many pointing devices use the VMD that's part of WIN386.EXE and don't need a separate file. As with the mouse driver filename, the disk number precedes the entry if it is a separate file from WIN386.EXE. In the example, this setting is x:\*vmd. The x means that you don't need to specify the disk number where the file is located.

*optional work section* refers to any additional information that's needed by Windows to use the driver file. In the preceding example, no additional information is required.

Add a custom choice by creating an entry for your pointing device following the syntax just described. By editing other sections of the SETUP.INF file, you can add other configuration choices to Windows Setup. For display adapters, edit the [display] section, for keyboard drivers edit the [keyboard.drivers] section, and for network drivers edit the [network] section.

**Streamlining User Options in Setup with SETUP.INF and CONTROL.INF** While editing the SETUP.INF lets you add custom choices to hardware menus, you can also use it to remove menu items so that users aren't overwhelmed by the number of choices. SETUP.INF lets you remove choices for the type of system, display, keyboard, network, and pointing device. Similarly, CONTROL.INF lets you remove menu options for any printers that you don't want.

If you've created a system settings file to completely automate Windows installation on your network, the myriad menu choices aren't a problem because the user never sees them. But if each user will be going through the Windows setup process on his or her workstation, you may want to make things easier by leaving only those choices that correspond to the available printers. That way users won't have to wonder if they're selecting the correct driver. For example, is it the HP LaserJet Plus, the HP LaserJet Series II, or the HP LaserJet IID?

To remove the choices you don't want, go to the corresponding section in the .INF file—for example, [machine]—and delete the choices you don't want to appear in Setup menus. For instance, to remove printer choices from the Setup menu and prevent them from appearing in the Control Panel printer list, follow these steps: Open CONTROL.INF in a text editor and locate the [io.device] section. You'll see a list like that in Figure 10.3. Then delete lines for any printers that you know you will never use or don't plan to add to the network. For example, you may decide to delete the lines for plotters, such as these for some of the Hewlett-Packard plotters. (Note that these lines are highlighted in the figure.)

```
6:HPPLOT.DRV,"HP 7470A [HP Plotter]","CONTINUOUSSCALING"  
6:HPPLOT.DRV,"HP 7475A [HP Plotter]","CONTINUOUSSCALING"  
6:HPPLOT.DRV,"HP 7550A [HP Plotter]","CONTINUOUSSCALING"  
6:HPPLOT.DRV,"HP 7580A [HP Plotter]","CONTINUOUSSCALING"  
6:HPPLOT.DRV,"HP 7580B [HP Plotter]","CONTINUOUSSCALING"  
6:HPPLOT.DRV,"HP 7585A [HP Plotter]","CONTINUOUSSCALING"  
6:HPPLOT.DRV,"HP 7585B [HP Plotter]","CONTINUOUSSCALING"  
6:HPPLOT.DRV,"HP 7586B [HP Plotter]","CONTINUOUSSCALING"  
6:HPPLOT.DRV,"HP ColorPro [HP Plotter]","CONTINUOUSSCALING"
```

Although deleting extraneous entries helps make Windows setup a more straightforward process for users, avoid deleting entries for any device that you may possibly want to add to the network one day.

As in the preceding tip for adding setup choices, edit the [display] section to remove display adapter choices, the [keyboard.drivers] section for keyboard choices, and the [network] section for network choices.

## **Copy Additional Windows Files to Workstations with**

**SETUP.INF** You can edit the SETUP.INF file so that you can copy additional Windows files to each workstation's directory. Perhaps you'd like to make sure that the Microsoft Diagnostics Utility is installed locally on all of the workstations so that they'll have access to it even when the network drive can't be accessed. (See Chapter 3, "Find Out about the Memory You Have in Your System" for more details on this handy program.)

To add MSD.EXE to each workstation when Windows is installed, you need to add an entry to the section that controls the files copied for your specific type of PC (networked or unnetworked, 286 or 386 and above).

Open SETUP.INF in a text editor and locate the first section pertaining to copying files, [win.copy]. Notice that the remark beneath it tells you that these entries are for 286 systems only. Below [win.copy] are sections called [win.copy.net], [win.copy.net.win386], and [win.copy.win386]. These are for networked 286 systems, networked 386 and higher systems, and stand-alone 386 or higher systems, respectively.

If your network was composed of all 386 and 486 systems, you'd make changes to the [win.copy.net.win386] section. Entries to the various copy-files sections can either be references to entire lists of files that are contained in their own sections or simply to an individual file. Regardless of the kind of entry, it must point to the user's Windows directory as the destination for the copied files. 0: is the number Windows uses to refer to the location of the system's WINDOWS directory. References to 1: through 7: stand for the original Windows disks.

There are two ways to add the single file that executes MSD.EXE from the network server to the workstation's WINDOWS directory. One way is to add the file to the bottom of the list under the [net] section, which, as you can see from the existing entries in the [win.copy.net.win386] section, is automatically copied to such systems. The new [net] section might look like this:

```
[net]
6:CONTROL.SRC, "Windows User Files"
5:WINVER
5:MSD.EXE
```

The other choice is to add the file to the [win.copy.net.win386] section itself. This entry would read

```
5:MSD.EXE, 0:
```

Note that the 5: that stands for setup Disk 5 is ignored by Setup because the file is actually located in the shared directory on the server.

If you choose to install existing groups of files to the workstation's disk, such as those in the [win.devices.win386] section (HIMEM.SYS, EMM386.EXE, SMARTDRV.EXE, and RAMDRIVE.EXE), you can do so by pointing to this group. For example, you can add the line **#win.devices.win386, 0 :** The # sign tells Windows that you are pointing to a group of files that is defined in the SETUP.INF file.

**Install Custom Apps with SETUP.INF** Take the previous tip one step further by having Windows copy non-Windows files to workstations during Setup. For instance, you can tell Windows to copy custom Windows applications, such as a database management program, or commercial Windows applications, such as a screen saver or utilities program.

The files you want to copy may be either on a floppy disk or in the same directory as the Windows files on the network server. If the files are on a floppy disk, you'll need to assign a disk number to it. Remember that the Windows 3.1 disks are numbered 1 through 7.

For example, to copy a set of custom Windows utilities called RCOUTILS (and made up of the four files, RPHONE.EXE, RINFO.EXE, RUTIL.HLP, and RUTILS.DLL), which are contained on a single floppy disk, you would do the following:

1.First assign this disk the number 8, as shown in Figure 10.4. Locate the [disks] section in SETUP.INF, and add a line assigning the number, like this:

```
8 =. ,"R. Company Utilities",disk8
```

2.The disk tag, disk8, corresponds to the disk's volume label, assigned in DOS when you format a floppy disk. If the R. Company Utilities disk doesn't have a volume label, you must assign one. Windows File Manager allows you to add a label to a disk that's already formatted by selecting the Label Disk choice from the Disk menu.

3.Next, add a reference to Disk 8 in the appropriate [win.copy] section for the workstations on your network. Since our network has all 386 and 486 workstations, we would use the [win.copy.net.win386] section. The addition for the utilities would look like these entries:

```
[win.copy.net.win386]
; copy this section for network setup on 386 machines
#net,    0:
8:rphone.exe, 0:
8:rinfo.exe, 0:
8:rutil.hlp, 0:
8:rutil.dll, 0:
```

If the utilities were located in the shared Windows directory on the network, in the preceding entry 8: would be replaced the with any number from 1: through 7:(since Setup ignores the number of the Windows disk if the files are located on the server).

If you want to copy many files to network workstations, or if your workstations include 286s as well as 386s and higher, you'll want to use a different method for copying files. As in the previous tip, where we placed a reference to a copy-files section ([win.devices.win386]) also defined in SETUP.INF, we can define a new copy-files section that contains our utilities and then simply reference this in the [win.copy] sections for our workstations.

For example, let's create a copy-files section named [rutils]. This new section should be placed near similar sections in the SETUP.INF file, such as [win.devices.win386], [win.other], and [win.shell]. The new section defining R. Company's utility collection would look like this:

```
[rutils]
8:rinfo.exe, "RInfo Utility", rinfo
8:rphone.exe, "RPhone Utility", rphone
8:rutil.hlp, "RHelp Utility"
```

The last file, RUTIL.DLL, would be added to the [win.dependents] copy-file section, which is used to copy DLLs to the corresponding programs (identified here by the profile strings rinfo and rphone) that use them. The new entry would look like these lines:

```
[win.dependents]
pbrush = 4:PBRUSH.DLL
recorder = 3:RECORDER.DLL
wintutor = 4:WINTUTOR.DAT
rinfo = 8:RUTIL.DLL
rphone = 8:RUTIL.DLL
```

Finally, you need to add a reference to the [rutils] section to the [win.copy.net.win386] section. It would look like this:

```
#net,    0:
```

#rutils, 0:rutils

The 0:rutils entry tells Windows to copy the files to the WINDOWS subdirectory RUTILS.

**Create Custom Program Groups** You can also customize SETUP.INF to create Program Manager groups that contain only the items that you want. You can change the contents of existing groups or create entirely new ones.

For new installations of Windows 3.1, you'll use the [progman.groups] section in the SETUP.INF file. If, however, you are upgrading from 3.0, you'll refer to the [new.groups] section. Remember, if you're using Windows Write it's quickest to use the Find command, as shown in Figure 10.5.

You'll notice that the last line of the [progman.groups] section identifies the StartUp group. To add a new group called Sales, you'd add a new entry like this:

```
group6=SalesTools
```

Now that the group has been established, you must create a section called [group6] that defines its contents. Scroll down in SETUP.INF file until you locate the place where other Program Manager groups are defined, with headings like [group3] and [group4]. Add the [group6] section, following the format of the other group listings. For example, you might enter the following:

```
[group6]
"Calendar", CALENDAR.EXE,,, calendar
"Calculator", CALC.EXE,,, calc
"Excel", F:\EXCEL\EXCEL.EXE,,, excel
```

Because the Windows application Excel is not located in the WINDOWS directory, you must specify its path.

Alternatively, you can edit the existing group definitions by deleting some programs from one group while adding them to another. For example, you could move the Clipboard Viewer from Group 3 (Main) to Group 4 (Accessories) by cutting the line from [group3] and pasting it into [group4].

On the other hand, you might want to get rid of some Program Manager group items altogether. For example, if your users will not be taking advantage of any of Windows' multimedia capabilities, you can create a less cluttered desktop by removing icons for applications such as the Sound Recorder and Media Player. If you decide to use these capabilities later, you can simply add the icons. Unlike the [dontinstall] section of the system settings file discussed earlier, the files for these applications are installed on your system; they just aren't visible in the default Program Manager groups.

**Create a Custom Working Environment** In the preceding section, you saw how to create Program Manager groups that contain exactly the items you need. You can customize and standardize the look of the Windows desktop by adding your own bitmaps to be used as wallpaper. For example, you might want to make your company logo the required wallpaper on your network. To do this, locate the [win.bmps] section of the SETUP.INF file; you'll see listings for the various bitmaps that are included with Windows. Then add an entry for the bitmap, in this case COLOGO.BMP, like this:

```
5:COLOGO.BMP, "Our Logo Wallpaper",898
```

The number 898 is the size of the COLOGO.BMP file, and as mentioned before, 5: refers to the Windows disk that contains the file. Since we've copied the COLOGO.BMP to the shared network directory, 5: just serves as a placeholder here and is ignored by Windows Setup.

While you're in the [win.bmps] section, delete any bitmaps that you don't want users to install as wallpaper on their desktops. You can also remove entries from the other section, [win.scrs], if you plan to use an alternate screen saver, such as a commercial Windows product like Intermission or After Dark.

**Replace Program Manager Before Windows Is Installed** If you plan on installing a third-party program such as Norton Desktop in place of the Windows shell Program Manager, you can make this switch while Windows is installed.

To do so, locate the [shell] section of the SETUP.INF file. It will look like this:

```
[shell]
progman.exe, "Program Manager"
```

Then replace it with the following:

```
[shell]
ndw.exe, "Norton Desktop"
```

Each network workstation will be started with Norton Desktop as the shell.

**Force Network Setup** If you want to make sure that network users set up a shared copy of Windows from the files located on the server, you can edit SETUP.INF to force this option. If a user forgets to use the /N switch (SETUP /N) for installing a shared copy of Windows on the workstation, Setup will automatically do so instead of installing a local copy of Windows.

To ensure a network setup, make sure that the line netsetup=true is in the [data] section of the SETUP.INF file. The default is netsetup=false.

## **Automating Setup with the APPS.INF and CONTROL.SRC Files**

Another .INF file you'll want to be familiar with is APPS.INF. This file contains information for installing applications with Windows. You can edit this file to install PIF information for your own DOS applications during Windows Setup on a network. If you're unclear about the contents of .PIF files, refer to Chapter 4 for a quick refresher before proceeding.

CONTROL.SRC is the master file for the Windows Control Panel's initialization file (CONTROL.INI). In Chapter 5's "Tips for Control Panel Control," we looked briefly at how you can edit CONTROL.INI once Windows is installed, to customize the Windows desktop. Here you'll see how to make these changes before Windows is set up.

**APPS.INF Contents** Like the CONTROL.INF file discussed previously, the APPS.INF file is in the WINDOWS\SYSTEM subdirectory. And like the other .INF files we've discussed, it is made up of sections defined by bracketed entries. You can use a text editor to make additions and changes to APPS.INF to provide for a custom Windows installation. You'll be able to add PIF information for DOS applications that aren't automatically recognized by Windows Setup. And if you will be using DOS applications that have common executable filenames, such as EDIT.EXE, you can make sure that APPS.INF recognizes them and configures them correctly.

APPS.INF is made up of five main sections:

**[dialog]** tells Windows what text should appear in the Setup Applications dialog box.

**[base\_PIFs]** contains instructions for creating a batch file that creates the \_DEFAULT.PIF file and settings for COMMAND.COM.

**[enha\_dosprompt]** contains memory requirement information for running the DOS prompt in Enhanced mode.

**[dontfind]** tells Windows which applications to ignore during Setup.

**[pif]** contains listings of PIF information for DOS applications that Windows will recognize and set up during installation. The information includes the executable filename, the .PIF filename, the window title, the startup directory, the close-window-on-exit flag, the icon filename, the icon number, a reference to PIF settings for Standard mode section, a reference to the section that defines applications with the same .EXE filename, and a reference to the section that contains optimized PIF settings.

**Automatically Install DOS Programs with APPS.INF** To add PIF information for DOS applications that will be used on your network, you need to add entries to the [pif] section of APPS.INF. If you are uncertain whether a DOS application is already listed in the APPS.INF file, scan the existing entries in the [pif] section. The applications are listed in alphabetical order according to executable filename. The syntax for entries in this section is as follows:

*exe file = PIF name, window title, startup directory, close-window-on-exit flag, icon filename, icon number, reference to standard PIF settings section, reference to enhanced PIF settings section, reference to ambiguous .EXE files section, reference to optimized PIFs section*

As a specific example, let's look at the first entry in the section:

```
123.COM = 123,"Lotus 1-2-3",,cwe,,3,std_gra_256,enha_123c
```

The following lists what each setting means. (Note that the first four settings refer to the first four text boxes that are defined for an application in the PIF Editor, as shown in Figure 10.6.)

**exe file** represents the name of the application's executable file. In the example it's 123.COM.

**PIF name** tells Windows what name to give to the program information file that contains the settings for running the application with Windows, in this case Lotus 1-2-3. In the example the PIF will be called 123.PIF.

**window title** is the title that will appear when the DOS application is running in a window, as pictured in Figure 10.7. In the example, the window will be entitled "Lotus 1-2-3."

**startup directory** is the working directory for the application. In the preceding example, no directory is specified (note the two consecutive commas). This means that Lotus 1-2-3 will start in the directory that Windows determines, usually the directory where the application's executable file is located. If you want to make sure that users save their data files in a specific directory that's different from the program directory, note it here.

**close-window-on-exit flag** tells Windows whether to close the application's window when the user exits from the application. In the example, the setting is cwe (close window on exit). If cwe is not specified, the application won't be closed on exit.

**icon filename** stands for the filename that contains the icon to be used for this application. In the example, the setting is blank, which means that Windows will use the default file PROGMAN.EXE as the icon source. Some of the icon choices for Lotus 1-2-3 are shown in Figure 10.8.

**icon number** is the number of the icon to be used. In the example, the setting is 3, which means that the third icon in the PROGMAN.EXE file will be used (as shown in Figure 10.8).

**reference to standard PIF settings section** tells Windows the name of the section in APPS.INF that contains PIF settings for running the application in Standard mode. In the example, the



section is titled [std\_gra\_256]. Scroll down through the APPS.INF to find this section. You'll find a section that looks like this:

```
[std_gra_256]
minconvmem = 256
videomode = gra ; Graphics mode app
```

This section tells Windows that the minimum conventional memory required by Lotus 1-2-3 is 256K (which corresponds to the Memory Requirements section of the PIF Editor), and the video mode setting to be used is Graphics/Multiple Text (which corresponds to the Video Mode section of the PIF Editor). See Figure 10.9 to locate these sections in the PIF Editor.

**NOTE** Settings aren't specified for the rest of the PIF Editor sections, such as XMS Memory and Directly Modifies, because Windows uses a section of the APPS.INF called [std\_dflt] as the default PIF settings. Locate this section and review the settings, comparing them to the PIF Editor settings. If you don't need to make any changes to the defaults for your application, leave the reference to a standard PIF settings section blank. You'll notice that other options for the settings are shown as comments in the section to help you determine if you need to make any changes.

The [std\_dflt] section looks like this:

```
[std_dflt]
; default is text mode app which does not directly modify COM ports
;; Other Possible options are given for reference
; ( (or) means entry corresponds to radio button group)
minconvmem = 128
videomode = txt ; (or) gra
xmsmem = 0,0 ; ##,, ## (min, max)
checkboxes = ; 1,c2,c3,c4,kbd,nse,pps,ata,aes,ces,psc,aps,nss
```

**reference to enhanced PIF settings section.** As with the default section for running DOS applications in Standard mode, APPS.INF contains the section [enha\_dflt], which contains default settings for running in Enhanced mode. In the preceding example, with Lotus 1-2-3, an addition section specifically for Lotus 1-2-3 is required: [enha\_123c]. If you don't specify an enhanced PIF settings section, Windows will refer to the [enha\_dflt] section, which contains the following settings:

```
[enha_dflt]
; default is as follows
;
;
; Other possible options are given for reference
; ( (or) means entry corresponds to radio button group )
;
convmem = 128,640 ; ##,## (Required, Limit)
emsmem = 0,1024 ; ##,## (Required, Limit)
xmsmem = 0,1024 ; ##,## (Required, Limit)
dispusage = fs ; (or) win
execflags = ; bgd, exc
multaskopt = 50,100 ; ##,## (Bgd Pri, Fgd Pri)
procmemflags = dit,hma ; eml,xml,lam
dispoptvideo = txt ; (or) lgr,hgr
dispoptports = hgr ; txt,lgr
dispflags = emt ; rvm
otheroptions = afp ; cwa,ata,aes,ces,psc,aps,asp,aen
```

**reference to ambiguous .EXE files section** tells Windows the name of the section that lists other applications with the same .EXE filename. In the Lotus 1-2-3 example, there is no reference. This means that there aren't any known programs that have an executable file with the name 123.COM.

**reference to optimized PIFs section** points to another section defined in the APPS.INF file, which contains settings for running the application with specific settings different from those defined in the [enhanced] or [standard] sections. In our example, no additional section is required. You might define an optional section when you want to create more than one PIF for a single application. For information on why you'd want to have multiple PIFs, see Chapter 4's "Make Two PIFs for the Same Application."

Now that you're familiar with the ins and outs of the APPS.INF file's [pif] section, you can add entries for custom DOS applications on your network.

**Automatically Ignore the DOS Apps You Choose at Setup** Windows uses the [dontfind] section of the APPS.INF to avoid installing duplicate copies of Windows applications—for example, Write, Paintbrush, and Calculator. The section consists of a listing of executable filenames that Windows should overlook when it's searching the hard disk for applications to install.

Put this section to work for you by adding the names of existing DOS applications on the network that you don't want to be set up for use with Windows. The section is organized alphabetically, as shown here:

```
[dontfind]
apm.exe
calc.exe
calendar.exe
cardfile.exe
charmap.exe
clipbrd.exe
clock.exe
```

Simply add the executable filenames of other DOS applications in the appropriate place.

**Exclude Sections of Control Panel by Editing CONTROL.SRC** CONTROL.INI, the Control Panel initialization file, contains settings for the Windows Control Panel, including those governing which of its modules (Colors, Desktop, and so on) are loaded. In Chapter 5 we showed how to edit this file to keep other users from being able to make any changes to a system by denying them access to specific sections of the Control Panel.

While CONTROL.INI isn't created until after Windows is installed, you can apply these same tricks before Windows is set up on your network by going right to the source: CONTROL.SRC, the master file for CONTROL.INI. If you've copied the Windows files to the network server you'll find CONTROL.SRC there.

You can add a section to the CONTROL.SRC file called [Don't Load]. Below this heading, list the modules of the Control Panel that you don't want users to have access to. For example, if you don't want users to waste time changing the look of a standard desktop, or if you don't want them to be able to change printers, you could exclude the Desktop and Printers sections from loading into Control Panel. The new section of CONTROL.SRC would look like this:

```
[Don't Load]
Desktop=1
Printers=1
```

The setting of 1 (0 can be used in its place) tells Windows not to load these sections of the Control Panel. Their icons will not be visible in the Control Panel, as shown in Figure 10.10. To grant access to these Control Panel sections later, you'll have to remove the [Don't Load] section from CONTROL.INI.

## **Windows Maintenance on a Network**

The preceding techniques for gaining the most control from a Windows network installation aren't just useful for Windows Setup. Use them to make system changes, such as hardware and software upgrades, a one-step process. We'll also provide new ways, such as editing the PROGMAN.INI file, to easily maintain Windows on a network.

**Quickly Setting Up Windows on Multiple Servers** If your business has more than one network server, you may find yourself installing the Windows files to each of these. Save yourself the trouble of shuffling through the Windows disks by copying the files from the original server to another shared server on the network. To do so, type **setup /a** from the directory that contains the Windows files on the original server. You can then specify a new directory on the network to install Windows into. The files will be copied to this location and marked as read-only. You'll also be prompted for a company name and group name, allowing you to make these entries distinct from those used in the Windows directory on the other server. Note that using the DOS COPY command will not yield the same results.

**Create Shared Program Groups** You can create a Program Manager group which is accessible to all Windows users on your network. To do so, create the desired group in Program Manager, adding all of the programs that you want it to contain. For example, say you've created a group called Orders, with the group filename, ORDERS.GRP. Copy the file ORDERS.GRP to the shared network directory that contains the Windows files, marking it as read-only.

Now each workstation user will need to select the File menu from the Program Manager and specify that they want to create a New Program group. In the Program Group Properties box, shown in Figure 10.11, each user need only specify the path and .GRP filename that you've already created, in this case Y:\WINDOWS\SALESTEA.GRP. The group will automatically be created on the workstation.

**Setting User Restrictions in the PROGMAN.INI File** PROGMAN.INI is the initialization file for the Windows Program Manager. Edit this file to restrict user actions—for example, to prevent users from being able to run programs that aren't represented by an icon, and to keep users from exiting Windows. You can accomplish this by adding a [restrictions] sections to the PROGMAN.INI. The [restrictions] section can contain the following:

NoRun=1 removes the Run command from the Program Manager's File menu, as shown in Figure 10.12.

NoFileMenu=1 removes the entire File menu from Program Manager, as shown in Figure 10.13.

NoClose=1 prohibits users from exiting Windows by making the Exit option unavailable, as shown in Figure 10.14.

NoSaveSettings=1 prohibits users from saving changes made to Program Manager. In Figure 10.15, note that the Save Settings on Exit choice in the Options menu is unavailable.

EditLevel= sets editing privileges for the user:

0 (the default) allows all Program Manager changes.

1 removes access to the New, Move, Copy, and Delete commands on the File menu when a program group is selected. Users won't be able to add, change, or delete existing Program Manager groups.

2 removes access to the New, Move, Copy, and Delete commands on the File menu at all times. Users won't be able to add, change, or delete Program Manager groups or program items.

3 provides for restrictions in EditLevel=2 and also restricts users from changing command lines in the Program Item Properties dialog box for selected program items or groups. It also restricts users from making any changes to Program Group Properties. (See Figure 10.16.)

4 provides all the restrictions in EditLevel=3 and restricts users from making any changes in Program Item Properties dialog boxes.

For example, to remove the File menu from the Program Manager and prevent users from making any changes to Program Manager items, you would change the [restrictions] section to look like this:

```
[restrictions]
NoFileMenu=1
EditLevel=4
```

### **Running Setup for an Update with a New System Settings File**

Create a system settings file (.SHH) that contains updated settings for the [configuration] section when you need to update system settings across the network, such as installing new drivers. Create the file as explained in the beginning of the chapter, but to ensure that Windows reinstalls the new drivers or devices that you've specified in the file, you'll want to alter the names used for drivers just slightly. Otherwise Windows will ignore new [configuration] settings in favor of those already installed under the driver's name.

Force Windows to accept the new setting by putting an exclamation point before the name of the new device. For example, to force a new display driver to be set up, you would use an entry in the [configuration] section like this:

```
display=!svga
```

## **Optimizing Network Performance**

Network performance is a gossamer goal. Every possible advantage should be brought to bear to attain it. If you've followed the steps so far, you should have a stable Windows network configuration. Use the tips that follow to maximize your system's performance in the three key areas: basic setup, SYS.INI management, and that old bugaboo, printing.

### **Fine-tuning Windows Performance on a Network**

Get better performance from Windows running on a network by keeping the following in mind:

Set the lastdrive= statement in the CONFIG.SYS to a drive that's toward the beginning of the alphabet (like J, as opposed to Z), if possible. (This isn't recommended for NetWare LANs, which use the back end of the alphabet for network drivers.) For example, if the network contains disk drives that you don't need access to, exclude them from the lastdrive= statement.

When setting up swap files for use on a workstation, avoid using a network drive for a temporary or permanent swap file.

Avoid loading network applications into high memory if you experience any problems running Windows.

Likewise, if you are loading network drivers and other device drivers into upper memory and you experience problems running Windows, remove these drivers and try loading them in conventional memory.

For a shared copy of Windows running from a network drive, make sure that the user's personal WINDOWS directory, as well as the shared WINDOWS directory, is included on the PATH statement in the AUTOEXEC.BAT. (Or, if you are using NetWare, make sure their location is mapped as a search drive in your login script.)

If you're having problems running a shared copy of Windows, make sure that the DOS command, SHARE, is not executed on the workstation.

### **Fine-tuning SYSTEM.INI for Network Use**

You may have mastered the intricacies of your PC's SYSTEM.INI file (as described in Chapter 2), but running Windows on a network may bring unexpected performance, making it necessary to reevaluate the current settings. The following tips highlight those entries that are relevant to optimizing Windows on a network. Some of what we discuss here lies in territory we advised you to stay away from in Chapter 2. Here, however, we assume somewhat greater sophistication, commensurate with the greater need for precise control that comes with networking. But our original warning holds: Be careful when you work with these areas of SYSTEM.INI. This is the system equivalent of open heart surgery!

**[Boot] Section Entries** The [Boot] Section contains information about basic settings that are established when Windows starts.

CachedFileHandles=*number* tells Windows the number of recently used executable files and dynamic link library files (DLLs) that may remain open at one time (*number* can be from 2 to 12; the default is 12). While we warned you in Chapter 2 against changing this setting, you may need to if you have problems running a shared copy of Windows from a server. Some networks limit the number of files that can be open on a server at one time; try setting this entry to a lower number.

Network.drv=*filename* tells Windows the name of the network driver to use. You can change this setting by running Windows Setup and selecting a new Network setting.

**[Standard] Section Entries** The [Standard] section of the SYSTEM.INI defines the parameters Windows uses when running in Standard mode.

Int28Filter=*number*, in Standard mode, determines the percentage of Int28h interrupts that are made visible to applications that were loaded before Windows. With the default setting of 10, Windows makes every tenth interrupt visible. Increasing the setting improves Windows' performance, but may interfere with network software. Experiment to find the best setting for your system.

NetHeapSize=*kilobytes* determines the size of the conventional memory buffer that Windows uses for transferring data over a network in Standard mode. (The default for kilobytes is 8.) If an application is not running correctly, it may be because your network needs a larger buffer than the default of 8. Increasing this value decreases available conventional memory, but may be necessary for some networks. On the other hand, many networks can run perfectly with a setting of 4, increasing the amount of conventional memory available to applications.

**[NonWindowsApps] Section Entries** The [NonWindowsApps] section provides Windows with some basic operating information for running DOS applications.

**NetAsynchSwitching=0|1** specifies whether your system can switch away from an application running in Standard mode after it has made an asynchronous network BIOS call. (The default setting is 0.) A setting of 1 allows this task switching, but use it only if you are certain that the application will not receive network messages while you are switched away from it.

**[386Enh] Section Entries** The [386Enh] settings help Windows take the fullest advantage of the 386 processor and define the settings for running Windows in 386 Enhanced mode. If you experience problems running Windows on a network in 386 Enhanced mode, you'll want to examine these settings.

**EMMExclude=paragraph-range** prevents Windows from scanning a designated range of memory. (There is no default.) Many network adapter cards require this entry to be set to the range of memory used by the card.

**FileSysChange=Off|On** determines whether File Manager is notified when a non-Windows application creates, deletes, copies, moves, renames, or modifies a file. (The default is On for 386 Enhanced mode, Off for Standard mode.) If you're having problems running a non-Windows network application, try disabling this setting.

**InDOSPolling=Off|On** is another entry we told you in Chapter 2 not to edit, but you may need to change this setting to On because some memory-resident software requires this setting. (The default is Off.)

**Int28Critical=Off|On**, when set to On (the default), indicates some memory-resident software needs this setting switched to On. This setting is required by some networks that use virtual devices to perform internal task switching in response to Int28h.

**Network=filename** tells Windows the type of network that will be running in 386 Enhanced mode. The filename is that of the network driver.

**NetAsynchFallback=Off|On** and **NetAsynchTimeout=seconds** determine whether Windows attempts to rescue a failing asynchronous NETBios request. (The former defaults to Off and the latter to 5.0.) Normally, if Windows doesn't have enough space in its global network buffer to handle this request, the request fails. If NetAsynchFallback is on, however, Windows allocates a buffer in local memory and prevents any other virtual machine from running until the time specified by NetAsynchTimeout.

**NetDMASize=kilobytes** specifies the size of the DMA buffer used for NetBIOS transport software, if a NetBIOS network is installed. (The default is 32 on MCA computers, 0 on all others.) Windows uses the larger of this setting and the DMABufferSize setting.

**NetHeapSize=kilobytes** determines the size of the conventional-memory buffer that Windows uses in 386 Enhanced mode for transferring data over a network. (The default is 12.) Increasing this value decreases available conventional memory, but may be necessary for some networks. The NetHeapSize is always rounded up to the next 4K boundary.

**ReflectDOSInt2A=Off|On** specifies whether Windows should reflect or consume Int2Ah interrupts. Some software requires this be set to On. (The default is Off.)

**TimerCriticalSection=milliseconds** tells Windows to go into a critical section around all timer interrupt code, specifying the time-out period. (The default is 0.) During this time-out period, multitasking or switching is prohibited. A non-zero positive value means that only one virtual

machine at a time receives timer interrupts. Some network software requires that this value be set. Check your documentation. Settings other than zero will slow Windows' performance.

**TokenRingSearch=Off|On** determines whether Windows searches for a Token-Ring network adapter on AT-architecture machines. (The default is On.)

**UniqueDOSPPSP=On|Off**, if On, tells Windows to start every application at a unique address. (The default is On for networks based on Microsoft Network or LAN Manager, Off for all other networks.) The amount of memory between the addresses of two applications is the PSPIncrement described next. Some networks use these addresses to identify network processes, and an Off setting might cause one application to fail when you exit another.

**PSPIncrement=number** sets the amount of additional memory that Windows should reserve in each successive virtual machine when UniqueDOSPPSP=On. (The default is 2.)

## Printing Tips

Printing in Windows can be a headache, printing in Windows on a network can be an office nightmare. Nothing makes users madder than printing a file, hiking down the hall to the network printer, standing there for 10 minutes, and then having nothing happen. That's not much of a productivity enhancement, either. Getting printing to work right is a critical element of any Windows network setup.

**Printing on a Diskless Workstation** If you're working on a diskless workstation, Print Manager needs a TEMP directory to spool print jobs to. If a diskless workstation does not have a TEMP directory set up in its AUTOEXEC.BAT file, Windows will not be able to spool print jobs. Normally, if no TEMP directory is set, Windows defaults to the root directory of drive C, but in the case of a diskless workstation, this location does not exist.

To avoid printing problems on such a diskless workstation, make sure that the AUTOEXEC.BAT contains a TEMP= statement that points to a valid network directory.

**Taking Advantage of Print Manager on a Network** One of the biggest advantages to running Windows on a network is access to more than one printer. Get the most from the printers on your network by maneuvering around Print Manager so that you can get your files printed exactly when you need them and right where you want them.

Print Manager lets you monitor network print queues, allowing you to decide whether to use your default printer or send your files to another network printer that you have access to when the print queue is full of other users' files. To check on the status of your own files on the network print queue, select Refresh from Print Manager's File menu (or press F5). If your files aren't in the process of printing yet, choose the Selected Net Queue command from the File menu. This will show you all the files, not just your own, waiting for the network printer. If the queue is too long for your liking, select the Other Net Queue command from the File menu and have a look at the other printer's queue.

If you'd rather print your files on the second printer, return to the original queue, select the files by highlighting them, and choose Delete. Now make sure you're connected to the second printer by choosing Network Connections and making the proper connection. Once this is set, return to the Windows application and choose Print again, sending the print jobs to the correct network printer.

**Troubleshooting Techniques for Printing** If you're unable to print from Windows on a network, check the following:

Are you using the correct version of the network driver? If you're not sure, contact the manufacturer to find out about the most recent version.

Are the network printers connected? In the Control Panel, choose the Printers icon, select Connect, select Network, and note if there are any established network connections. If there aren't, select Browse, choose the correct printer, and establish a connection.

Can you print over the network outside of Windows? If not, there is a problem with the network, not Windows.

Try printing without the Print Manager.

Choose LPT1.DOS rather than LPT1 as your printer port.

From Windows, try printing a file to disk. This can help determine whether the problem is in Windows or the network. If you select an application such as Write, there is a Print to File option available in the Print dialog box, as shown in Figure 10.17.

If the test application doesn't have a Print to File option, you'll first need to set up this option in Control Panel:

1. Choose Printers.
2. Select Add.
3. Choose Generic/Text Only from the list of printers.
4. Select Install.
5. Choose Connect.
6. Select File from the Ports box.
7. Select OK.
8. Select Set as Default Printer.
9. Exit Control Panel and print a file to disk.

Once the file is printed to disk, exit Windows and try copying the file to the network printer using the command COPY filename LPT1. If it prints, there's most likely a problem with your network configuration. If the file does not print from DOS, try reinstalling the Windows printer driver on your system.

**A Printing Workaround** If you cannot print from Windows on your network, but you can print from DOS, here's a workaround to try until you get the problem solved:

1. From the Control Panel, select Printers.
2. Locate your printer in the list of supported devices, and select Connect.
3. In the Connect dialog box, make sure the Fast Printing Direct To Port check box is off.
4. Then select OK to return to the Printers dialog box, and select Close.

Changing the Fast Printing Direct to Port option sends the print job directly to DOS to be processed. This trick may not work for large files, however.



## Tips for Specific Networks

No two networks run the same way, so Windows can't interact with different networks in precisely the same way. General network tips, such as we've had so far in this chapter, are valuable. However, no one actually implements a general network. You will be choosing and living with a specific networking product. Use the following tips with the network you've chosen, or use them to get a feel for different network systems (and the problems they bring) before you buy.

### Novell NetWare Tips

NetWare has been the dominant PC networking system for nearly a decade. Known for superb file-handling capabilities, huge capacity, hardware independence, and almost mind-numbing complexity, NetWare takes a lot of care and feeding, but offers great payback for the effort. Many of these tips focus on avoiding the inevitable confusion that comes with NetWare's complexity.

**Make Sure Your Shell Is Current** If you're running Windows on a Novell network, make sure the NetWare shell you are using is version 3.26 or later. Windows 3.1 includes this updated version, NETX.COM.

**Running in Standard Mode** If you plan to run Windows in Standard mode on a Novell network, you'll have to remember to load the TSR, TBM12.COM. So that you don't have to remember to unload it from memory once you've exited Windows, create a batch file that does this for you. The file would contain the following lines:

```
TBM12
WIN
TBM12/U
```

**Logging In and Out** Novell Netware users should log in to the network before starting Windows and log out after exiting Windows. Do not attempt to log out from a DOS session within Windows.

**Increase FILES Access** Netware only allows you access to 40 files at a time. Working in Windows means that those can be used up before you know it. To increase the available file handles, you need to edit the SHELL.CFG file, as well as your CONFIG.SYS file. Open SHELL.CFG in a text editor and add the following line:

```
file handles=60
```

Next, open your CONFIG.SYS file in a text editor and locate the FILES= statement. Increase the current number to 60. The line would look like this:

```
Files=60
```

**SYSTEM.INI Settings** To make sure a PC running Novell NetWare with Windows doesn't time-out a network connection, you may need to make the following addition to the [386Enh] section of the SYSTEM.INI:

```
TimerCriticalSection=10000
```

**Settings for Diskless Workstations** A major improvement to performance for workstations without hard disks is to reduce the size of the temporary swap file's paging size to 128 or 256K. This prevents huge packet runs over the net. Also don't forget Show Dots=On, in SHELL.CFG. Otherwise you will not see .and .. in directory listings.

**Troubleshooting Techniques for Printing** If you're having problems printing from Windows on a NetWare network, check the following:

Do you have a valid SET TEMP statement in your AUTOEXEC.BAT? Windows won't be able to print correctly unless the DOS TEMP variable is set to a valid directory to which you have complete access.

Is the printer configured correctly? Verify that the correct Windows printer driver is installed and that the appropriate port is selected for output.

Do you have the correct NetWare shell version? Windows 3.1 with DOS 5.0 requires the NetWare shell version 3.26. If that is not loaded, check with your NetWare dealer or support representative for an update.

Is the Windows NetWare driver installed? Run Windows SETUP from Program Manager and check that Novell NetWare is shown as the installed network driver.

Are you using the proper NetWare print configuration? Check to make sure your settings match the following using the Netware PRINTCON utility:

Suppress Form Feed should be set to Yes. Windows places a form feed command at the end of its print jobs, so Netware does not need an additional form feed.

Since Windows is often printing bitmap graphics, File Contents must be set to Byte Stream to avoid NetWare's automatic expansion of tab characters. Byte values resembling tab characters occur randomly within bitmap graphic output; if these are expanded to spaces, the output will be garbled.

The Print Banner should be disabled (set to No) to avoid garbled printing.

Setting Auto Endcap to No prevents NetWare from timing out the print job while it is in progress.

Enable Timeout should also be set to No to avoid timing out the print job while in progress.

If you experience fragmented or garbled print jobs, make sure the Print Banner, Auto Endcap, and Enable Timeout settings are correct.

If you cannot set these options globally by using the PRINTCON utility, you may specify them on your CAPTURE command line. The command would look like this:

```
CAPTURE NB NA TI=0 NFF NT
```

NB is used to specify that no banner page be printed, the NA (No Automatic endcap) option is the same as setting Auto Endcap to No in PRINTCON. TI=0 (timeout) is the same as setting Enable Timeout to No. NFF (no form feed) is the same as Suppress Form Feed set to Yes in PRINTCON. NT (no tab expansion) is similar to setting File Contents to Byte Stream.

### **3Com Networks**

3Com's networking system comes in more flavors and has gone through more changes than NetWare. These are tips that apply to most 3Com software. 3Com's networks tend to be most popular where a variety of different kinds of hardware sit on users' desks.

**SYSTEM.INI Settings** If you're using 3Com's 3+Share or 3+Open network, add the following entries to the SYSTEM.INI's [386Enh] section:

TimerCriticalSection=10000  
UniqueDOSPSP=True  
PSPIncrement=5

**Check Your PATH** Windows Setup makes changes to a workstation's AUTOEXEC.BAT file, including making sure that the WINDOWS directory is located ahead of any conflicting directories in the PATH statement. But 3Com's 3+Share network may change the path when a user logs in. You may need to change the network log-in procedure to guarantee that the PATH statement is set correctly to run Windows on the workstation.

**3Station and All ChargeCards** If your 3Com network contains 3Com 3Station diskless workstations with All ChargeCards, make sure that the system is not configured to load the network drivers in the upper memory area. This configuration causes the system to run unreliably in Standard mode.

### **Artisoft LANtastic Tips**

LANtastic is a peer-to-peer network, which means that all the linked PCs share management chores, rather than a single central server handling them. Lauded as the easiest network to install and use, LANtastic is great for small offices and PC-only networks with moderate power requirements.

**SYSTEM.INI Settings** When Windows is installed on an Artisoft LANtastic network, an EMMExclude= entry is added to the [386Enh] section of the SYSTEM.INI file. This setting is required for the LANtastic Enhanced 2mbps network adapter. And if you change the memory address setting of this adapter, you'll need to change this setting to reflect the new address of the adapter.

The exact entries for LANtastic versions 3.x are

EMMExclude=D800-DFFF  
InDOSPolling=True  
NetHeapSize=76  
NetAsynchFallback=True  
NetAsynchTimeout=50

For version 4.x they are

EMMExclude=D800-DFFF  
NetHeapSize=64  
NetAsynchFallback=True  
NetAsynchTimeout=50  
PerVMFiles=0

**How to Create a Permanent Swap File** When Windows is running on a LANtastic server, you will be unable to create a permanent swap file for a workstation. To do so, restart the workstation but don't load the server software. Create the swap file in Windows and then restart the network software.

**Format Floppies** When Windows is run from a LANtastic server, users will be unable to format floppy disks from File Manager. You can, however, shell to DOS, and use the FORMAT command to format a floppy disk.

**Printing in LANtastic 3.x** To successfully print on LANtastic 3.x network over Windows, you'll need to turn off the Fast Printing Direct to Port option in the Control Panel's

Printers section. Also, set your printer port to LPT n.DOS, where n is the LPT port number. You should also disable the Print Manager.

### **Banyan VINES Tips**

VINES is a Unix-based network that excels at wide-area connectivity. In VINES you can literally click on a printer in London from your PC in San Francisco and print halfway around the globe. Since VINES already goes to some lengths to mask the complexity of its inner workings, it has a somewhat more arm's-length interaction with Windows, at least so far. Windows and VINES will work on the same system, but they don't really work together.

**Start VINES First** Remember to log onto VINES before starting Windows. If you don't, you'll receive an error message telling you that you won't be able to use VINES functions from within Windows.

**SYSTEM.INI Settings** For VINES versions 4.0x, the following changes are made to the [386Enh] section of the SYSTEM.INI:

```
TimerCriticalSection=5000
UniqueDOSPSP=True
PSPIncrement=5
```

For VINES versions 4.1x, these changes are made to the [386Enh] section of the SYSTEM.INI:

```
Network=*dosnet, *vnetbios, vvinesd.386
TimerCriticalSection=5000
UniqueDOSPSP=True
PSPIncrement=5
```

**Printing in VINES 4.0x** If you're using VINES 4.0x and have print jobs that are timing out, use the VINES SETPRINT utility to remove the time-out on your printer port. The command would be as follows:

```
SETPRINT LPTx servicename/D:INFINITE
```

x is the printer port and *servicename* is the network print queue.

**NetBIOS, Standard Mode, and DOS apps** If you are running Windows in Standard mode and load NetBIOS before Windows, you may not be able to access any DOS applications. You'll have to load Windows in Standard mode first and then load NetBIOS from within Windows.

### **Microsoft LAN Manager Tips**

Microsoft's LAN Manager hasn't been a very popular networking choice. It has, however, been a rather influential product all the same. That's because many other products, including recent versions of 3Com and Ungermann-Bass networks, have been based to varying degrees on LAN Manager. In other words, the following tips may apply to products of many names. Check your network documentation to find out if the networking software you use is based on LAN Manager and, if so, which exact version.

**SYSTEM.INI Settings** For LAN Manager Basic, the following changes are made to the [386Enh] section of the SYSTEM.INI:

```
network=*dosnet, *vnetbios, lanman10.386
```

```
TimerCriticalSection=10000
UniqueDOSPPSP=True
PPSPIncrement=5
```

For LAN Manager 2.0 Enhanced, this change is made to the [386Enh] section of the SYSTEM.INI:

```
network=*vnetbios, *dosnet
TimerCriticalSection=10000
UniqueDOSPPSP=True
PPSPIncrement=5
```

For LAN Manager 2.1 Enhanced, this change is made to the [boot] section of the SYSTEM.INI:

```
network.drv=lanman21.drv
```

while the following changes are made to the [386Enh] section of the SYSTEM.INI:

```
TimerCriticalSection=10000
UniqueDOSPPSP=True
PPSPIncrement=5
```

**LAN Manager and Expanded Memory** If you run Windows in 386 Enhanced mode on LAN Manager, it's a good idea to avoid using expanded memory. If you do use expanded memory, watch out for the system slowing down and occasionally locking up on you. To turn off the use of expanded memory, you'll need to edit the LANMAN.INI file.

Open LANMAN.INI in a text editor and locate the [workstation] section. Find the line lim= and change it to read lim=no. Save the file and restart the workstation.

**Removing WinPopup** LAN Manager Enhanced has a pop-up feature that allows you to receive incoming messages broadcast on the network. To run compatibly through Windows, these must be accessed through LAN Manager's WinPopup utility. To disable this utility and support for the pop-up services (since messages popping up in the middle of critical work may infuriate users), do the following:

1. Open LANMAN.INI in a text editor such as Notepad.
2. Locate the [workstation] section and find the line WRKSERVICES=.
3. Remove NETPOPUP or MINIPUP from the WRKSERVICES= line, but retain the MESSENGER entry.
4. Save the file, exit from the network, and start it again.

WinPopup will no longer appear, and any messages received will not be displayed in Windows but will be written to the MESSAGES.LOG file that is in the LOGS subdirectory under the LANMAN.DOS directory. You can view MESSAGES.LOG using a text editor such as Notepad if you want to review the messages received during a Windows session.

## DEC Pathworks

DEC Pathworks stands a notch above other networks in terms of what it aims to do. Pathworks wants to provide an entire network environment, much as Windows provides a desktop environment. Pathworks, coming as it does from DEC, presumes a much more varied, multiplatform system than most PC-centric local area networks do. Since both systems involved

are complex in their own right, the interface between Pathworks and Windows requires careful tending.

**SYSTEM.INI Settings** For DEC Pathworks 4.0 and above, the following changes are made to the [386Enh] section of the SYSTEM.INI:

```
Network=*dosnet, denet.386, decnb.386  
TimerCriticalSection=10000
```

If you have problems running NetBIOS applications, try replacing the preceding decnb.386 entry with \*vnetbios. The entry would look like this instead:

```
Network=*dosnet, denet.386, *vnetbios
```

**A CONFIG.SYS Change You May Need to Make** If you're having difficulty starting Windows on a DEC Pathworks network, or if you're having problems opening and copying files, you may need to increase the FILES= setting in your CONFIG.SYS file to FILES=30.