WIN.INI AND SYSTEM.INI: A COMPLETE REFERENCE GUIDE

THE MULTIFACETED NATURE OF WINDOWS IS CONCENTRATED WITHIN TWO files: WIN.INI and SYSTEM.INI. These files function as intricate control panels through which the sophisticated user can fine-tune Windows in innumerable ways. On the one hand this is a brilliant stroke of program design and one of Windows' big wins. It makes the operating system easily accessible to user, manager, and programs. On the other hand, the density of options in Windows results in two huge, complex files that can prove daunting to the uninitiated. So, as a natural first step on our journey toward total Windows power, we are devoting this early chapter to deciphering the intricacies of Windows' two most important files.

NOTE Both Windows 3.0 and Windows 3.1 have WIN.INI and SYSTEM.INI files, but not every notation is the same in both. We have aimed this chapter at Windows 3.1, the latest version, but we note where a particular aspect is new or changed from Windows 3.0.

WIN.INI and SYSTEM.INI have distinctly different purposes. In broad terms, WIN.INI's purpose is to record your preferences about how you want your system to look and behave. Entries in WIN.INI, for example, tell Windows whether you want to be beeped when you err, what wallpaper you want to use, whether you want your wallpaper tiled or centered, and myriad other details. SYSTEM.INI, in contrast, mostly records information about the nature of your system - the kind of display you're using, the type of network you're connected to, the names of device drivers that Windows will use, and so on.

This distinction, preferences versus system components, is not absolute. Printer and communication ports, for example, are listed in WIN.INI, and SYSTEM.INI handles one crucial matter of preference - the name of the program that will act as your Windows shell. But generally you can think of WIN.INI as the place where customization choices are recorded. If you change the equipment you're using to run Windows, or if you get involved in troubleshooting your system, you'll probably be working with SYSTEM.INI.

How .INI Files Are Structured

WIN.INI and SYSTEM.INI are both plain ASCII text files. Each consists of several sections, and each section is headed by a word or phrase enclosed in brackets, like this:

 [Desktop]

Beneath the bracketed heading come one or more lines consisting of a keyword, an equal sign, and zero or more parameters. A typical entry might look like this:

 KeyboardSpeed=31

If a line has no parameter to the right of the equal sign, Windows uses that line's default value. So, for example, the line

 KeyboardSpeed=

would be equivalent to the line

 KeyboardSpeed=31

because 31 is KeyboardSpeed's default value.

In most cases, Windows also uses a line's default value if that line is omitted. (There are exceptions. If certain lines are removed from SYSTEM.INI, Windows doesn't use defaults; it crashes.) If you want to specify a nondefault value for a line that isn't now in your WIN.INI file, you can simply add the line. For example, the [Desktop] section of your WIN.INI file may not include the lines

 WallpaperOriginX=

and

 WallpaperOriginY=

By adding these lines and specifying values other than 0 (the lines' default value), you can make a wallpaper image appear at some position other than the center of your screen.

Many .INI-file entries require Boolean values. For instance, you can enter True, Yes, On, or 1 to enable an entry's function, and False, No, Off, or 0 to disable the entry's function. (Case doesn't matter.) Any line that begins with a semicolon is treated as a "comment" and disregarded by Windows. You can add as many comment lines as you like. Within any section, the order in which lines appear is immaterial. And within the initialization file as a whole, the order in which sections appear is also immaterial.

Who Does What to Whom?

The majority of the lines in both WIN.INI and SYSTEM.INI are created and maintained by various elements of your Windows software. You do not need to change these items directly; Windows will do it for you.

For example, entries in the [Desktop] section of WIN.INI specify details about the way you want your system to look. Most of these entries are made automatically when you change settings in the Desktop section of Control Panel. As soon as you click OK after making selections in this part of Control Panel, Control Panel updates your WIN.INI file. The Windows applications you install, like Word for Windows or Ami Pro, may also add lines to WIN.INI to define preferences and other items.

Some customizing choices can only be made by editing your .INI files, however. If you want to change the font used by Windows to

display the text beneath your icons, for example, you'll need to modify WIN.INI yourself.

As a rule, changes that are made to an initialization file by some component of Windows, such as Control Panel, take effect immediately. Changes you make yourself are usually not effective until the next time you start Windows. (The Setup program can make changes to SYSTEM.INI that don't take effect until the next startup. But in these cases, Setup presents a button that you can click to automatically restart Windows.)

How to Edit an .INI File

You should remember two rules when editing any of Windows' initialization files: Make a backup first, and don't use a formatting editor. Making a backup gives you a chance to recover if you make a serious mistake - or simply change your mind. Formatting editors (such as most word processing programs) introduce nontext characters into documents and are a good way to damage an .INI file.

The safest way to edit WIN.INI or SYSTEM.INI is to use the System Editor that Microsoft ships with Windows. The formal name for this Windows application is SYSEDIT.EXE. If you don't find an icon for it in your Program Manager group, add one. As an editor, SYSEDIT is a bit primitive; for example, it has a Search command but no Replace. The program has two virtues, however: It creates backups automatically (with the extension .SYD), and it puts no formatting characters in your file.

So, to edit WIN.INI or SYSTEM.INI, follow these steps:

1. Make a backup of the file you're editing. (Although SYSEDIT automatically backs up any file you use it to change, it keeps only the version last saved, which may not be a working version. As a precaution, always back up the .INI file you're editing before making any changes.)

2. Run SYSEDIT from either the File Manager (File, Run) or by double-clicking on its icon.

3. Four files appear ready for your perusal, each in its own window: WIN.INI, SYSTEM.INI, AUTOEXEC.BAT, and CONFIG.SYS. Click the window that contains the file you want to edit, and modify the text as you like.

4. Save the file. You can either use SYSEDIT's File, Save command or exit SYSEDIT (Alt+F4) and click on Yes when it asks if you want to save the current changes.

5. Restart Windows so that your changes will take effect.

Tips for Using SYSEDIT

You might think that a text editor built into an operating system that only works with four files isn't of much value. In the case of Windows and SYSEDIT.EXE, you couldn't be more wrong. Certainly, you could call up WIN.INI, SYSTEM.INI, AUTOEXEC.BAT, or CONFIG.SYS in any text editor when you need to change them. But, in Windows, these files are so important, and you'll be checking them so often, that having all four ready in editable form is a great boon. SYSEDIT also assures that the files are saved in pristine text format, something you might have to double-check when using a word processor. In short, SYSEDIT represents

the kind of small, beneficial detail that has set Windows apart as a user-responsive environment.

We explain the basics of using SYSEDIT in the tip "How to Edit an .INI File." Here are some more advanced tricks to use with this unique editor. Try it, you'll like it.

Install SYSEDIT as an Icon If you like to optimize your system setup by messing around with your AUTOEXEC.BAT CONFIG.SYS, WIN.INI, and SYSTEM.INI files, you can save steps by adding an icon for SYSEDIT to your desktop, as shown in Figure 2.1. Create the icon by choosing New from the File menu in Program Manager and specifying SYSEDIT.EXE as the command line. Be sure to always back up these system files before doing any edits.

Quicker .INI Editing When making changes to your WIN.INI, PROGMAN.INI, and SYSTEM.INI files, you probably find yourself scrolling through the document trying to locate the appropriate section. Save time by using the Search function in Notepad, SYSEDIT, or whatever text editor you are using. Search for the opening bracket character ([) that starts each of the files' sections. By using the keyboard equivalent to search for the Next occurrence (F3 in SYSEDIT and Notepad) you'll move through the file much more quickly. Likewise, if you're looking for multiple references to the same application, use the Search function to find them quickly.

Back Up Your INI Files before Installing a New Windows Application Because many Windows applications make changes to the INI files, it's a good idea to back them up before you install a new application. If you run into a problem and need to remove the application, it's handy to have a reference to see what changes were made to the INI files. With the backups as your guide, you can compare the two files and remove all references to the application.

Use SYSEDIT without a Mouse If you open up SYSEDIT and attempt to start working on your files without using your mouse, you'll notice that you can't activate the cursor. To do so, switch out of SYSEDIT momentarily using the Alt+Tab key combination. Then press Alt+Tab again to return to SYSEDIT. The cursor will now be active and you can edit without the aid of a mouse.

A Word about Syntax

The WIN.INI file is divided into distinct sections, which are always referred to in brackets: [Windows], [Desktop], and so on. Each entry has an equal sign-for example, Device=. Following the equal sign are the parameters that can be used with the entry. If the parameters for the entry consist of a choice between a limited number of discrete values, the choices are demarcated with a pipe character (|), like so: Yes | No.

NOTE If you must supply a specific value for a parameter, the type of value is noted in italic-for example, *seconds* or *kilobytes*. For each entry, we include the default value in parentheses immediately following the entry-for example, (default: 3). Table 2.1 includes a brief description of each section of the WIN.INI file.

TABLE 2.1 Sections of the WIN.IN File and What They Do

Section   Description

[Windows]Affects elements of the Windows environment, including which applications start when you load Windows, the warning beep, printing, the window border width, keyboard speed, mouse settings, and the definition of files as documents or programs

[Desktop]Controls the appearance of the desktop and the position of windows and icons

[Extensions]Associates specified types of files with corresponding applications

[Intl]Describes how to display items for countries other than the United States

[Windows Help]Lists settings used to specify the default size, placement, and text colors of the Help window and dialog boxes

[Ports]Lists all available output ports

[FontSubstitutes]Lists pairs of fonts that are recognized by Windows as interchangeable

[TrueType]Describes options for using and displaying TrueType fonts

[Sounds]Lists the sound files assigned to each system event

[MCI extensions]Associates specified types of files with Media Control Interface devices

[Compatibility]Includes lines that resolve compatibility issues arising from minor differences between Windows 3.1 and Windows 3.0

[Fonts]Describes the screen font files that are loaded by Windows

[Network]Describes network settings and previous network connections

[Embedding]Lists the server objects used in Object Linking and Embedding (OLE)

[Colors]Defines colors for the Windows display

[PrinterPorts]Lists active and inactive output devices to be accessed by Windows

[Devices]Lists active output devices that provide compatibility with earlier versions of Windows applications

[Programs]Lists additional paths that Windows will search to find a program file when you try to open an associated data file

        We discuss the sections of WIN.INI in the order in which they appear in the table, which is how they should appear in a default

Windows setup. Within each of these sections, we include line-by-line options in alphabetical order for easy reference.

WIN.INI Line by Line

This is a section you should *not* skim over. This is real meat. Read it. Savor it. Chew it. If you want to understand how Windows works and what you can do with it, you must understand .INI files *in detail*. Here we examine all the many different sections of WIN.INI and each line within those sections that you may ever want to change. At first glance this may seem like an awful lot of detail to plow through, but persevere. Where Windows is concerned, .INI knowledge is power.

Tips for Editing WIN.INI

Windows sets up its .INI files at setup. However, if you are a sophisticated user-or even just a curious one-you'll want to tinker with the arrangement Windows starts out with. Here are some ideas for editing WIN.INI.

Automate .INI Backups with SYSEDIT The SYSEDIT program that comes with Windows is a handy way to access your WIN.INI, as well as your SYSTEM.INI, CONFIG.SYS, and AUTOEXEC.BAT files. But there's an added bonus to using SYSEDIT: automatic backups. Every time you make a change to a file with SYSEDIT, it keeps the previous version of your file and marks it with the .SYD extension. So if you're editing your WIN.INI and realize that you've made some mistakes that have overwritten your original file, don't panic. Just rename WIN.SYD to WIN.INI and you're back in business.

Familiarize Yourself with WIN.INI's README Files The WININI.TXT file that comes with Windows 3.0 and the WININI.WRI file that comes with Windows 3.1 are important documents for those who will be editing their WIN.INI. Skim them on-line, or print a copy of their contents for future reference.

Why It Pays to Keep WIN.INI Slim Every time you select an option in the Windows Control Panel, the entire WIN.INI file is written to disk. If your file contains unnecessary entries, or if it contains a lot of blank lines between sections, this will slow the process down. Keep WIN.INI as slim as possible, and you won't always be waiting for your hard-disk light to turn off.

How to Make WIN.INI Changes Take Effect without Restarting Windows If you make changes to your WIN.INI file, you'll have to exit Windows before these changes take effect. And of course this isn't very convenient. To fool Windows into updating the WIN.INI file without exiting Windows, go into the Control Panel. Select an option such as the Keyboard icon, and click on it. Don't change anything; just select OK. You'll hear some hard-disk activity, which means that just by opening the option you've rewritten the WIN.INI file. Your previous changes should now be in effect.

[Windows]

Lines in the [Windows] section affect a variety of factors concerning the appearance and behavior of your system. Almost all of these lines are managed for you by Control Panel.

Beep=Yes | No (default: Yes) Makes Windows beep to signal various kinds of user errors. If the beeping annoys you, you can turn it off by changing the Beep=line from Yes to No. To do this, open the Sound section of Control Panel and clear the Enable System Sounds check box shown in Figure 2.2.

BorderWidth=number (default: 3) Tells Windows how wide to draw the borders around sizable windows. This value can range from 1 to 49. The default is 3, which is also the value that Windows always uses for nonsizable windows (dialog boxes, for example). To change the BorderWidth value, open the Desktop section of Control Panel. Enter a new number in the Border Width field at the lower-left corner of the dialog box.

 If you're using Windows on a low-contrast LCD screen and have difficulty clicking on window borders, you might want to increase the border width. A setting of 5 or 6 will make the borders much more visible. If you prefer the Twiggy look, reduce the value to 1.

CoolSwitch=0 | 1 (default: 1) Affects the behavior of the Alt+Tab and Alt+Shift+Tab keyboard combinations. mWith CoolSwitch on (set to 1), when you switch programs by pressing Alt+Tab, Windows displays a box in the center of your screen showing the name and icon of the program you're about to switch to. Alt+ Shift+Tab steps you through your apps in reverse order. With CoolSwitch off, Windows displays the program's title bar and window borders (or the icon and icon label, if the program is minimized). As its more formal name (Fast "Alt+Tab" Switching) suggests, CoolSwitch can speed up the switching process because it requires the system to do less graphical work. But its main value is that it lets you see exactly where you're going. If you want to turn CoolSwitch off, open the Desktop section of Control Panel and clear the check box labeled Fast "Alt+Tab" Switching.

Boolean Values for Editing .INI Files

Most of entries in Windows' .INI files are set using Boolean values. You should note that there is more than one acceptable value for enabling or disabling the setting.

 To *activate* a setting you can use

- True

- Yes

- On

- 1

 To *disable* a setting you can use

- False

- No

- Off

- 0

The values are not case-sensitive.

CursorBlinkRate=*milliseconds* (default: 530) Governs the speed at which Windows blinks the cursor in text documents, spreadsheet cells, and other editable fields. The default is 530 milliseconds per blink. You might want to slow it down, if you find the cursor blink distracting, or speed it up if you're working at a screen where the cursor is sometimes hard to find. To change this value, open the Desktop section of Control Panel and adjust the scroll bar in the lower-right corner of the dialog box (shown in Figure 2.3).

The range of values available when you use Control Panel's scroll bar is 200 through 1200 (although those values are not visible in Control Panel). A value of 200 gives you a very fast blink, and 1200 gives you almost none at all. You may use blink rates outside that range by modifying WIN.INI directly. For example, you could make the cursor seem to stop blinking altogether by setting CursorBlinkRate to some large number, such as 50000, but this is not advisable. The cursor has two phases of equal duration-on and off. If you make the on phase long, you will have equally long periods when the cursor is invisible. For best results, stick to the range that Control Panel offers.

DefaultQueueSize=*number* (default: 8) Specifies the maximum number of messages that can be held in an application's message queue. Your WIN.INI file probably does not include this line (which means your system is using the default value). This line is included primarily for the benefit of developers who want to test their applications under varying conditions. As an end user, you should have no need to modify this line.

Device=*output-device-name, device-driver, port-connection* Records your current default printer setting, as specified in the Printer section of Control Panel. The line consists of three arguments separated by commas, like this:

 Device=HP LaserJet Series II,HPPCL,LPT1:

The first argument is the name of the printer; this is how youÕll see your printer identified in your applications' printer-selection dialog boxes. The name must match an entry in the [Devices] section of WIN.INI (described later in this chapter). The second argument is the filename of your printer's driver file, minus its .DRV extension. The third argument is the name of the printer port to which the default printer is assigned. This entry must match an entry in the [Ports] section of WIN.INI (also covered later in the chapter).

DeviceNotSelectedTimeout=*seconds* (default: 15) Specifies the default amount of time that Windows will wait for an off-line printer to be connected before issuing a timeout error. To increase or decrease this setting for all your printers, edit this line in WIN.INI. To

override the default for one particular printer, use the Connect option under the Printers icon in Control Panel.

Documents=*extensions* (default: none) Specifies the extensions of files to be regarded by Windows as documents. Specify each extension without a period, and use spaces to separate extensions, like this:

 Documents=abc xyz 123

If the Documents=line in your WIN.INI file includes extensions that also appear in the [Extensions] section (covered later), you may safely remove them from this line. All [Extensions] extensions are automatically treated as documents.
        You might want to put an extension on the Documents= line if that extension is associated with more than one application. If you have two programs that create .DOC files, for example, you might find it inconvenient to associate .DOC with either program. By adding that extension to the Documents= line, however, you can still include your .DOC files in File Manager views that are filtered to display documents only.

DosPrint=Yes | No (default: No) Makes Windows send printer output directly to hardware ports, rather than printing via the DOS Interrupt 21 file-access services. This results in faster performance, but requires complete hardware compatibility. If you are able to print from DOS but not from Windows (and you have a properly installed Windows printer driver), try changing this setting from No to Yes. You can do that by opening the Printers section of Control Panel, selecting any installed printer driver, clicking the Connect button, and clearing the Fast Printing Direct to Port check box shown in Figure 2.4. (Note: This setting was introduced with Windows 3.1.)

DoubleClickHeight=*pixels* (default: 4)
DoubleClickWidth=*pixels* (default: 4) Specify the amount you can move the mouse between the two clicks of a double-click. If you move the mouse more than 4 pixels (by default) in either direction, Windows regards your action as two single clicks. To make Windows more tolerant, add these lines to WIN.INI and specify values greater than 4.

DoubleClickSpeed=*milliseconds* (default: 452) Determines the maximum time that can elapse between the two clicks of a double-click. If the amount of time between double-clicks exceeds this value, Windows regards your action as two single clicks. To increase the double-click tolerance, open the Mouse section of Control Panel and adjust the Double Click Speed scroll bar.

KeyboardDelay=*milliseconds* (default: 2) Specifies the length of time you have to press a key before it begins to repeat. (Note: This line was introduced with Windows 3.1.) To make your keyboard begin repeating sooner, reduce this value or open the Keyboard section of Control Panel and move the top scroll bar to the right.

KeyboardSpeed=*number* (default: 31) Determines the speed at which a key repeats when you hold it down. The higher the number, the faster your keyboard repeats. The default setting, 31, is also the highest

accepted value. To change the repetition speed, open the Keyboard
section of Control Panel and adjust the lower scroll bar, shown in
Figure 2.5.

Load=*filename(s)* (default: none) Can be used to name any programs
that you want to run minimized at the beginning of each Windows session.
If the program is not stored in your WINDOWS directory, your SYSTEM
directory, or another directory in your DOS path, include the full path.
Use the space character to separate programs if you specify more than
one.

       If you're using Program Manager as your Windows shell, you don't
need to add startup programs on this line. To load an application or
document automatically at startup, put its icon in a program group
called Startup. The Load= line is included for compatibility with
earlier versions of Windows. The installation routines of certain
applications may add entries to this line (or to Run=, discussed later
in the chapter).

       If your default shell is not Program Manager, you may be able to
use the Load= line to specify startup programs. File Manager, MS-DOS
Executive, Norton Desktop for Windows, and some other shell programs
look for information on the Load= line at startup to load the specified
programs. Not all shell programs do this, however.

MenuDropAlignment=0 | 1 (default: 0) Tells Windows whether to
align drop-down menus flush left or flush right with their titles. The
default setting produces left alignment. If you want to be different,
you can edit WIN.INI and change the setting to 1. (You will probably
need to add the line to your [Windows] section.) See Figure 2.6 to see
what the result looks like.

**MenuShowDelay=*milliseconds* (default: 0 on 386 and 486
computers, 400 on 286 computers)** Determines the length of time
Windows waits after you click any menu bar item before it drops the
menu. On 286 computers, Windows uses a 400-millisecond delay to avoid
jerky menu behavior as you pass the mouse across the menu bar with a
menu already open. (If you move the mouse beyond a menu before it drops,
the drop is canceled.) If you find the delay irksome, reduce the value
in this line in WIN.INI. (You will probably need to add the line, and
then specify a nondefault parameter value.)

MouseSpeed = 0 | 1 | 2 (default: 1)
MouseThreshold1=*pixels* (default: 5)
MouseThreshold2=*pixels* (default: 10) Govern the acceleration
characteristics of your mouse. If MouseSpeed is 0, the relationship
between mouse movement and pointer movement is constant. If MouseSpeed
is 1, you get one level of acceleration, meaning anytime you move the
mouse more than Threshold1 pixels between mouse interrupts, the pointer
begins traveling twice as fast. If MouseSpeed is 2, the ratio of pointer
travel to mouse travel quadruples whenever the mouse is moved more than
Threshold2 pixels between interrupts.

       The easiest way to adjust these settings is by choosing options
in the Mouse section of Control Panel. Because not all variations of the
three settings are available via Control Panel, however, you may want to
fine-tune your mouse's behavior by editing WIN.INI.

MouseTrails=*number* (default: 0) Causes Windows to display trailing images of the mouse pointer as it moves across the screen when a MouseTrails setting of between 1 and 7 is used. The number parameter specifies the number of images that appear. Using MouseTrails may help you follow the pointer on an LCD display. The only Windows-supplied display drivers for which this option is available are EGA, VGA, and SuperVGA. Depending on your mouse driver, you may be able to change the setting via the Mouse section of Control Panel. If you do not see a Mouse Trails option there, or if the Mouse Trails option is dimmed, edit the MouseTrails line in WIN.INI.

NetWarn=0 | 1 (default: 1) Displays a warning message any time the network is unavailable at the start of a session if you have configured Windows to run on a network and NetWarn is set to 1. If you don't want the warning, disable the Network Warnings option in the Network section of Control Panel.

NullPort=*string* (default: "None") Earlier versions of Windows did not permit you to assign more than one printer to the same printer port. If you had two printers and only one port, you had to assign one of them to the null port. The NullPort line in WIN.INI specified the name of the null port. Its default was "None," but you could change it to *zip*, *nada*, or anything else. In Windows 3.1, this line is strictly an anachronism. If it appears in your Win 3.1 WIN.INI file, you may safely remove it.

Programs=*extensions* (default: com exe pif bat scr) Specifies the extensions of files for Windows to regard as programs. Specify each extension without a period, and use spaces to separate extensions, like this:

```
 Programs=abc xyz 123
```

Only file types identified as programs can be executed from within Windows. Thus, a simple way to make your system inoperable is to remove the exe extension from the Programs= line. (Be sure you have a backup of WIN.INI handy before you try this.)
        As long as pif is included in the list of program extensions, from File Manager you can run a .PIF file to launch the DOS application with which it is associated. Just double-click on the .PIF filename.

Run=*filename(s)* (default: none) Names any programs that you want to run in open windows at the beginning of each Windows session. If the program is not stored in your WINDOWS directory, your SYSTEM directory, or another directory in your DOS path, include the full path. Use the space character to separate programs if you specify more than one.
        If your default shell is not Program Manager, you may be able to use the Run= line to specify startup programs. File Manager, MS-DOS Executive, Norton Desktop for Windows, and some other shell programs look for information on the Run= line at startup and load the specified programs. Not all shell programs do this, however.

ScreenSaveActive=0 | 1 (default: 0) Determines whether Windows should activate one of its screen savers when your system is idle. The default turns off use of a screen saver. To turn screen-saving on, use

the Desktop section of Control Panel. If you're using a third-party screen saver, leave this switch off.

ScreenSaveTimeOut=*seconds* (default: 120) Specifies the length of time Windows waits when your system is idle before activating the current screen saver. (Note: This setting was introduced with Windows 3.1.) The default is two minutes. To change this value, use the Desktop section of Control Panel.

Spooler=Yes | No (default: Yes) Determines whether printing is managed by Print Manager. If you change the value to No, Windows may complete a print job more quickly, but you will be able to print only one job at a time. To turn spooling off, open the Printers section of Control Panel and clear the Use Print Manager check box, as shown in Figure 2.7.

SwapMouseButtons=0 | 1 (default: 0) Specifies your primary mouse button. If SwapMouseButtons is 0 (the default), the left button is primary. To make the right button primary, check the box labeled Swap Left/Right Buttons in the Mouse section of Control Panel.

TransmissionRetryTimeout=*seconds* (default: 45) Specifies the default amount of time that Windows will wait for acknowledgment while transmitting data to a printer. (If the printer in use is a PostScript device, Windows waits twice the length of time specified by TransmissionRetryTimeout.) If this time elapses, Windows issues a timeout error. To increase or decrease this setting for all your printers, edit this line in WIN.INI. To override the default for one particular printer, modify the printer's RetryTimeout entry in the [PrinterPorts] section of WIN.INI (see below), or change the Transmission Retry value in the Connect dialog box of the Printers option in Control Panel.

[Desktop]

The [Desktop] section records preferences about the appearance of your Windows desktop. Most of the items in this section are managed automatically by the Desktop section of Control Panel.

GridGranularity=*number* (default: 0) Lets you position application windows anywhere you please. If you set GridGranularity to a nonzero value, your windows are forced to align along an imaginary grid. Values from 1 to 49 are permitted; each increment increases the coarseness of the grid by 8 pixels both horizontally and vertically. This feature can help if you need precise positioning, but it has no effect on the placement of icons or document windows. To change the setting in Control Panel's Desktop dialog box, change the Granularity number near the lower-left corner (in the Sizing Grid box).

IconSpacing=*pixels* (default: 77) Determines the horizontal spacing used by Windows in response to an Arrange Icons command. If you want more space between your icons, increase the IconSpacing value. To do so, use the Spacing setting in Control Panel's Desktop dialog box.
        If you've used a version of Windows earlier than 3.1, you probably sometimes needed to increase the default value to avoid having icon labels overlap. In version 3.1, a simpler solution is to let

Windows wrap icon titles. (IconTitleWrap is covered in a moment.) Even with IconTitleWrap on (set to 1), however, you may sometimes experience overlapping icon titles, particularly if you minimize applications containing maximized document windows. In that case, the icon title text includes the name of the current document as well as the name of the application. The solution? Increase the horizontal distance between icons, as explained here, and the vertical distance, too, as explained shortly.

IconTitleFaceName=*fontname* (default: MS Sans Serif) Specifies the font that Windows uses to display icon titles. For a different look, edit this line in WIN.INI and specify another font (from the list that appears in the [Fonts] section of WIN.INI).

IconTitleSize=*number* (default: 8) Specifies the point size of the font used to display icon title text. If you have trouble reading icon titles on a high-resolution screen, try editing this line of WIN.INI to increase their value.

IconTitleWrap=0 | 1 (default: 1) Wraps long icon titles onto two or three lines, if necessary, to fit within the space defined by IconSpacing. If you object, clear the Wrap Title check box in the Desktop section of Control Panel.

IconVerticalSpacing=*pixels* (default: depends on other settings) Determines the vertical spacing Windows uses in response to an Arrange Icons command. The default depends on your IconTitleSize setting and display driver. To increase or decrease the default spacing, edit this line of WIN.INI.

Pattern=b1 b2 b3 b4 b5 b6 b7 b8 (default: "(None)") Records any desktop background pattern you specify via the Desktop section of Control Panel. The numeric values represent a bitmap 8 pixels high and 8 pixels wide. To change the pattern, use Control Panel's Desktop dialog box. You can either select a built-in pattern from the drop-down list or click the Edit Pattern button and design your own.

TileWallpaper=0 | 1 (default: 0) Determines whether the current wallpaper is centered (the default) or tiled. To change the setting, use the Desktop section of Control Panel.

Wallpaper=*filename* (default: "(None)") Specifies the filename and path of your current wallpaper bitmap. To change the setting, use the Desktop section of Control Panel.
        Although the Desktop dialog box does not include a Browse button, you are not limited to bitmaps stored in your Windows directory. Just add the drive and directory when you specify the filename.

WallpaperOriginX=*number* (default: 0)
WallpaperOriginY=*number* (default: 0) Tile wallpaper images beginning at the upper-left corner of your screen and continuing rightward and downward until the screen is filled. By specifying values other than zero for WallpaperOriginX= and WallpaperOriginY=, you can begin the tiling at some other location. The *number* parameter for WallpaperOriginX specifies a horizontal offset, in pixels, from the

upper-left corner. The *number* parameter for WallpaperOriginY specifies a vertical offset from the upper-left corner.

These lines also affect the position of centered wallpaper. If the coordinates are 0,0 (or the lines are omitted from WIN.INI), your wallpaper is centered around the midpoint of your screen. Nonzero coordinates produce alternative centering points measured from the upper-left corner. Low values for both coordinates, for example, would position the image near the upper-left corner, as shown in Figure 2.8.

[Extensions]

The [Extensions] section can contain any number of lines of the form

 extension=command-line

Each line associates a class of document files with an application. A line associating .INI files with Notepad, for example, would look like this:

 ini=notepad.exe ^.ini

When you run a document file whose extension is listed here, Windows launches the associated application, and the application loads your document. You can run a file by double-clicking on it in a File Manager window or by means of a shell program's File, Run command. You can associate as many extensions as you want with a specific application here, but you can't associate a given extension with more than one application.

The Windows Setup program automatically creates associations for many of the Windows "applets." Your [Extensions] section therefore may already include entries for .WRI, .TXT, .CAL, and others. Other applications' installation programs also commonly modify this section.

Many associations are also recorded in the Windows registration database. (The *registration database* is a binary file, REG.DAT, that records both simple file associations and information used by Windows for object linking and embedding.) If an extension is associated in both the [Extensions] section of WIN.INI and the registration database, Windows disregards the entry in [Extensions].

The safest way to create or modify an association, therefore, is to use File Manager's File, Associate command. That command updates both the registration database and the [Extensions] section of WIN.INI.

[Intl]

Settings in the [Intl] section record your preferences regarding the display of numbers, dates, and times. Entries that begin with an "i" must be integers. Entries that begin with an "s" must be strings (text). All entries can be set from the International section of Control Panel. Defaults listed here are for systems installed in the United States.

Certain Windows programs that existed originally in DOS versions have their own extensive international formatting commands and may use these in preference to your WIN.INI settings. Lotus 1-2-3 for Windows is an example.

iCountry=*country-code* (default: 1) Specifies the country whose standard settings you want to use. The *country-code* should be the country's international telephone code in all cases except Canada. To

specify Canada, use the number 2. The default setting, 1, specifies the United States.

iCurrDigits=*number* (default: 2) Specifies the number of digits that should appear after the decimal separator in currency-formatted values. (The decimal separator is either a period or a comma, depending on the iCountry setting.)

iCurrency=*number* (default: 0) Specifies the position of the currency symbol. (The symbol itself is determined by the sCurrency line.) Use 0 to put the symbol before the number, 1 to put it after the number, 2 to put it before the number (with a space between the symbol and the number), or 3 to suppress the display of the currency symbol.

iDate=*number* (default: depends on iCountry setting) Specifies the date format Windows should use. Values may be 0 (mm/dd/yy), 1 (dd/mm/yy), or 2 (yy/mm/dd). This line is provided for compatibility with Windows 2.x. If you have it in your WIN.INI file, you may safely delete it.

iDigits=*number* (default: 2) Specifies the number of digits to be displayed to the right of the decimal separator.

iLZero=0 | 1 (default: 0) Specifies whether a single leading 0 should appear to the left of the decimal separator in values between -1 and +1.

iMeasure=0 | 1 (default: 1) Specifies your choice of metric (0) or English (1) measurement.

iNegCurr=*number* (default: 0) Governs the display of negative currency values. The currency symbol and decimal separator are determined by the sCurrency and iCountry settings. Table 2.2 lists each value and how it displays negative currency values.

TABLE 2.2 Negative Currency Value Settings Used with the iNegCurr= Setting in WIN.INI's [Intl] Section

Value	Display
0	($123.45)
1	-$123.45
2	$-123.45
3	$123.45-
4	(123.45$)
5	-123.45$
6	123.45-$
7	123.45$-
8	-123.45 $
9	-$ 123.45
10	123.45 $-

iTime=*number* (default: 0) Specifies your choice of a 12-hour (0) or 24-hour (1) clock.

iTLZero=*number* (default: 0) Specifies whether a leading 0 should appear before single-digit time values. Change this value to 1 if you want the zero.

s1159=*string* (default: AM) Specifies the characters that should appear after ante-meridian times when a 12-hour clock is in use. Some programs (Excel, for example) will use only the first four characters of *string*.

s2359=*string* (default: PM) Specifies the characters that should appear after post-meridian times when a 12-hour clock is in use.

sCountry=*string* (default: United States) Specifies the name of the country whose standard values you want to use.

sCurrency=*string* (default: $) Specifies your choice of currency symbol.

sDecimal=*string* (default: .) Specifies your choice of decimal separator.

sLanguage=*string* (default: enu) Specifies your choice of standard language. Some Windows applications use this information to determine which sorting sequence or spelling checker to use. Acceptable values include those in Table 2.3.

TABLE 2.3 Acceptable Values for the Windows Standard Language Setting in the [Intl] Section of WIN.INI

Value	Language
dan	Danish
deu	German
eng	International English
enu	American English
esn	Modern Spanish
esp	Castilian Spanish
fin	Finnish
fra	French
frc	Canadian French
isl	Icelandic
ita	Italian
nld	Dutch
nor	Norwegian
ptg	Portuguese
sve	Swedish

sLis=*string* (default: ,) Specifies your choice of punctuation to separate items in a list.

sLongDate=*format* (default: dddd, MMMM dd, yyyy) Specifies your choice of long-date format. Acceptable symbols and their meanings are listed in Table 2.4.

sShortDate=*format* (default: M/d/yy) Specifies your choice of short-date format. The symbols are the same as for sLongDate (see Table 2.4).

TABLE 2.4 Symbols Used for the Date Settings in WIN.INI's [Intl] Section

Symbol Meaning

d Day number, without leading 0
dd Day number, with leading zero
ddd Weekday, abbreviated
dddd Weekday, spelled out
M Month number, without leading 0
MM Month number, with leading 0
MMM Month name, abbreviated
MMMM Month name, spelled out
yy Year, in two-digit form
yyyy Year, in four-digit form

sThousand=*string* (default: ,) Specifies your choice of punctuation to appear between three-digit groups in numbers greater than 999.

sTime=*string* (default: :) Specifies your choice of punctuation to use in time values, such as 12:34.

[Windows Help]
The [Windows Help] section of WIN.INI records settings that determine the size and placement of the Help window and its dialog boxes, including the color of text that displays a Help macro, pop-up window (used for glossary definitions), or new screen of information.

M_WindowPosition=[upper_left_x, upper_left_y, width, height, 0 | 1] Specifies the default position and size of the main Help window. The first two parameters specify the x and y coordinates of the window's upper-left corner. The third and fourth parameters specify the width and height of the window, in pixels. The fifth parameter is 1 if the window is maximized and 0 if it isn't (if 0, Winhelp uses the default width and height). The five parameters are enclosed within a pair of brackets. You do not need to edit WIN.INI to set these values. Simply position the Help window the way you want it to appear in the future. When you close the Window, Winhelp updates your WIN.INI file.

H_WindowPosition=[upper_left_x, upper_left_y, width, height, 0 | 1] Specifies the default position and size of the Help system's History dialog box. The parameters work as they do for M_WindowPosition, except that the fifth parameter has no effect because the History dialog box cannot be maximized.

A_WindowPosition=[upper_left_x, upper_left_y, width, height, 0 | 1] Specifies the default position and size of the Help system's Annotate dialog box. The parameters work as they do for M_WindowPosition, except that the fifth parameter has no effect because the Annotate dialog box cannot be maximized.

C_WindowPosition=[upper_left_x, upper_left_y, width, height, 0 | 1] Specifies the default position and size of the Help system's Copy dialog box. The parameters work as they do for M_WindowPosition, except that the fifth parameter has no effect because the Copy dialog box cannot be maximized.

IFJumpColor=_r g b_ Specifies the color of Winhelp's "inter-file" jump text. This is the text that, when clicked, leads to a new panel of information taken from a different help file. The three parameters specify red, green, and blue color values and must be in the range 0 to 255. If you choose a combination of settings that would produce a dithered color on your display adapter, Windows uses the nearest available solid color.
        To change this setting, you must edit WIN.INI. The first time you change it, you'll probably need to add the line to your WIN.INI file. If you want to restore the original color settings, you can either delete the line from your file or just delete the three color values, leaving the keyword and equal sign with no parameters.

IFPopupColor=_r g b_ Specifies the color of Winhelp's "inter-file" pop-up hot text-the text that, when clicked, generates a pop-up window taken from another file. The parameters work as they do for IFJumpColor.

JumpColor=_r g b_ Specifies the color of Winhelp's "intra-file" jump text. This is the text that, when clicked, leads to a new panel of information taken from the same help file. The parameters work as they do for IFJumpColor.

MacroColor=_r g b_ Specifies the color of text that, when clicked, runs a Help-file macro. The parameters work as they do for IFJumpColor.

PopupColor=_r g b_ Specifies the color of Winhelp's "intra-file" pop-up hot text-the text that, when clicked, generates a pop-up window that usually defines terms. The parameters work as they do for IFJumpColor.

[Ports]
The [Ports] section of WIN.INI lists your system's serial and printer ports, as well as the communication parameters in effect for your serial ports. You may list as many as ten ports.
        Use the Ports section of Control Panel to specify communications parameters. WIN.INI will record your settings, using the following syntax:

 portname:=baud-rate, parity, word-length, stop-bits [[,p]

Acceptable entries include the following:

 LPT_n_:=
 COM_n_=
 EPT:=
 FILE:=
 filename.ext=
 LPT_n.ext_=
 LPT_n.ext=path_

Use LPT*n*:= to specify a parallel printer port. Be sure to include the colon and the equal sign. This line takes no parameters.

Use EPT:= to specify an IBM Personal Pageprinter. This line takes no parameters.

Use FILE:= if you want to be able to print to disk. This line takes no parameters. If the current printer is attached to a port named FILE:, Windows prompts you for a filename and then directs output to that file. You can use File Manager or the DOS COPY command to transfer the file's contents to a physical printer.

Use *filename.ext*= if you want to print to a particular filename. Windows will not prompt in this case.

Use LPT*n.ext*= to bypass Windows' normal method of printing direct to port. (See the DOSPrint= line, under the [Windows] heading, discussed earlier.) For example, if you sometimes want to use the MS-DOS Interrupt 21 services when printing to your first parallel port, you could include the line

 LPT1.DOS=

(Do *not* include a colon after the port number. This line doesn't take any parameters, either.) It makes no difference which three characters you use for the extension. (If you have upgraded from earlier versions of Windows, you may find a line such as LPT1.OS2 in your WIN.INI file. This is exactly equivalent to LPT1.DOS, which was introduced with Windows 3.1.)

Because the name LPT1.DOS does not include a colon, Windows regards it as a filename. Output directed to such a name is handed off to DOS as though it were redirected to an ordinary disk file. However, DOS sees the first four characters as a printer device and handles the data accordingly. The result is output directed to the printer via the DOS interrupts.

[FontSubstitutes]

The [FontSubstitutes] section, introduced with Windows 3.1, provides a list of font aliases. Each entry has the following form:

 font-name=font-name

Windows includes four entries by default:

```
Helv=MS Sans Serif
Tms Rmn=MS Serif
Times=Times New Roman
Helvetica=Arial
```

Helv and Tms Rmn were bitmapped fonts included with Windows versions prior to 3.1. Their Windows 3.1 equivalents are MS Sans Serif and MS Serif. So, for example, if you open a document formatted in Helv under Windows 3.0, Windows will use MS Sans Serif instead (provided you haven't removed these default [FontSubstitutes] entries).

If you regularly receive documents showing bitmapped fonts that remain on your system, substitute them for TrueType equivalents. Name all the fonts you receive in documents from other people, and pair each with a TrueType alternative. Note, however, that if you type the name of a non-TrueType font resident on your system into an application's font

box, it will still format correctly. In other words, you'll see the non-TrueType font, not the TrueType equivalent you specified in WIN.INI.

[TrueType]

The [TrueType] section of the WIN.INI file was added with version 3.1 because this version marked the first appearance of the TrueType fonts it describes. TrueType fonts are Microsoft's answer to PostScript. Both produce lettering that is scalable: Characters can be sized, altered, and manipulated in ways that standard, static characters cannot.

OutlineThreshold=*pixels-per-em* (default: 256) Specifies a "crossover" point at which TrueType characters are rendered from outlines into bitmaps via GDI (Graphics Device Interface) calls, rather than via the TrueType rasterizer. The TrueType rasterizer is faster.
    If this threshold is set too high, characters with a large point size (say, 61-point type-particularly in highly decorative fonts-on a 300 dpi printer) may not be rendered. Microsoft recommends that you not set the threshold higher than 300. If you get blank spaces on a printout where you expect to see characters, try editing WIN.INI and reducing the OutlineThreshold value.

TTEnable=0 | 1 (default: 1) Determines whether TrueType fonts are available. The default setting enables TrueType fonts for your applications. To disable TrueType, use the Fonts section of Control Panel, click the TrueType button, and clear the Enable TrueType Fonts check box, as shown in Figure 2.9.

TTIfCollisions=0 | 1 (default: 0) Determines which font Windows uses in the rare circumstance when a TrueType font and a non-TrueType font have identical names and identical character sets. The default setting causes Windows to use the non-TrueType font.

TTOnly=0 | 1 (default: 0) Determines whether only TrueType fonts appear in your applications' font dialog boxes. The default setting makes all fonts on your system available to your applications. Windows 3.1's built-in TrueType fonts look great-they're scalable to any point size between 4 and 127 points, and they can be downloaded to your printer for true WYSIWYG output. If you like what you see, banish the jaggies forever by specifying that your Windows applications list only TrueType fonts: Open the Fonts section of Control Panel, click the TrueType button, and select the check box labeled Show Only TrueType Fonts in Applications.

[Sounds]

The [Sounds] section records the mapping of system events to sound files established via the Sound section of Control Panel. The section may contain one or more lines of the format:

 system-event=filename,description

A typical line might look like this:

 SystemStart=C:\WINDOWS\MMDATA\GONG.WAV,SystemStart

In this line, the event known internally to windows as SystemStart has been mapped to a waveform file named GONG, stored in the directory C:\WINDOWS\MMDATA. When the SystemStart event occurs, Windows plays the specified waveform file. The last parameter on the line, also SystemStart in this example, is the description of the event as it appears in the Events list in Control Panel's Sound dialog box. Normally, this last parameter is identical to the keyword at the beginning of the WIN.INI line, but you could change it to something else if you wanted to. You could change SystemStart to Startup, for example.

Ordinarily, however, there's no particular reason to edit this section of WIN.INI. Simply assign events to sound files in Control Panel, and WIN.INI will be updated for you.

You must have a properly installed Windows-supported sound board or the Microsoft speaker driver to be able to play .WAV files. If your computer does not have either of these, the Events and Files list boxes in the Sound section of Control Panel will be dimmed, as shown in Figure 2.10, and you will probably not have a [Sound] section in your WIN.INI file.

[MCI extensions]

The [MCI extensions] section, introduced in Windows 3.1, may include one or more lines of the following format:

 extension=mcidevice-type

Typical examples look like this:

 wav=waveaudio
 mid=sequencer
 rmi=sequencer
 mmm=animation

These lines, which are entered automatically by the Windows Setup program, tell Windows how to render multimedia files. You should never need to edit this section.

[Compatibility]

The [Compatibility] section includes lines that resolve compatibility issues arising from minor differences between Windows 3.1 and Windows 3.0. The keyword on each line is the name of an application, and the parameter is a hexadecimal value, like this:

 CHARISMA=0X2000

The Setup program automatically creates more than two dozen [Compatibility] entries when you install Windows 3.1 (whether or not you have the affected applications on your system). Other applications' installation routines may add lines of their own. You should not edit or delete the entries in this section.

[Fonts]

The [Fonts] section records the names of the screen fonts loaded by Windows at the start of each session. Each entry has the following format:

```
 font-name=font.filename
```

The *font-name* is the name of the font as it appears in Windows dialog boxes. The *font.filename* is the name of the file that contains the font's specifications.

Be aware that simply adding a new entry to the [Fonts] section of WIN.INI does not make the font available. To make a screen font available, you must install it via the Fonts section of Control Panel.

[Network]

Unlike DOS, which never really got its arms around networking, Windows accepts the presence of LANs as part of its natural milieu. This section of WIN.INI allows pinpoint description of key aspects of a network connection for a Windows session.

drive=*network-server-and-share* Tells Windows which network connections to restore at the start of each session. To change this line, use the Network Connections command on File Manager's Disk menu.

InRestoreNetConnect=0 | 1 (default: 1) Tells Windows whether to restore network connections at startup. To change, use the Networks section of Control Panel. If the Network dialog box for your network doesn't include a reconnect option, change this line to 1 to reconnect to the network at startup.

port=*network-printer-path* Defines the path and port for a network printer. To change, open the Printers section of Control Panel, select the appropriate printer driver, click the Connect button, and then click the Network button.

[Embedding]

The [Embedding] section, added to WIN.INI in Windows 3.1, may include one or more lines of the following format:

```
 object=description, description, program file, format
```

These lines define classes of objects that may be embedded in OLE client applications. A typical example might look like this:

ExcelWorksheet=Microsoft Excel Worksheet,Microsoft Excel Worksheet,C:\EXCEL

The information in this section duplicates information kept in the registration database (REG.DAT) and is provided for compatibility with applications that don't assume you're using Windows 3.1. You should never have to edit this section of WIN.INI.

Although this section may eventually become an anachronism, don't delete it from your WIN.INI file. Many OLE client programs (to allow for the possibility that you're using Windows 3.0) read object information from WIN.INI, not from the registration database.

[Colors]

The [Colors] section records the color preferences that you have specified in the Color section of Control Panel. The section contains 21 lines of the following format:

component=r g b

The *component* keyword names an element of your Windows user interface,
and the *r*, *g*, and *b* parameters specify red, green, and blue color
values. These parameters must be in the range 0 to 255.
        Because Control Panel takes care of this section of WIN.INI for
you, there is ordinarily no reason to edit this section yourself. You
might conceivably want to go into this part of WIN.INI if you needed to
match a custom color scheme on someone else's machine. In that case, you
could copy the [Colors] values from your colleague's WIN.INI file into
your own.
        Named color schemes-the ones that Windows supplies and any you
create yourself-are not recorded in WIN.INI. Instead, they're stored in
the [Color schemes] section of the CONTROL.INI file.

[PrinterPorts]

The [PrinterPorts] section specifies which port(s) each of your printer
drivers is attached to, and what timeout settings are in effect.
        Control Panel fills out this section for you automatically when
you connect printer drivers to printer ports and adjust their timeout
settings. Certain applications, such as Winfax and Publisher's Powerpak,
add virtual ports for themselves in this section as well. You should
never need to edit this section directly.
        The [PrinterPorts] section may include one or more lines of the
following format:

 device=driver,port,DeviceTimeout,RetryTimeout[, other ports ...]

A typical entry might look like this:

 HP LaserJet Series II=HPPCL,LPT1:,15,45

        Device is the name of your printer, as it appears on a device=
line in the [Windows] section of WIN.INI.
        Driver is the name of your printer's driver file, without
the .DRV extension. If this file is not in your WINDOWS\SYSTEM
subdirectory, the full path must be included.
        Port is the name of a port listed in the [Ports] section of your
WIN.INI file.
        DeviceTimeout specifies the amount of time that Windows will
wait for this printer to be put on line before issuing a timeout error.
        RetryTimeout specifies the default amount of time that Windows
will wait for acknowledgment while transmitting data to a printer.
        If you have a printer connected to more than one port, each
connection will appear on that printer's device= line. For example, if
your LaserJet Series II is attached to both FILE: and LPT1:, the line
might look like this:

 HP LaserJet Series II=HPPCL,FILE:,15,45,LPT1:,15,45

[Devices]

The [Devices] section is included for compatibility with Windows 2.x
applications. In Windows 2.x, it served a purpose comparable to that now
served by [PrinterPorts]. You should never need to edit this section
directly. Do not delete or comment out this section, however.

Applications that do not require Windows 3.1 may look for information here rather than in the [PrinterPorts] section.

The [Devices] section may contain one or more lines of the following format:

device=driver,port,[,other ports ...]

The *device* parameter is the name of a printer as identified on a *device*= line in the [Windows] section of WIN.INI and as it appears in your applications' printer-selection dialog boxes. The *driver* parameter is the name of your printer's driver file, without its .DRV extension.

If you have a printer connected to more than one port, each connection will appear on that printer's *device*= line. For example, if your LaserJet Series II is attached to both FILE: and LPT1:, the line might look like this:

HP LaserJet Series II=HPPCL,FILE:,LPT1:

[Programs]

The [Programs] section, not to be confused with the Programs= line of the [Windows] section, provides a mechanism for expanding your DOS PATH environment variable. If you try to run a document file (by double-clicking in a File Manager window, for example) and Windows cannot locate the document's associated .EXE file, you will be prompted to provide the path of that .EXE file. After launching your file, Windows will then create an entry like the following in the [Programs] section of WIN.INI:

program file=drive:directory\program file

For example, suppose you move NOTEPAD.EXE to C:\WINDOWS\ TINYAPPS, a directory not included in your AUTOEXEC.BAT file's path= statement. The next time you try to run JOTTINGS.TXT, Windows will ask you where NOTEPAD.EXE is located. After you tell it, Windows will create a [Programs] section in WIN.INI, if one is not there already, and will add the following line:

notepad.exe=c:\windows\tinyapps\notepad.exe

You should never need to edit this section directly.

SYSTEM.INI Line by Line

Windows' SYSTEM.INI file primarily records information about the makeup of your PCÑits display, drivers, network, and so on. As a result, anyone who wants to understand the deep, inner workings of Windows needs a full knowledge of SYSTEM.INI. The following pages list the lines in SYSTEM.INI that you are most likely to want to change at some point in your use of Windows 3.1. Study them well; they are vital for a full understanding of this environment.

Because SYSTEM.INI describes the deep workings of Windows, it's potentially dangerous. You should change settings in SYSTEM.INI only when you completely understand what you are doing and why. Some parts of SYSTEM.INI should never be changed by the user. We list those sections in this chapter. Some of these entries are modified for you automatically by the Windows Setup program, Control Panel, or the installation routines included with third-party applications.

If you look through your own SYSTEM.INI file, you will probably find a number of lines that are not listed here. Some of these install drivers essential to your system. Others (those whose parameters begin with an asterisk) identify virtual device drivers for 386 Enhanced mode that are actually incorporated into WIN386.EXE. All of the lines we chose to omit are lines that you should not tinker with or delete, unless you're willing to reinstall Windows altogether if something goes awry.

A Word about Syntax

Like WIN.INI, the SYSTEM.INI file is divided into distinct sections, the names of which are always enclosed within brackets ([Boot], [386Enh], and so on). Each entry has an equal sign-for example, Shell=. Following the equal sign are the parameters you can use with the entry. If the parameters for the entry consist of a choice between a limited number of discrete values, the choices are demarcated with a pipe character (|), like so: Off | On. If you must supply a specific value for a parameter, the type of value is noted in italic-for example, filename or seconds. For each entry, we include the default value in parentheses immediately following the entry-for example, (Default: 16).

Table 2.5 provides a brief description of the sections that appear in SYSTEM.INI.

TABLE 2.5 The Sections That Make Up SYSTEM.INI

Section Description

[Boot] Lists drivers and Windows modules
[Keyboard] Contains information about the keyboard
[Boot.description] Lists the names of devices you can change using Windows Setup
[386Enh] Contains information used by Windows in 386 Enhanced mode
[Standard] Contains information used by Windows in Standard mode
[NonWindowsApp] Contains information used by non-Windows applications
[MCI] Lists Media Control Interface (MCI) drivers (this section was introduced with Windows 3.1)
[Drivers] Contains a list of aliases (or names) assigned to installable driver files (this section was introduced with Windows 3.1)

We discuss the sections of SYSTEM.INI in the order in which they appear in Table 2.5, which is how they should appear in a default Windows setup. The line-by-line discussion includes only four sections, however-[Boot], [386Enh], [Standard], and [NonWindowsApp]-because it is inadvisable to edit the other sections, as we will discuss next. We have described those sections of SYSTEM.INI that well-informed users can tamper with to their benefit. Within each of these sections, we include line-by-line options in alphabetical order for easy reference.

Abandon Hope, All Ye Who Edit Here

As we mentioned at the beginning of the chapter, it's important to realize that the majority of the settings in the SYSTEM.INI file should not be casually edited. Such tinkering could literally kill your Windows session. In Table 2.6, to make sure there is no question about it, we list those sections and lines of SYSTEM.INI that you should touch only at your peril.

TABLE 2.6 Sections and Specific Entries of SYSTEM.INI That You *Should Not* Edit

Section   **Entry**

[Boot]   286grabber=*filename*
                         386grabber=*filename*
                         CachedFileHandles=*number*
                         Display.drv=*filename*
                         Drivers=*filename* | *aliasname*
                         keyboard.drv=*filename*
                         language.dll=*libraryname*
                         mouse.drv=*filename*
                         network.drv=*filename*
                         system.drv=*filename*

[Keyboard]   keyboard.dll=*filename*
                         oemansi.bin=*filename*
                         subtype=*number*
                         -type=*number*

[Boot.description] The whole section

[386Enh]   A20EnableCount=*number*
                         BkGndNotifyAtPFault=0 | 1
                         Device=*filename* | **devicename*
                         Display=*filename* | **devicename*
                         EBIOS=*filename* | **devicename*
                         EISADMA=0 | 1 | *channel.size*
                         HardDiskDMABuffer=*kilobytes*
                         IdleVMWakeUpTime=*seconds*
                         IgnoreInstalledEMM=0 | 1
                         InDOSPolling=0 | 1
                         Keyboard=*filename* | **devicename*
                         KybdPasswd=0 | 1
                         Local=*devicename*
                         LPTnAutoAssign=*seconds*
                         LRULowRateMult=*number*
                         LRURateChngTime=*milliseconds*
                         LRUSweepFreq=*milliseconds*
                         LRUSweepLen=*length-in-pages*
                         LRUSweepLowWater=*number*
                         LRUSweepReset=*milliseconds*
                         MapPhysAddress=*range*
                         MaxBPs=*number*
                         MaxDMAPGAddress=*address*
                         MinUnlockMem=*kilobytes*
                         Mouse=*filename* | **devicename*
                         Network=*filename* | **devicename*
                         NMIReboot=0 | 1
                         NoWaitNetIO=0 | 1
                         OverlappedIO=0 | 1
                         PageOverCommit=*megabytes*
                         PerformBackfill=0 | 1
                         PSPIncrement=*number*
                         ReserveVideoROM=0 | 1
                         ROMScanThreshold=*number*
                         ScrollFrequency=*number*
                         SGrabLPT=*port-number*
                         SyncTime=0 | 1

```
                        SystemROMBreakPoint=0 | 1
                        SysVMEMSLimit=number | kilobytes
                        SysVMEMSLocked=0 | 1
                        SysVMEMSRequired=kilobytes
                        SysVMV86Locked=0 | 1
                        SysVMXMSLimit=number | kilobytes
                        SysVMXMSRequired=kilobytes
                        TimerCriticalSection=milliseconds
                        TranslateScans=0 | 1
                        TrapTimerPorts=0 | 1
                        UniqueDOSPSP=0 | 1
                        UseInstFile=0 | 1
                        VideoBackgroundMsg=0 | 1
                        VideoSuspendDisable=0 | 1
                        WindowUpdateTime=milliseconds
                        WOAFont=font-filename
                        XlatBufferSize=kilobytes
                        XMSUMBInitCalls=0 | 1
[Standard]              PadCodeSegments=0 | 1
                        StackSize=kilobytes
[NonWindowsApp]         GlobalHeapSize=kilobytes
[MCI]                      The whole section
[Drivers]                  The whole section
```

[Boot]

The [Boot] section contains information about some basic settings that are established when Windows starts. Most importantly, this is where the shell-the program that controls the Windows screen-is loaded. Usually this is the Windows Program Manager, although alternatives are available, as discussed in Chapter 5.

Comm.drv=*filename* (default: comm.drv) Specifies the filename of the serial communications driver in use. The Windows 3.1 default driver is named COMM.DRV. A number of third-party communications programs install their own replacement drivers for COMM.DRV.

     If you upgrade a Windows 3.0 system that uses a third-party communications driver, the Windows 3.1 Setup program does not replace that driver with Microsoft's COMM.DRV, because your software may depend on the third-party driver. However, if you are running Windows 3.1 in Enhanced mode with a communications driver designed for Windows 3.0, you may need to add the line COMMDrv30=On to the [386Enh] section of SYSTEM.INI (discussed later in the chapter).

     To replace a third-party communications driver with Microsoft's COMM.DRV, you may need to make other changes to the [386Enh] section of SYSTEM.INI (because some third-party products also install their own virtual communications drivers). You should contact your communications vendor for details.

Fixedfon.fon=*filename* (default: depends on display driver)
Specifies the system font used by Windows 2.x running under Windows 3.1. If you have older applications that use monospaced fonts for all their text, you may be able to modernize them by changing this line.

     To make Notepad use a proportionally spaced font on a VGA system, try setting Fixedfon.fon to VGASYS.FON. Valid entries are raster

font files with a .FON extension. The specified file should be in your
SYSTEM directory.

Fonts.fon=*filename* (default: depends on display driver)

Specifies the system font Windows uses for menus and dialog boxes.
Windows is shipped with versions of this font for various resolutions of
display. You may switch to any other available system font by specifying
its name on the Fonts.fon= line. The specified font must be classified
by Windows as a system font and must be in your SYSTEM directory. (The
Windows-supplied system fonts are named EGASYS.FON, VGASYS.FON, and
8514SYS.FON.)

If you specify an invalid font, Windows will not start, and you
will have to reboot your computer.

For another way to change the Windows menu font (one that is not
limited to fonts of the system class), see the entry describing the
[Fonts] section of WIN.INI, earlier in the chapter.

Oemfonts.fon=*filename* (default: depends on display driver)

Specifies the name of the font the Clipboard Viewer program uses to
display text in OEM Text format. This is a monospaced font that uses the
character set native to DOS, rather than the ANSI character set used by
all other Windows-supplied fonts (except Symbol). When you copy a block
of text to the Clipboard, Windows ordinarily stores it in OEM Text
format as well as in Text format. If you then paste into a non-Windows
program, the Clipboard supplies the data in OEM Text format instead of
Text format. Wherever possible, the appropriate character-value
conversions are made so that, for example, accented characters appear
the same in the non-Windows program as they did in your Windows
application.

The font specified by Oemfonts.fon= is simply the font that the
Clipboard Viewer uses when you choose its Display command and select OEM
Text. Windows supplies three OEM-class fonts, named EGAOEM.FON,
VGAOEM.FON, and 8514OEM.FON.

The Setup program installs the font appropriate for your display
resolution and initializes the Oemfonts.fon= line accordingly. You can
change this setting by editing SYSTEM.INI, although there's probably no
good reason to do so. If you do edit this line, the specified font must
be categorized by Windows as an OEM font, or else Windows will not start
and you will need to reboot.

Shell=*filename* (default: PROGMAN.EXE)

Specifies the name of your
default shell, which is the program that starts and ends your Windows
sessions. This is a required line, so don't delete it or its parameter
value. The Windows Setup program automatically makes Program Manager
your default shell, but you can switch it to File Manager or another
program if you like. (To make File Manager the default, specify
WINFILE.EXE.) If the program you specify is not in your WINDOWS
directory, your SYSTEM directory, or a directory included in your DOS
PATH statement, be sure to supply the full path name.

Nearly any program can act as your default shell. But if the
program you specify is not designed to launch other programs, you will
turn your Windows system into a single-application environment. That may
be convenient at times. If you work only in Excel, for example, you can
save yourself some startup time and steps by making Excel your default
shell.

Vendors of third-party shells (such as the Norton Desktop for Windows) usually modify the Shell= line for you automatically if you choose to make their programs the default. To restore Program Manager as your default shell, edit the Shell= line and specify PROGMAN.EXE. (Note that you must include something on the Shell= line. If you leave it blank or delete the line, Windows won't start.)

Sound.drv=filename (default: depends on your system's sound hard-ware) Specifies the filename of the sound driver in use.TaskMan.Exe=filename (default: TASKMAN.EXE) Specifies the program that appears when you press Ctrl+Esc, double-click on the desktop, or choose an application's Switch To command. If this line is omitted or has no parameter value, Windows uses TASKMAN.EXE. Some third-party shells come with their own task-switching programs and modify this line. You may choose to modify it yourself if you want a different program to pop up in response to Ctrl+Esc.

If you make a change to this line and later decide to restore TASKMAN.EXE as your task switcher, either delete the line altogether or enter TASKMAN.EXE as the filename parameter. If you do not enter anything to the right of the equal sign, you get no task switcher.

A trick: You can modify this line for easier access to Solitaire. Just add the line

```
 TaskMan.Exe=SOL.EXE
```

to the [Boot] section of your SYSTEM.INI file. Now you can press Ctrl+Esc to get right into your favorite card game, as shown in Figure 2.11.

[386Enh]

Windows runs somewhat differently in 386 Enhanced mode than in Standard mode. These changes are primarily aimed at taking the fullest possible advantage of the powers of the 386 processor. The [386Enh] section defines the parameters Windows uses when running in 386 Enhanced mode.

32BitDiskAccess=Off | On (default: Off) On systems with hard-disk controllers that are 100-percent compatible with the Western Digital 1003 standard, this line determines whether Windows uses its "Fastdisk" technology to bypass the ROM-BIOS disk access services. As the name suggests, Fastdisk may result in quicker disk access. It may also let you run more non-Windows applications at once in 386 Enhanced mode.

When you install it, Windows checks to see if the hardware you're using could benefit from 32-bit disk access. If it can, a check box entitled Use 32-Bit Disk Access is included in the Control Panel's Virtual Memory dialog box. If you use Windows to run DOS applications in 386 Enhanced mode, 32-bit virtual memory is a godsend. It speeds up virtual memory access and doubles your PC's capacity to run DOS apps with background processing. There are some caveats, however: 32-bit disk access works only with Western Digital 1003-compatible hard-disk controllers (which account for 90 percent of them), and it's not safe on battery-operated PCs with power-saving facilities. But unless you're using a laptop or ESDI or SCSI hard disk, use 32-bit access.

To turn Fastdisk on, open the 386 Enhanced section of Control Panel, choose Virtual Memory, choose Change, and then select the check

box labeled Use 32-Bit Disk Access. Control Panel records your action on the 32BitDiskAccess line of SYSTEM.INI.

AllEMSLocked=Off | On (default: Off) If On, this entry prevents Windows from swapping expanded memory to disk and overrides any PIF settings that leave expanded memory unlocked. In most cases, swapping expanded memory to disk does not create problems.
      Change this line to On if you use a disk-caching program or DOS device driver that uses expanded memory.

AllVMsExclusive=Off | On (default: Off) If On, this line forces all non-Windows programs in Enhanced mode to run full screen, regardless of their PIFs. Microsoft recommends using this setting if you're running network or TSR software that isn't fully compatible with Windows 3.1.
      Setting this line to On also renders Alt+Enter ineffective at toggling between full and windowed-screen DOS apps.

AltKeyDelay=*seconds* (default: .005) Specifies the length of time Windows waits to process a keyboard interrupt after it has processed an Alt interrupt. Try increasing this value if you have applications that don't handle the Alt key correctly.

AltPasteDelay=*seconds* (default: .025) Specifies the length of time Windows waits to paste characters after the Alt key has been pressed. Some applications may require a higher setting than the default.

AutoRestoreScreen=Off | On (default: On) Specifies whether, when you switch back to a non-Windows program running in 386 Enhanced mode, the program's screen is restored by Windows or by the program. Affects only programs running in VGA display modes that identify themselves to Windows as being able to restore their own screens.
      Changing this setting to Off reduces memory use but may also degrade performance, since Windows can usually restore the screen more quickly.

CGANoSnow=Off | On (default: Off) If On, this entry causes Windows to do some processing to avoid "snow" on a CGA display.

CGA40WOA.FON=*filename* (default in U.S.: CGA40WOA.FON) Specifies the fixed-pitch font that Windows uses to display non-Windows applications that use a 40-column display with 25 or fewer lines.

CGA80WOA.FON=*filename* (default in U.S.: CGA80WOA.FON) Specifies the fixed-pitch font Windows uses to display non-Windows applications that use an 80-column display with 25 or fewer lines.

Com1AutoAssign=*number* (default: 2, if port is present)
Com2AutoAssign=*number* (default: 2, if port is present)
Com3AutoAssign=*number* (default: 2, if port is present)
Com4AutoAssign=*number* (default: 2, if port is present)
Determine how Windows behaves when two non-Windows applications or a non-Windows application and a Windows application both want to use a particular COM port at the same time. If *number* is -1, Windows always

displays a warning and asks you to decide which application gets the port. If *number* is 0, Windows doesn't arbitrate-any application can use the port at any time. If *number* is greater than 0, Windows gives the port to the application that claims it first. After that application has finished, Windows waits *number* seconds before allowing another application to use the port. *Number* may not be greater than 1000.

NOTE Windows always arbitrates device contention between Windows applications. The AutoAssign value affects only contention between virtual machines-that is, contention in which at least one of the applications is a non-Windows program.

To change the AutoAssign value for any COM port, open the 386 Enhanced section of Control Panel, shown in Figure 2.12. In the Device Contention section of the dialog box, select the port you want to change. Then choose Always Warn, Never Warn, or Idle.

Normally, it's not a good idea to set number to 0. However, certain non-Windows print spoolers may require a 0 setting for printers attached to serial ports. Your spooler vendor should advise you if this setting is necessary.

COM1Base=*address* (default: port value in BIOS data area)
COM2Base=*address* (default: port value in BIOS data area)
COM3Base=*address* (default: 3E8h)
COM4Base=*address* (default: port value in BIOS data area) Let you set the base (starting) address for each serial port, in both Standard mode and Enhanced mode. To change them, use the Ports section of Control Panel. Double-click the icon for the port you want to adjust, then click the Settings button and then the Advanced button to specify a base address.

COM1Buffer=*number* (default: 128)
COM2Buffer=*number* (default: 128)
COM3Buffer=*number* (default: 128)
COM4Buffer=*number* (default: 128) Specify the number of characters that Windows buffers on each port. You may need to increase the buffer size if you are losing characters. Before trying this, however, Microsoft recommends that you try setting the Com*x*Protocol line (described in a moment) to XOFF.

If you are using a 16550 Universal Asynchronous Receiver Transmitter (UART) and a non-Windows communications program running in Enhanced mode, you may need to set the communications buffer to 0 to take advantage of the UART's buffer.

COM1FIFO=Off | On (default: On)
COM2FIFO=Off | On (default: On)
COM3FIFO=Off | On (default: On)
COM4FIFO=Off | On (default: On) Determine whether the FIFO buffer of a 16550 UART will be enabled. If you are using an older 16550 UART that does not properly support the FIFO buffer, and you experience problems, try disabling the FIFO buffer for the appropriate port by changing this setting to Off.

COM1IRQ=*number* (default: 4)

COM2IRQ=*number* (default: 3)
COM3IRQ=*number* (default: ISA and EISA, 4; MCA, 3)
COM4IRQ=*number* (default: 3) Specify the interrupt request line
(IRQ) that each serial port uses in both Standard and Enhanced mode. To
change them, use the Ports section of Control Panel. Double-click the
icon for the port you want to adjust, then click the Settings button and
then the Advanced button to specify the IRQ you want to use. To disable
input for a serial port, specify an IRQ of -1.

COM1Protocol=XOFF | (blank) (default: blank)
COM2Protocol=XOFF | (blank) (default: blank)
COM3Protocol=XOFF | (blank) (default: blank)
COM4Protocol=XOFF | (blank) (default: blank) Determine whether
Windows, when running in 386 Enhanced mode, stops simulating characters
in a virtual machine on receipt of an XOFF character. If these lines are
set to any value other than XOFF, or if the lines are omitted, Windows
ignores XOFF characters. If a line is set to XOFF, Windows stops sending
characters when it gets an XOFF and resumes when it receives the next
character, whatever that may be. You should turn off the protocol
setting for any serial port used for binary file transfers. If you are
losing text characters at high baud rates, try setting the protocol to
XOFF. If you still lose characters, try increasing the associated
COM*x*Buffer line (described previously) to a higher value.

COMBoostTime=*milliseconds* (default: 2) Specifies the length of
time a virtual machine is allowed to process a COM interrupt. Try
increasing this value if you're losing characters on the screen in a
communications program.

COMIrqSharing=Off | On (default: MCA and EISA, On; ISA, Off)
Specifies whether serial ports can share interrupt request lines. If
COM1 and COM3 or COM2 and COM4 use the same IRQ on your system, this
line should be set to On.

COMdrv30=Off | On (default: Off) If this line is Off (the
default), Windows' virtual communications driver maintains its own copy
of the serial communications driver's interrupt handler.
        If you're using a communications driver designed for Windows 3.0
(specified by the [Boot] section's COMM.DRV= line discussed earlier),
you probably need to change this setting from Off to On. Add the line to
SYSTEM.INI if it isn't already there.

DMABufferIn1MB=Off | On (default: Off) Specifies whether the
direct memory access (DMA) buffer should be located in the first
megabyte of memory, for compatibility with 8-bit bus-mastering cards.

DMABufferSize=*kilobytes* (default: 16) Specifies the amount of
memory to be reserved for the DMA buffer. Some backup systems may
require a higher value than 16. Central Point Software, for example,
recommends a setting of 64.
        If a NetBIOS network has been installed before Windows starts,
Windows uses the larger of this size and the size specified by the
NetDMASize line (described in a moment).

If you are using EMM386.EXE with a D= parameter, be aware that Windows ignores this parameter, using the DMABufferSize parameter instead.

DOSPromptExitInstruc=Off | On (default: On) With this line set to On (or omitted), displays five lines of instructions near the top of the screen when you run a copy of COMMAND.COM (or choose the DOS Prompt icon in Program Manager) in 386 Enhanced mode. These instructions (shown in Figure 2.13) tell you to type Exit to quit the DOS session, to press Alt+Tab to switch back to Windows or another application temporarily, or to press Alt+Enter to switch the DOS session between a window and a full screen.

If you find this message vexing, change the DOSPromptExitInstruc line to Off. (You may need to add the line to the [386Enh] section of SYSTEM.INI.)

Note that with the prompt turned off, you will still get a copyright message from Microsoft near the top of your DOS session screen. And you will still get a polite message if you try to return from a DOS session by typing **Win**.

DualDisplay=Off | On (default: depends on display adapter) Set this line to On if you want EMM386 to make the B000-B7FF address range available as an upper memory block (UMB).

If you set DualDisplay to On, you must also include the parameter i=B000-B755 in the DEVICE=EMM386.EXE statement of your CONFIG.SYS file.

EGA40WOA.FON=*filename* (default in U.S.: EGA40WOA.FON) Specifies the fixed-pitch font Windows uses to display non-Windows applications that use a 40-column display with more than 25 lines.

EGA80WOA.FON=*filename* (default in U.S.: EGA80WOA.FON) Specifies the fixed-pitch font Windows uses to display non-Windows applications that use an 80-column display with more than 25 lines.

EMMExclude=*paragraph-range* (default: none) Prevents Windows from scanning a designated range of upper memory. This line is most often used for troubleshooting purposes. If you suspect that Windows is trying to locate its expanded-memory page frame or translation buffers in an area already occupied by a hardware adapter or device, you may want to exclude all of high memory by adding the statement EMMExclude=A000-EFFF. If that enables your system to work, you can then try excluding a smaller area of upper memory, continuing in this fashion until you locate the address where the conflict is occurring.

To use EMMExclude, specify the lower and upper boundaries of the region you want to exclude, in hexadecimal notation, with a hyphen separating the two addresses. Windows will round your lower boundary down and your upper boundary up, if necessary, so that the region is an even multiple of 16K. To exclude a smaller region of upper memory, use the ReservedHighMemory= setting, discussed later in this chapter.

EMMInclude=*paragraph-range* (default: none) Explicitly instructs Windows to scan the designated upper-memory range for free UMBs, overriding any overlapping area specified in an EMMExclude line. You can

use this line in conjunction with EMMExclude to locate the source of a UMB conflict.

As it does with EMMExclude, Windows rounds the specified area, if necessary, to cover an even multiple of 16K. To specify a smaller region, use the UsableHighArea setting (described later in this chapter).

Windows will ignore an EMMInclude range if it overlaps with a range excluded by means of EMM386's X= parameter.

EMMPageFrame=*paragraph* (default: none) Explicitly specifies the starting address for Windows' expanded-memory page frame.

EMMSize=*kilobytes* (default: 64) Allocates all available memory as expanded memory for the benefit of non-Windows applications that can use expanded memory. The EMMSize line lets you specify a maximum amount toÊbe used as EMS. You would want to do this if an application you're using claims all the available expanded memory, leaving you unable to start any other applications.

If you set this line to 0, Windows allocates no expanded memory but still loads its expanded memory manager. If you really want no expanded memory, use NoEMMDriver=On instead (described in a moment).

FileSysChange=Off | On (default: On for 386 Enhanced mode, Off for Standard mode) Determines whether File Manager is notified when a non-Windows application creates, deletes, copies, moves, renames, or modifies a file. Performance is faster with this service off. If you run Windows in 386 Enhanced mode and can live without immediate updating of File Manager's records, change this line from On to Off. (You may need to add this line to your [386Enh] section.)

Global=*devicename* (default: none) Specifies that an MS-DOS device is to be global, which means that one copy of its state is maintained in memory for all virtual machines. By default, all DOS devices other than CON are global. But some device drivers may try to make themselves local. The Global= switch is provided to override any drivers that may cause the system to fail if they are local.

The device-name must match the case of the device name exactly as it's listed in CONFIG.SYS, or this entry will not work.

Int28Critical=Off | On (default: On) If On, indicates that a critical section is needed for handling Int 28h interrupts used by memory-resident software. The default is On because some networks use virtual devices that perform internal task switching in response to Int 28h. If you're not using such software, you may get quicker task switching by setting this switch Off.

IRQ9Global=Off | On (default: Off) If On, converts IRQ 9 masks to global. Try setting this switch On if you are unable to read from floppy disks in 386 Enhanced mode.

KeyBoostTime=*seconds* (default: .001) Specifies the length of time an application gets increased priority when it receives a keystroke. Increase this value if your foreground programs respond sluggishly to keystrokes while other programs are running in the background.

KeyBufferDelay=*seconds* (default: .2) Specifies the length of time Windows waits before pasting keyboard input after the keyboard buffer is full. Try increasing this value if some of your non-Windows programs don't paste Clipboard text correctly.

KeyIdleDelay=*seconds* (default: .5) Specifies the length of time Windows ignores idle calls after simulating a keystroke to a virtual machine. Reducing this value may speed up keyboard input but may also cause some applications to slow down.

KeyPasteCRSkipCount=*number* (default: 10)
KeyPasteSkipCount=*number* (default: 2) Govern the delay loop Windows uses to slow down pasting into a virtual machine. Specifically, they determine how many Int 16h calls should return "empty" after a carriage return has been pasted (KeyPasteCRSkipCount) or any other character has been pasted (KeyPasteSkipCount) before Windows pastes another character. Try increasing both values if you are losing characters while pasting into a non-Windows program.

KeyPasteDelay=*seconds* (default: .003) Specifies the minimum time Windows waits after pasting a character before pasting another character. Try increasing this setting if you are losing characters while pasting.

KeyPasteTimeout=*seconds* (default: .003) Specifies the length of time Windows gives an application to make the BIOS keyboard-reading calls before Windows switches from its fast-paste (Int 16h) to its slow-paste (Int 9h) mechanism.

KybdReboot=Off | On (default: On) Specifies whether Windows uses a keyboard controller command to reboot the computer in response to Ctrl+Alt+Del. If your computer hangs when you attempt to reboot, change this setting to Off.

LocalLoadHigh=Off | On (default: Off) Determines whether Windows leaves some UMBs free for local use of virtual machines under DOS 5. Don't set this On unless you need local use of UMBs for particular virtual machines.

LocalReboot=Off | On (default: On) If On, enables the Ctrl+Alt+Del keyboard combination to terminate a hung application without rebooting your computer. Leave this switch On unless you relish the prospect of losing data when a program fails.

MaxCOMPort=*number* (default: 4) Specifies the maximum number of communications ports that Windows will support. Increase this value if you use more than four communications ports.

MaxPagingFileSize=*kilobytes* (default: 50% of available disk space) This is one of two SYSTEM.INI lines that govern the maximum size of a temporary swap file, relative to the amount of space available on the designated drive. (The other is MinUserDiskSpace, discussed in a moment.) Windows normally does not permit you to create a temporary swap

file larger than half the available disk space. If you're low on disk space, you might want to override this requirement. You can do that by adding the line MaxPagingFileSize to the [386Enh] section of SYSTEM.INI and specifying the number of kilobytes you want to be permitted to use. Then, after restarting Windows, go to the 386 Enhanced section of Control Panel, choose Virtual Memory, choose Change, and specify the size and location of your temporary swap file.

MaxPhysPage=*hexadecimal page number* (default: depends on amount of physical memory Windows detects at startup) Specifies the highest physical page number in memory that Windows will use. Windows initializes this value at startup depending on the amount of physical memory it finds in your system. If you use hardware devices that cannot recognize all of your system's physical memory, you might want to set an explicit ceiling with MaxPhysPage=. (An example of such a device might be an ISA DMA network card that cannot use memory beyond 16Mb.)

MCADMA=Off | On (default: On for MCA computers, Off for all others) Determines whether Windows uses the Micro Channel extensions to DMA. Turn this switch off if you use a Micro Channel computer on which the DMA extensions have not been implemented.

MessageBackColor=*vga color attribute* (default: 1)
MessageTextColor=*vga color attribute* (default: F) Specify the background and foreground colors, respectively, that Windows uses for certain full-screen text messages-like the message that appears when you press Ctrl+Alt+Del. The defaults give you white characters on a blue background. To change these values, specify a color attribute between 0 and 7 for the background and between 0 and F (in hexadecimal notation) for the foreground.

MinTimeSlice=*milliseconds* (default: 20) Governs the minimum amount of time a virtual machine is allowed to run before other virtual machines are given the opportunity to claim the processor. Setting this value lower can make multitasking appear smoother, but it will degrade overall performance because Windows will spend more time switching. Note that because all Windows applications run within a single virtual machine, this setting has no effect on the division of processor time between Windows applications.
        To change the MinTimeSlice setting, open the 386 Enhanced section of Control Panel and adjust the Minimum Timeslice value. Control Panel will update SYSTEM.INI for you.

MinUserDiskSpace=*kilobytes* (default: 2000) Specifies the minimum amount of disk space that Windows will leave free when creating a temporary swap file. (This entry has no effect if you have a permanent swap file.) The default, 2K, leaves room for temporary files created by Windows applications as well as for new and enlarged data files. Decrease this value cautiously if you need more room for a temporary swap file.

MouseSoftInit=Off | On (default: On) Change this setting from On to Off (by editing SYSTEM.INI) if your cursor and screen information

sometimes appear distorted when you use a mouse with a non-Windows program running in a window.

NetAsynchFallback=Off | On (default: Off)
NetAsynchTimeout=*seconds* **(default: 5.0)** Determine whether Windows attempts to rescue a failing asynchronous NetBIOS request. Normally, if Windows doesn't have enough space in its global network buffer to handle the 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 elapses.

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

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

NoEMMDriver=Off | On (default: Off) If On, prevents Windows from loading the driver that makes expanded memory available to non-Windows programs running in 386 Enhanced mode.
　　　If you never require expanded memory for your non-Windows programs, you should change this line to On. Doing so reduces the likelihood that Windows will be unable to place its translation buffers in upper-memory blocks, and may therefore free up more conventional memory for your non-Windows programs. Note that if Windows has to put translation buffers in conventional memory, the buffers reduce available conventional memory in all virtual machines.

PageBuffers=*number* **(default: 4)** Specifies the number of 4K page buffers that Windows uses to store asynchronous read and write pages when 32-Bit Disk Access is in use. You may get better Windows performance by increasing this setting.

Paging=Off | On (default: On) If Off, disables Windows' use of virtual memory. You should change this to Off only if you need the disk space that would be used by a temporary or permanent swap file. To change this line, open the 386 Enhanced section of Control Panel, choose Virtual Memory, and choose Change. Then, in the Type list, choose None.

PagingFile=path-and-filename (default: C:\WINDOWS
WIN386.SWP) Specifies the location and name of your temporary swap file. Because this line includes the full path of the file, it overrides any contradictory setting in PagingDrive= (described next). There's probably no reason to change the PagingFile value.

PagingDrive=*drive-letter* **(default: the drive on which your**
SYSTEM.INI file is stored) Specifies the drive that Windows uses for a temporary swap file. The setting is ignored if you use a permanent

swap file. To change this setting, click the Virtual Memory and Change buttons in the 386 Enhanced section of Control Panel and choose a new drive.

PermSwapDOSDrive=*drive-letter* (default: none) Specifies the drive that Windows uses for a permanent swap file. The setting is ignored if you use a temporary swap file. To change, click the Virtual Memory and Change buttons in the 386 Enhanced section of Control Panel and choose a new drive.

PermSwapSizeK=*kilobytes* (default: none) Specifies the size of your permanent swap file. To change this setting, click the Virtual Memory and Change buttons in the 386 Enhanced section of Control Panel and specify the desired size in the New Size box.

PerVMFiles=*number* (default: 10) Sets the number of local file handles allowed for each virtual machine. Note, however, that this setting is ignored if the DOS SHARE utility is installed. Also be aware that the total file handles opened by all virtual machines, plus the file handles specified in your CONFIG.SYS's FILES statement, may not exceed 255.

If a non-Windows application gives you an error message instructing you to increase the FILES= statement in your CONFIG.SYS, ignore this advice and increase PerVMFiles.

ReflectDOSInt2A=Off | On (default: Off) Specifies whether Windows should reflect or consume Int 2Ah interrupts. Windows consumes the interrupts and runs more efficiently with this switch off. Turn it on only if you are running memory-resident software that depends on Int 2Ah calls.

ReservedHighArea=*paragraph range* (default: none) Prevents Windows from scanning a designated range of upper memory. Specify the lower and upper boundaries of the region you want to exclude, in hexadecimal notation, with a hyphen separating the two addresses. Windows rounds your lower boundary down and your upper boundary up, if necessary, so that the region is an even multiple of 4K. This line is similar to EMMExclude (described previously), except that it allows you to exclude smaller areas of memory.

ReservePageFrame=Off | On (default: On) Tells Windows whether to give preference to the expanded-memory page frame or to translation buffers, if insufficient UMB space for both is available. If this switch is On and Windows can't find enough room in upper memory for both the EMS page frame and the translation buffers, Windows locates the page frame in upper memory (if a contiguous 64K space is available) and puts the translation buffers in conventional memory, thereby reducing the amount of conventional memory available to all virtual machines.

If you never use expanded memory in your DOS applications, it's a good idea to change this setting to Off.

TokenRingSearch=Off | On (default: On) Determines whether Windows searches for a Token-Ring network adapter on AT-architecture machines. Turn this switch off if you're not using a Token-Ring network and the search interferes with another device.

UsableHighArea=*paragraph range* (default: none) Explicitly instructs Windows to scan the designated upper-memory range for free UMBs, overriding any overlapping area specified in an EMMExclude line. You can use this line in conjunction with EMMExclude to locate the source of a UMB conflict.

Windows rounds the specified area, if necessary, to cover an even multiple of 4K. To specify a larger region, use EMMInclude.

Windows will ignore a UsableHighArea range if it overlaps with a range excluded by means of EMM386's X= parameter.

UseROMFont=Off | On (default: On) Determines whether Windows uses the soft font in video ROM to display text in full-screen non-Windows programs, as well as text used in messages that appear when you switch away from non-Windows programs. You should set this switch to Off if you see random dots and shapes on your screen.

According to Microsoft, you should also turn this switch off if you use the VGASwap option with Qualitas's 386Max or BlueMax.

VGAMonoText=Off | On (default: On) If your applications don't use the monochrome display mode of the VGA adapter, you should set this switch Off. Doing so allows Windows access to an additional upper-memory block, B000 to B7FF. This switch has no effect with systems that don't use a VGA display.

VirtualHDIrq=Off | On (default: On for AT-compatible systems, Off for all others) If On, allows Windows to virtualize the hard-disk interrupt line, bypassing the ROM BIOS. Virtualization provides better performance but is incompatible with certain hard-disk controllers, as well as certain software products.

Among the products requiring VirtualHDIrq to be set to Off, according to Microsoft, are the Super PC-Kwik cache utility, Borland's Reflex, the Toshiba HARDRAM, various games from Broderbund, Plus Development's HardCard Plus 80 II, and some ESDI hard-disk controllers.

WindowKBRequired=*kilobytes* (default: 256) Specifies the amount of conventional memory that must be free for Windows to start. If you have trouble starting Windows, try reducing this value.

WindowMemSize=-1 | *kilobytes* (default: -1) Specifies the amount of conventional memory Windows may use for itself. The default, -1, allows Windows as much conventional memory as it needs.

If you get insufficient-memory messages when you try to run virtual machines on a 2Mb system, try setting WindowMemSize to a value less than 640. (Alternatively, use PIFs and set smaller values for Kb Required.)

WinExclusive=0 | 1 (default: 0) Allows you to prevent non-Windows applications running in 386 Enhanced mode from operating in the background. With this line set to 1, Windows suspends all non-Windows programs whenever a Windows program is running in the foreground. To change this line, open the 386 Enhanced section of Control Panel and set or clear the Exclusive in Foreground check box.

WinTimeSlice=*foreground, background* (default: 100,50)
Determines the priority given to the Windows virtual machine relative to the non-Windows virtual machine(s) when the Windows virtual machine is running in the foreground and background, respectively. (When both Windows and non-Windows applications are open in 386 Enhanced mode, all the Windows applications together run in one virtual machine, while each non-Windows program runs in its own separate virtual machine.)

Unlike the MinTimeSlice line, whose parameter is measured in milliseconds, WinTimeSlice specifies relative values only. For example, suppose the WinTimeSlice values are 100 and 50 (their defaults) and no DOS programs are running. In this case, the Windows virtual machine gets 100 percent of the processor time, since there are no non-Windows virtual machines.

Now suppose you start two non-Windows programs whose own priority values are also set to 100 and 50. (These values are determined by the application's PIF. If you don't run the program from a PIF, Windows uses the values set in _DEFAULT.PIF, which are 100 and 50 unless you modify them.) While the Windows virtual machine is in the foreground, it will get approximately half of the processing time, as shown by the following expression:

or, in this case,

While the first non-Windows program is running in the foreground, all your Windows applications together will get about one-third of the processing time, as shown by this expression:

or

If you leave your DOS applications' priorities set at 100 and 50 (that is, if you don't specify other values in _DEFAULT.PIF or in any of your applications' PIFs), you can reduce the drag that non-Windows programs running in the background exert on your foreground Windows programs by increasing the *foreground* parameter on the WinTimeSlice line. To do that, open the 386 Enhanced section of Control Panel and adjust the Windows in Foreground value upward. Values from 1 to 10000 are accepted. Similarly, if you find the foreground performance of your non-Windows programs unsatisfactory, you can reduce the WinTimeSlice line's *background* priority by adjusting the Windows in Background value in Control Panel. (To boost the performance of a particular non-Windows program, however, you may find it simpler to increase its foreground priority-or just to run it in Exclusive mode. You can do either of those things by modifying the applicationÕs PIF.)

[Standard]
The [Standard] area of SYSTEM.INI defines the parameters Windows uses when running in Standard mode.

FasterModeSwitch=0 | 1 (default: 0) Allows most 286 computers to switch more quickly between Protected mode and Real mode, thereby improving throughput for Windows running in Standard mode. (This line has no effect on 386 or 486 computers.) The default is 0 because a value of 1 may cause certain early-model IBM PC-ATs and compatibles to hang when Windows starts. You can improve performance on most 286 systems by changing FasterModeSwitch to 1. (Add this line to the [Standard] section if it is not already there.)

You should definitely try setting FasterModeSwitch to 1 if you are losing characters while typing on a Zenith Z-248, losing control of the mouse on an Olivetti M-250-E, or losing characters or experiencing slow performance with Winfax Pro 2.0 on a 286 computer.

Int28Filter=*number* **(default: 10)** In Standard mode, this line determines the percentage of Int 28h interrupts that are made visible to applications that were loaded before Windows. With the default setting, Windows makes every tenth interrupt visible. Setting this value higher may improve performance but may also cause problems with certain memory-resident programs.

MouseSyncTime=*milliseconds* **(default: 500)** Determines the number of seconds that can elapse between mouse data bytes before Windows assumes a data packet is complete. (This line only affects systems with an IBM PS/2 mouse interface running in Standard mode.)

NetAsynchSwitch=0 | 1 (default: 0, unless an application is running that supports the use of the Task Switcher API by NetBIOS) Determines, in Standard mode, whether you are allowed to switch away from an application after that application has made an asynchronous NetBIOS call. Because switching away from some applications under these circumstances can cause your system to fail, Windows prevents switching by default.

NetHeapSize=*kilobytes* **(default: 8)** Determines the size of the conventional-memory buffer that Windows in Standard mode uses for transferring data over a network. If an application is not running correctly, your network may need a larger buffer than Windows' default. Increasing this value decreases available conventional memory but may be necessary for some networks.

Stacks=*number* **(default: 12)** Determines the number of reflector stacks DOSX.EXE uses to map DOS or BIOS interrupts from Real mode to Protected mode.
If you get a "Standard Mode: Stack Overflow" error message, try increasing this value. Acceptable values range from 8 to 64.

[NonWindowsApp]

Windows requires some basic operating information that it can use to run DOS applications. This information focuses primarily on fundamentals for running DOS itself, operating mice in DOS, and other such housekeeping items.

CommandEnvSize=*bytes* **(default: 0, with DOS 3.0 or 3.1; with DOS 3.2 or later, the default matches the /e: parameter of the SHELL statement in your CONFIG.SYS, if any)** Specifies the size of your DOS environment space. This line can help you out if you get a "not enough environment space" message while trying to run a non-Windows program under Windows. When you launch a DOS application in Windows' 386 Enhanced mode, Windows creates a virtual machine and gives it a copy of DOS with the same amount of environment space as you allocated in your CONFIG.SYS file. The default is 160 bytes or whatever you may have specified with a statement such as this:

```
    SHELL=C:\DOS\COMMAND.COM /E:1024
```

        You can increase the environment space for all non-Windows
applications by adding a CommandEnvSize= line to the [NonWindowsApp]
section of SYSTEM.INI. To allocate 2K, for example, enter the line

```
 CommandEnvSize=2048
```

        You may also use PIFs to increase the environment space for
particular non-Windows applications. For example, to run the program
WHIZBANG.EXE in a virtual machine with 2,048 bytes of environment space,
you could create a PIF with the following entries:

```
 Program Name: COMMAND.COM
 Program Title: Whizbang
 Program Parameters: /E:2048 /C C:\WHIZ\WHIZBANG.EXE
 Startup Directory: C:\WHIZ
```

If the environment space specified via a PIF is larger than that given
on the CommandEnvSize= line, Windows allocates the larger amount-but
only for the virtual machine launched by that PIF.

DisablePositionSave=0 | 1 (default: 0) Records the window's
position, whether it's maximized, and the font it uses, when you quit a
non-Windows application running in a window in 386 Enhanced mode. This
information is stored in an initialization file called DOSAPP.INI. You
can ask Windows not to record this information by choosing the Fonts
command on the application's Control menu (while the app is in a
windowed display) and clearing the check box labeled Save Settings on
Exit.
        If you change the DisablePositionSave line in SYSTEM.INI from 0
(its default) to 1, Windows never saves this position and font
information. You may need to add the line to SYSTEM.INI to do this. If
this line is omitted, Windows assumes its parameter is 0.

**FontChangeEnable=0 | 1 (default: 1 on systems that use
Windows 3.1 grabbers; 0 on systems that use Windows 3.0
grabbers)** Lets you change the display font used for a non-Windows
application running in a window in 386 Enhanced mode. If you are unable
to do this, it may be because you are using a display driver and grabber
file designed for Windows 3.0. (The grabber is a file that supports the
exchange of data in video memory between non-Windows applications and
Windows 3.1.) You may be able to rectify the problem by entering
FontChangeEnable=1 in the [NonWindowsApp] section of your SYSTEM.INI
file. This fix does not work on all display systems, however, and it may
cause your mouse pointer to be displayed incorrectly. If it doesn't
work, reset FontChangeEnable to 0.

LocalTSRs=*TSRname1,TSRname2*... (default: dosedit, ced, pced)
Specifies a list of TSR (terminate-and-stay-resident) programs that may
be copied into each running virtual machine. Certain TSRs do not
function correctly in virtual machines if they have been loaded before
Windows starts. DOSEDIT, for example, assumes that there is only one
keyboard buffer, not a separate buffer in each virtual machine. If a TSR
that is in memory when you start Windows is included on the LocalTSRs=

line, Windows puts a copy of it in each virtual machine you create. Note, however, that some TSR programs do not work if copied into virtual machines in this manner.

MouseInDosBox=0 | 1 (default: 1 if supported mouse driver is loaded; otherwise, 0) In Windows 3.1, when you run a non-Windows application in a window in 386 Enhanced mode, mouse events are normally passed through to the application. (This is true for most mouse drivers that have the extension .COM or .SYS, provided that your display system uses a Windows 3.1 grabber file.) If you want to select text and copy it to the Clipboard, you must first use the Edit and Mark commands on your application's Control menu. If you prefer to be able to select text directly (without going to the Control menu) and don't mind forgoing your application's own mouse interface, edit the SYSTEM.INI file to change the MouseInDosBox line from 1 to 0.

Screenlines=*number* (default: 25) Specifies the number of lines displayed on a non-Windows application running full-screen. The default is 25. Provided your display system permits, you may select a 43- or 50-line display by editing the Screenlines line in SYSTEM.INI. This setting may be overridden by the application itself, however.

Swapdisk=*drive:\directory* (default: the directory pointed to by your DOS TEMP environment variable; if no TEMP variable, the root directory of your first hard disk) When you switch away from a non-Windows application in Standard mode, Windows swaps that program's data to disk, storing it in the directory indicated by the Swapdisk= line. To maximize the efficiency of this process, you can set a DOS environment variable named TEMP to point to your fastest and least-fragmented hard disk. Or you may edit SYSTEM.INI to enter a particular drive and directory on the Swapdisk= line.

   If you run Windows in Standard mode and use a RAM disk, you might want to point your TEMP variable to the RAM disk and set Swapdisk= to some other drive and directory. That way, applications that put temporary files in the TEMP directory will be able to use the RAM disk, but you won't be trying to swap entire applications in and out of a RAM drive. (If you point Swapdisk= to a RAM disk that isn't large enough, Windows may force you to reboot.)

   The Swapdisk= line has no effect in 386 Enhanced mode.