

# **EXPERT MANAGEMENT FOR DOS UNDER WINDOWS**

A KEY TO THE STUNNING SUCCESS OF WINDOWS SINCE THE RELEASE OF VERSION 3.0 has been the choices offered to users: You can use Windows (either 3.0 or 3.1) as an entirely new graphical computing environment, complete with sophisticated new Windows applications, or you can use it as a splendid graphical memory manager and program launcher for familiar DOS applications.

This duality has made it much easier for people to begin experimenting with Windows, and has given users a flexibility they not only crave, but require. You can use Windows without making any changes in the applications that handle your critical computing chores. As time goes on, you can migrate smoothly from DOS applications to Windows applications when new programs offer enough significant new benefits to make the change seem worth your while.

Windows' dual nature has been a boon, no question. But it means that DOS remains a significant element in the lives of many, if not most, Windows users.

While Windows does a splendid job of handling DOS applications and providing DOS services to users, it's not perfect. How could it be? This is, after all, gnarly, nasty, difficult old DOS we're talking about. DOS is a tough environment on users, and on system software such as Windows as well.

Friction along the boundaries between Windows and DOS creates more confusion and consternation than any other aspect of the new environment, so this chapter offers dozens of effective secrets for taming DOS on your Windows PC, including these:

Tips for installing and launching DOS applications under Windows, with information on how Windows' modes affect DOS program performance

The ins and outs of Program Information Files (PIFs), which control every aspect of DOS program operation; includes a guide to troubleshooting common problems

Advice on establishing effective DOS sessions, with details on how to optimize your AUTOEXEC.BAT and CONFIG.SYS files and how to interpret some common error messages

Step-by-step instructions on copying and pasting between DOS and Windows applications

Lots of shortcuts and ingenious workarounds for managing DOS sessions under Windows

We even show how to set up your system so that you know whether you're in native DOS or you're running DOS under Windows, a confusion that's a potential cause of fatal errors such as crashing the system by running DOS commands forbidden under Windows.

## **Installing and Launching DOS Apps under Windows**

The proper handling of DOS applications under Windows is somewhat akin to the correct method for handling rare poisonous snakes-take care and beware. Although adding DOS applications to Windows is simple, the day-to-day practice of actually getting DOS programs to work well with Windows without crashing the system can be vexing. We recommend a step-by-step approach to DOS application management. To begin, let Windows do the work. If things don't go smoothly, get more involved yourself.

### **Ask Windows Setup to Search Out DOS Apps**

Your first opportunity to bring DOS programs into Windows occurs when you install Windows with the Setup program. The Setup program asks if you want it to search your hard disk for programs

that have already been installed on your computer. If you answer Yes, Setup looks for the more than 80 DOS applications-including all the chart-leading favorites-that it recognizes, plus any Windows applications. It then lists the programs it finds and asks you to select which ones to install in Program Manager (see Figure 4.1). To add DOS applications, click on the application name in the list on the left and then click on the Add button.

Don't be alarmed if Setup "finds" applications that aren't on your disk. All that means is that one of your programs has the same executable filename as the one Setup thought it found. For instance, the name MP.COM is used by both MousePerfect, a mouse driver for WordPerfect, and Multiplan, Microsoft's spreadsheet. When Setup finds MP.COM, it assumes that it has located Multiplan. Also, the setup routine may identify arcane DOS utilities that are of little value to most users, such as LINK or BASICA. If you don't use them regularly, don't install them in Program Manager-they'll just create clutter. You can easily add them later if necessary.

Once you've indicated your choices, Setup creates two program groups, one called Windows Applications and one called Non-Windows Applications. Then it generates a program item for each application you've selected. In addition, Setup creates a PIF for each DOS application that it installs. The PIF tells Program Manager where to find the application (that is, in which DOS subdirectory on your hard disk it is stored), what the application's executable file is called, and other data relating to the application, including the minimum amount of RAM that must be available before the program can be launched. (For more on PIFs, see the section "PIFs: DOS Program Control Centers" later in this chapter.)

## **Installing DOS Programs after Setup**

You can install more DOS programs at any time after the initial setup. Just select the New command on the Program Manager's File menu and fill in the two-line dialog box with the title you want for the program item and the location and filename of the executable file (or the PIF, if you've created one) needed to run it. Your program will appear with the default DOS icon supplied by Windows, or you can choose a nicer-looking icon, as described in Chapter 5.

To change the icon, highlight it and choose Properties from the File menu. Select Change Icon, and a dialog box will appear. The filename for the icon source file is PROGMAN.EXE by default, meaning that the icons available in Program Manager are the default selection. If you're using Windows 3.1, try MORICONS.DLL, located in the WINDOWS subdirectory.

## **Six Ways to Launch DOS Applications from within Windows**

You can hang on to your beloved DOS applications even if you're using Windows, but stop wasting time exiting Windows to run them. There are at least six ways to run DOS apps from within Windows. Choose the one that works best for you:

Add an icon to launch your DOS application from the Program Manager. Select New from the Program Manager File menu, specify that you're creating a program item, supply the name you want to appear with the icon, and indicate the path and command you would normally use to execute the program.

Use File Manager to launch a DOS program by locating the subdirectory containing its executable file and double-clicking on the filename entry (see Figure 4.2).

If the program has a PIF file-either one you've created or one that Windows put together during Setup-locate it and double-click on the PIF's name in the File Manager.

Use the Run command on the File menu in Program Manager. When prompted you for the name of the program you want to use, type the name of your application's executable file (including its path) or the name of the PIF that launches it, and then press Enter.

Select the DOS prompt icon in Program Manager to get to the DOS command line, and then launch your application with the same commands you'd use outside of Windows.

Or, finally, use third-party alternatives to Program Manager, such as hDC's Power Launcher or Norton Desktop for Windows.

### **Launch a DOS Application with a Data File Loaded**

Windows can launch many DOS programs with a certain file already loaded into them. This capability is automatically enabled for most Windows programs when you install them, but for DOS applications you'll have to modify the WIN.INI file. The [Extensions] section of the WIN.INI file tells Windows what to do when someone tries to run data files with certain filename extensions. Part of that section looks like this:

```
[Extensions]
txt=notepad.exe ^.txt
```

All this little piece of WIN.INI says is that when you double-click on a .TXT file, Windows should launch Notepad and load that file. If you decide that you'd rather edit .TXT files in your favorite DOS editor—say, Norton's NED.EXE—replace the above line in WIN.INI with the following:

```
txt=ned.exe ^.txt
```

Keep in mind that this works only if the executable file is in your current DOS path. If it isn't, change the entry to read

```
txt=ned.pif ^.txt
```

Then create a PIF file for NED.EXE that tells Windows where to find the program.

You can also add new filename extensions to the list to launch other types of programs. For example, to add .WKS files that launch a DOS-based version of 1-2-3, put the following line in WIN.INI's [Extensions] list:

```
wks=123.exe ^.wks
```

WIN.INI is an ordinary text file, so you can use any text editor, including Windows' own Notepad, to modify it.

### **Windows' Modes and DOS Apps**

DOS applications interact with Windows differently in each of Windows' modes. Real mode, which exists only in Windows 3.0, has become outdated. Since the introduction of Windows 3.1, Standard mode represents the common way Windows is used on 286 and low-powered 386 machines. 386 Enhanced mode brings DOS applications much closer to the graphical, shared-screen, multitasking world of Windows by taking advantage of the power of the 80386 microprocessor.

**Windows 3.0 Real Mode Isn't Great for DOS Applications** Here's a great reason to upgrade to Windows 3.1, if you haven't already: Don't expect much when running your DOS favorites in Windows 3.0's Real mode. DOS applications in Real mode have access to the same 640K of RAM that's available to them under DOS. Unfortunately, Windows itself consumes 60 to 75K of conventional DOS memory when running in Real mode. Only one program can run at a time, whether it's a DOS or a Windows application, and switching from one program to another requires moving data from RAM to disk and back. In fact, in Real mode, Windows 3.0 (and any DOS applications running under it) are hobbled by many of the same performance and memory limitations that plagued earlier versions of Windows.

Standard mode in either Windows 3.0 or 3.1 is a more realistic environment for task-switching among DOS applications. (Task switching means switching from one application to another without having to exit one program and start up the next.)

### **Run DOS Applications in Standard Mode for Faster**

**Performance** On some occasions, you might actually want to run a lesser Windows mode than your machine can handle. Owners of 386 PCs who are willing to give up 386 Enhanced mode's special multitasking capabilities (see the following tip) will find that Windows applications run 10 to 15 percent faster in Standard mode than in 386 Enhanced mode.

Task-switching in Standard mode is simple enough. Launch one program, and then use it in the foreground for a while. When you're done, minimize its window, and then launch another application. To reactivate the first program, double-click on its icon to bring it into the foreground. You can also use Windows keyboard commands to switch between programs. Pressing Alt+Esc or Alt+Tab switches you from one loaded application to another, and pressing Ctrl+Esc brings up the Windows Task List, a list of loaded applications from which you can choose.

Unfortunately, although Standard mode allows Windows and Windows applications to access all the memory installed on the PC (up to 16Mb), DOS applications are still restricted to the same 640K limitation found in Real mode. Switching among loaded applications is a fairly rapid process, but when memory becomes overloaded, inactive programs go to disk, slowing your work considerably.

### **Run DOS Applications in 386 Enhanced Mode for True**

**Multitasking** 386 Enhanced mode is far and away the most versatile alternative for running DOS applications under Windows. It adds both a complete, disk-based virtual memory manager and background processing to Standard mode's task-switching capabilities. Its superior memory management means that DOS applications use about 10K less memory than they would running on the same machine using standard DOS.

In 386 Enhanced mode, one or more DOS or Windows programs can run in the background while another DOS or Windows application runs in the foreground. Therefore, your PC can execute a large database sort and conduct a file transfer in the background while you continue to work at word processing or some other task in the foreground. (There will be some compromise in performance, of course.)

When you're in Windows and have a DOS application loaded, the application is represented by a DOS icon at the bottom of the screen. You can reactivate the DOS program that the icon represents by double-clicking on it, or you can use the Alt+Tab or Alt+Esc keyboard commands to switch between the DOS program and other programs you have running.

### **Toggle Enhanced Mode Applications between Windowed and**

**Full-Screen Size** In 386 Enhanced mode, you can run DOS applications either full-screen size or in a resizable window. Press Alt+Enter to toggle between these two modes. If you want your DOS app to start up in a window by default, go to the program's PIF (or create a PIF if there isn't one already, as is described in the next section of this chapter), and select Windowed as the Display Usage option.

### **Mouse Tricks**

Many DOS applications were developed BTM (before the mouse). Back then men were men, women were women, and long keyboard commands were considered sexy. No longer. Today, women are strong, men are sensitive, and application services are expected to be just a mouse click away. Windows offers methods savvy users can employ to bring mouse power to old DOS programs. With them you can install or launch a DOS application with a quick click.

**Use the Mouse in DOS Applications in Windows 3.1** In Windows 3.1, you can use a mouse in DOS applications whether they're running full-screen or in a window. But you need to load a DOS mouse driver into memory to do so.

Make sure you're using the most recent version of MOUSE.COM, the Microsoft mouse driver for DOS. Version 8.2 of MOUSE.COM is included with Windows 3.1, but is not automatically installed during setup.

If you retrieve the driver from the Windows disks (it's on Disk 4), remember that you need to expand it before you can use it. To do this, copy EXPAND.EXE from Windows 3.1 Disk 3 to your \WINDOWS directory. Then, assuming Windows Disk 4 is in drive A, type

```
EXPAND A: MOUSE.CO_ MOUSE.COM
```

EXPAND will create a usable version of the driver in your \WINDOWS directory.

**Load the DOS Mouse Driver from a Batch File** If you won't always need the mouse in DOS applications, or if only one specific DOS application calls for it, load the driver from a batch file. You can create a PIF for a batch file that loads just the mouse driver, or, in the case of a single mouse-supported DOS application, loads the driver and application in one batch file.

## **PIFs: DOS Program Control Centers**

Before Windows, DOS programs never had to describe themselves to their PC's system or to any other running applications. When operating, a DOS program was its PC's system and there were no other applications running. To integrate these recalcitrant DOS applications into Windows sessions, you must place them in an envelope of descriptive information. This envelope is called a Program Information File, or PIF. Windows provides the framework for building PIFs in a utility called the PIF Editor, which resides in the Program Manager's Main program group. Using it wisely can make DOS applications as "windowable" and as cooperative with other programs as their natures allow.

### **Check for Predefined PIFs**

The Windows APPS.INF file (in Windows 3.1) and the SETUP.INF file (in Windows 3.0) each contain PIF settings for many popular DOS applications; Windows uses these files during setup if you tell it to look for your existing applications. If one of your DOS applications is not listed in these files, check with the DOS application's manufacturer. They may have a PIF available for your application (many manufacturers' bulletin boards will provide these). If not, it may have information on the PIF settings that you should use. Finding this information will save you time trying to fine-tune the performance of your favorite DOS application.

If you didn't let Windows set up PIFs for your DOS applications during the initial setup, it's not too late to do so now. In Program Manager, choose the Setup icon (located in the Main program group) and choose Set Up Applications from the Options menu. Setup will search your disk for all the Windows and DOS applications it recognizes and then give you the option of installing any or all of them. If you want to check whether APPS.INF or SETUP.INF contains a PIF for your DOS application before you rerun Setup, you can open the .INI file in a text editor such as Notepad. These .INF files are located in Windows' SYSTEM subdirectory. If your DOS application isn't listed in the [pif] section, don't waste time running Setup.

### **How to Write the Perfect PIF**

Program Information Files are small settings files that tell Windows where to find and how to load non-Windows applications. When you select Install Applications during Windows setup, Windows creates PIFs for all the non-Windows applications it recognizes on your disk. Here's how to write perfect PIFs for the DOS applications Setup misses:

1. Open the Windows PIF Editor. (It's found in the Main program group in Windows 3.1 and in the Accessories group in Windows 3.0.)
2. Enter the path and filename for your application's executable file on line 1.
3. Enter a title to appear with your application's icon on line 2.
4. On line 3, enter any command-line parameters you'd like to pass to the application.

5. On line 4, enter the directory where the application's executable file is stored.

6. Save the PIF. Change nothing else.

That's it. You're done. You've just created the perfect PIF, or something close enough to perfect that you'll never know the difference. This holds true for about 90 percent of DOS-based applications.

### **Make PIFs for Both Standard and 386 Enhanced Mode**

When you're setting up a PIF file for a DOS application, the available configuration options depend on which Windows mode you're currently running. To have access to all the PIF options available in Standard and 386 Enhanced mode, first go into the Windows PIF Editor (in the Main group in Windows 3.1 and in the Accessories group in Windows 3.0) and fill in all the currently available options. When you're done, select the name of the mode that isn't already checked from the Mode menu at the top of the PIF Editor dialog box. For example, if you're currently running Windows in 386 Enhanced mode, a check mark will appear next to the item 386 Enhanced; click on Standard. A message will tell you that you are not currently running in this mode and that PIF options may not be appropriate; you'll be asked if you want to continue. Choose Yes to display a new PIF dialog box with other options for you to configure. The options that are identical in both modes will already be filled in, based on the options you selected in the first PIF box. See Figure 4.3.

### **Make Two PIFs for the Same Application**

For DOS applications you use frequently, consider making two separate PIFs (and program icons) that allow you to specify different starting directories and memory options. For example, this would come in handy with a DOS spreadsheet program that you use for both work and personal purposes. Start the PIF Editor and enter the program information for the first set of options. Choose Save from the File menu, giving the PIF a descriptive name that will distinguish it from the second one. Now select New from the PIF Editor's File menu and fill in the information for the second set of options, treating it as if it were a separate application (the Program Filename box will contain the same executable file as the first PIF). In Program Manager, add the icons for each PIF to the desired program groups, naming each appropriately so that you'll be able to easily distinguish between the two.

### **Make PIF Settings Uniform**

If everyone at your workplace runs the same DOS applications under Windows, you might want to fine-tune the PIFs for each application and distribute them for everyone's use, especially if the applications have been known to cause problems. This way everyone will be running DOS applications optimized for Windows performance. Likewise, if your company develops its own DOS applications, it's a good idea to add PIFs with the correct settings for running them under Windows.

### **Protect Shortcut Key Sequences in DOS Apps**

If one of your DOS applications uses some of the same shortcut key sequences as Windows itself (like Alt+Tab, Alt+Esc, and Ctrl+Esc), you can reserve these keys for use only by the application by changing the PIF settings. The Standard mode PIF dialog box includes five keyboard combinations that you can reserve for your DOS application; the 386 Enhanced mode Advanced Options dialog box contains seven key combinations that you can reserve for your DOS application. Note that reserving these keys only affects that particular DOS application. Once you return to Windows, the key combinations will revert to their Windows actions.

### **Recover Original PIFs**

If you've modified a PIF file that Windows installed for you at setup, and you then decide to restore the original settings, all is not lost. Simply rerun Windows Setup from the Main program group and select Set Up Applications from the Options menu. You can then reinstall the original PIF over the one you've modified.

## **PIF Troubleshooting**

Any DOS application that doesn't work with the standard PIF settings (see the earlier tip "How to Write the Perfect PIF") will do one of three things: Refuse to load when you use the PIF, crash, or crash Windows. In any of these events, turn to these troubleshooting tips.

And remember, if you're having trouble determining the correct PIF settings for a DOS application, use context-sensitive help. Pressing F1 brings up help for setting each section of the PIF, as shown in Figure 4.4.

**Not Enough Memory** Does your program start to load, and then flash a "Not enough memory" message on the screen? You can increase the amount of memory available to your application by shedding a few TSRs or device drivers. Standard mode users can reserve a little more memory for their applications by checking the No Screen Exchange and Prevent Program Switch options in the PIF Editor dialog box. However, this prevents you from switching from your program while it's operating and from copying the contents of its screen to the Windows Clipboard.

**Not Enough Memory for High-Resolution DOS Graphics in 386 Enhanced Mode** Do you receive an out-of-memory error when running a high-resolution DOS graphics application? Check the program's PIF file: Under Video Memory, select High Graphics, and in the Advanced Options dialog box select Retain Video Memory from Display Options. Also select the Full Screen option.

**Maintaining Access to Serial Ports** Does your program use one of your PC's serial ports? If you're running in Standard mode, check the appropriate Directly Modifies box in the PIF Editor dialog box for the COM port that it uses. Note, however, that doing so will prevent you from minimizing the program while it's running. Once you've checked any of the Directly Modifies boxes, you'll have to quit your program before you can return to Windows.

**Mouse and Keyboard Problems** Does your mouse no longer work with your program? Write a batch file that loads your mouse driver and then the program, and change your PIF to run the batch file instead of running your program directly.

Do the shortcut key combinations (such as Alt+Tab and Ctrl+Enter) used by your program no longer work? Check the appropriate combinations in the Reserve Shortcut Keys section of the PIF Editor dialog box. If you're running in Standard mode and the special key sequences in your program still don't work, try selecting the Directly Modifies Keyboard check box in the PIF Editor dialog box. (Be aware, however, that this last step will prevent you from switching from your application while it's loaded.)

**Putting Data Files Where You Want Them** Do you want to save files created by your DOS program to a directory other than the one in which your application's executable file is stored? Although the first line of the PIF will include the full path of your application's executable file (as in C:\TEXT\TEXTEDIT.EXE), change the Startup directory in line 4 to be the path in which your data files can be found (C:\TEXT\DATAFILS).

**Clearing Up a Garbled Display** Is your application's screen garbled or incorrectly restored when you switch back to it after it has been iconized? Select the Graphics/Multiple Text option on the Video Mode check box in the PIF Editor dialog box if you're running in Standard mode. (Careful-this will reduce the amount of RAM available to your application.) If you're running in 386 Enhanced mode, click on Advanced Options in the PIF Editor dialog box, and then change

the Monitor Ports and Video Memory options one by one until the problem is solved. Also make sure the Monitor Ports options is checked and that Emulate Text Mode is not checked.

**Lost or Garbled DOS Data** Does your DOS data become garbled or lost when you're running a DOS application under Windows? Check the Directly Modifies options in the PIF Editor. If the options for the keyboard and COM ports are checked, the DOS application has exclusive control over these resources. Another application cannot simultaneously use the same serial port, and Windows cannot switch from the application. You can also select No Screen Exchange to prevent Windows from updating the screen when you switch back to the DOS application.

**Expanded Memory** Does your DOS application require EMS memory that Windows isn't providing? Standard mode users should first try changing the XMS KB Required setting in the program's PIF to -1 (which will allot any EMS memory you've installed to your application, as needed), and then to a specific number (such as 500 for 500K) if that doesn't work. If you want to reserve XMS memory for other applications, enter an upper limit for the EMS RAM your application should be allowed to access in the KB Limit box. In 386 Enhanced mode, Windows automatically provides EMS and XMS memory to applications. However, you can adjust both the Required and Limit settings in the PIF Editor if your application is not operating correctly.

**Background Execution and Windowed Display** Some applications may crash if suspended while another program becomes active. If your application must continue execution while other applications are running in the foreground, check the Background Execution check box in the PIF Editor dialog box.

Do you want your application to load in a window, not full-screen? Select Windowed for the Display Usage option.

**Pasting Data to and from the Clipboard** Does your application hang or beep loudly when you paste data into it from the Windows Clipboard? Or are some of the characters not showing up? Deselect the Allow Fast Paste check box in the PIF Editor's Advanced Options dialog box.

**Pasting Graphics to and from the Clipboard** Are you unable to paste graphic screens from DOS applications to the Clipboard? If you're running in Standard mode, make sure that the No Screen Exchange option in the program's PIF file is not checked. This option prevents Windows from using PrtSc and Alt+PrtSc in DOS applications to paste to the Clipboard. Also check the Reserve Shortcut Keys settings. Make sure that PrtSc and Alt+PrtSc are not checked as reserved keys by the DOS application.

**Task Switching** Are you unable to switch from a DOS application into Windows? Check the application's PIF file to make sure that Prevent Program Switch isn't checked. If this item is checked, you won't be able to go back to Windows until the DOS session has ended. Also, if you've checked any of the Directly Modifies boxes, task-switching from your DOS application will be disabled; that is, you'll have to quit your DOS program before you can return to Windows.

**Running Multiple DOS Applications** Can you run DOS applications under Windows individually, but not run two or more DOS programs at the same time? You may have a problem with a conflicting device driver or TSR. One of the DOS applications may have a driver or TSR that requires expanded or extended memory, but the other DOS application may have the EMS Memory Locked or XMS Memory Locked option selected in its PIF file. If so, and if you want to run these DOS applications at the same time, turn off these memory options in the PIF.

**DOS Communications Programs** If you have problems keeping a modem connection while running a DOS communications program and trying to switch back to Windows, first consider if you're running in Standard mode. Standard mode doesn't allow DOS applications to



run in the background; any DOS program is put on hold when you switch to Windows and the connection is terminated.

If you're operating in 386 Enhanced mode, try making these changes to the communications' program PIF: Under Execution, select the Background check box, and in the Advanced Options dialog box, make sure that the Detect Idle Time check box is not checked in the Multitasking Options section.

**Last Resorts** If you try these adjustments and your application still doesn't work well under Windows, you may have a serious problem on your hands. Try calling Microsoft's Windows technical support line: (206) 637-7098. Or try calling the application program's developers, because they've no doubt already heard from other frustrated Windows users and may either have suggestions on how to make it work or may be working on an updated version that solves the problem. Finally, if you have access to CompuServe, check the Windows forums-the Windows New Users Forum (Go WINNEW) and the Windows Advanced Users forum (Go WINADV)-or any forum specific to your application to find out whether other users have encountered (and solved) the problem.

## **Establishing Effective DOS Sessions**

DOS doesn't go away when Windows is running; it goes underground. The entire DOS structure becomes a substructure beneath Windows, just a click on the DOS prompt button away from reasserting itself. This means that you must take care to set up DOS properly if you want to get the most from Windows. This is particularly true if you are running DOS applications, which may look like they are part of your Windows screen groups, but actually operate in their traditional DOS sessions. Setting up the proper parameters in the DOS control files, such as CONFIG.SYS and AUTOEXEC.BAT, is essential to getting the most out of Windows.

### **Optimizing CONFIG.SYS and AUTOEXEC.BAT**

Here are some straightforward rules of thumb to consider when setting up your DOS sessions under Windows. In total, they represent something of a refresher course in DOS basics. That may seem out of place in a Windows book, but it's not. When a contractor builds a house, even if it's his 10,000th house, he very carefully measures every angle, moving slowly, and sets the cornerstone with care. The same is true for DOS sessions in Windows. You may know all this material, but review it anyway to make sure you are setting your Windows cornerstone properly.

**Upgrade to DOS 5.0** Upgrade to DOS 5.0, and then load it into the high memory area on 386 PCs by making sure HIMEM.SYS and EMM386.SYS are loaded and adding the line DOS=HIGH or DOS=HIGH,UMB to your CONFIG.SYS file just after the lines that load these files. (HIMEM and EMM386 are discussed fully later in this chapter.) If you specify DOS=HIGH,UMB you are also telling DOS to utilize any available sections of reserved memory (called upper memory blocks). Loading DOS high or in upper memory will give you more conventional memory for your DOS applications running under Windows. This procedure leaves only about 19K of DOS code in conventional memory.

**Check Your Memory with DOS 5.0's MEM Command** Before you start experimenting with loading drivers, programs, and TSRs into high memory by tinkering with CONFIG.SYS, run the MEM command with the /C switch to see just which programs would benefit most. The /C switch is new to DOS 5.0 and gives you a detailed rundown of how much conventional memory drivers and TSRs are using. If certain programs are taking up too much memory, you can place them in the upper memory blocks by following the steps outlined later in this chapter or using the LOADHIGH and DEVICEHIGH commands.

**Load HIMEM.SYS First** Make sure that the line of your CONFIG.SYS file that loads HIMEM.SYS comes before any other commands that load applications or drivers into extended memory.

**Set FILES=30 in CONFIG.SYS** Set FILES=30 in your CONFIG.SYS unless you have an application that requires a higher number. Applications that can take advantage of a higher number include databases, which use a large number of file handles.

**Set BUFFERS=10 in CONFIG.SYS** If you use Windows' SMARTDrive disk cache, set BUFFERS=10 in your CONFIG.SYS. If you don't use SMARTDrive, set it to 20. Using a higher number of buffers may improve disk access time, but it also uses up more conventional memory. (For more information on Windows's SMARTDrive disk cache, see Chapter 3.)

**Free Up Conventional Memory by Loading Device Drivers High** If you're using DOS 5.0, for each device you want to load into the upper memory blocks, your CONFIG.SYS must contain a DEVICEHIGH= command, as in

```
DEVICEHIGH=C:\DOS\RAMDRIVE.SYS
```

Likewise, for each TSR that you want to load into upper memory, your AUTOEXEC.BAT must contain a LOADHIGH command, as in

```
LOADHIGH C:\DOS\DOSKEY.COM
```

**If You Can't Load Programs in High Memory under DOS 5.0** To run programs in high memory, the following conditions must be met:

The PC must be a 386 or greater.

At least 350K of extended memory must be available.

CONFIG.SYS must contain the line DEVICE=HIMEM.SYS before any other DEVICE= or DEVICEHIGH= commands.

CONFIG.SYS must contain either a DOS=UMB or DOS=HIGH, UMB command.

**Troubleshooting a Specific Device Driver** If you cannot load a specific driver with the DEVICEHIGH command in DOS 5.0, check that the device driver itself works by loading it into conventional memory first. Next try changing the loading order of drivers and TSRs, keeping in mind that some programs require extra memory to load, often up to twice the amount of space they will actually use when resident.

**Load EMM386 to Access Expanded Memory** If you run DOS applications that require expanded memory, or if you want to access the upper memory blocks, load Windows' EMM386 memory manager in your CONFIG.SYS file. (The expanded memory manager is EMM386.EXE in Windows 3.1, EMM386.SYS in Windows 3.0. The following examples assume you're using Windows 3.1.)

Using the RAM switch with the line that loads EMM386.EXE, as in

```
DEVICE=EMM386.EXE RAM
```

tells the memory manager to simulate expanded memory and manage the upper memory blocks. On the other hand, using the NOEMS switch, as in

```
DEVICE=EMM386.EXE NOEMS
```

prevents access to expanded memory. If none of your DOS applications require expanded memory, use the NOEMS switch to gain an extra 64K in the upper memory blocks that would otherwise be reserved for the expanded memory page frame.

**Troubleshooting EMM386 Error** If you try to run Windows in 386 Enhanced mode and get the error message "Unable to start Enhanced Mode Windows due to invalid path specification for EMM386," here's what's going on: When you start Windows in 386 Enhanced mode with DOS 5.0's EMM386.EXE driver loaded, Windows attempts to locate the driver. The error message you see is an indication that EMM386.EXE is not in the same location as when the system was started. (This usually happens when you boot from a floppy disk that contains EMM386.EXE and then remove the floppy before starting Windows.)

There are two ways to fix the problem:

If possible, modify your CONFIG.SYS so that EMM386.EXE is loaded from the hard disk. For example, if you currently boot from a floppy that contains the statement `DEVICE=EMM386.EXE NOEMS`, and there is a copy of EMM386.EXE in the DOS directory of drive C, you should change the statement to read

```
DEVICE=C:\DOS\EMM386.EXE NOEMS
```

Then, when Windows looks for EMM386.EXE, it will find it.

Use the EMM386.EXE driver's sparsely documented /Y switch to explicitly spell out the path you want to use for EMM386. For example, if it's imperative that you load EMM386.EXE from the floppy, you can still direct Windows to the copy of the driver in C:\DOS by modifying the CONFIG.SYS file to read

```
DEVICE=EMM386.EXE NOEMS /Y=C:\DOS\EMM386.EXE
```

This, too, will eliminate the error message.

**Load Only What You Need** In the AUTOEXEC.BAT and CONFIG.SYS files, load only the TSRs, drivers, and programs that you will use regularly. And conserve conventional memory by loading as much as possible into high memory and the upper memory blocks. You can also eliminate the DOS mouse driver altogether; if you have DOS applications that use a mouse, write batch files that load the mouse driver with the application.

**Use DOSKEY with Windows** To have access to all of the macros that you've created with DOSKEY, load DOSKEY and the macros you want to use *before* you start Windows. You can do this automatically by adding lines invoking your DOSKEY macros to your AUTOEXEC.BAT file *before* the line that loads Windows. This way, no matter how many DOS sessions you have running, you'll always be able to use your favorite DOSKEY macros.

**Optimize Memory Usage via PIFs** An addendum to the memory control possible through CONFIG.SYS is to optimize memory usage by allocating the expanded and extended memory required by DOS applications in a custom PIF. Put specific values in the EMS and XMS memory boxes in the PIF Editor to fine-tune the application's memory needs. By adjusting these settings you may be able to squeeze more applications into high memory by fitting them tightly into the available space.

**Streamline Your PATH Statement** Your PATH statement, located in your AUTOEXEC.BAT file, tells DOS where to look for command files. When you install a new DOS or Windows program on your PC, the setup routine often adds the program's directory to your PATH statement. If you have many applications installed, a good chunk of the environment space

allocated to the PATH statement is being used up (DOS allocates 127 bytes to the PATH statement). To make the most of this precious real estate, here are a few things you can do:

**Keep directory names short.** Your application will usually suggest a standard name for the default directory, but you can specify one that's much shorter. For example, instead of PROCOMM, you could opt for just PRO. If applications that you've already installed have long directory names, rename them in the Windows File Manager and change the name of the directory in the AUTOEXEC.BAT's PATH statement and the program's PIF. For Windows applications, you can rename any references in the WIN.INI file or the application's own INI file.

**Eliminate the drive letter.** If all of your applications are in a single drive (in most cases C), you can eliminate the drive letter that precedes every directory listing. For example, instead of

```
PATH=C:\DOS;C:\WINDOWS;C:\PROCOMM;C:\EXCEL,
```

your PATH could simply read

```
PATH=\DOS;\WINDOWS;\PROCOMM;\EXCEL
```

**Keep applications in branches off the root directory rather than in subdirectories that are more deeply nested.** When you install applications, make sure that you do so in a directory that branches directly from the root directory. That way, if the application's directory needs to be in the PATH statement, it will be shorter than if it were nested. For example, instead of installing PROCOMM as a subdirectory of the directory DOSAPPS, so that the PATH statement would have to include C:\DOSAPPS\PROCOMM, install it in just the PROCOMM directory so that the PATH would simply include C:\PROCOMM. If you already have applications installed in subdirectories, move them to new directories and change your PATH statement, PIFs, and INI files to reflect their new locations.

## **Troubleshooting Difficulties with DOS Apps**

As with many areas of Windows, you may feel you've done everything right in setting up your DOS applications, and the system will still give you fits. Here are some of the more common hidden problems to look for as you begin troubleshooting DOS session woes.

**Where to Put WINA20.EXE** When you install DOS 5.0 on your system, DOS Setup places a file called WINA20.386 in your root directory. If you move it out of the root directory and into a subdirectory, you may no longer be able to run Windows in 386 Enhanced mode. Windows aborts with the error message "You must have the file WINA20.386 in the root of your boot drive to run Windows in Enhanced Mode."

You can put the WINA20.386 file anywhere you wish, but you must let DOS and Windows know where it is by performing these two steps:

**1.**Add a SWITCHES=/ statement to CONFIG.SYS.

**2.**Add a DEVICE statement to the [386Enh] section of SYSTEM.INI naming the drive and directory where WINA20.386 is stored.

Suppose you want to move WINA20.386 to C:\SYSTEM. First, copy the WINA20.386 file to the SYSTEM subdirectory. Then use an ASCII text editor to add the statement SWITCHES=/W to your CONFIG.SYS so that DOS will know the file has been moved. Next, go to the directory where Windows is stored, call up the SYSTEM.INI file with your text editor, locate the [386Enh] section, and add the statement

```
DEVICE=C:\SYSTEM\WINA20.386
```

Save the SYSTEM.INI file and then reboot your system. Windows should run just fine.

### **Troubleshooting the "Unexpected MS-DOS Error #11" Message**

If you receive the "Unexpected MS-DOS Error #11" error message when you try to run a DOS application under Windows 3.1, Windows has tried to execute a file that has an invalid format and the Windows files that support non-Windows applications are corrupted. To fix the problem, copy the grabber file and either WINOLDAP.MOD (if this happens in Standard mode) or WINOA386.MOD (in Enhanced mode) from your original Windows disks. The grabber filename will differ depending on which Windows and video mode you are running in. For example, it's CGA.2GR or VGACOLOR.2GR in Standard mode, and EGA.3GR or VGA.3GR in Enhanced mode. All the grabber files are on Windows 3.1 Disk 1; WINOLDAP.MOD is on Disk 5, and WINOA386.MOD is on Disk 4.

You need to expand these files before you can use them. The files are in compressed form on the disks, with the last character of the uncompressed filename's extension replaced with an underscore. To expand the VGA.3G\_ file, for example, copy EXPAND.EXE from Windows 3.1 Disk 3 and type

```
EXPAND VGA.3G_ VGA.3GR
```

EXPAND will create usable versions of the files.

**How to Delete a DOS Application** If you want to delete a DOS application, one point to remember about program items is that they merely direct Windows to the location of your application. Nothing you do to a program item—including moving it, deleting it, or renaming it—affects the underlying program file. Conversely, Windows has no way of knowing if you change or move the program files, so it's up to you to keep program items up-to-date. Deleting a DOS application file doesn't change its Windows icon; double-clicking on the icon will simply return a "Program Not Found" error. Also you can still access the application through the File Manager. If you want to actually delete the application, you can do so through the File Manager.

### **If DOS Application Doesn't Start in the Correct Directory in Windows 3.1**

If you're running a DOS application from a PIF and it doesn't start in the directory you've specified, the Working Directory setting in the file's Properties settings may be different from the one you've specified. To check, highlight the program icon in Program Manager and choose Properties from the File menu. The Working Directory setting (shown in Figure 4.5) overrides the Start-up Directory setting in the application's PIF.

### **Troubleshooting Font Changes for a Windowed DOS Application**

If you're unable to change the font size for a DOS application running in a window, the application may be running in graphics mode. You can only change the font for a DOS application running in text mode. If the application is running in text mode and you still can't change the font, there may be a problem with the display driver. For an SVGA monitor, make sure that you are using the SVGA driver that came with Windows, because it supports font changes in windowed DOS applications. If the Windows driver doesn't work with your display and your driver doesn't seem to support font changes, try adding this line to your SYSTEM.INI file:

```
FontChangeEnable=1
```

Note that this setting may cause problems with displaying the mouse pointer.

## **Customizing Sessions Run from Windows' MS-DOS Prompt**

Even though you should be cautious in establishing DOS sessions running under Windows, you can still play around a little. Gourmet users of DOS under Windows can use these tips to spice up their DOS sessions.

## **Start DOS in a Window**

To start DOS in a window rather than full-screen in 386 Enhanced mode, you have to change the DOSPRMPT.PIF file (in Windows 3.1) or create a new PIF (in Windows 3.0).

In Windows 3.1, the DOSPRMPT.PIF file (the file that runs DOS when you click on the MS-DOS icon in the Main group) is located in the WINDOWS subdirectory. Simply open the PIF in the PIF Editor (located in the Main group in Windows 3.1 and in the Accessories group in Windows 3.0) and change the Display Usage selection from Full Screen to Windowed.

In Windows 3.1, highlight the gray DOS icon in the Main group and select Properties from the File menu. Change the Command Line entry in the dialog box from COMMAND.COM to COMMAND.PIF. Then use the PIF Editor to create COMMAND.PIF. Specify COMMAND.COM as the Program Filename, and set the Display Usage option to Windowed. The next time you click on the DOS icon, DOS will run in a window.

Here's another tip: Certain programs can't be run from a windowed command line. If you need to switch to a full screen temporarily, you can select Settings from the DOS window's Control menu (in the upper-left corner) and choose Window under Display Options.

You can also press Alt+Enter to switch between a small window and a larger full-screen view. See Figure 4.6.

## **Disable DOS Prompt Instructions**

Every time you select the DOS prompt icon to start a DOS session in Windows 3.1, you'll be given instructions on how to exit the DOS session and how to switch to other applications. To disable this message, edit your SYSTEM.INI. The line should read

```
DOSPromptExitInstruc=Off
```

## **Change the DOS Default Directory**

Here's a case where Windows 3.1 is a big improvement over 3.0. When you start a DOS session from Windows 3.0, you end up in the WINDOWS directory by default. To specify a different working directory, highlight the DOS icon in Program Manager and choose Properties from the File menu. The Command Line text box will contain the DOS executable filename COMMAND.COM. The path, including the name of your DOS directory, may also be there. Change the path to include the directory you want DOS to start up in. For example, if the Command Line box reads C:\DOS\COMMAND.COM or simply COMMAND.COM, change it to read

```
C:\DATA\COMMAND.COM
```

if you want DOS to start in the DATA subdirectory. For this trick to work, the working directory you specify must be included in your DOS PATH statement.

In Windows 3.1 this is even easier to do. You can specify a working directory for your DOS session (or for any application) by highlighting the DOS prompt icon, choosing Properties from the File menu, and specifying the Working Directory in the Program Item Properties dialog box that appears.

## **Expand the DOS Environment**

For a DOS application running in 386 Enhanced mode, the environment space available in a DOS session is much less than that set in your CONFIG.SYS file. To have access to the full environment, edit the application's PIF file to launch DOS, specifying the application's executable file and your standard DOS environment as optional parameters. For example, to run Procomm

Plus with a DOS environment of 1,024 bytes, the first four lines of the PIF would read like those in Figure 4.7.

## **Copying and Pasting between DOS and Windows**

You can copy and paste data between DOS and Windows applications in Real or Standard mode, but doing so can be a tedious process because you're restricted to copying a full screen at a time from the DOS applications. Windows' 386 Enhanced mode, on the other hand, allows DOS applications to run in resizable, movable windows, so you can copy and paste only as much data as you need.

Use the following tips for easy cut-and-paste operations, no matter what Windows mode you're working in.

### **Copy and Paste Text from Full-Screen DOS Applications**

The full-screen presentation of DOS programs in Real or Standard mode means that the only amount of data that the Windows Copy command and Clipboard will understand is "all of it." To copy data from a full DOS screen and paste it into a Windows application, press the **PrtSc** key; this copies the entire DOS screen to the Windows Clipboard. Then switch to the desired Windows program and select **Paste** from its **Edit** menu to insert the data from the Clipboard. You might need to clean up the pasted screen by cutting material you didn't really want, such as the menu bar, status line, and Ready indicator from a Lotus 1-2-3 screen.

### **Copy and Paste Text in Standard Mode**

Pasting text from a Windows application to a full-screen DOS application (in Real or Standard mode) is more complicated, and you can't transfer graphics this way. First, place the cursor at the point in the DOS document where you want to paste text. Then use **Alt+Tab** or **Alt+Esc** to switch to your Windows application and select and copy the data. Your DOS document will be iconized on the Windows desktop at this point—click on its icon once to activate its **Control** menu. Choose **Paste**, and Windows will paste the text at the cursor location in your DOS file.

### **Copy and Paste in 386 Enhanced Mode**

When your DOS application is running in a window (that is, when you're working in 386 Enhanced mode), the copy and paste capabilities work much as they do in a Windows application. Select **Edit**, **Mark** from the DOS program's **Control** menu. See Figure 4.8. (To get to the **Control** menu, click once on the box in the upper-left corner.) Then, use your mouse to select the range of data you want to copy. Next, select **Edit**, **Copy** from the DOS program's **Control** menu. To paste the selected data into a DOS file, place the cursor at the desired location in that file and use **Edit**, **Paste** from the DOS application's **Control** menu or **Paste** from any Windows program's **Edit** menu.

### **Copy DOS Graphics to Windows via the Clipboard**

While running a DOS application in Windows 386 Enhanced mode, you can copy DOS graphics to the Windows Clipboard by pressing the **PrtSc** key. This copies a bitmap of the DOS screen to the Clipboard. From there you can paste it into Paintbrush or another Windows application and save it. For a windowed DOS application you can copy the entire active window by pressing **Alt+PrtSc**.

To copy only part of a DOS screen, also run the application in a window. From the **Control** menu (click once on the box in the upper-left corner of the window), select **Edit** and choose **Mark**. With the mouse, select the part of the screen you want to copy. Then choose **Edit**, **Copy** from the **Control** menu.

### **Use the Clipboard to Send ASCII Files in DOS Communications Programs**

You probably still use your DOS communications program even though you've switched to Windows: Why rewrite your custom log-on scripts? But the Windows Clipboard can eliminate your having to save Windows word processing files as "Text Only" before sending them via ASCII-based E-mail. Launch the Windows PIF Editor and load the PIF for your DOS communications program. Select Windowed as the Display Usage option, and then click the Advanced command button and turn on the Allow Fast Paste check box near the bottom of the dialog box. Save your changes to the PIF.

Your DOS communications program will now appear in a resizable window when you launch it. To send a message or document composed on your word processor, copy the message to the Clipboard. Then switch to your communications program, click on its Control menu (in the upper-left corner), and select Edit, Paste. Your message will be entered, with no intermediate text file required.

## **Tips for Working in DOS Programs Running under Windows**

DOS has driven a decade of users mad by being weak and unpredictable, but it has given a generation of users great joy by being flexible. So it is with DOS under Windows. We could offer as many tips for handling DOS in the Windows world as we could for handling DOS on its own. DOS longs to be experimented with. Since we can't run on forever about this subject, however, here is a handful of tips that we thought particularly useful or interesting.

### **Change Fonts and Font Sizes for DOS Applications in Windows 3.1**

Make font sizes larger in windowed DOS applications to make text easier to read, or make the font size smaller to get more text into view at one time. While you're in a DOS application window, click once on the Control menu box in the upper-left corner and select Fonts from the drop-down Control menu. You'll see a menu of screen fonts and sizes that will work in a character-mode DOS screen. (See Figure 4.9.) Unlike the graphical Windows environment, DOS supports few font options. If you have the Save Settings box checked when you exit Windows, the font you choose will be your new default for DOS applications.

### **Launch a Batch File from Windows and End in DOS**

When you run a batch file from Windows (either from a PIF, from the File menu's Run command, or from a program item for a .BAT file), you'll end up back in Windows once the batch file has executed. To remain in DOS instead, make COMMAND.COM the last line of your batch file for starting a new DOS session. From the DOS prompt, type **exit** to return to Windows.

### **Unzip Files in Windows with Drag and Drop**

Follow these steps to unzip files that you've downloaded with your Windows communication program without going into DOS:

1. Create a separate directory for this purpose and make sure that PKUNZIP.EXE is in it.
2. Download your zipped files into this directory or move them into it after you're off-line.
3. Go into File Manager and drag the zipped file onto the filename PKUNZIP.EXE.
4. A dialog box will ask "Are you sure you want to start PKUNZIP.EXE using *filename*.ZIP as the initial file?" Choose Yes.
5. Windows starts a DOS session to expand the files and returns to the File Manager when it's all done.



## Turn Your Favorite Batch Files into Windows Icons

Experienced Windows users have probably passed this stage already, but those who've recently switched to Windows from plain old DOS may still be exiting Windows, opening a DOS window, or slogging through directories with the File Manager to use the batch files they once started with a few simple keystrokes. Don't.

You can run your old batch file simply by clicking on an icon. If you have a 386 machine or better and run Windows in 386 Enhanced mode, you'll get a bonus when you run batch files because Windows can multitask DOS programs. This means that the batch file can accomplish work in the background, performing hard-disk backups and the like as you continue your other tasks.

To create an icon that runs a batch file, use the PIF Editor to make a PIF that tells Windows how to run DOS applications. Enter the full path and name of the batch file, as well as a Window Title. If your batch file needs parameters, enter them in the Optional Parameters box. Also enter a Start-up Directory name if a program invoked by your batch file must be started in a particular directory. See Figure 4.10.

In many cases, you won't need to touch the other options because they're already set for a generic DOS program, but it's a good idea to look through them just in case. For instance, in Standard mode, if the batch file runs any graphics programs, select the radio button that says Graphics/Multiple Text. Likewise, in either mode, if you run any program that needs more than the default amount of RAM (128K), tell Windows to allocate more RAM by changing the number in the Memory Requirements text box. If you're running Windows in 386 Enhanced mode, you can choose whether the batch file runs full-screen or in a window.

Now that you have a PIF for your batch file, add it to your desktop as an icon: Select New from the File menu, specify Program Item, and enter the description you wish to appear under the batch file's icon (if you enter nothing, the name of the batch file, sans extension, is used to label the icon). Then, in the Command Line text box, enter the name of the PIF (for example, BATCH.PIF). Finally, click on the Change Icon button and browse through the available icons using the View Next button. When you find one you like, click on OK.

That's all there is to it. Now you can run your batch file simply by double-clicking on its icon.

## DOS Commands to Avoid

APPEND, ASSIGN, SUBST, and JOIN are DOS commands that you should avoid when running DOS under Windows. They're likely to cause problems because Windows gets confused as to which drives and files are which and where they are. APPEND tells DOS where to look for data files. ASSIGN refers requests for one disk drive to another. JOIN allows one entire drive to be treated as a subdirectory of a separate drive. SUBST (substitute) lets you assign a drive letter to a subdirectory. Windows 3.1 Setup scans your CONFIG.SYS and AUTOEXEC.BAT files for any of these commands and warns you about using them, but if you install them after your initial setup, Windows does not warn you of any danger in using them.

When you're in a DOS session, avoid running CHKDSK/F, FDISK, RECOVER, SELECT, FORMAT C: (or any other drive letter), UNDELETE, or other commands that reorganize a hard disk.

## Don't Use the MS-DOS Task Swapper

If you're running Windows, avoid the DOS Task Swapper. The Windows Task List will do the same thing without costing you more conventional memory. Better yet, use Alt+Tab to navigate from task to task.

## Use DOS 5.0's Built-in Help System

If you're in a DOS session and you forget the syntax of a DOS command, type

*HELP command name*

or

*command name /?*

and DOS will provide a description of the command, the proper syntax, and the optional switches available.

## **Three TSR Rules for DOS Applications**

To avoid unnecessary hassles with memory-resident DOS programs, follow these rules:

If you have a TSR (memory-resident program) that doesn't need to interact with other applications, run it as you would any other DOS application under Windows.

If you have a TSR that's required by all DOS programs, load it before you start Windows.

If you have a TSR that's required by all Windows applications and you're running in 386 Enhanced mode, load the TSR using a batch file called WINSTART.BAT (stored in your Windows subdirectory). This way, the TSR will be loaded and available when Windows starts or your AUTOEXEC.BAT file loads it, but it won't take memory away from your DOS programs.

## **Run Local TSRs**

To have access to TSR programs in each DOS session that you run under Windows 3.1, you have to edit the SYSTEM.INI file. In the [NonWindowsApp] section, look for the line LocalTSRs=. You may not already have this line in the file. If you don't, add it to the [NonWindowsApp] section, and after the equal sign add the names of the TSRs you want available during every DOS session, separating each name with a space. Before adding a TSR to the list, you should make sure that it has no problems running under Windows.

If you don't want to edit your SYSTEM.INI file, or if you're running Windows 3.0, you can create a batch file that launches all of your TSRs in a single DOS session. Create the batch file with Notepad and then create a PIF for it. Make sure to include the setting Allow Close When Active in the PIF. This will let you exit from the DOS session but leave the TSRs loaded into memory.

## **Start DOS Applications with Graphic Logo Full-Screened**

Many DOS applications have graphical startup or logo screens, although the application itself may be text-based. These programs, including Lotus 1-2-3, cannot be started in a window in 386 Enhanced mode. The best solution is to start the application at full-screen size (check Full Screen in the application's PIF) and then press Alt+Enter to switch to a window.

Another possibility is to check the DOS program's documentation for parameters that will start the application without displaying the graphics screen.

## **Tab Through Applications from a Full-Screen DOS Application**

When switching out of a DOS program running in full-screen mode, hold down the Alt key and press Tab successively-a title bar at the top of a blank screen will cycle through the names of all applications to which you can switch. This method is quicker than using Alt+Esc, which switches entire windows, loading complete programs as you cycle through applications. It's also faster than pressing Ctrl+Esc, which will switch you to the Windows desktop and call up the Task List.

## **Three Ways to Speed Up Printing for DOS Applications**

There are three things you can do to speed printing from DOS applications running under Windows:

Don't use Print Manager (a good general rule for DOS apps).

If you're running in 386 Enhanced mode, make sure that the PIF setting has your DOS application running in Exclusive mode. This will force Windows to allocate the maximum available resources to the DOS print job. If you choose to do this, however, don't print if something is actively running in the background (such as a communications download), because background processing will come to almost a complete stop.

Load Windows with the /S parameter at the DOS command line to start Windows in Standard mode. You won't be able to multitask your DOS applications, but printing from them should go faster.

## **Assign Hotkeys to Your DOS Applications**

Windows 3.1 allows you to assign hotkeys to your DOS applications, allowing for quick task switching.

In each program's PIF, go to the Advanced Options dialog box and find the last option, Application Shortcut Key. Click on this box and a cursor will appear. Press the key combination that you want to assign to the application. The keystrokes are recorded and appear in the box. Your combinations can include the Ctrl, Alt, and Shift keys. Make the hotkeys easy to remember—use the first letter of the application's name if possible.

When you want to assign a new hotkey to the application, clear away the previous one by placing the cursor in the Application Shortcut Key box and pressing Ctrl+Shift+Backspace.

## **Easy-to-Remember Hotkeys**

If you think you'll forget the hotkeys that you've assigned to DOS applications, you can jog your memory every time you launch the application by annotating the application's icon title with the hotkey combination. To do so, highlight the program's icon in Program Manager, select Properties from the File menu, and in the Description text box, type the name of the hotkey after the program name—for example, type **Excel\_Alt+Ctrl+E**. See Figure 4.11. In Windows 3.1, icon titles automatically wrap to the next line, so add blank spaces until the hotkey combo appears in the line beneath the actual description.

## **Run DOS Programs in the Foreground**

For better overall performance when running DOS applications in 386 Enhanced mode, try running them in the foreground only. To do this, go to the Execution section of the program's PIF and make sure that the Background box is not checked. If you leave this option unchecked, the application will stop what it's doing when you switch to another task. However, if you want the DOS application to calculate data or print documents while you're doing something else, the Background box should be checked.

If you'll be running DOS applications in the background and want to experiment with the PIF's background and foreground settings, click on the Advanced command button in the PIF Editor and refer to the Multitasking Options at the top of the dialog box. Try increasing or decreasing the values for Background Priority and Foreground Priority by small increments (say 10) at a time.

## **Stop Update Messages to File Manager**

Set FileSysChange=no in the [386Enh] section of SYSTEM.INI to avoid sending update messages to File Manager from non-Windows applications. Turning this setting off prevents File Manager from receiving updates every time a DOS application creates, renames, or deletes a file. When this setting is on, system performance often suffers.

## **Switch between DOS and Windows Applications with the Task List**

Windows 3.0 lets you switch easily between standard DOS and Windows applications during a multitasking session. When you are working in a DOS program loaded from Windows, pressing Ctrl+Esc temporarily exits that program and displays the Windows Task List, from which you can select the Windows application you want. When it's time to switch back to your DOS application, simply click on the DOS icon that Windows created for that program when you pressed Ctrl+Esc.

## **Exit DOS Sessions before Powering Down Your Computer**

Never shut off your system while still in a DOS session-you may cause file damage. Always exit the DOS session and return to Windows (by typing exit at the DOS prompt), and then exit the Windows session.

## **Create a DOS Prompt to Remind You that You're in Windows**

Some programs, such as Windows 3.0 and XyWrite III Plus, let you temporarily exit, or shell, to DOS without reminding you that you've done so. In such cases, it's easy to forget that the program is still loaded and that there's little room to run another application. You may also forget how to return to the original application once you've shelled out to DOS.

To remedy this situation, modify the batch file that starts the DOS application. First, add a line that changes your prompt to a warning message. Position this line just above the command that invokes the application. After the line that runs the application, include a line that returns the prompt to its normal setting.

The following batch file, which starts Windows 3.1, illustrates this technique:

```
CD \WINDOWS
PROMPT=Type EXIT To Return To
Windows $_$P$G
WIN
PROMPT=$P$G
```

The second line changes the prompt and remains in effect while Windows is running. if you shell to DOS, you'll see the message "Type EXIT To Return To Windows" on one line and the current drive and directory on a second line. The last line resets the prompt when you exit Windows.

## **Create Fun Prompts**

Personalize your DOS environment by setting a special DOS prompt when you're in Windows. Sure, you can add a message reminding you that you're in Windows (as you just learned), but how about something a little more fun? You can create the following prompts using the DOS device driver ANSI.SYS, which controls the display of colors and special graphics characters under DOS. To take advantage of these, the ANSI.SYS driver must be loaded before Windows is run. You can do this by adding the following line to your CONFIG.SYS:

```
DEVICE=C:\DOS\ANSI.SYS
```

Or, if you're using DOS 5.0, load it high:

```
DEVICEHIGH=C:\DOS\ANSI.SYS
```

With ANSI.SYS loaded, you can in turn load Windows from a batch file that sets up a customized DOS prompt at the same time.



\$e[xx;xx;...m

where xx is the number for the color attribute (from Table 4.2), and semicolons separate each number. Note that the letter "m" must end the metastring. Table 4.3 lists the display attributes you can use in ANSI metastrings; Table 4.4 lists the ANSI metastrings for cursor positioning.

**TABLE 4.2 Color Attributes for Use in ANSI Metastrings**

Color	Foreground	Background
Black	30	40
Red	31	41
Green	32	42
Yellow	33	43
Blue	34	44
Magenta	35	45
Cyan	36	46
White	37	47

**TABLE 4.3 Display Attributes for Use in ANSI Metastrings**

Attribute	Meaning
0	Normal
1	Highlight
4	Underline
5	Blink
7	Inverse

**TABLE 4.4 ANSI Metastrings for Cursor Positioning\***

**Action What You Enter**

Moves cursor up **\$e[rA**  
 Moves cursor down **\$e[rB**  
 Moves cursor right **\$e[cC**  
 Moves cursor left **\$e[cD**  
 Moves cursor to precise position **\$e[r;cf** or **\$e[r;ch**  
 Moves cursor to upper-left corner **\$e[H**  
 Clears screen and leaves cursor in upper-left corner **\$e[2J**  
 Erases current line from cursor to end of line **\$e[K**  
 Records cursor's position **\$e[s**  
 Moves cursor to recorded position **\$e[u**

**Graphics Characters**

ANSI.SYS can also access the graphics characters at the high end of the ASCII character set (see Table 4.5). These include lines, simple geometric shapes, scientific symbols, and the like. They are listed by number, and can be found in the technical reference manuals for most PCs or in any listing of the ASCII character set, which is the standard alphabet PCs use. To enter any ASCII graphics character into your prompt, hold down the Alt key and then type the character's number using the numeric keypad only (not the numbers across the top of the keyboard). Then release the Alt key. The character should appear.

**TABLE 4.5 Use the Numeric Keypad on Your Keyboard to Generate Graphic Characters in Your Fun Prompts**

Character Generated	ASCII Code	Character Generated	ASCII Code	Character Generated	ASCII Code
-	196	±	177	×	215
¿	191	Í	205	Ð	208
Ú	218	»	187	,	184
À	192	É	201	Õ	213
Ù	217	È	200	Ô	212
Ã	195	¼	188	½	189
´	180	ì	204	Æ	198
Å	197	¹	185	µ	181
	179	î	206	Ø	216z
Â	194	º	186	Ñ	209
Á	193	Ë	203	Ï	207
©	169	Ê	202	î	238
ª	170	·	183	à	224
ß	223	Ö	214	ï	248
Ü	220	Ó	211	õ	245
Ý	221	½	189	ô	244
þ	222	Ç	199		
Û	219	¶	182		