CORIOLIS GROUP BOOK

WINDOWS® 3.1 INSIDER

THE GUIDE TO HARD-TO-FIND AND UNDOCUMENTED FEATURES

HOT TIPS AND SHORTCUTS FOR WORKING FASTER AND SMARTER

TECHNIQUES FOR CUSTOMIZING WINDOWS

ADVICE ON AVOIDING COMMON PITFALLS AND HANG-UPS

KEITH WEISKAMP
RON PRONK

Windows® 3.1 INSIDER

THE WILEY INSIDER SERIES

If you think this book is just another computer book, stop right there. I am proud to tell you that this book kicks off a stunning new series created by the Coriolis Group and published by John Wiley & Sons. What you're holding is a "tour-de-force" of Windows insider tips, secrets, undocumented features, shortcuts, and technical advice that has never been available in a book format like this.

I've teamed up with longtime Windows wizard and consultant Ron Pronk to develop our first INSIDER book. Ron has spent every waking moment since the release of Windows 3.1 interviewing users and searching for the best technical information available. But the real kicker is that the unique format of this book will help you understand and use the tips and techniques presented as if a friendly expert were at your side, ready to answer any question you might have.

In creating the new INSIDER series, we wanted to break with tradition and develop books that go way beyond the typical "documentation approach" of most computer books. In each INSIDER book, we'll show you how to customize your software, put new features to work right away, work smarter and faster, and solve difficult problems.

What you're holding represents an innovative and highly practical guide that was developed by some of the best minds in computer book publishing. We hope you enjoy each and every INSIDER guide.

—Keith Weiskamp
INSIDER Series Editor

CORIOLIS GROUP BOOK

Windows® 3.1 INSIDER

Keith Weiskamp
Ron Pronk

John Wiley & Sons, Inc.
New York • Chichester • Brisbane • Toronto • Singapore

Recognizing the importance of preserving what has been written, it is a policy of John Wiley & Sons, Inc. to have books of enduring value published in the United States printed on acid-free paper, and we exert our best efforts to that end.

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold with the understanding that the publisher is not engaged in rendering professional service. If expert assistance is required, the services of a competent professional person should be sought. FROM A DECLARATION OF PRINCIPLES JOINTLY ADOPTED BY A COMMITTEE OF THE AMERICAN BAR ASSOCIATION AND A COMMITTEE OF PUBLISHERS.

Library of Congress Cataloging-in-Publication Data

Weiskamp, Keith.
Windows 3.1 insider / Keith Weiskamp.
p. cm. -- (Wiley insider's guide series)
Includes index.
ISBN 0-471-57984-X (pbk.)
1. Windows (Computer programs) 2. Microsoft Windows (Computer file) I. Title. II. Series.
QA76.76.W56W4514 1993
005.4'3--dc20 92-26855

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

To Ruth and Christina.

And our many talented friends of Suite 204.

Acknowledgments

Writing any book (and especially developing a series of books) requires a small army of talented and hardworking people. At the top where it all begins, we're very grateful to be working with Stephen Kippur and Katherine Schowalter at John Wiley & Sons.

We'd also like to thank a number of creative and insightful publishing people including Frank Grazioli, an extraordinary managing editor who went way beyond the call of duty to help us create a first rate-book; Paul Farrell, a highly-skilled and sensitive editor who made us stop and think every step of the way; Pat Vincent, a talented technical editor and reviewer who provided numerous valuable insights; Kenneth J. Iversen for his vlauable contribution with the networking material; Christina Bauman for rushing back from Canada to help with proofreading; Rob Mauhar and Brad Grannis of the Coriolis Group, who created the illustrations; Jeff Duntemann, master editor and author who championed the new INSIDER series; and last but not least, Bob Ipsen whose job is just beginning.

Contents

Introduction

Microsoft bestowed the term "Windows" on its graphical environment long before Windows 3.1, or even before Windows 3.0. In fact, the first version of Windows was a flop in the marketplace. Why?

Microsoft envisioned a software environment that could manage programs without the need to load each program from DOS, and could help out the user with a graphical environment that replaced cryptic DOS commands (like COPY C:\WINWORD\INSIDER\CHAP1.DOC A:). Microsoft's vision was on target, but was gazing too far into the future. At the time, IBM PCs and compatibles had limited graphics capabilities, and were so limited in terms of processing power and memory capacity that a user-friendly environment like Windows just slowed things down to a crawl. At that time (mid-1980s), PC performance was pretty sluggish by today's standards, and a program like Windows just made computing life a few pulses slower.

Fortunately, Microsoft stuck with its graphical user interface (GUI) technology and upgraded it as IBM-PC processor technology improved. As PC technology matured, a GUI like Windows began to make more and more sense. In fact, Apple's Macintosh environment was already devoted to GUI technology, and was rapidly making the IBM-PC look—to a lot of users—like a needlessly complex machine.

Apple and its Macintosh systems just might have eclipsed the IBM PC standard had it not been for the introduction of Windows 3.0 in 1989, which brought a friendly GUI to all systems that used a 286 or 386 processor. Windows 3.0 could also run on 8086 or 8088 systems, but this was ridiculously impractical.

With Windows 3.1, released in 1992, Microsoft has essentially discarded the 8086/8088 processor standard, recognizing that most users who need to maximize performance on their PCs are using *at least* a 286-based system, and more likely a 386- or 486-based system.

So what does the 286/386 minimum processor standard mean, in terms of processing power, to users like yourself? Two facts come to mind:

Fact 1 Under Windows, many, if not all, of your programs can now exploit the extended memory-addressing capabilities of 286/386/486 machines.

Fact 2 You can open and use multiple programs—without having to close one program in order to switch to a different program.

Windows and Your Computer

When you view your computer through the Windows environment, what do you really see? It's our view that Windows is actually misnamed; its name made sense in earlier versions of the program, but today "Windows" doesn't really describe the true power of the operating environment.

The value of Windows lies in its ability to manage multiple programs that are essentially running at the same time. The Windows 3.1 program is really a one-armed juggler. It uses a single CPU (one arm) to keep multiple programs running (in the air), without "dropping the ball."

Windows manages multiple programs by taking advantage of the memory addressing capabilities of the 286/386/486 processors. Although you can use Windows without understanding how Windows 3.1 uses these CPUs, you'll be at a disadvantage if you want to *optimize* Windows. In other words, if you want Windows to juggle many, many objects (programs) of different sizes and origins—without dropping the ball—you'll probably want to improve Windows' juggling dexterity.

That's the purpose of this book. Windows will do a good job for you even if you never read or apply a single paragraph in this book. But you can get Windows to do some amazing juggling feats by understanding the way the Windows "arm" works and can be manipulated to improve its performance.

A Few Goals of This Book

We don't pretend that this book is going to be all things to all Windows users, but we do believe that it will have *something* to say to every Windows user.

To make this book as useful as possible, we want to provide you with the same guidelines, as a reader, that we followed in putting together the book:

1. *If the Windows* User's Guide *covers a topic (in our humble opinion) incompletely or too cryptically, we'll elaborate on the topic for you.* Microsoft was limited by space in its *User's Guide*, and we understand that. However, you might not be so sympathetic. Case in point: it's 11:30 P.M.; you're trying to improve the way your system uses conventional and upper memory, and you've tried for the umpteenth

time to make heads or tails out of the *User's Guide*'s cryptic sections on optimization. We'll come to the rescue, with detailed step-by-step explanations of numerous optimization techniques, along with descriptions that explain *why* and *how* these techniques actually work.

2. *If you can do it in Windows, but the Windows* User's Guide *ignores the technique, you'll probably find it described in this book.* We've interviewed hundreds of users, in dozens of different office and personal-use settings, to determine how people use Windows and how they would like to use Windows. And that's the kicker. This book is geared toward results. If we give you a tip, we'll also suggest some practical situations for using the tip. At the very least, we'll always explain how you can benefit from any concept, tip, or warning given in the book.

3. *If the Windows* User's Guide *covers a topic satisfactorily, you won't find the topic discussed here.* We're not out to reinvent the wheel. Microsoft's *User's Guide* is a good piece of writing. You'll find a lot of good information and tips there. So, you won't find anything on the Paintbrush or Notepad applications in this book, nor will you find an explanation of how to create a Program Group. If you don't have the Windows *User's Guide*, you can find this kind of basic information in any good tutorial book on Windows 3.1.

4. *We present tips, warnings, and other information not available at the time Windows 3.1 was released.* Microsoft is constantly releasing news bulletins about hardware problems and solutions, new software for Windows, updated device drivers, and other information that can help you get the best possible performance from Windows. We've included much of this information in this book. In cases where information is available via modem or by phone, we'll tell you how to contact these sources.

Features of This Book

We don't expect you to read through this book page by page, cover to cover. We know that most readers of computer books have specific questions or problems that they want resolved, or they want to learn how to use specific features. If this what you expect from a computer book, you also want to be able to find information quickly and easily. This book is designed with this kind of easy access in mind.

Each chapter contains several major topics, such as *Using TrueType*, *Using DOS 5 to Optimize Windows*, and *Using Swap Files.* For each topic, we'll first provide some general information. Then, the topic is broken down further into subtopics, with each subtopic describing a particular Windows concept in more detail or showing you how to use a particular feature or solve a specific problem. Each subtopic begins with an informative heading, followed by a few brief sentences (printed in bold italic type) that tell you quickly what you can expect to learn by reading through the rest of the material in the topic.

HOT TIP

We've also included several *Hot Tips* throughout each chapter. Each Hot Tip provides you with information that you can use immediately to solve some of the more sticky Windows problems or to put to use some little-known techniques that we think every Windows user should know.

Although this book includes a lot of reference material regarding Windows 3.1, it's not your run-of-the-mill reference book. We want you to know *how* Windows works—from the ground floor up—and we want you to enjoy and understand what you're reading. Above all, we want this book to be *practical*. So, general reference sections, such as detailed listings of PIF options, are not included in the book. In the body of the book, we'll explain specific PIF, WIN.INI or SYSTEM.INI settings as they relate to the topic at hand. In this way, you can quickly find a specific topic or feature and learn how to put it to work for you; you won't have to rummage through pages of technical listings or exhaustive entries for every situation under the sun.

You might also want to take a quick look at the two troubleshooting chapters—Chapters 16 and 17. Chapter 16 provides hardware troubleshooting strategies and solutions for a wide range of commonplace Windows-related problems. Chapter 17 identifies specific software problems that have been discovered since the April 6, 1992 release of Windows 3.1. Both chapters provide workarounds and alternatives.

One final note: since you're reading this book, you've probably already installed Windows 3.1 and put the program to some use. However, if you need to install Windows on a different system or on a network, you might encounter problems that didn't occur with your initial installation. If so, we suggest you use Appendix A to help you do some troubleshooting.

The INSIDER Disks

ON DISK

A set of disks is available (see Appendix B for detailed information) that includes an assortment of Windows enhancement tools, utilities, and hard-to-find troubleshooting information. Some examples of the utilities are discussed throughout the book. Just look for the "On Disk" icon.

Ten Great Windows 3.1 Tips

To get you started, we've gathered together ten tips that exploit features that are new in Windows 3.1. You'll find most of these ten techniques mentioned elsewhere in the book. But by presenting these techniques together, we think we can give you a good sense of the direction we take in this book, and you'll get a good feel for the improved performance of Windows 3.1 over Windows 3.0.

TIP 1

Suppress Your StartUp Group

You've probably already discovered the power and usefulness of Windows 3.1's StartUp group. The concept is simple: when you place an application in the StartUp group, Windows will start the application automatically, each time you start Windows itself. You don't need to tinker with the load= and run= lines in your WIN.INI file as you did in Windows 3.0. (For compatibility, these lines in WIN.INI are still available in Windows 3.1, but they're completely independent of the StartUp group.) You can place as many applications as you want in the StartUp group. However, this fact can have its downside.

When you add several programs to the StartUp, Windows can be slow to start. Also, when you use the StartUp group to load programs, you could be consuming quite a lot of memory. So, there will probably be times when you want Windows to load the programs in your StartUp group, and other times when you don't want to use the StartUp group. Fortunately, Windows 3.1 provides a way for you to do this—with just the touch of a key. Here's how:

1. At the DOS prompt, start Windows as you normally would, by typing **WIN** and then pressing **Enter.**
2. *Immediately* after you press **Enter**, press and hold the **Shift** key.
3. Keep the **Shift** key held until your Program Manager window appears. You'll notice that the programs in your StartUp group haven't been loaded.

TIP 2

Use Alt+Tab to Cycle through Open Applications

Although you can use the Task List to list and switch between open applications, there's a much easier way to move from one application to another—the **Alt+Tab** key combination.

The **Alt+Tab** combination was available in Windows 3.0, but it wasn't as useful, fast, or attractive as it is in Windows 3.1. When you hold down

Alt and then tap the **Tab** key, Windows will display the name of the previously opened Windows application. You can continue to tap the **Tab** key (you should keep the **Alt** key pressed the entire time) to cycle, in reverse order, through the names of all open applications. When you release both the **Alt** and **Tab** keys, Windows will switch to the currently selected application.

You'll get the greatest benefit out of the **Alt+Tab** feature if you're running Windows in enhanced mode. The feature works essentially the same in standard and enhanced modes, but is more visually appealing in enhanced mode.

To introduce yourself to this feature, try following these simple steps:

1. Open at least three applications. (You can minimize any or all of the applications if you want, and you can use both Windows and non-Windows applications.)
2. Hold down **Alt** and then tap the **Tab** key to display the name of the previously opened application.
3. Release the **Alt** key to display the named application.
4. Hold down **Alt** and tap **Tab** repeatedly to cycle through the names of all open applications, in the reverse order in which they were last opened or used. Release the **Alt** key when Windows displays the name of the application that you want to display.

Chapters 4, 9, and 15 provide more information about running multiple applications in Windows 3.1. Chapter 4 explains the internal techniques that Windows uses to manage multiple applications. Chapter 9 provides some specific tips and techniques for running non-Windows applications in tandem with Windows applications. Chapter 15 explains how to use the new object linking and embedding (OLE) capabilities of Windows 3.1.

TIP 3 Save the Program Manager or File Manager Settings without Exiting

In Windows 3.0, the only way for you to save new configuration settings for the Program Manager or File Manager was to check the Save Settings on Exit box. This approach was troublesome because if you wanted to change, say, Program Manager settings, you usually had to exit Windows immediately after you configured the Program Manager—a real hassle if you weren't done working in Windows but wanted to specify a new Program Manager configuration.

Another annoying aspect of the Windows 3.0 technique was that you had to remember to *turn off* the Save Settings on Exit check box the next time you started Windows. If you failed to make this change and then modified your Program Manager enviroment, this new environment would be saved when you exited Windows—overwriting the Program Manager configuration that you had wanted to keep permanent!

In Windows 3.1, you can easily save Program Manager settings without exiting Windows, and you can save File Manager settings without exiting the File Manager. Here's how:

1. In the File Manager or Program Manager, make sure the Save Settings on Exit option (in the Options menu) is *not* checked. If this option is checked, any changes that you make will be saved each time you exit—which is probably not what you want.
2. Make any desired changes to the Program Manager or File Manager. For example, in the Program Manager, you can resize group windows, open or close different group windows, and change the contents of your StartUp group. In the File Manager, you might want to change the default directory that appears whenever you start the File Manager, or you might want to change the size of the Directory Tree/Directory List window split.
3. Hold down the **Shift** key, then click on File Exit. Windows will save your new settings without exiting the File Manager or Program Manager.

Chapter 8 provides more information and tips for using the Program Manager and File Manager effectively.

TIP 4 Use MORICONS.DLL and PROGMAN.EXE to Assign a Different Icon to an Application

In Windows 3.0, your DOS applications had to use one of a few generic—and ugly—icons that were stored in PROGMAN.EXE. In Windows 3.1, you can choose from among almost 150 icons for DOS applications, or even for Windows applications whose icons don't satisfy you.

Windows 3.1 stores icons in two different files: PROGMAN.EXE (as was done in Windows 3.0) and in MORICONS.DLL. PROGMAN.EXE provides 46 icons, while MORICONS.DLL provides 100 icons—most of which are designed for use by DOS applications.

To assign one of these icons to an application that already has an icon, follow these steps:

1. In the Program Manager, click on the existing icon for the application.
2. Click on File Properties, then click on the Change Icon button.
3. In the File Name box, enter either **PROGMAN.EXE** or **MORICONS.DLL**, then click on OK.
4. Scroll through the Current Icon display, then click on the icon that you want to select. If you don't see any icon that you like, enter the file name that you didn't select the first time (**PROGMAN.EXE** or **MORICONS.DLL**), then click on OK to view the other set of icons.
5. After you've highlighted an icon that you want to use for the application, click on OK to associate the new icon with the application.

If the application doesn't already have an icon, replace Steps 1 and 2 above with the following step:

- Click on File Properties, make sure the Program Item is selected, then click on the OK button. Enter an appropriate description and command line in the Program Item Properties dialog box.

TIP 5 Use Drag-and-Drop to Create Document Icons and Start Applications

In Windows 3.1, you can create an icon for a document—as long as the document has an extension that's associated with a Windows application. Document icons are useful because you can put them in one of your Program Manager group windows, then start the document instantly from within the Program manager.

In other words, when you want to start the document, you just double-click on the document icon from within the Program Manager. Windows will start the application for the document, and will load and display the document within the application.

To create a document icon, follow these steps:

1. Start the File Manager.
2. In the directory tree, click on the directory that contains the document.
3. In the left half of the window, click on the document name.
4. Press **Ctrl+Esc** to display the Task List, then click on the Tile button to display the File Manager and Program Manager side-by-side. (If you have other applications open, these will also be tiled; you might want to close other applications, then re-tile, before you continue.)
5. Drag the highlighted document from the File Manager into the desired group window within the Program Manager. Windows assigns the

document the same icon that you use to start the application. Now you can start the document directly from within the Program Manager, simply by double-clicking on the document icon.

Note: To create a document icon, the document's extension must have already been associated with the desired application. For more information on creating an association, see Chapter 8.

The new drag-and-drop features provide some interesting twists and additions to the procedure we've just described. For instance, by now you've probably already experimented with the StartUp group and realized that any application that you place in the StartUp group will run each subsequent time that you start Windows.

What you might not know is that you can drag applications directly from their group window to the StartUp group; you don't need to go through the old Windows 3.0 procedure of creating a program item icon each time you want to place an application's icon in a different group. In fact, you can use this same technique to copy or move icons easily from any group window to any other group window.

You can also use drag-and-drop techniques to drag a document from the File Manager into the StartUp group. This trick works great when you want to take up tomorrow where you decide to quit at the end of the day. If you drag a document icon into the StartUp group, Windows will start the document and its application the next time you start Windows. To use this technique, just follow the steps described above, except make sure you drag the document name into the StartUp group (or onto the StartUp group icon).

You can even use drag-and-drop to start documents that are *not* associated with an application. To do so, just find the document name in the File Manager. You can then drag the document name on top of the EXE file (in any directory window) for the application you want to use to edit the document. When you do this, the application will start and will open the document. To achieve this same result, you can also drag a document name from the File Manager onto an application icon in the Program Manager.

Chapter 8 provides more information on creating associations for documents and on using drag-and-drop techniques.

TIP 6 Print Documents from Within the File Manager

Windows 3.1 lets you print a document from within the File Manager without having to open the document's application. There are two ways to do this:

Technique 1—Use the File Menu Under this approach, you use the File Menu to print a selected document. Follow these steps:

1. Start the File Manager.
2. Use the directory tree and list of files to find the document that you want to print.
3. Click on the name of the document to be printed.
4. Click on File Print.
5. In the Print dialog box, click on the OK button.

Technique 2—Drag and Drop to the Print Manager If the Print Manager is already in use, you can drag a document from the File Manager into the Print Manager, which automatically tells Windows to print the document. Here are the steps to follow:

1. Make sure the Print Manager is already open (although it can be minimized as an icon), then start the File Manager.
2. Reduce the size of the File Manager window so that the Print Manager window or icon is visible.
3. Use the directory tree and list of files to find the document that you want to print.
4. Drag the file to the Print Manager. (For some applications, you might need to verify print options in an additional Print dialog box.)

Chapters 11 through 13 provide more information on printing with Windows 3.1—including tips for using TrueType and for working with printers.

Tip 7 Use File Open from the Program Manager to Return to Your Previously Used Application

Whenever you click on File Open in the Program Manager, Windows reopens the application that you most recently exited. This is a useful feature if you want to quickly return to a previously used application.

Suppose you're working in Word for Windows, and you need to make some quick calculations. You switch to the Program Manager, then find and start the Windows Calculator, make your calculations, close the Calculator, and then return to Word. A few minutes later, you realize you need to make some additional calculations. You can save yourself a few steps by trying this technique:

1. From within Word (or whatever the current application), press **Alt+Tab** to switch to the Program Manager.
2. Click on **File New** to start the previously used (but closed) application—in this case, the Calculator.

TIP 8

Display the Contents of the File Manager in Any Font and in Any Font Size

The File Manager in Windows 3.1 lets *you* decide how text and icons should appear—including the size and font. You're not stuck with the System font used in Windows 3.0. In fact, you can select from any installed screen font or TrueType font.

If you select a TrueType font as the File Manager display font, you can display text in sizes ranging from eight points to 36 points. This flexibility is useful if you have poor eyesight, or even if your eyes are just tired of reading that seemingly tiny 11-point default font. Windows will enlarge icons in the File Manager to match the point size that you select for text.

To change the File Manager's display font and/or size, follow these steps:

1. Start the File Manager.
2. Click on Options Font.
3. In the Font dialog box, select a font, font style, and font size. (Use the Bold font style if you want to make the File Manager display as easy on your eyes as possible.)
4. Click on the OK button to exit the Font dialog box.

The new File Manager settings are now permanent, and will remain in effect until you change them. Chapters 11 and 12 provide additional information on using fonts in Windows 3.1.

TIP 9

Tip 9: Change the Font Size for a DOS Application Displayed in a Window

In both Windows 3.0 and 3.1, you can display a DOS Application in a Window by pressing **Alt+Enter** or by turning on the Display Usage—Windowed button in the application's PIF.

But Windows 3.1 gives you an additional level of control over the DOS applications that appear within a window. Specifically, you can change the size of the display font—a useful feature if you want to view more text at a time in a relatively small DOS window, or if you want to enlarge the size of text when your DOS application is windowed.

To change the font size in a DOS window, follow these steps:

1. Start the DOS application and make sure the application is displayed in a window (rather than full screen).
2. Click on the window's Control-menu button, in the upper-left corner of the window.
3. In the menu that appears, click on Fonts.
4. Use the Font list box to select a different aspect ratio (the relative width-to-height, measured in pixels) of each character. An example of the selected font, or aspect ratio, will appear near the bottom of the dialog box.
5. If you want this setting to be permanent for this application, make sure the Save Settings on Exit box is checked.
6. Click on the OK button to exit the Font Selection dialog box.

Chapter 9 includes more information on running non-Windows (DOS) applications in Windows 3.1.

TIP 10 Use the MSD Program to Examine and Diagnose Your System's Memory and Hardware

Microsoft bundled a valuable diagnostics program called Microsoft Diagnostics, or MSD, with Windows 3.1. You can use this program at the DOS prompt to examine your system's memory usage, your hardware configuration and settings, and much more. To use MSD, exit Windows, then type **MSD** at the DOS prompt. The program will take a minute to examine your system's memory and hardware, and then will display a menu allowing you to view and diagnose different parts of your system. For more information on MSD, see Chapter 2. Chapters 1, 2, and 3 provide additional information, techniques, and tips for managing and optimizing your system's memory to support Windows 3.1.

Windows® 3.1 INSIDER

Part Overview

PART 1

Inside Windows

The chapters in Part I are designed to let you "get under the hood" and see how the different components of Windows work in coordination with applications and your computer's hardware—especially with memory. You'll also have an opportunity to try out some selected Windows 3.1 tips.

The Introduction helps orient you toward the layout and features of this book, and includes ten of our favorite Windows tips. Chapter 1 introduces you to how memory is structured on personal computers and explains how Windows uses this memory. If you're running Windows 3.1 under DOS 5, you'll find Chapter 2 to be especially useful, because it explains how to use DOS 5 features to optimize your system's memory for use by Windows. Chapter 3 introduces you to the uses and benefits of third-party memory managers, such as QEMM-386 and 386MAX. Chapter 4 explains, in great detail, the way Windows uses your system resources to manage multiple applications.

CHAPTER

1

The Memory Connection

Windows is so friendly that it can install itself with almost no help from you. On many computers, Windows can also manage your system's memory and resources very effectively. But if you want to get optimal performance out of Windows, *you're* going to have to take charge.

This chapter will start you on your way to learning the inner workings of Windows. Since the key to optimizing Windows is to make full use of memory and disk-management capabilities, we'll begin there.

If you feel that some of this information is too basic for your skill level, feel free to skip sections. On the other hand, if you feel that much of the material in this chapter is too technical for you, don't give up on us. You don't need to be an auto mechanic to be a good driver. And you don't need to be a Windows guru in order to get the most out of Windows. If you don't think you're quite ready to spend time learning about your system at the hardware level or if you don't plan to spend a lot of time optimizing your system's memory, you can skip all or part of these early chapters and jump instead to some of the more hands-on-oriented material in later chapters. You can also return to the first three chapters later. In any event, make sure you check out the *Ten Great Windows 3.1 Tips* in the introduction.

If your knowledge of processor operations and PC memory is a little sketchy or if you just want an accurate Windows-oriented view of computer performance and memory use, this chapter was written with you in mind.

Windows and the CPU

Before we can wind our way through Windows and memory management, we need to touch on some basic *CPU* concepts. We're not going to offer a primer on the operations of the Intel 8086/286/386 family of processors—the so-called 80x86 family used in all IBM-PC-compatible computers. Instead, we would like to make sure you're clear on a few CPU-related concepts and terms.

It can be difficult, if not impossible, to adequately explain how Windows operates and manages memory unless we use some fairly technical terms. If you'll allow us to get just a bit technical here, we promise to make it worth your while later. Of course, if some or all of the concepts we introduce are already familiar to you, feel free to skip those sections.

Riding the CPU Bus

All IBM PCs are based on an Intel *bus architecture.* The *bus* (also called the *system bus*) is essentially a set of parallel lines that connect the CPU to main memory. The bus allows electrical currents to pass between the CPU and memory. The CPU, which stands for *central processing unit,* is the chip on your computer's main board (or motherboard) that manipulates all program instructions and data. Essentially, the CPU—also called the *processor* or *microprocessor*—is the heart and soul of your computer system.

Main memory is more commonly know as RAM (Random Access Memory) and is typically a collection of memory chips that are plugged into your motherboard. Some computers also have memory chips that are plugged into an *expansion board,* which in turn is plugged into one of the expansion-card slots of your motherboard. RAM is used to store program instructions and data temporarily, prior to or following processing.

The lines on the system bus are divided into three parts—*data bus* lines, *address bus* lines, and *control bus* lines. Data bus lines are used to move program instructions and data between RAM and your system's CPU. Address bus lines tell the CPU where to find specific sections of program instructions or data within the memory chips. Control bus lines tell the CPU whether it is supposed to be reading information from memory into the CPU or writing processing results from the CPU back to memory. (The control lines also perform other control and monitoring functions, but these functions are beyond the technical scope of our book.)

All Intel CPUs are not created equal, and much of the inequality involves the number of bus lines available with each CPU. That's why it's important for you to recognize the capabilities of the system bus on different computers. Generally, the more lines on a bus, the better. Windows provides program and memory management features on the basis of the number of bus lines in a CPU. The more bus lines that are available, the easier it is for Windows to transfer program instructions and data to and from the CPU. Actually, the category of processor chip in your system (286, 386SX, 386DX, or 486) also determines what Windows can and cannot do.

The Data Bus: A Helpful Analogy

Since the data bus lines are used to move program instructions and data between the CPU and RAM, you should understand what

these bus lines do. Here's an analogy that might make the data bus's purpose clear.

Imagine a city that has a rather bizarre rapid transit system. In this city, each highway has eight one-way lanes, and each lane has its own line number. One bus can travel down each lane (line) at a time. To add foolishness on top of this complexity, consider that each bus can carry *only one passenger at a time.*

So, to move passengers from one destination to another, eight vehicles arrive at a bus stop simultaneously, pick up one passenger apiece, and carry their passengers—each vehicle traveling at the same speed—to the same destination, as illustrated in Figure 1.1.

It doesn't sound like a very efficient approach in our scenario, but when data and program instructions are all traveling from a common departure point in memory to a common destination (within the CPU), it works well enough.

In the data lines of a CPU bus, each "vehicle" (line) carries one bit (one binary digit—0 or 1) at a time. The number of lines in a bus that can be used to transfer data between the CPU and memory determines the *data width* of the CPU. The actual *system bus width* (total number of lines in the bus) is larger than the data width of the CPU. But the other lines (the address and control lines) have different jobs to perform. (We'll explain how the address lines are used in the next section.)

An 8088 (XT) processor (ancient by today's standards) has an eight-line data bus, so it can transfer up to eight bits (one byte) of data or program instructions at a time between memory and the CPU. The 8088 is said to be an eight-bit processor because it uses an eight-bit data width.

The 8086 (also now obsolete) and 80286 (AT) processors have 16 data lines within the system bus and thus can transfer 16 bits of data and program instructions at a time between memory and the CPU. As you might guess, 80286 systems are said to use a 16-bit architecture. The 80386SX processor also has 16 data lines. However, a 386SX is said to support 32-bit processing because its general-purpose registers each can store 32 bits at a time. We'll explain this subtle distinction a bit later in the chapter.

An 80386DX (a "true" 386) or 80486 processor can transfer 32 bits at a time between memory and the CPU. This 32-bit processing architecture is currently the fastest in use by personal computers. Table 1.1 shows the hierarchy among 80x86 processors.

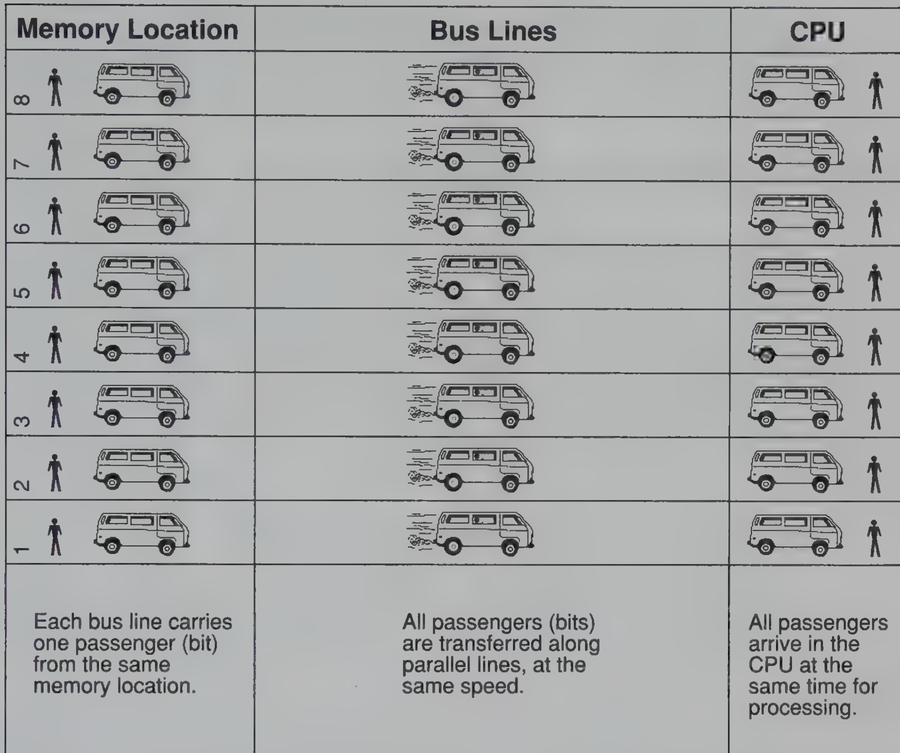


Figure 1.1 The data bus transfers a set of 8, 16, or 32 bits of data at a time along parallel lines.

Table 1.1 Data Bus Widths of 80x86 Processors

Processor	*Supports 8-bit Processing*	*Supports 16-bit Processing*	*Supports 32-bit Processing*
8086/8088	Yes	No	No
80286	Yes	Yes	No
80386	Yes	Yes	Yes
80486	Yes	Yes	Yes

Putting PC Memory to Use

Memory is really just an array of electronic storage locations. Each location can store one byte of data (eight bits) and has a unique value, or *address*, that the processor uses to find program instructions and data stored in that location.

Memory addresses are passed between memory and the CPU along the address lines within the system bus. Remember, the address lines are separate from the data lines in the bus, and serve a different purpose.

Memory Addressing Schemes

We'll explain the purpose and importance of address lines in a moment. But first, just to make sure the concepts are clear to you, here's a recap of the basics we've just explained. A processor can access, or use, only as much memory as it can address. An address, again, is just a pointer to a location in memory. The number of addresses available to a CPU is dependent on the number of address lines provided in the system bus. We'll explain this concept next.

The Address Bus: Another Analogy

The CPU uses the address bus to figure out where in memory a particular segment of code or data can be found. Here's an analogy that should help explain this concept.

Imagine that the passengers in our buses (from our earlier example) are all headed to a particular location in a particular neighborhood in RAM. Each set of eight vehicles arrives at a location in the neighborhood, carries a total of eight passengers to the same destination in the CPU, drops them off, then turns around to get more passengers. However, on their next trip to the neighborhood, the vehicles need to pick up passengers at a new location.

So, before each set of vehicles departs to pick up its passengers, all of the vehicles must be told where the destination will be within the neighborhood.

Now, in this hypothetical neighborhood, each house can have as many as 20 digits in its address. But there's a catch. Each digit in the address is limited to the numbers 0 and 1. (This is the binary number system, of course. Since computers only recognize *on* and *off* electrical signals, each *on* signal can be assigned the binary value 1, while each *off* signal can be assigned the binary value 0.)

Under this addressing scheme, the neighborhood can have up to 2^{20}, or 1,048,576 houses (two possible values for each of 20 binary digit positions within an address). In computer terms, this means that 20 address lines in the system bus can address up to 1,048,576 bytes, or 1MB, of storage space. Remember: each address can point to one byte (or one

Table 1.2 Addresses Available as a Function of the Number of Address Lines

Number of Digit Positions (Address Lines)	*Total Number of Addresses*	*Possible Address Combinations*
1	2 (1^2)	0,1
2	4 (2^2)	00,01,10,11
3	8 (2^3)	000,001,010,100,011,101,110,111
4	16 (2^4)	0000,0001,0010,0100,1000,1001, 1010,1100,0011,0110,1100,0111, 1011,1101,1110,1111
.	.	.
.	.	.
.	.	.
20	1,048,576 (2^{20})	00000000000000000000, 00000000000000000001, 00000000000000000010, etc.

character) in RAM. Table 1.2 shows how the number of available binary digits affects the number of available addresses.

In fact, 20 bits is the *address size* of the 8086/8088 processors, which means they have 20 lines in the address portion of the system bus, and for this reason can address up to *and no more than* 1MB of memory. This fact is crucial to the way Windows works with DOS applications. In a nutshell, here's why: most DOS applications were originally designed to use the features of the 8086/8088 processor standard, and limit their operations to the memory limitation of these processors. We'll have more to say on this limitation—lots more—later in this chapter.

HOT TIP

Round 1,024 Bytes to 1,000

Life in the computer lane is a lot easier if we round byte values to the nearest thousand. A storage address can point to a minimum of one byte of data, which is 8 bits, essentially a sequence of 8 binary (0 and 1) digits—that is, one of two possible values per bit position. For this reason, computer storage is often calculated in multiples of 8^2 (multiples of 64).

In turn, the nearest multiple of 64 to the number 1,000 is 1,024 (8 x 128). So we say that 1,024 bytes is equal to 1 kilobyte (1K), or approximately one thousand bytes. So 10K (roughly 10,000 bytes) is actually 10,240 bytes; one megabyte (1MB) is approximately one million bytes, or 1,048,576 bytes (1,024 x 1,024).

In terms of memory addressing capabilities, the 80286 processor is a significant advancement over the 8088, because it has an address size of 24 bits. In other words, the 80286 has 24 address lines and can address up to 2^{24} bytes of memory, or 16MB.

The 80386 processor, on the other hand, puts the 80286 process to shame with its address size of 32 bits (Table 1.3). That is, the 80386 processor can address up to 2^{32} bytes, or 4,294,967,296 bytes (4 gigabytes) of memory. (A gigabyte is 1 billion bytes.)

The point to keep in mind is that each processor has limitations on the amount of memory it can address. As a result, data and program code can be stored only in locations that the CPU can address. As you'll soon see, Windows can work with all three processor types—8088/8086 (although Windows 3.1 does not run on these machines), 80286, and 80386/80486 CPUs, and can switch its operations to emulate a processor other than the one actually installed on your motherboard. This feature has profound implications for the way you can use memory under Windows.

The Importance of Registers

To understand the limitations and advantages of different processors, we need to review some facts about CPU registers and the 80x86 (Intel standard) memory addressing scheme.

A *register* is a storage area *within* the CPU (not in RAM). The storage capacity of registers is minuscule by comparison with RAM storage capacities. For instance, even the largest-capacity registers in a 286 processor can each store no more than 16 bits at a time. However, registers are extremely valuable because they allow data and program instructions to remain within the CPU during processing, rather than having to be continually transferred between the CPU and RAM.

Table 1.3 Memory Addressing Capabilities in the 80x86 Series

Processor	*Address Lines*	*Total Addresses*
8086/8088	20	1,048,576 (1MB)
80286	24	16,777,216 (16MB)
80386	32	4,294,967,296 (4 gigabytes)

Remember our analogy of buses and bus architecture; all that transferring of instructions and data between the CPU and memory takes time. And, for a computer, a fraction of a section is comparable to a day or more in the lives of us non-machine mortals. In fact, the CPU parcels out processing time in *nanoseconds,* or billionths of a second.

HOT TIP

Define Your System's Clock Speed with Registers

Ever wonder what your system's *clock speed* really represents? It's an important term, since computers are sold on the basis of processor type and clock speed.

Your CPU has its own clock that fires pulses on a regular basis. Each pulse triggers a processing step required to complete a machine instruction. Several different steps (actions within the CPU) might need to be performed to complete one machine instruction.

The amount of time between two pulses of the CPU clock is called one clock *cycle. A clock cycle is the amount of time required to transfer a value from one register to another within the CPU.* The number of clock cycles (pulses) that occur each second is used to set the speed (pulse rate) of the CPU clock. Specifically, clock speed is measured in megahertz (MHz), or millions of cycles per second. So, a CPU clock that operates at 33 MHz fires 33 million pulses per second. In turn, the amount of time between each cycle is measured in nanoseconds (billionths of a second).

Processors that have the same basic design (such as a 386) can have different models—each with slightly more or slightly less efficient design features. These slight differences can substantially affect the speed at which values can be transferred between registers. For instance, although all 386 CPUs have 32-bit general registers, their clock speeds can vary. A 33 MHz 386 computer can process instructions at twice the speed of a 16 MHz 386 machine.

Each CPU has a specific number of built-in registers; some are *general-purpose registers,* which means they can be used to store instructions for several different kinds of processing operations. Other registers are designed to store data or instructions that serve a particular processing purpose.

The processing power of a CPU is limited in many ways by the size of its general-purpose registers. The size of the 8088, 8086, and 80286 chips' general registers is 16 bits. In an 80386 or 80486 processor, the general registers are 32 bits.

The size of a system's general registers is especially important when you consider how the 80x86 family of processors calculates memory addresses. A specific *segment register* called CS (for Code Segment) is used to specify the starting location of a region in memory where a chunk of program instructions can be stored. This region of memory is the *code segment.*

In an 80x86 processor, the actual memory location of a particular program instruction is calculated by finding out how far the instruction is from the start of the entire code segment. This distance is called the *offset.* Offsets are stored in general-purpose registers.

Now, here's where register size becomes a serious problem. The total number of offsets (actual memory addresses) that can be represented in a 16-bit register is 2^{16}, or 65,536. So, on a 286 processor, the total number of addressable bytes in a code segment can't exceed 65,536 bytes, or 64K. In other words, the largest chunk of program instructions that can be readied for processing in one fell swoop is a 64K segment. Therefore, programs designed for 8-bit and 16-bit machines have to be designed so that code segments no larger than 64K can be swapped into and out of memory.

Now enter the 386 processor, which has 32-bit general registers. With a 386 (or 486) machine, the largest possible offset is 2^{32}, or 4 gigabytes. So, theoretically, a code segment for a 386 processor can be as large as 4 gigabytes—the total available size of memory! Such a large code size blows the lid off of the program limitations that the 8088/8086 and 80286 processors have to abide by—and offers tremendous possibilities for multitasking, which allows the CPU to manage processing for two or more applications simultaneously. Table 1.4 sums up the CPU features that we've discussed so far.

Note: Even though an 80386SX CPU has a 16-bit data bus (16 data bus lines), its general registers can store 32-bit instructions. Since processing speed is technically a measure of how fast values can be transferred between registers within the CPU, a 33 MHz 386SX can internally process a 32-bit instruction as fast as a 386DX. So, 386SX systems are considered

Table 1.4 CPU Characteristics in the 80x86 Series

CPU Type	*Total Addressable Memory*	*Data Bus Width*	*General Register Size*
8086/8088	1MB	8 bits	8 bits
80286	16MB	16 bits	16 bits
80386SX	4 gigabytes	16 bits	32 bits
80386DX	4 gigabytes	32 bits	32 bits

to be legitimate 32-bit processing machines. However, in real time, a 33 MHz 386SX is at best half as fast as a 33 MHz 386DX, because only 16 bits, rather than 32 bits, can be transferred at a time between the CPU and RAM.

Understanding PC Memory

Now that we've got some of the fundamentals out of the way, it's time to take hold of your PC's "mind"—that is, memory usage. In fact, you need to have a strong understanding of PC memory to understand how Windows can be optimized.

Why Memory Management Is so Important with Windows

The first responsibility of Windows' built-in memory managers is not to make optimal use of memory, but to prevent conflicts with software.

One of the basic features of Windows is its ability to continually juggle parts of programs and even parts of multiple programs between your hard disk and RAM—and between different parts of memory. So Windows has to work hard to make sure that different applications and drivers don't try to access the same memory at the same time. For this reason, memory optimization takes a back seat to the protective safeguards required to keep multiple programs running without a hitch.

The good news is that you can manipulate memory management in a number of ways to optimize the performance of Windows—*if* you understand how your hardware and software (including Windows) use memory.

Using the Hexadecimal Number System to Address Memory

Although we've explained that a memory address just indicates a storage location in RAM, there's a technical drawback here. Memory addresses are typically represented in a hexadecimal (or just hex) numbering system.

Hexadecimal is a base-16 numbering system that uses the ten digits 0 through 9, and the six letters A through F to represent values. Keep in mind that if we were to use binary notation to refer to every string of bits in memory, you would have to get used to seeing big numbers like 1001110010101110. This scheme would keep your fingers snugly wrapped around the Tylenol bottle.

Hexadecimal numbering makes it possible to represent lengthy memory addresses in a shorthand notation. In hex code, the digits 0 through 9 represent values that correspond to the base-10 numbering system. For a binary value that would correspond to the decimal number 10, we use the value A. The value B, in turn, corresponds to a binary number that would be represented in decimal numbering as 11. We continue this approach through to the letter F, which is equivalent to the base-10 value 15. For instance, the binary value 111011011110101111001110 can be represented in hex format as EDEBCE. Because hex code uses 16 values rather than the 10 in the decimal numbering system, hex numbers let us represent large decimal numbers with just a few hex digits. For example, the hex number EDEBCE becomes 15,592,398 in decimal format. Since PC memory address values can run into the billions, hex numbering has become a standard method for representing both memory addresses and the size of an application stored in memory.

If all of this seems confusing, don't worry. You don't really have to understand hex formats to understand PC memory. However, you will have to get used to looking at hex numbers when you examine memory addresses, which are universally represented in hex format. You can find a detailed explanation of the hexadecimal numbering system in just about any book on assembly language programming. If you want more practice in comparing hex numbers and decimal numbers, take a look at the following two tips.

HOT TIP

Easy Conversion of Hex Values to Decimal

The hexadecimal numbering system uses the same *place value* concept that you studied in grade school when you learned the decimal numbering system. (Remember the ones place, the tens place, the hundreds place, and so on?) The difference, though, is that each place column in hexadecimal is a power of 16, rather than 10—the ones place, the 16s place, the 256s place, the 4096s place, and so forth.

So, converting hex numbers to their decimal equivalents really isn't too difficult if you remember your place value math and if you keep in mind that 16 hexadecimal numbers can be used in each place column, not just 10. In the following example, we'll convert the hex value EDEBCE to a decimal number:

```
No. of      No. of   No. of   No. of   No. of  No. of
1,048,576s  65,536s  4096s    256s     16s     1s

E (14)      D (13)   E (14)   B (11)   C (12)  E (14)
_______________________________________________________
14,680,064+851,968 + 57,344 + 2,816 + 192 +  14 = 15,592,398
                                                  (decimal)
```

Yes, this procedure is a pain, and yes, you shouldn't have to go through this type of conversion whenever you see a hex number. In fact, as long as you have the Windows calculator running, you don't have to make these conversions yourself. And many hand-held calculators also have features for converting between hexadecimal and decimal formats. The conversion math is still good to know, though, if for some reason you have to rely on pencil, paper, and sheer brainpower to convert addresses.

HOT TIP

Use the Windows Calculator to Convert Hex Values

The Windows Calculator, available from the Accessories group window, includes useful features (in Scientific view) that allow you to convert hex values to decimal values, and vice versa. To use this feature:

1. Start the Calculator.
2. Select Scientific from the View menu to view the Scientific version of the calculator.
3. Enter a hex or decimal value.
4. If you want to convert a hex value to decimal, click the Dec button. If you want to convert a decimal value to hexadecimal, click the Hex button.

Types of Memory

Memory is not memory is not memory—and that's not a misprint. We just need to stress that your system is capable of recognizing different types of RAM. And the types of RAM available on your system determine the ways in which you can use Windows.

So let's begin with a review of the available types of RAM. PC memory can be divided into five categories:

- Conventional memory
- Upper memory area
- Expanded memory
- Extended memory
- Virtual memory

All PCs have conventional memory. Just about all PCs that can run Windows 3.1 also have an upper memory area and some extended memory. Expanded memory is used on some older AT-class (286) computers that were designed before extended memory chips became affordable. Virtual memory is actually hard disk space that 386 and 486 computers (and, to a limited extent, 286 computers) can use as though it was extended memory. We'll take a detailed look at all five types of memory in the next sections.

Conventional Memory

DOS still rules the roost with PCs, regardless of whether Windows is running on a particular machine. Windows and all other applications must work cooperatively with DOS and its—yes—limitations. Much of the Windows architecture is designed specifically to give a "face lift" to the limitations and lackluster personality of the aging DOS. In a memory sense, when DOS "forgets," Windows "reminds."

DOS was originally designed for computers that used an 8086 or 8088 processor, which could address up to 1MB of memory. In the relatively prehistoric years when the 8086/8088 processor standard was in use, 1MB seemed like more than enough memory to run programs.

Since DOS was developed with this standard in mind, your operating system—regardless of the version—can directly address only 1MB. The lower 640K of your computer's memory is called *conventional memory*, or sometimes *DOS memory*. The vast majority of DOS applications have to be run within the 640K conventional-memory limit.

Upper Memory Area

If DOS can address up to 1MB of memory, why can't programs use the 384K of addressable memory above 640K (up to 1024K, or

1MB)? By default, DOS reserves the upper 384K portion of memory for storing video driver programs, code from the ROM BIOS (the built-in part of the operating system that controls basic input and output operations), and drivers for other hardware adapters. In other words, DOS says to your applications: this is our domain—keep out!

The reserved range of addresses from 640K to 1024K is called the *upper memory area (UMA)*. Figure 1.2 shows how the UMA is typically organized on PCs. Notice that the ROM BIOS is always loaded into the top end of the UMA (from 960K to 1024K, or F0000 to FFFF0 hex). Blocks for video memory are reserved from the bottom up, with different video cards typically using different address ranges in the UMA.

Figure 1.2 reveals an interesting feature of the UMA: much of this "reserved" area is not really reserved for anything! This fact holds a great deal of potential for optimizing Windows, as you soon will see.

HOT TIP

What's the ROM BIOS?

ROM stands for Read-Only Memory, and refers to chips within your computer that contain permanent instructions ("burned in" to the circuitry).

One part of ROM contains instructions that control basic operations of your input and output devices (especially your keyboard and monitor). In fact, BIOS stands for Basic Input/Output System. Needed instructions in your ROM BIOS are stored in the UMA when your computer boots, and these instructions control certain operations of your peripheral hardware devices.

HOT TIP

Troublesome Aliases of the Upper Memory Area

You'll read or hear several different terms used to refer to the UMA—including *reserved memory* and the *adapter segment*. Those two terms have a consistent meaning. However, two other terms sometimes used to refer to the UMA, *upper memory block* and *high memory*, are somewhat misleading since they have multiple meanings. When you see either of these two terms in magazine articles and books, make sure they actually refer to the full range of addresses in the UMA, and not to some other memory concept. Throughout this book, we use the term *upper memory area* (or simply *upper memory*) to refer to the range of addresses from 640K to 1024K.

	Address Range (Hexadecimal)	Address Range (Decimal)				
Upper Memory Area (UMA)	FC000-FFFF0	1008-1024K	ROM BIOS			
	F8000-FBFF0	992-1007K				
	F4000-F7FF0	976-0991K				
	F0000-F3FF0	960-975K				
	EC000-EFFF0	944-959K	PS/2 VGA			
	E8000-EBFF0	928-943K				
	E4000-E7FF0	912-927K				
	E0000-E3FF0	896-911K				
	DC000-DFFF0	880-895K	Reserved but often unused			Often available but sometimes used by network cards
	D8000-DBFF0	864-879K				
	D4000-D7FF0	848-863K				
	D0000-D3FF0	832-847K				
	CC000-CFFF0	816-831K				
	C8000-CBFF0	800-815K	8514/A cards			
	C4000-C7FF0	784-799K		VGA (except PS/2s)		
	C0000-C3FF0	768-783K			EGA	
	BC000-BFFF0	752-767K	EGA/VGA text and low res.	Hercules cards (page 2)	CGA cards	
	B8000-BBFF0	736-751K				
	B4000-B7FF0	720-735K	MDA cards	Hercules graphics cards (page 1)		
	B0000-B3FF0	704-719K				
	AC000-AFFF0	688-703K	High-resolution EGA/VGA memory			
	A8000-ABFF0	672-687K				
	A4000-A7FF0	656-671K				
	A0000-A3FF0	640-655K				
Conventional Memory	9C000-94C00	624-639K	COMMAND.COM, TSRs, and other DOS applications			
	98000-9C400	608-625K				
	94000-97C00	592-607K				
	•	•				
	•	•				
	•	•				

Figure 1.2 The upper memory area (UMA) is typically reserved for video drivers, ROM BIOS code, and other hardware driver programs.

Expanded Memory

Expanded memory provides a way for DOS to use more than 1MB of memory by switching data and program instructions from a separate memory board into the UMA.

Several years ago, to overcome the addressing limitations of 8086/8088 computers, a consortium of software and hardware engineers from Lotus, Intel, and Microsoft cooperated to develop a standardized way to add memory to those early PCs. Under this approach, called the LIM 3.2 standard, an expansion board could be plugged into the motherboard of the PC, and the *expanded memory* on this board could be accessed through a utility program called an *expanded memory manager (EMM)*. The trick, though, was to get DOS to recognize the memory. This problem was solved by having the expanded memory manager access unused addresses in the upper memory area.

Here's how it works: an application that is specifically designed to use expanded memory makes a request for additional memory. The EMM intercepts the request and moves the application's data into an unused block of UMA. Specifically, the data is moved in 16K sections called *pages*.

Under the LIM 3.2 standard, an EMM can move as many as four pages (64K) of data into the UMA (so that the data can be located by DOS)—provided the pages can all be stored in contiguous (adjoining) locations.

This paging feature means a single, contiguous 64K block of unused addresses, called a *page frame*, must be available in order to move four pages of data from the expansion board into the UMA. This concept is illustrated in Figure 1.3.

EMM386.SYS is an example of an expanded memory manager, and is supplied with DOS 5 and with Windows. Other third-party expanded memory managers are available, including Quarterdeck's QEMM-386 and Qualitas' 386MAX. We'll discuss these memory managers in Chapter 3.

HOT TIP

Windows Supplies Its Own Version of EMM386.SYS

Although DOS 5 includes a version of the EMM386 expanded memory manager, the Windows 3.1 version of EMM386 is a more efficient "tweaked" manager.

When you install Windows 3.1 under DOS 5, the setup program will automatically adjust the DEVICE line in your CONFIG.SYS file so that EMM386.SYS loads from the WINDOWS directory, rather than from the DOS directory. (The CONFIG.SYS file is explained in more depth in Chapter 2.)

In any case, if you use EMM386.SYS, load the version supplied with the most current version of Windows. For instance, if Windows is stored in the WINDOWS directory of drive C of your hard drive, the DEVICE line below should appear in your CONFIG.SYS file:

```
DEVICE=C:\WINDOWS\EMM386.EXE (other parameters might be included
                              here)
```

and the following line *should not* appear:

```
DEVICE=C:\DOS\EMM386.EXE (other parameters might be included
                          here)
```

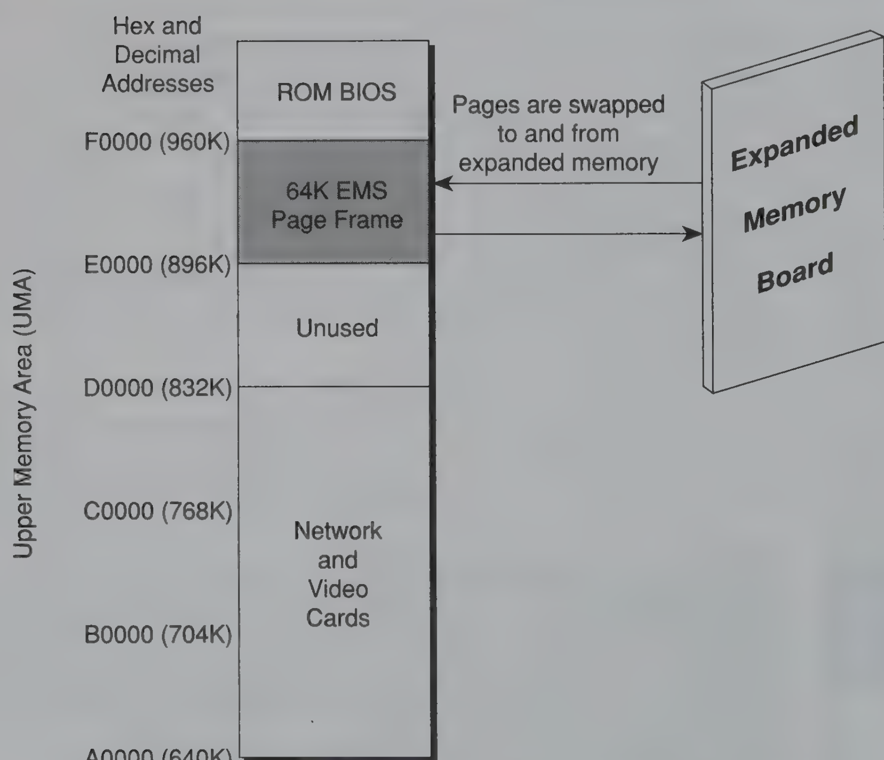


Figure 1.3 Expanded memory is typically supported by setting up a 64K page frame in the UMA and then swapping data between the expansion board and the UMA.

As you might suspect, this "expanded" approach to memory enhancement is a little like plowing a 40-acre field to grow a few dozen stalks of corn; not very efficient.

The Lotus-Intel-Microsoft group later developed an improved expanded memory management standard, called LIM 4.0, that overcomes some of the limitations of LIM 3.2. The basic differences were that LIM 4.0 could move up to 64 pages into noncontiguous locations and—more importantly—could use a method called *backfilling* to store pages in unused conventional memory as well as in the UMA. Also, a contiguous 64K page frame was not required.

Unfortunately, the LIM 4.0 standard got caught in the transition from 8086/8088 machines to 80286 machines. The newer 80286 machines could recognize up to 16 megabytes of memory, mostly *extended memory* (discussed in the next section). Many software companies didn't bother to upgrade their applications to use the LIM 4.0 standard (and instead stuck with their LIM 3.2 code and designed newer versions to use extended memory). The new 80286 chip was making expanded memory obsolete.

So why is expanded memory even worth discussing if it's yesterday's news? Well, it's not entirely obsolete. A few very popular DOS applications, including some CAD programs and some versions of Lotus 1-2-3, use or even require expanded memory (typically under the LIM 3.2 standard). Also, most expanded memory managers, including the EMM386 manager provided with Windows, strangely enough can provide excellent optimization features for systems that don't have expanded memory, as we'll explain later.

HOT TIP

Expanded Memory Managers Work Differently on 286 and 386 Processors

If your system uses a 286 processor, you can only use expanded memory if you have an expanded memory add-on board plugged into your motherboard. However, if you have a 386 or 486 processor, EMM386 (and other expanded memory managers) can actually be used as an expanded memory *emulator*. That is, EMM386 uses the features of the 386 processor to make extended memory look like expanded memory to DOS applications that require true expanded memory.

Note: Under Windows, expanded memory emulation is only available when you are running in enhanced mode.

Extended Memory

Extended memory is available for virtually all 286, 386, and 486 computers and allows applications to directly address memory above 1024K (1MB).

Remember, the 286 processor can address up to 16MB of memory, while the 386 and 486 processors can address up to 4 gigabytes of memory. Most of this memory is extended memory, which starts at the 1MB level, where conventional memory ends.

Unlike expanded memory, with extended memory it isn't necessary to move page frames in and out of the UMA so that DOS can recognize the addresses. Instead, applications that use extended memory rely on some form of *DOS extender* technology, which uses an *extended memory manager* to supply requested memory directly to the application—effectively bypassing DOS when requesting the memory. So, extended memory can be managed much more efficiently than expanded memory. HIMEM.SYS, supplied with DOS 5 and with Windows, is an example of an extended memory manager that uses a DOS extender technology.

Extended memory is probably the single most important hardware requirement for Windows. The more extended memory you've got on your system, the faster and more efficiently Windows and Windows applications will run.

HOT TIP

Windows Comes Equipped with Its Own HIMEM.SYS

Although DOS 5 includes a HIMEM.SYS extended memory manager, the Windows 3.1 version of HIMEM.SYS is a more efficient "tweaked" manager.

When you install Windows 3.1 under DOS 5, the setup program will automatically adjust the appropriate DEVICE statement in your CONFIG.SYS file so that HIMEM.SYS loads from the Windows directory, rather than from the DOS directory. (The CONFIG.SYS file is explained in more depth in Chapter 2.)

In any case, always make sure you are using the HIMEM.SYS file that is supplied with the most current version of Windows. For instance, if Windows is stored in the WINDOWS directory of drive C of your hard drive, this DEVICE line should appear in your CONFIG.SYS file

```
DEVICE=C:\WINDOWS\HIMEM.SYS
```

and the following line *should not* appear:

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

Understanding VCPI and DPMI Extended Memory Specifications

VCPI and DPMI are the two major extended memory specifications used by DOS extenders. But these two specifications have very different capabilities.

VCPI, created by Phar Lap Systems, stands for Virtual Control Program Interface and provides a way for DOS programs that require extended memory to coexist with expanded memory managers.

In other words, VCPI allows you to run a DOS extender (so that a DOS application that uses extended memory can be loaded) while an expanded memory manager is also running.

Note: VCPI only works on 386 systems and only allows one application to use extended memory at a time.

DPMI, a creation of Microsoft, stands for DOS Protect Mode Interface and serves essentially the same interface purpose as VCPI. However, DPMI allows DOS extenders to load multiple programs in extended memory, and supports the 286 processor as well as the 386 and 486 processors.

DPMI was created primarily to allow Windows to take advantage of the virtual memory capabilities of the 286 and 386 processors.

Note: HIMEM.SYS runs all DOS applications that conform to the DPMI specification, but might not work with some VCPI-based applications.

High Memory Area

Most extended memory managers, including HIMEM.SYS, configure the first 64K of extended memory in a way that makes the memory addressable by DOS.

This first 64K of extended memory is called the *high memory area (HMA)*. Essentially, the HMA adds 64K to the 1MB pool of memory that DOS can address.

DOS 5 can load itself (or at least most of itself) into the HMA when your computer boots. In fact, this is the chief value of the high memory area. With most of the command portion of DOS loaded in the HMA, you free up more than 60K of conventional memory for use by other applications.

Cache Memory

A disk cache is an area of RAM (usually extended memory) that has been temporarily mapped to emulate the sector and track geometry of a disk.

A disk cache is set up by a utility program when your computer boots. Both DOS 5 and Windows 3.1 install the SMARTDrive cache utility.

HOT TIP

Always Use the SMARTDrive Program Supplied with the Current Version of Windows

The SMARTDrive version 4.0 supplied with Windows 3.1 offers some safeguards that aren't available with the DOS 5 or Windows 3.0 version of SMARTDrive, so you should use the version supplied with Windows 3.1. This SMARTDrive version is installed automatically in the WINDOWS directory when you install Windows 3.1. So your AUTOEXEC.BAT file should contain a line similar to the one below:

```
C:\WINDOWS\SMARTDRV.EXE (other parameters can be included here)
```

SMARTDrive 3.1, which is supplied with DOS 5, requires a line in CONFIG.SYS. If your CONFIG.SYS file contains a DEVICE line for SMARTDrive, this is probably the older, 3.1 version. If you are running Windows 3.1, you should delete this line from your CONFIG.SYS file. Then, make sure your AUTOEXEC.BAT file contains a statement to load SMARTDrive from the WINDOWS directory, *not* from the DOS directory.

So what benefits do you receive from caching? Plenty. Every time an application requests data or program instructions from your hard disk, the mechanical arms and heads on your hard drive are activated. In the realm of personal computers, anything "mechanical" should be avoided like a bad flu. Why? Mechanical activities are usually *at best* hundreds of times slower than activities that use only the electrical currents (circuits) of your computer.

A RAM caching utility like SMARTDrive takes the data and program instructions that have been most recently read from your hard disk and stores them in the cache area of extended memory. Then, if an application should request these same program instructions or data again, they are supplied from the cache rather than from the hard disk. This approach saves time, because the mechanical components of the hard drive don't have to be activated. SMARTDrive is explained in more depth in Chapter 2.

Note: Don't confuse a disk cache like SMARTDrive with a memory cache that is actually part of the CPU. A CPU-based cache serves a similar function as a disk cache, but is built into the motherboard of the computer and is controlled by the ROM BIOS—*not* by a TSR or other cache utility program. SMARTDrive will work in conjunction with (not in place of) a memory cache.

Virtual Memory

The last type of memory we need to discuss—virtual memory—actually isn't memory at all. It's just hard disk space.

One of the features of the 286 and 386 processors is their ability to make hard disk space look like extended memory—in other words, *virtual memory.* Figure 1.4 illustrates the different categories of memory that we've described so far. Virtual memory, where the code and data are swapped between extended memory and available hard disk space, is made possible by an operating mode built into the 286 and 386 processors called *protected mode.* The word "protected" refers to the ability of the processor to protect areas of physical memory from being used by more than one program at a time.

These protection features become important when you recognize that protected mode, in combination with virtual memory capabilities, makes it possible for multiple applications to run at the same time—a fact that has far-reaching implications for all Windows users.

But there is a world of difference between the 286 processor's protected mode (called *standard mode* by Windows), and the 386 processor's protected mode (called *enhanced mode* by Windows). In fact, an important key to understanding and optimizing Windows lies in understanding how Windows uses the *operating modes* available with the 80x86 family of CPUs.

As an experienced Windows user, you've no doubt heard the terms *protected mode, real mode, standard mode,* and *enhanced mode* liberally tossed about. Although you might have a good general understanding of the capabilities of different operating modes, you can improve your ability to use Windows by learning how and why each CPU uses its available operating modes. Table 1.5 shows which CPUs support which

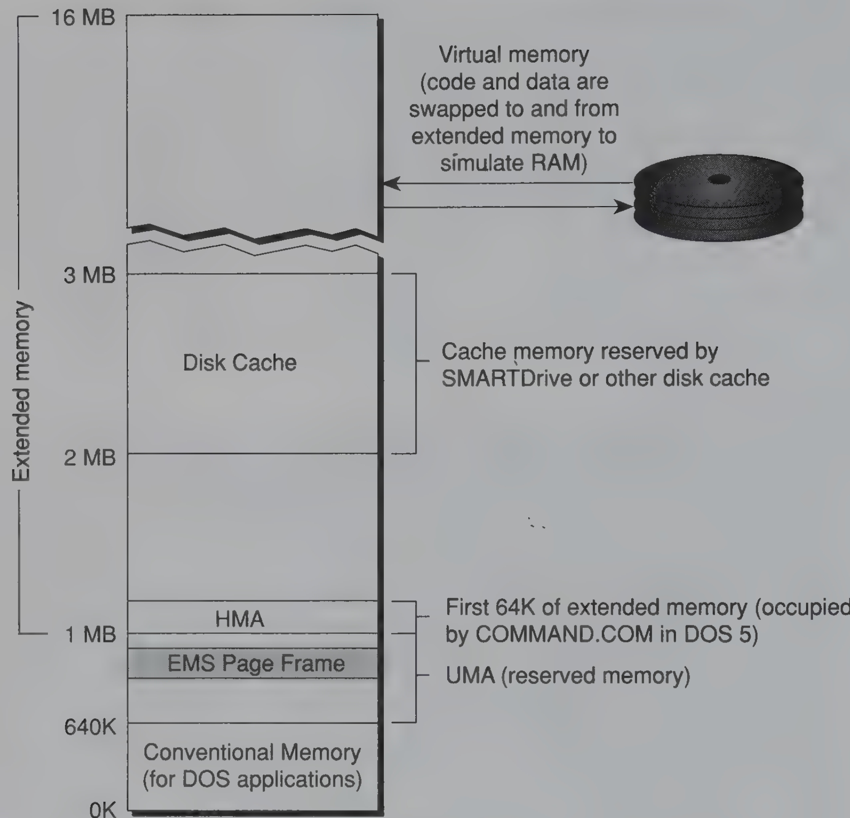


Figure 1.4 This diagram demonstrates the hierarchy and types of memory discussed in this chapter and used by Windows 3.1.

Table 1.5 Support for Windows in the 80x86 Series

	8086/8088	*80286*	*80386*
Real-mode support	Yes	Yes	Yes
Protected-mode support	Yes	Yes	Yes
Windows standard-mode support	No	Yes	Yes
Runs Windows 3.1	No	Yes	Yes
Expanded memory support	Yes	No†	Yes
Extended memory support	No	Yes	Yes
Virtual memory support	No	Yes	Yes
Virtual 86 mode support	No	No	Yes

† with a few exceptions

categories of memory and operating modes. We'll take a look at these operating modes next.

Windows and Real Mode

Real mode is the only operating mode available with the 8088/8086 CPUs. Although Windows 3.0 ran on these machines, Windows 3.1 doesn't. This limitation is by design, and makes sense when you consider the inherent limitation of real mode.

When Windows 3.0 was released, there were still a lot of 8086/8088 machines in use in business and in homes. Microsoft didn't want to lock out users of these machines from the benefits of Windows, so Windows 3.0 supported real mode. But this was probably more of a marketing move than anything else.

Many Windows applications designed explicitly for Windows 3.0 made use of the additional registers and other processor features available with the 286 CPU. (Excel 3.0 is an example of an application that was designed for Windows 3.0, and must run in standard mode or enhanced mode.) This restriction prevented 8086/8088 machines from running these Windows applications (like Excel 3.0), even though Windows 3.0 itself was running.

Also, the 8086/8088 processors' 640K memory addressing barrier and the lack of a protection scheme to prevent programs from fighting over memory addresses meant that Windows could not have multiple applications open when running in real mode. The bottom line: running Windows 3.0 on anything less powerful than a 286 system was impractical to the point of being absurd.

So, Microsoft abandoned real mode with the release of Windows 3.1 in April, 1992. By this time, most users who wanted the kind of processing power that Windows could offer had upgraded to 286 or 386 systems. However, real mode still has meaning in Windows 3.1, chiefly because 80286, 80386, and 80486 machines can emulate real mode to run DOS applications.

Running Windows in Standard Mode

Standard mode is designed to take advantage of the extended memory addressing capabilities of the 286 CPU, as well as the 286's protected mode. In protected mode, the CPU enforces a set of read/write privileges that prevent one program from accessing memory space that is currently occupied by another program.

To understand both the features and limitations of the 286 CPU's protected mode, you will need to know a few facts about the way the 286 addresses and manages memory.

We've already explained that the 286 can address up to 16MB of memory. The first megabyte of addressable memory is the first 640K of conventional memory plus the 384K of upper memory. The remaining 15MB can be extended memory or hard disk space (virtual memory). To access memory above the first megabyte of addressable memory, your system needs to use an extended memory manager. Both DOS 5 and Windows supply the extended memory manager HIMEM.SYS.

HOT TIP

The 286 *Does* Provide Virtual Memory Capabilities

Despite what you might have heard, the 286 processor *can* support virtual memory, and, in fact, does support virtual memory under Windows. Keep in mind, though, that we're referring to the general concept of swapping programs and data to the hard disk temporarily.

Microsoft uses the term "virtual memory" only to refer to the virtual paging capabilities of the 386 processor, with Windows running in enhanced mode. This capability allows you to run Windows and DOS applications simultaneously. Microsoft's virtual memory approach is only available on 386 machines.

The 286 CPU enforces the code segment + offset addressing technique to swap segments from physical memory to disk. Since offsets in the 286 can address a code segment no larger than 64K, swapping of code to disk is limited to one 64K segment at a time. This is a serious limitation, since Windows applications are often 640K or larger. Despite this limitation, the addressing capabilities of the 286 chip do—within some important limits—allow applications to be swapped to disk (virtual memory).

With the 8086/8088 processors, all of the code segments required to run a program have to be placed in contiguous addresses, making it impossible to swap code segments to disk. The 286 processor offers a significant advancement over the 8086/8088 processors by allowing different segments of a program to be placed in noncontiguous locations—including locations on the hard disk.

Windows uses standard mode differently for Windows applications and DOS applications. The following sections explain why.

Running Windows Applications in Standard Mode

All Windows applications must be written with the 286 protected mode (standard mode) in mind.

In particular, the code in any Windows application can be identified by its *segment attribute.* These attributes fall into three categories:

- Movable code segments
- Discardable code segments
- Swapable code segments

A movable code segment (sometimes called a *relocatable* code segment) can be moved from one place in memory to another. Movable code segments are typically moved between *conventional memory* and *extended memory* when DOS applications (which must use conventional memory) are started.

A discardable code segment is not essential to the operation of the program, so the address space it occupies can be overwritten by more important code segments. The discardable segment can be read from disk later if it is needed again.

A swapable code segment can be swapped to disk temporarily if no physical memory is available, and then read back to memory as soon as physical memory becomes available.

You can think of these three types of code segments as chunks of a program that can be juggled between conventional memory, extended memory, and the hard disk as necessary to keep multiple programs running under limited memory conditions.

Running DOS Applications in Standard Mode

Since Windows now has a built-in DOS extender technology (HIMEM.SYS), Windows can store DOS programs in extended memory—but only if a DOS application has been written specifically to request extended memory from a DOS extender.

The problem is that most DOS applications, which were designed to run on 8086/8088 processors, don't request extended memory. So Windows standard mode has a dilemma: an open DOS application can either be run in conventional memory or can be swapped—in total—to hard disk temporarily.

Standard Mode Performs Task Switching

***The technique that Windows uses to swap DOS applications to and from conventional memory is called* task switching**

When you start Windows in standard mode, Windows loads a file called DOSX.EXE, which sets up a portion of extended memory that emulates real mode. This emulated real-mode environment allows DOS applications to operate, even though Windows is actually running in protected (standard) mode.

When you are running Windows applications in standard mode, the DOSX real-mode environment remains essentially dormant. Windows uses a file called WSWAP.EXE to swap the different Windows code segments between conventional memory, extended memory, and the hard disk.

However, when you start a non-Windows (DOS) application, Windows switches to the DOSX environment to emulate real mode, forcing any open Windows applications into the background. Also, standard mode only allows one DOSX environment to operate at a time. For this reason, only one DOS application can be running at a time. (Remember, real mode does not allow multiple programs to run simultaneously.)

In standard mode, each time you switch from a DOS application back to the Windows environment, Windows swaps the DOS application and its data to your hard disk, within *application swap files.* These swap files are temporary, and are deleted whenever you exit Windows.

HOT TIP

Maximize Your Application Swap File Space

Windows can only swap a DOS application to the hard disk if free space is available on the disk. If you run in standard mode and want to maximize the number of DOS applications you can have open at one time, delete any unnecessary programs and data files from your hard disk. This will free up additional space for storing application swap files.

ON DISK

Use WINZIP to Compress and Uncompress Files

Several programs are available for *compressing* files stored on your hard disk. These programs use a special compressional algorithm to reduce the amount of space required to store files on disk. Compression programs can also uncompress files so that they can be read from disk.

The *Windows Insider Disks* include a set of compression/uncompression programs, collectively called WINZIP, designed specifically for use with Windows. You can use this program to compress the programs and files on your hard disk, thereby increasing the amount of hard disk space available for swap files. You also must use WINZIP (or PKUNZIP from the DOS prompt) to uncompress most of the programs stored on the Insider Disks. The WINZIP programs are located in the WINZIP directory of the *Windows Insider Disks*.

Running Windows in Enhanced Mode

The 286 CPU is designed to support virtual memory by detecting when a code segment being loaded into memory can't actually fit within physical memory—in other words, when conventional and extended memory are full. When this happens, the operating environment that manages virtual memory (such as Windows) generates a *segment fault*.

When a segment fault occurs, the operating environment determines which code segment currently in memory can be swapped to hard disk to make room for the incoming segment of code. This approach is similar to the one used by Windows in standard mode to support multiple applications.

The 386 CPU makes it possible for you to use a more sophisticated virtual memory concept called *demand paging*. Under this approach, available hard disk space is divided into fixed-sized blocks called pages. Each page is 4K in size.

Windows uses a built-in virtual memory manager (VMM) to set up and manage the use of virtual memory pages. The VMM is assisted by an additional *pageswap* device that is built into the file WIN386.EXE.

HOT TIP

Use Swap Files in Enhanced Mode

The Windows VMM sets up a temporary swap file, called WIN386.SWP, but only if you have used the Virtual Memory dialog box (available from the Control Panel) to specify a temporary swap file.

By default, when you install Windows 3.1, a permanent swap file called 386PART.PAR is created in the root directory of your hard disk. A permanent swap file reserves available contiguous tracks and sectors for use as virtual memory. You cannot use this hard disk space for any other purpose unless you change the swap file type to Temporary or None in the Virtual Memory dialog box. We'll discuss swap files in more depth in Chapter 4.

Here's how demand paging works:

1. When Windows is started, the VMM divides physical memory (conventional and extended memory) into 4K *physical pages*, and divides available hard disk space into 4K *virtual pages*. The amount of available conventional and extended memory determines the number of physical pages that are created, while the amount of available hard disk space determines the number of virtual pages that are created. (Windows will not use all of your available hard disk space for virtual memory; some space will be ignored so that you can save or create other files during your Windows session.)
2. The VMM sets up a page table that maps physical-page addresses to virtual pages, and tracks which pages are currently in memory and which pages have been swapped to the hard disk's swap file.
3. When an application needs one or more pages that are not currently in physical memory, the VMM calls the pageswap device, which swaps physical pages to the hard disk to make room for the incoming pages.

Under this approach, Windows juggles the pages between physical and virtual memory to keep several Windows and non-Windows applications open and running simultaneously. The number of applications that can remain open depends on the total amount of conventional, extended, and virtual memory that is available.

HOT TIP

Determining Available Memory in Enhanced Mode

In Windows 3.0, you could only find out the total amount of free conventional, extended, and virtual memory by choosing About from the Program Manager's Help menu.

In Windows 3.1, however, you can find out the amount of available memory by choosing **About...** from the Help menu in *almost any Windows application.*

How the VMM Determines Which Pages to Swap to the Hard Disk

The page table also tracks which physical pages are accessed and which are "dirty" (altered). If a page in physical memory has the accessed attribute turned on, the VMM interprets this to mean an

application or some other process has referred to this region of physical memory since the page was first stored there.

If a physical page has the dirty attribute turned on, the VMM interprets this to mean some application or process has written to, or altered, the contents of that page since it was first stored there. A dirty page automatically implies the accessed attribute, since writing to the page qualifies as a "reference."

Windows uses these attributes to determine which physical pages can be swapped to the hard disk when virtual pages are requested. When such a request occurs, Windows searches the table for pages that have not been used recently. The intent is to keep pages in physical memory that are most likely to be used again, thereby reducing the number of page swaps that will be required later. Specifically, Windows searches for pages that do not have the accessed or dirty attributes turned on.

When Windows finds enough "clean" pages, it swaps them to virtual memory, and brings in the virtual pages requested by the application. If Windows can't find enough clean pages on its first pass, it will make additional passes and swap physical pages to virtual memory regardless of whether they have been accessed or read to during the current Windows session.

Using Virtual Machines

When you run DOS applications under Windows in enhanced mode, Windows creates a separate DOS environment in extended memory for each DOS applications.

Remember that all DOS applications that don't make use of DOS extender technology (and that's most DOS applications) must run in real mode. In standard mode, the DOSX.EXE program can set up one emulated real-mode environment at a time. This approach limits you to one DOS application running at a time under standard mode.

In enhanced mode, Windows can use a feature of the 386 CPU called *virtual-86* mode. Under this approach, the *virtual machine manager* that is built into the file WIN386.EXE can set up an emulated real-mode environment (the 8086 and 8088 processor environments) for each DOS application that you start. This environment is called a *virtual machine*, because DOS thinks that each virtual machine is running on a separate system, essentially a different 8086 or 8088 CPU.

HOT TIP

You Can't Page a DOS Application to the Hard Disk

Under Windows 3.0, all open DOS applications had to run in extended memory or not at all. In other words, you couldn't page part of a DOS application to the temporary or permanent swap file used as virtual memory. The reasons for this are somewhat complex and will be explained in Chapter 4.

For now, recognize that in Windows 3.1 you *can* page part of a DOS application to virtual memory, but *only* if the Use 32-Bit Disk Access check box (in the Virtual Memory dialog box accessed from the 386 Enhanced icon in the Control Panel) is turned on. This feature allows you to open and run DOS applications in enhanced mode even when extended memory is full.

HOT TIP

Enhanced Mode vs. Standard Mode Performance

With Windows 3.0, Microsoft advised you to run in standard mode if you would be using Windows applications exclusively. The overhead required to support virtual memory and virtual-86 mode (for DOS applications) caused enhanced mode to operate about 15 to 20 percent slower than standard mode on most machines.

Windows 3.1 is designed to use an increased amount of code that can exploit 32-bit addressing. For this reason, Windows enhanced mode runs faster than standard mode in almost all cases.

CHAPTER 2

Optimizing Windows at the DOS Level

Chapter 1 covered most of the basic concepts you need to know in order to make memory work for you when you run Windows. If you've finished digesting this information, you're probably ready to use some specific memory-management techniques.

Many of the terms used in this chapter are reviewed in Chapter 1. If you need to brush up on your knowledge of memory management concepts and terms, you might want to review Chapter 1 before you continue.

Interestingly, many of the memory management techniques for optimizing Windows must be done at the DOS level. Several of the more valuable optimization techniques are specifically designed to improve your ability to run as many virtual DOS machines as possible under enhanced-mode Windows. This chapter explains the memory management optimization techniques you can use at the DOS-prompt level.

Why Optimize Conventional Memory?

Several of the memory-management features that are available at the DOS level are intended to free up conventional memory. The basic concept here is to load as many device drivers and TSRs as possible into the upper memory area.

A *TSR* is a *terminate-and-stay resident* program that typically loads into memory whenever you start your computer but doesn't use your CPU resources unless you or some application makes a request of the TSR. Examples of TSRs include a DOS utility like DOSKEY or even SMARTDrive, a desktop utility like Borland's Sidekick, or a self-managed utility like a screen blanker (also called a screen saver). Commands to start TSRs usually are placed in your AUTOEXEC.BAT file.

Device drivers (small programs that control the operations of peripheral devices like your monitor or mouse) work in a manner similar to TSRs. However, entries that configure and run device drivers are usually placed in your CONFIG.SYS file.

AUTOEXEC.BAT and CONFIG.SYS files are stored in the root directory of your boot disk (usually your hard disk). DOS automatically looks for these two files whenever you start your computer.

Keep in mind that Windows in enhanced mode creates a virtual machine for each DOS application that you run under Windows. When Windows creates a virtual machine, it uses the conventional-memory environment that existed before Windows was loaded. *That's why it is* *important to load as many device drivers and TSRs as possible into upper memory.* Every DOS machine inherits the DOS environment that existed in conventional memory before Windows was loaded. So, the more drivers and TSRs that you can move out of conventional memory, the more memory you make available to each virtual DOS machine.

Understanding Your CONFIG.SYS and AUTOEXEC.BAT Files

The device drivers and TSRs that your system uses are loaded through the use of instructions contained in either your CONFIG.SYS or AUTOEXEC.BAT file.

If you're familiar with these files and their contents, you can skip to the next section in this chapter. If you would like to learn more about CONFIG.SYS and AUTOEXEC.BAT, please read on.

Your CONFIG.SYS file is stored in the root directory of your hard disk. This file is responsible for loading the driver programs that control peripheral devices, and for allocating memory for *buffers* and *file handles.* A buffer is a small-sized area in RAM (typically 512 bytes) used to keep frequently used data in memory, so that it doesn't have to be repeatedly accessed from disk. A file handle provides identifying and addressing information for files that are open (in other words—they've been read into memory from disk). Device drivers are configured with the entry DEVICE=.

Note: The DEVICE= line for HIMEM.SYS must appear before the DOS=HIGH line because DOS cannot load itself into the HMA (high memory area) until HIMEM.SYS has actually created the HMA.

In general, the order of other CONFIG.SYS entries is not important, although some vendor-supplied drivers must be placed before or after certain entries. You will need to consult the documentation for your drivers if you suspect a problem with their location within CONFIG.SYS.

These lines, or similar ones, are commonplace in a CONFIG.SYS file:

```
files= 60
BUFFERS=20
DEVICE=C:\DOS\SETVER.EXE
DEVICE=C:\WINDOWS\himem.sys
device=\gmouse.sys *6
DOS=HIGH
STACKS=9,256
SHELL=C:\DOS\COMMAND.COM C:\DOS\ /e:2048 /p
```

HOT TIP

Increase the Number of File Handles to 60

For years, a FILES= value of 30 in CONFIG.SYS provided plenty of file handles for any DOS application and even for many multitasking environments. But Windows 3.1 can use 20 or more file handles solely for booting and can require several dozen more handles to support a complex multitasking environment. So, you should increase your FILES= value to 50 or 60 to ensure that DOS provides enough handles to support Windows and your applications. A file handle takes up very little space in memory, so increasing the number of handles is just cheap, sensible memory-management insurance.

Your AUTOEXEC.BAT file is stored in your hard disk's root directory and contains statements for loading programs, including TSRs, automatically when your system starts up. TSRs never appear in CONFIG.SYS; however, it's also possible to write other batch files that allow you to start certain TSRs at the DOS prompt after your system has been booted or even from within Windows.

AUTOEXEC.BAT also contains instructions for modifying the way DOS works—such as the PROMPT command to set the appearance of the DOS prompt. The following simple listing shows an example of an AUTOEXEC.BAT file. (SMARTDRV.EXE is the WINDOWS 3.1 TSR that sets up a disk cache in RAM.)

```
C:\WINDOWS\SMARTDRV.EXE
@ECHO OFF
PROMPT $p$g
PATH C:\;C:\WINWORD;C:\WORKS;C:\EXCEL;C:\WINDOWS;C:\DOS
DOSKEY
```

Other lines in the AUTOEXEC.BAT file include the PATH line, which identifies all the directories that DOS will search whenever you issue a startup command at the DOS prompt. In other words, you can start any program stored on your hard disk from within any directory if the location of the program is included in the PATH line of your AUTOEXEC.BAT file.

You can also use the AUTOEXEC.BAT file to set the TEMP environment variable. Some programs, including Windows, often store temporary files during their operation. The TEMP variable determines the directory where these programs store files temporarily.

HOT TIP

Perform a Clean Boot Before You Troubleshoot Any Windows-Related Problems

Many problems that you might encounter in trying to run Windows stem from device drivers, TSRs, and other configuration information that an application might enter into your DOS CONFIG.SYS or AUTOEXEC.BAT files. For this reason, your first troubleshooting step should be to modify your CONFIG.SYS and AUTOEXEC.BAT files to perform a *clean boot.* When you use a clean boot to start your computer, your CONFIG.SYS and AUTOEXEC.BAT files contain *only* those entries that are necessary and safe for Windows 3.1.

For DOS 5, Microsoft recommends the following CONFIG.SYS and AUTOEXEC.BAT files to support a clean boot:

```
CONFIG.SYS
FILES=60
BUFFERS=20
DEVICE=C:\DOS\SETVER.EXE
DEVICE=C:\WINDOWS\HIMEM.SYS
SHELL=C:\DOS\COMMAND.COM C:\DOS\ /e:2048 /p

AUTOEXEC.BAT
PROMPT $p$g
PATH C:\WINDOWS;C:\DOS;C:\
SET TEMP=C:\WINDOWS\TEMP
```

To perform a clean boot with earlier versions of DOS, CONFIG.SYS should contain only the FILES=, and BUFFERS= lines.

After you've performed a clean boot, start Windows and see if the problem you've been having still exists. If the problem doesn't occur, you can be sure that the problem lies in one (or more) of the entries in your CONFIG.SYS or AUTOEXEC.BAT files. You can troubleshoot by returning your CONFIG.SYS and AUTOEXEC.BAT files to their original appearance—*one line at a time*—until you've isolated the offending entry. Of course, you'll need to restart your computer and then restart Windows each time you modify your CONFIG.SYS or AUTOEXEC.BAT file.

Note: Read SETUP.TXT (installed in your WINDOWS directory and stored in uncompressed format on the Windows Installation Disk #1) if you think a TSR might be preventing you from installing or running Windows successfully. SETUP.TXT includes a list of all TSRs that are known to cause problems for Windows 3.1.

Viewing and Editing CONFIG.SYS and AUTOEXEC.BAT at the DOS Level

Although there are several ways to view and edit CONFIG.SYS and AUTOEXEC.BAT in Windows, you will frequently need to work with these files at the DOS level.

Many of the optimization tips that we'll provide in this chapter involve modifying your CONFIG.SYS and/or AUTOEXEC.BAT. In addition, you'll often need to open and modify these files while you are outside of Windows.

If you are using DOS 5, use the MS-DOS Editor to edit these files. (Stay away from the clunky EDLIN line editor unless you're hopelessly stubborn and have already come to know and love it.) Make sure that your DOS directory is included in the PATH statement of your AUTOEXEC.BAT file. That way, you'll be able to start the Editor from any directory.

To run the Editor and open a file in one step, use this simple format from the DOS prompt:

```
EDIT <PATH><FILENAME>
```

So, to start the Editor and load AUTOEXEC.BAT, just type:

```
EDIT C:\AUTOEXEC.BAT
```

and press **Enter**. Most of the Editor's menus and options are intuitive and simple, so we won't delve into them here.

HOT TIP

Always Back Up Your AUTOEXEC.BAT and CONFIG.SYS Files before You Modify Them

Occasionally you'll make a change to your AUTOEXEC.BAT or CONFIG.SYS file that will wreak havoc on your system. In fact, it is possible to make incorrect entries (into either file) that will prevent the system from booting successfully. And if you need to modify your AUTOEXEC.BAT or CONFIG.SYS files to support a clean boot or to troubleshoot, you might find yourself creating several different versions of these files.

For these reasons, always make backup copies of your AUTOEXEC.BAT and CONFIG.SYS files before you modify them. To do this, simply assign a different extension to each file—such as AUTOEXEC.BAK and CONFIG.BAK. You'll need to make sure some application hasn't already used these names to back up previous versions of the files. For added security, copy these backup files to a diskette.

HOT TIP

Create a Bootable Diskette for Use in Restarting Your System from a Floppy

If your AUTOEXEC.BAT or CONFIG.SYS file will not allow you to boot DOS successfully, you will have to load DOS from a diskette. So before you make any changes to your AUTOEXEC.BAT or CONFIG.SYS files, make a *bootable diskette*, which contains a copy of the COMMAND.COM DOS environment.

Then, if corrupted entries in your AUTOEXEC.BAT or CONFIG.SYS files prevent DOS from booting successfully from your hard disk, you can start your computer using the bootable diskette in the A: drive. To create a bootable diskette in Windows 3.1:

1. Start the File Manager.
2. Insert a blank diskette (or any diskette that does not contain files that you want to keep) into drive A or B.
3. In the menu bar, click on Disk to display the Disk menu, then click on Make System Disk to display the standard Make System Disk dialog box.
4. Make sure the correct drive letter is specified, then click on OK to instruct DOS to copy the system files to the diskette.

Using DOS 5 to Optimize Windows

If you're running Windows on a DOS version prior to 5, you really should consider upgrading. Many features of DOS 5 were created specifically to support Windows memory-management capabilities. For this reason, the following sections pertain chiefly to users who are running DOS 5 on their systems. In turn, most of the optimization tips we'll provide in these sections are useful only if you have a 386 or 486 system and are running in enhanced mode.

Using DOS 5 to Evaluate Your Memory Resources

Before you take steps to optimize your system's memory usage, you need to examine the way memory is currently configured and used on your machine.

DOS 5 provides an upgraded MEM command that lets you view the contents of conventional memory and (if you have a 386 or higher processor) the upper memory area. Although the MEM command has been around since DOS version 4.01, it hasn't been widely used, probably because there wasn't all that much you could do to change your memory configuration even after you examined it.

However, DOS 5 contains some new and extremely useful parameters for optimizing memory use. The MEM command and its two switches—/C and /P—provide some detailed glimpses of memory, and can get you started in planning your memory-optimization strategy. For non-programmers, the /C switch is the easiest to use.

When you type **MEM**, with no switches, at the DOS prompt, DOS displays a few cryptic lines similar to the following:

```
655360 bytes total conventional memory
655360 bytes available to MS-DOS
544560 largest executable program size

3145728 bytes total contiguous extended memory
      0 bytes available contiguous extended memory
2031616 bytes available XMS memory
      MS-DOS resident in High Memory Area
```

At first glance, you might think this listing doesn't offer much more information than the old DOS standby command, CHKDSK. And you would be correct. Using MEM with no switches doesn't tell you *what* is loaded *where* in memory, which is what you really need to know to optimize memory. However, this vanilla MEM command does offer a few useful tidbits.

First, take a look at the line "544560 largest executable program size." We've got 655360 bytes (640K) of total conventional memory, but only 544560 bytes available for storing programs. So, one or more memory hogs have already started eating at our conventional-memory trough. We'll track down the culprits in a moment.

Now note that we've got 3145728 bytes (3MB of total extended memory available), but none of it is available to DOS. Remember: DOS can only access conventional memory directly. We need an extended memory manager (like HIMEM.SYS) to get at our extended memory, which brings us to the next line in the MEM listing:

```
2031616 bytes available XMS memory
```

We haven't done anything except boot the computer, and already we've lost 1MB of our total 3MB of extended memory. (XMS stands for *extended memory specification*, but is used synonymously with the simpler term *extended memory*.) So, who or what is controlling that precious megabyte?

The answer, of course, is SMARTDrive. When you installed Windows 3.1 to your hard disk, Windows automatically inserted a command in your AUTOEXEC.BAT file to load SMARTDrive whenever your system boots.

SMARTDrive configures up to 2MB of your system's extended memory as a RAM cache. Usually, you'll want to leave the SMARTDrive settings alone, although there are situations where you will want to increase or decrease your RAM cache. (We'll have more on SMARTDrive in Chapter 4.)

HOT TIP

SMARTDrive Automatically Adjusts Its Cache Size

If your system has less than 6MB of extended memory, SMARTDrive will automatically reduce its cache size when you start Windows. The Windows operating environment is essentially a multitasking environment with a built-in DOS extender technology. Since Windows relies heavily on extended memory to store applications, SMARTDrive will turn over some of its extended cache space to Windows. In other words, SMARTDrive maintains one cache size under DOS, and a different cache size under Windows. Table 2.1 shows the criteria SMARTDrive uses to adjust its cache size. Note that these values are based on true RAM. SMARTDrive does not use or consider virtual memory.

Table 2.1 SMARTDrive Cache Sizes

Total Extended Memory	*Cache Size under DOS*	*Cache Size under Windows*
Up to 1MB	All extended memory	No cache
1MB up to 2MB	1MB	256K
2MB up to 4MB	1MB	512K
4MB up to 6MB	2MB	1MB
6MB or more	2MB	2MB

Using MEM /C

You can use MEM with the /C switch to get a good "snapshot" of your system's memory use.

When you type MEM /C, DOS lists the current contents of conventional and UMA memory, in this format:

```
Program  Size        Size             Size
Name     (in bytes)  (in kilobytes)   (in hexadecimal)
```

MEM /C lists only those programs that are loaded into memory from the hard disk. So, anything loaded from the ROM BIOS, Video ROM, or other ROM memory won't appear in the listing.

Here's an example of what DOS might display when you type **MEM /C**:

```
MSDOS        15296  ( 14.9K)         3BC0
SETVER         400  (  0.4K)          190
HIMEM         1072  (  1.0K)          430
GMOUSE       14912  ( 14.6K)         3A40
COMMAND       2624  (  2.6K)          A40
SMARTDRV     26768  ( 26.1K)         6890
GRAB         23808  ( 23.3K)         5D00
DOSKEY        4128  (  4.0K)         1020
FREE            64  (  0.1K)           40
FREE            64  (  0.1K)           40
FREE        565920  (552.7K)        8A2A0
               .
               .
               .
```

Now you've got a good picture of your system's conventional memory—from 0K to 640K. But what about your UMA usage (the 384K from 640K to 1MB)? Unfortunately, MEM /C only provides you with information about conventional memory—unless an expanded memory manager has been installed to configure the UMA for use by DOS. (You'll see how to get MEM /C to recognize the UMA in a moment.)

Using MSD to Examine Memory

Before you can put the UMA to use, you need to know how much available space you have in the UMA. Remember: DOS automatically reserves part of the UMA for your system's BIOS and video BIOS. (Take a look at Figure 1.2 to verify this.) So you know that

part of your UMA is being used, but how do you find out how much UMA is being used?

Windows 3.1 comes with a utility called *MSD* (which stands for *Microsoft Diagnostics*). Although MSD was designed specifically to help support professionals (including Microsoft product support personnel), it also does a good job in assessing memory usage. By using the MSD Memory options, you can get a picture of all of your system's physical memory.

When you install Windows 3.1, MSD is automatically installed in the Windows directory. Because Windows places the Windows directory in your AUTOEXEC.BAT file's PATH line, you can start MSD from any directory when you are at the DOS prompt.

To start MSD, type **MSD** at the DOS Prompt and press **Enter**. The program will take about five to ten seconds to examine your hardware and software. Then, you'll see a menu similar to the one shown in Figure 2.1.

HOT TIP

Always Exit Windows before You Use the MSD Utility

MSD is a DOS application, not a Windows application. Windows manages memory by taking control of many services that DOS normally provides. For this reason, Windows does not thoroughly report its memory modeling information to DOS.

The bottom line is that the information that MSD reports if Windows is running may not be accurate. Since you'll usually use MSD to

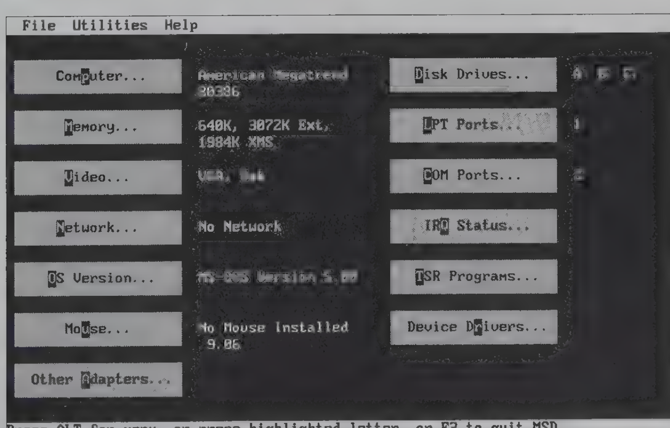


Figure 2.1 Use this screen to assess your computer hardware and software, including memory usage.

optimize or troubleshoot the way DOS is configuring memory, you should always exit Windows before starting MSD.

Don't run MSD using the MS-DOS Prompt application available in Windows. If you try this technique, Windows creates a 640K virtual machine when you double-click on the MS-DOS Prompt icon. As a result, MSD will only "see" the conventional memory that's available to the virtual machine it is running in; because a virtual machine inherits the conventional memory that existed *before* Windows is loaded, MSD won't be able to report on any Windows code that is currently stored in conventional memory.

Click on Memory to display a map of the UMA, as shown in Figure 2.2.

Warning: MSD uses a hex notation that deletes the trailing 0 from each hex value. So the value C000 is actually C0000 (decimal 786,432), the value F800 is actually F8000 (decimal 1,015,808), and so on. Although this is a confusing practice, it's a standard one in the computer industry—especially with Microsoft. Hex addresses in most of Microsoft's Windows documentation (including .TXT and .WRI files) are missing the trailing 0. You'll need to remember to add a 0 to hex addresses and hex sizes before you use the Windows Calculator (or other calculator) to convert hex values to decimal values.

The left side of Figure 2.2 maps UMA usage in a format similar to a stacked bar chart. The chart portrays memory from the top down, with memory from 1024K down to 768K shown in the initial display. If you want to view memory usage from 768K down to 640K, use your mouse along with the scroll bar at the right edge of the screen to scroll down.

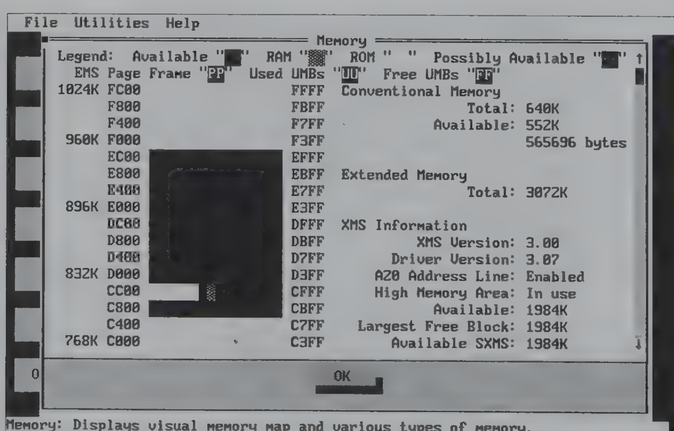


Figure 2.2 This MSD memory map displays information about current UMA usage.

The right side of Figure 2.2 displays lines of text that describe memory use. Note that 552K of the total 640K is available (unoccupied). That means about 90K of conventional memory is being occupied by device drivers and TSRs. (We've already confirmed this fact by using the MEM /C command at the DOS prompt.)

However, the memory map to the left shows a large black area from about 780K (hex address C300 on the memory map) up to almost 960K (hex address F000 on the memory map). The black area, as you can see by examining the legend along the top line of the screen, indicates available memory.

In a sense, then, we have more than 150K of the UMA space being "wasted." We could make a much more efficient use of our conventional memory if we could just load some of our drivers and TSRs into this unused UMA space. By doing so, we would free up valuable conventional memory for use in creating virtual DOS machines in enhanced mode. (Remember: Windows 3.1 doesn't actually create virtual machines in conventional memory; instead, the conventional memory environment is used to create each virtual machine.)

Using EMM386 to Configure the UMA

Recall that an expanded memory manager (EMM) is designed specifically to create pages in the unused portions of the UMA. However, most expanded memory managers, including EMM386, can also emulate expanded memory. This feature allows you to use EMM386 to configure the UMA, even if you don't have an expanded memory board.

Originally, the creation of 16K page frames in the UMA was valuable for swapping memory into the UMA from an expanded memory board. But most 286 and 386 systems in use today don't have expanded memory boards and don't need to use expanded memory because extended memory is available. DOS 5 takes this fact into account and makes it possible for an EMM to claim UMA space even though an expanded memory board is not present.

After you use an EMM to take control of available blocks within the upper memory area (called *upper memory blocks*, or *UMBs*), you can then configure DOS 5 to recognize these upper memory blocks. In turn, you can modify your AUTOEXEC.BAT and CONFIG.SYS files to load as many drivers and TSRs as will fit within the available upper memory blocks.

Since EMM386 is available to all Windows 3.1 users who have a 386, we'll use this expanded memory manager to perform our optimization tricks.

Note: EMM386 establishes expanded memory and allocates UMA space for DOS. However, when Windows 3.1 starts up, it puts EMM386 "to sleep" and uses its own internal memory manager. However, Windows 3.1 won't disturb any of the UMA space that has already been allocated by EMM386. (Your TSRs and device drivers also won't be disturbed.) In fact, Windows 3.1 only uses the UMA in enhanced mode, and only to support the Windows translation buffers. Windows never stores applications in the UMA.

To load the EMM386 expanded memory manager, add the following line to your CONFIG.SYS file, if it doesn't already exist:

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

The NOEMS parameter tells the EMM386 manager that you will not be using any expanded memory—in other words, you don't have an expanded memory board. EMM386 then assumes you want to reserve the UMBs that it manages for use by DOS programs. This is precisely the meaning that you want to convey to EMM386.

Note: The NOEMS parameter only works if you are running DOS 5.

HOT TIP

Use the RAM Parameter If You Want to Emulate Expanded Memory

If you do have a DOS application that requires expanded memory, use the RAM parameter in place of the NOEMS parameter in your DEVICE line for EMM386. The RAM parameter sets up a 64K page frame that conforms to the LIM 3.2 EMS, and allows DOS to use the remaining upper memory blocks to store device drivers and TSRs.

Note: This parameter is only available if you are running DOS 5.

Telling DOS to Recognize the Upper Memory Area

After you have used EMM386 to configure the UMA, you still need to let DOS know that you've got more than just BIOS code and video memory stored in the upper memory blocks of the UMA.

To tell DOS to map the UMA for use by programs—including device drivers and TSRs—add the following line to your CONFIG.SYS file:

```
DOS=UMB
```

This command tells DOS to look for programs in the UMA as it manages your system resources.

HOT TIP

You Can Combine the DOS=UMB and DOS=HIGH Entries on One Line

When you install DOS 5, DOS automatically adds the line DOS=HIGH to your CONFIG.SYS file. This line instructs DOS to load a portion of its command environment into the 64K HMA (high memory area) that is configured by HIMEM.SYS. You can add the DOS=UMB entry to this line by typing **,UMB** after DOS=HIGH. Your DOS= line should then look like this:

```
DOS=HIGH,UMB
```

Using MSD to View the Contents of the UMA

Assume that you've made changes to your CONFIG.SYS file so that EMM386.EXE is installed when you boot and so that DOS will recognize that the UMBs can contain DOS programs. At this point, you need to reboot your computer so that the new CONFIG.SYS settings take effect. Then, you can load the MSD program to look at the contents of the UMA.

Figure 2.3 shows how the MSD memory map for the UMA now appears in our example. Note the following lines in the figure:

```
       Total UMBs: 87K
  Total Free UMBs: 61K
Largest Free Block: 45K
```

The amount of total UMBs available in this example is 87K. However, the line "Total Free UMBs: 61K" indicates that only 61K is available to us. (The "Largest Free Block" line just indicates the size of the largest contiguous series of addresses in upper memory.) What happened to the other 26K of UMB space? A hint is available by looking at the Conventional Memory numbers in Figure 2.3.

Notice that 570K of conventional memory is now available. Before we installed EMM386, only 552K of conventional memory was available. And, even though we've actually *added* to the contents of conventional memory by loading MSD, we've still got more conventional memory than

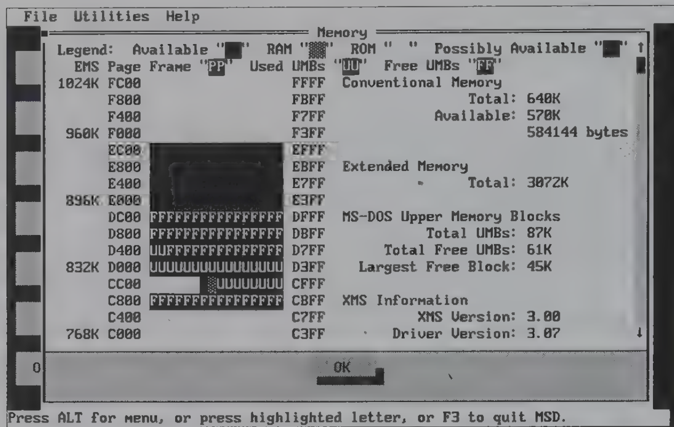


Figure 2.3 After you've used EMM386 to configure the UMA, MSD will report on the contents of the UMA.

we had before we installed EMM386. So, some program that previously was loaded in conventional memory at startup time detected that UMBs in the UMA portion of memory were now available, and decided to load itself there. But which program?

The answer can be found in SMARTDrive 4.0, the Windows 3.1 RAM caching utility. If possible, SMARTDrive will try to get its "footprint" out of conventional memory by loading itself automatically into the UMA. As soon as we installed EMM386 with the NOEMS parameter, SMARTDrive detected available space in the UMA, and loaded itself there to free up conventional memory.

HOT TIP

Some DOS Utilities Automatically Load in Upper Memory

A few other DOS utilities, such as MIRROR, will automatically try to load in available UMA.

Determining What to Load into the UMA

If you've been following along with our example, the available conventional memory space has now been increased, but the available UMBs have been reduced from 87K to 61K. However, 61K is still quite a lot of memory for small programs like device drivers and TSR utilities. The problem now becomes: which drivers do we load into the remaining available UMA?

To answer this, you need to see a list of installed device drivers and TSRs, along with their sizes. It's a good bet (although not a sure thing) that if you identify a list of drivers and TSRs whose total size is less than the total UMB space available, you'll be able to fit these programs into the upper memory area.

In our case, we have 61K of upper memory available. So we now need to determine which, if not all, of our drivers and TSRs can fit into the UMA.

You can determine much of this information from within MSD, but the technique is a bit tricky. Follow these steps:

1. Display the MSD File menu, as shown in Figure 2.4. Notice that this menu allows you to view your AUTOEXEC.BAT and CONFIG.SYS files (as well as your Windows SYSTEM.INI and WIN.INI files).
2. In the File menu, click on AUTOEXEC.BAT, and identify the TSR programs that are included in this file. You should write these programs on a sheet of paper so that you don't forget them. (Remember that you do not need to concern yourself with the SMARTDRV.EXE utility; this program automatically loads itself into the UMA if EMM386 has made the space available.)
3. Click on OK to exit the AUTOEXEC.BAT file display.
4. Display the File menu, then click on CONFIG.SYS.
5. Identify the device drivers that are loaded by CONFIG.SYS. (These drivers are identified by the DEVICE= entry.) You should write the names of these drivers on a sheet of paper so that you don't forget them.

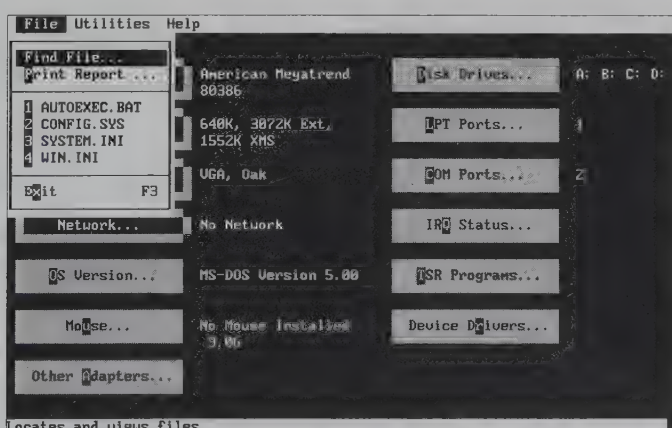


Figure 2.4 The MSD file menu lets you display the contents of your CONFIG.SYS, AUTOEXEC.BAT, WIN.INI, and SYSTEM.INI files.

6. Click on OK to exit the CONFIG.SYS file display.
7. Display the MSD main menu, then click on TSR Programs. This option, shown in Figure 2.5, displays a list of all TSR programs and device drivers currently stored in memory and indicates their sizes in both hex and decimal. Actually, MSD considers anything that is loaded into memory at startup time to be a TSR, so you will see many more entries here than are actual TSRs and drivers.
8. Match the TSR names from your AUTOEXEC.BAT file with their decimal sizes. Ignore the MSD.EXE entries; these belong to the MSD program, which you are only using temporarily and which isn't in your AUTOEXEC.BAT file.
9. Match the device names from your CONFIG.SYS file with their decimal sizes.
10. Determine which drivers and TSRs can fit within the available amount of UMA.

HOT TIP

MSD Does Not Always Provide a Complete List of Resident Programs

MSD won't list some DOS utilities separately, such as DOSKEY. If your AUTOEXEC.BAT file includes TSRs that MSD can't identify, you can use the MEM /C command (at the DOS prompt) to identify the sizes of these additional TSRs.

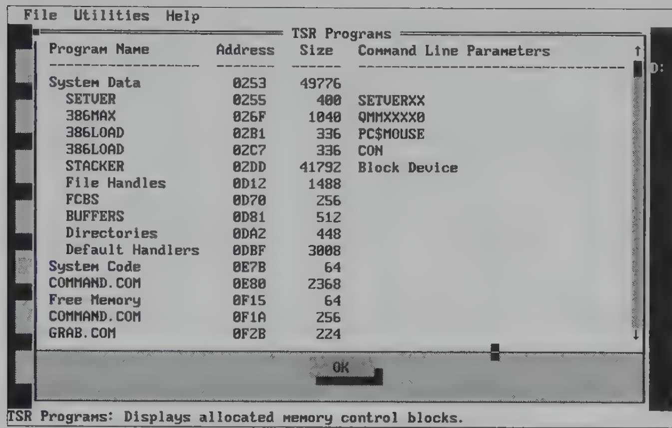
The following programs cannot be loaded into the UMA:

MSDOS Any item that has this name is used by the COMMAND portion of DOS and cannot be moved.

HIMEM This is the Windows extended memory manager, and must remain in conventional memory.

EMM386 This is, of course, the DOS and Windows expanded memory manager; since it is responsible for configuring UMA for use by DOS and Windows, it must first be loaded into conventional memory before it can configure the UMA.

COMMAND This a portion of the DOS command environment that must remain in conventional memory so that programs designed for earlier versions of DOS can request DOS services.


File Utilities Help

TSR Programs

Program Name	Address	Size	Command Line Parameters
System Data	0253	49776	
SETVER	0255	400	SETVERXX
386MAX	026F	1040	QMMXXXX0
386LOAD	02B1	336	PC$MOUSE
386LOAD	02C7	336	CON
STACKER	02DD	41792	Block Device
File Handles	0D12	1488	
FCBS	0D70	256	
BUFFERS	0D81	512	
Directories	0DA2	448	
Default Handlers	0DBF	3008	
System Code	0E7B	64	
COMMAND.COM	0E80	2368	
Free Memory	0F15	64	
COMMAND.COM	0F1A	256	
GRAB.COM	0F2B	224	

OK

TSR Programs: Displays allocated memory control blocks.

Figure 2.5 This MSD screen displays information about all TSRs and device drivers currently stored in memory.

Using MEM /C to Determine What to Load in the UMA

The MEM /C command provides a more reliable listing of your TSRs and drivers than MSD. So, we'll explain this alternative approach to determining UMA optimization.

If we return to the DOS prompt at this point in our memory setup, then type **MEM /C**, DOS displays the following screen:

```
Conventional Memory:
  Name         Size in Decimal        Size in Hex
-----------------------------------------------
  MSDOS      15312  ( 15.0K)           3BD0
  SETVER       400  (  0.4K)            190
  HIMEM       1072  (  1.0K)            430
  EMM386      8400  (  8.2K)          .20D0
  GMOUSE     14912  ( 14.6K)           3A40
  ANSI        4192  (  4.1K)           1060
  COMMAND     2624  (  2.6K)            A40
  GRAB       23808  ( 23.3K)           5D00
  DOSKEY      4128  (  4.0K)           1020
  FREE          64  (  0.1K)             40
  FREE          64  (  0.1K)             40
  FREE      603760  (589.67K)         8A2A0

Total Free: 580208  (566.6K)
```

```
Upper Memory:

  Name      Size in Decimal           Size in Hex
-------------------------------------------------
  SYSTEM    172064  (168.0K)          2A020
  SMARTRDV   26768  ( 26.1K)           6890
  FREE          32  (  0.0K)             20
  FREE       16272  ( 15.9K)           3F90
  FREE       46912  ( 45.8K)           B740

Total FREE:  63216  ( 61.7K)

              .
              .
              .
```

This display confirms what we saw in the MSD program: about 61K is available in upper memory. The listing also provides us with an idea of the programs that we can load into upper memory:

```
Program        Size
SETVER          0.4K
GMOUSE         14.6K
ANSI            4.1K
GRAB           23.3K
DOSKEY          4.0K
```

We've identified four programs on our system that can be moved to the UMA: SETVER, GMOUSE (the mouse driver), ANSI, GRAB (a screen grab utility) and DOSKEY. The total size for all five programs is about 48K bytes. We have about 61K available in the upper memory area, so we can easily fit all five programs into the UMA.

Moving Programs to the UMA

In CONFIG.SYS, you use the DEVICEHIGH command to load drivers into upper memory. In AUTOEXEC.BAT, you use the LOADHIGH command to load TSRs into upper memory.

The format for DEVICEHIGH is simple:

```
DEVICEHIGH=<PATH\DRIVER_NAME>
```

So, the following existing lines in our CONFIG.SYS file

```
DEVICE=C:\DOS\SETVER.EXE
DEVICE=C:\GMOUSE.SYS *6
DEVICE=C:\DOS\ANSI.SYS
```

can be changed to:

```
DEVICEHIGH=C:\DOS\SETVER.EXE
DEVICEHIGH=C:\GMOUSE.SYS *6
DEVICEHIGH=C:\DOS\ANSI.SYS
```

Here's the LOADHIGH format that you can use in AUTOEXEC.BAT:

```
LOADHIGH <PATH\DRIVER_NAME>
```

So, the following lines in our AUTOEXEC.BAT file

```
C:\HSG\GRAB.EXE
C:\DOS\DOSKEY
```

can be changed to:

```
LOADHIGH C:\HSG\GRAB.EXE
LOADHIGH C:\DOS\DOSKEY
```

HOT TIP

DOS 5 Uses an Allocate-Largest-Block-First Algorithm

A good strategy is to begin with the largest driver or TSR that will fit into the amount of free UMB space, then load successively smaller programs there. The reason: some programs require more space to load than they actually use when they are resident. For instance, the GRAB.EXE program can require more than 40K to load, even though only 23.3K remains resident after the program has been initialized.

By loading this GRAB.EXE before DOSKEY, we ensure that the UMBs that DOSKEY will use are available for GRAB.EXE when it loads. After the GRAB program shrinks to its resident size, DOS can load DOSKEY into the remaining UMA. If we reversed the order of these two utilities in the AUTOEXEC.BAT file, there might not be enough UMB space for GRAB to initialize itself. (If this problem occurs, DOS loads the utility in conventional memory instead of the UMA.)

At this point, if we reboot the computer and use MEM /C to view memory usage, we get the following listing:

```
Conventional Memory:
  Name          Size in Decimal         Size in Hex
-----------------------------------------------
  MSDOS       15312  ( 15.0K)             3BD0
  SETVER        400  (  0.4K)              190
  HIMEM        1072  (  1.0K)              430
  EMM386       8400  (  8.2K)             20D0
  COMMAND      2624  (  2.6K)              A40
  FREE           64  (  0.1K)               40
  FREE       627264  (612.6K)            99240

Total Free: 627328  (612.6K)

Upper Memory:

  Name       Size in Decimal            Size in Hex
-----------------------------------------------
  SYSTEM     172064  (168.0K)            2A020
  GRAB        23808  ( 23.3K)             5D00
  DOSKEY       4128  (  4.0K)             1020
  GMOUSE      14912  ( 14.6K)             3A40
  ANSI         4192  (  4.1K)             1060
  SMARTRDV    26768  ( 26.1K)             6890
  FREE          208  (  0.2K)               D0
  FREE        11728  ( 11.5K)             2DD0
  FREE         4144  (  4.0K)             1030

Total FREE:  16080  ( 15.7K)

              .
              .
              .
```

Before we began optimizing memory, we had only 544,560 bytes available for storing programs in conventional memory. Now, we have 627,040 bytes available—an additional 82,480 bytes, or 80.5K. And we can do even better, as we'll show next.

HOT TIP

You Can't Use EMM386 in Standard Mode

EMM386 can emulate expanded memory in enhanced mode only. If you've used EMM386 to configure the UMA, you cannot start Windows in standard mode. If you try to start Windows in standard mode with

EMM386 installed, Windows will display an error message and insist that you run Windows in enhanced mode.

You can get around this problem by adding a semicolon (;) or the letters REM (for Remark) in front of the DEVICE=EMM386.EXE line in your CONFIG.SYS file. When you do this, DOS ignores the line when your system boots up. If EMM386 is not loaded, any DEVICEHIGH entries in CONFIG.SYS are automatically converted to DEVICE entries; any LOADHIGH entries in AUTOEXEC.BAT are automatically ignored (causing TSRs to load in conventional memory).

ON DISK

Use BOOT.SYS to Maintain Different CONFIG.SYS and AUTOEXEC.BAT Setups

The BOOT.SYS program, provided on the Insider Disks, displays a menu environment whenever you boot your system. You can make choices from these menus to determine which parts of your CONFIG.SYS and AUTOEXEC.BAT files you want to load for your current session. The files for BOOT.SYS are stored in the BOOTSYS directory on the disks.

BOOT.SYS is extremely valuable if you configure your hardware and memory differently for different sessions. Without BOOT.SYS, you would have to alter your AUTOEXEC.BAT or CONFIG.SYS files whenever you want to use a different setup environment (or keep and rename multiple AUTOEXEC and CONFIG files with different extensions—a real pain); then, you would have to reboot your computer.

Using EMM386 to Map over Unused ROM

If you look back to Figure 2.3, you'll notice that EMM386 did not try to claim any of the address space from hex addresses E000 to F000 (896K to 960K). In our example, this 64K range of UMA is not being used by ROM, by EMM386, or by any other hardware or software. It is, essentially, wasted space. Unless.

Remember at the start of this chapter we said that DOS 5 was designed with Windows in mind. We also explained that Windows manages memory to prevent device conflicts *first*, and then—and only then—does it work to optimize memory.

When EMM386 configures the UMA, it searches for unused address space from hex address C000 (768K) up to hex address E000 (896K).

Although other address space might be available, EMM386 takes a conservative approach to prevent itself from conflicting with addresses that might be used by a video display driver or some other hardware adapter.

However, if you know that your hardware does not use this UMA region—and it very likely does not—you can instruct EMM386 to recognize it. You do this by using the I (for Include) switch in the EMM386 line of your CONFIG.SYS file. You simply "include" the desired range before the NOEMS or RAM parameter. (You can even include multiple ranges of UMA, but we'll get to that in a moment.) Here's the format:

```
DEVICE=<PATH>EMM386.EXE <I=[RANGE]> <NOEMS> <RAM>
```

The RANGE is defined by the starting and ending hex addresses, separated by a hyphen. So, to include the range E000-EFFF, you would create this line in your CONFIG.SYS file:

```
DEVICE=C:\DOS\EMM386.EXE I=E000-EFFF NOEMS
```

Warning: You cannot include the range E000-EFFF if you have a PS/2 computer. Also, EMM386 (and most Microsoft applications) doesn't include the trailing 0 on hex addresses. So, strictly speaking, even though you specify E000-EFFF, these are actually the addresses E0000 and EFFF0.

Another range that might be safe to include is the 32K range from hex address B000 to B7FF (704K to 736K). This region is reserved for monochrome display drivers. If you have a color monitor, you can include this range, even for PS/2 systems.

You include multiple ranges of UMA memory by adding a new I=<RANGE> specification for each range to include. So, if you don't have a PS/2 system *and* you don't have a monochrome monitor, you could specify the following EMM386 line in your CONFIG.SYS file:

```
DEVICE=C:\DOS\EMM386.EXE I=E000-EFFF I=B000-B7FF NOEMS
```

This line adds a precious 96K of addressable UMA for use by your device drivers and TSRs.

Note: Network cards also use the UMA. If a network card is installed on your computer, you can't include the UMA region that the card is using. This is true even if you don't actually use a network. (If a network card is installed on your system, it gets configured when your system boots, regardless of whether you log onto the network.)

Windows and Its Translation Buffers

When you run DOS applications in enhanced mode, Windows creates translation buffers in memory so that data can be passed from protected mode to real mode, and from real mode to protected mode. Windows will place its translation buffers in the UMA if sufficient space is available.

In enhanced mode, Windows must switch to real mode whenever it writes to disk (unless you're using 32-bit disk access) or to write to a network device. In real mode, of course, DOS can use only the first 1MB of memory. So, before Windows can write data to disk or to a network, the data must first be stored somewhere in the first 1MB of memory so that DOS can recognize it.

So, whenever you run Windows in enhanced mode, Windows sets aside *translation buffers* in order to store data below 1MB. Although you might never have to worry about the location of these translation buffers, they might become an issue if you need as much conventional memory as possible for your DOS applications. Remember, each virtual DOS machine inherits the conventional-memory environment. If Windows reserves space in conventional memory for the translation buffers, this space is reserved in every DOS machine.

Each Windows translation buffer is 4K in size. Windows allocates two translation buffers (8K total) by default when you start Windows in enhanced mode. If a network driver is also installed on your system, Windows sets up an additional four translation buffers (24K total) for network support.

If enough contiguous space in the UMA is available when Windows creates translation buffers, the buffers will be allocated in the UMA. If sufficient space is not available in the UMA, the buffers are created in conventional memory. Again, if Windows has to allocate the buffers in conventional memory, the buffers will occupy space in each subsequent virtual machine that is created.

If your system is running on a network, it is a good idea to reserve up to 24K of the UMA so that this space can be used for translation buffers. (If you aren't running Windows on a network, the location of the translation buffers is less important since Windows only reserves 8K for the buffers.) To provide this space, you need to understand how Windows uses the UMA. Specifically, EMM386 makes upper memory available to DOS. Under enhanced mode, Windows uses its own internal memory drivers to allocate *any* upper memory blocks that are available, regardless of whether EMM386 has been installed.

However, if you have used the I parameter in your CONFIG.SYS file's EMM386 line to map over *all* unused upper memory blocks, Windows will recognize that this space has been allocated to DOS and will not allocate translation buffers in the UMA.

If necessary, modify the Include range(s) in your EMM386 line so that at least 24K remains free, for use by Windows to allocate its translation buffers. Use the MSD memory map to determine if you need to modify your Include range(s).

HOT TIP

Excluding UMA Blocks for Use by Translation Buffers Can Be a Catch-22

When you exclude a range of addresses so that Windows can use the range to allocate translation buffers, you might very well be reducing the amount of UMA space available for storing your TSRs and device drivers.

The result? If you are forced to load one or more TSRs or drivers into conventional memory to make room for translation buffers, these TSRs and drivers will occupy space in every virtual machine that Windows creates. This could defeat the purpose of reserving UMA space for translation buffers.

Here are some general rules to follow to ensure that you are using UMA space wisely:

- Don't "Include" more UMA space than is required by your TSRs and device drivers.
- The more DOS applications you typically run during each Windows session, the more important it becomes to exclude UMA space for use by translation buffers.
- If you rely on several TSRs and rarely use DOS applications in enhanced mode, reserve all available UMA space for use by TSRs and let Windows create translation buffers in conventional memory.

Note: If one or more of your DOS applications *must* use expanded memory, you might need to ensure that an EMS page frame gets created *before* Windows allocates translation buffers. Remember, in enhanced mode, Windows creates expanded memory in case a DOS application should request it. But Windows can create this page frame only if 64K of contiguous memory is available in the UMA.

If you've already used EMM386 or some other memory manager to set up a page frame before Windows starts, you're all set. However, if your'e relying on Windows to create a page frame, you should make sure that translation buffers don't eat up the available UMA space. Do this by setting the ReservePageFrame= entry (in the [386enh] section of SYSTEM.INI) to "true", like this:

```
ReservePageFrame=true
```

With this setting, Windows will allocate an EMS page frame whenever it starts in enhanced mode, *before* it allocates translation buffers. If there is still room in the UMA to reserve the translation buffers, Windows will put them there, but only after the page frame for expanded memory has been created.

CHAPTER 3

Using Third-Party Memory Managers

Third-party memory managers like Quarterdeck's QEMM-386 and Qualitas' 386MAX have been around ever since the Intel 386 processor was introduced. Third-party memory managers won't work on most 286 computers because they were designed specifically to support the 386 processor's protected and virtual-86 modes. QEMM-386 and 386MAX (the two most popular managers) both manage expanded and extended memory on 386 systems, and both can optimize virtual machines by loading TSRs and drivers into upper memory.

Essentially, third-party memory managers offer an alternative to the memory-management capabilities of EMM386.EXE and HIMEM.SYS, and usually offer more effective memory-management features. However, third-party managers cost money; they require you to make a software purchase over and above the purchase of Windows 3.1. Do you really need to shell out more money for software? Can you optimize memory well enough using the Windows-supplied EMM386 and HIMEM managers, or would you be better off using QEMM-386 or 386MAX? How do you decide?

In this chapter, we'll explain how memory managers work and describe the benefits of using a third-party manager over the Windows-supplied managers. We'll also describe the basic features and some advanced features offered by QEMM-386 and 386MAX. Toward the end of the chapter, we'll provide a demonstration of 386MAX so that you can get a clear idea of the features of third-party managers. With this information, you'll find it easier to determine whether a third-party manager is for you.

If you already own a version of QEMM-386 or 386MAX, you might be looking for specific tips and troubleshooting techniques. This chapter is just an overview of third-party managers. For specific tips and troubleshooting techniques for QEMM-386 and 386MAX, refer to Chapter 17 in the Troubleshooting section of this book.

Common Components of Third-Party Memory Managers

EMM386 and HIMEM.SYS, along with some features of DOS, perform memory-management services similar to those you'll get with QEMM-386 or 386MAX. HIMEM.SYS configures extended memory at the DOS level, and EMM386 configures the upper memory area (UMA) to support an expanded memory (EMS) page frame and lets you load device drivers and TSRs in upper memory. But third-party managers like QEMM-386 and 386MAX are "smarter" than EMM386; that is, these third-party managers automatically determine the best method to use your system's upper memory, whereas you need to specify memory settings and configuration information for EMM386.

We'll begin explaining third-party managers by pointing out the way most memory managers perform their tricks.

A Memory Manager Requires a UMB Provider and a UMB Loader

A memory manager configures, or provides, upper memory blocks for use by drivers and TSRs (and optionally for an EMS page frame), and it actually loads drivers and TSRs into upper memory. In fact, the two major components of any memory-management package are the UMB provider and the UMB loader.

A *UMB provider* is one or more programs that make upper memory blocks (UMBs) in the UMA available to DOS. As we explained in Chapter 2, EMM386.EXE and HIMEM.SYS combine to provide this feature for DOS 5 and for Windows 3.1.In QEMM-386, the UMB provider is a program called QEMM-386.SYS. In 386MAX, the UMB provider is the program 386MAX.SYS.

Regardless of the memory manager you use, the UMB provider information must be specified on DEVICE= lines in your CONFIG.SYS file. So, if you are using EMM386 and HIMEM.SYS to provide access to the UMBs, your CONFIG.SYS file must include lines similar to these:

```
DEVICE=C:\WINDOWS\HIMEM.SYS
DEVICE=C:\WINDOWS\EMM386.SYS
```

If you're using QEMM-386, your CONFIG.SYS file should include a line similar to this one:

```
DEVICE=C:\QEMM386\QEMM386.SYS
```

And, of course, 386MAX requires this line (or a similar one) in your CONFIG.SYS:

```
DEVICE=C:\386MAX\386MAX.SYS
```

A *UMB loader* is one or more utilities responsible for loading TSRs and drivers into available upper memory blocks. In DOS 5, the UMB loader is part of the DOS command environment and is called by using the

DEVICEHIGH command for each device driver in CONFIG.SYS and by using the LOADHIGH command for each TSR program in AUTOEXEC.BAT.

In QEMM-386, the UMB loader is accessed using the LOADHI.SYS and LOADHI.COM utilities. LOADHI.SYS is used in CONFIG.SYS to load device drivers, while LOADHI.COM is used in AUTOEXEC.BAT to load TSRs. Specifically, a DEVICE=LOADHI.SYS line (in CONFIG.SYS) is required for each device driver to be loaded into upper memory. A LOADHI line is required in AUTOEXEC.BAT for each TSR to be loaded into upper memory.

The 386MAX UMB loader approach is just about identical to the one used by QEMM-386. In 386MAX, the UMB loader is accessed using the 386LOAD.SYS and 386LOAD.COM utilities. A DEVICE=386LOAD.SYS line (in CONFIG.SYS) is required for each device driver to be loaded in upper memory. In a similar fashion, a 386LOAD line is required in AUTOEXEC.BAT for each TSR to be loaded into upper memory.

HOT TIP

Use DOS=HIGH with Third-Party Managers

When you install DOS 5, DOS automatically adds the line DOS=HIGH to your CONFIG.SYS file. Even if you install QEMM-386 or 386MAX, you should keep this line in CONFIG.SYS. Only one program can have access to the high memory area (HMA) at a time. So, if you delete DOS=HIGH, QEMM-386 or 386MAX will be able to use the HMA. But the DOS command environment will be forced to load within the first 640K of memory, adding about 60K to each virtual DOS machine. QEMM-386 and 386MAX cannot place the DOS command environment in upper memory. DOS uses the 64K high memory area very efficiently. Therefore, you can help out your third-party memory manager by allowing DOS to load itself into the HMA.

Basic Features of QEMM-386 and 386MAX

The features and performance of QEMM-386 and 386MAX are very similar, although not identical. We aren't going to endorse one product over the other, because frankly we don't think the differences between QEMM-386 and 386MAX are substantial. However, for each package we'll point out a few unique features that are particularly helpful to Windows.

To give you a feel for the basic capabilities available with both products, the following sections explain some of the common features of QEMM-386 and 386MAX that are not available with EMM386.

QEMM-386 and 386MAX Provide Automated UMB Loaders

Both QEMM-386 and 386MAX can determine how to use their UMB provider and loader—without needing your help.

This is one of the most important benefits of using a third-party manager rather than EMM386. With EMM386, it is up to *you* to insert DEVICEHIGH and LOADHIGH commands in your CONFIG.SYS and AUTOEXEC.BAT files. As we explained in Chapter 2, you have to manually enter one of these commands for each device driver and TSR that you want to load in upper memory.

When you install QEMM-386 or 386MAX, the install program places the UMB provider in your CONFIG.SYS file, then uses the provider to identify available upper memory and set up upper memory blocks. The install program then searches your CONFIG.SYS and AUTOEXEC.BAT files for drivers and TSRs and *automatically* inserts the appropriate load command for each program so that it can be loaded into upper memory. You don't have to do a thing.

Using Third-Party Managers to Configure the UMA

Third-party managers like QEMM-386 and 386MAX relieve you of the burden of finding and configuring unused UMA space.

When you install EMM386, it searches for unused upper memory space only in the 128K range of hex addresses C0000 (768K) to E0000 (896K). To include other address ranges, you have to use the DOS MEM command or a memory viewer to determine unused address space. Then, you have to add an INCLUDE setting in your CONFIG.SYS file for each additional range of addresses that you want to include.

QEMM-386 and 386MAX search the entire 384K range of upper memory to find unused addresses, then configure the upper memory blocks automatically. You don't have to lift a finger to help.

EMM386 Forces You to Do a Lot of the Work

Before we explain how third-party memory managers load programs into the UMA, take a second look at all the work you have to do with EMM386.

As we explained in Chapter 2, EMM386 leaves it to you to determine the order in which TSRs and drivers are loaded into the UMA. Usually, it's best to load the largest programs into upper memory first, because that's when the required amount of upper memory is most likely to be available.

This approach is only a little more efficient than throwing darts blindfolded. Chances are that your available upper memory is not all in one contiguous block, but is broken into two or more areas of upper memory. Also, some TSRs require more than twice as much memory to load (initialize) as they actually need to run. These factors suggest that determining the optimal load order can require some effort.

When DOS prepares to load a program into upper memory, it just searches for the first available block that the program will fit into, then stores the program there. DOS doesn't attempt to compare the available UMBs with the storage requirements of the programs to be loaded from CONFIG.SYS and AUTOEXEC.BAT. To add insult to injury, DOS is blind to the difference between the initialization size required by a TSR and the actual size of the TSR once it is resident in memory.

As you can see, true upper memory optimization can require a lot of hand wringing. Even if you *could* use EMM386 to shuffle programs into the UMA in the most effective way possible, you wouldn't want to. Think about it: the number of load-order options increases exponentially depending on the number of programs you want to load and the number of contiguous UMBs that are available in upper memory. For even half a dozen programs and two or three different blocks in upper memory, the number of load-order possibilities can number in the hundreds.

Third-Party Managers Determine the Best Way to Load Programs into the UMA

A third-party memory manager like QEMM-386 or 386MAX rescues you from the load-order limitations of EMM386. When you install one of these managers, the installation program will evaluate your upper memory resources and the memory requirements of your TSRs and drivers. The install program can even determine the difference between the initialization and resident sizes of each TSR.

Generally, here's how the installation process works: after the installation program identifies and configures your upper memory, it examines your CONFIG.SYS and AUTOEXEC.BAT files and sets up programs and drivers so they can be loaded into the UMA. Then the installation program reboots your computer to determine how much memory each TSR requires during initialization and how much memory it uses after it is resident.

Next, the memory manager uses the combined information it has obtained about your UMA and your TSRs and drivers to calculate the most efficient load order. The memory-management software might reboot your system a few more times to verify and finalize the load order. The entire process requires only a few minutes to complete.

HOT TIP

Reconfigure Upper Memory Periodically

Your TSR and driver requirements might change over time, as you add or delete new TSRs and hardware devices. Each time you make a change, you should have your memory manager reconfigure your upper memory area. Normally, this takes only a few minutes to do and ensures that your manager maintains the most efficient load order.

In QEMM-386, you use the Optimize utility to configure upper memory and determine load order. In 386MAX, you use the Maximize utility to perform these services.

Third-Party Managers Don't Run While Windows Is Running

Memory managers like EMM386, QEMM-386, and 386MAX don't operate while Windows is running. Windows has its own built-in memory managers, which were codeveloped by Quarterdeck and Microsoft. When you load Windows, any third-party manager that you've loaded from within CONFIG.SYS gets "put to sleep," so that the Windows drivers can take charge.

Windows' built-in memory managers work a lot like QEMM-386 and 386MAX in the sense that they manage expanded and extended memory and try to use any available upper memory. When you exit Windows or run a virtual DOS machine in enhanced mode, your third-party manager again takes charge.

Because Windows doesn't really allow a third-party manager to do its job with Windows applications, you might conclude that QEMM-386 and 386MAX aren't really worth using with Windows.

Not so. Windows' built-in memory managers can't manage your TSRs and device drivers. You still need a system like EMM386, QEMM-386, or 386MAX if you want to optimize your upper memory area to support virtual DOS machines.

HOT TIP

Third-Party Memory Managers Work with Earlier Versions of DOS

You don't need DOS 5 to use a third-party manager. Both QEMM386 and 386MAX support DOS versions 3.0 and higher. If you already own one of these managers but haven't upgraded to DOS 5, you'll still be able to optimize memory to support Windows 3.1.

However, if you're trying to make an either/or decision—whether to upgrade to DOS 5 or to buy a third-party memory manager—we recommend that you first upgrade to DOS 5. The DOS 5 upgrade package costs about half as much as a third-party memory manager, and DOS 5 contains a number of built-in features designed specifically to boost the performance of Windows. And, of course, you can still use DOS to perform many memory optimization features.

QEMM-386 and 386MAX Footprints

The footprint required by QEMM-386 and 386MAX is much smaller than the footprint required by EMM386.

A *footprint* is the amount of space that a program requires within the first 640K of conventional memory (also called low memory). The smaller a program's footprint, the more memory will be available for each virtual DOS machine that Windows creates.

EMM386 uses about 10K of low memory, so it has a 10K footprint. QEMM-386 does much better, requiring a footprint of only 3K to 4K. 386MAX gets by with a relatively tiny 1K footprint.

Viewing and Analyzing Memory Use

Both QEMM-386 and 386MAX provide menu-driven memory viewers that you can use to examine memory use and to look for possible memory conflicts.

Windows supplies a good menu-driven memory viewer within the MSD program (discussed in Chapter 2). QEMM-386 and 386MAX each provide a comparable memory viewer. QEMM-386's memory viewing and analysis program is called Manifest. 386MAX's memory viewing and analysis program is called ASQ. (We'll show you some features of ASQ later in this chapter.)

You'll probably find that you need to use a memory viewer much less frequently with a third-party memory manager than if you are using

QEMM-386. The reason, of course, is that you don't have to configure upper memory yourself, nor are you responsible for altering your CONFIG.SYS and AUTOEXEC.BAT files to load programs and drivers into the UMA.

However, memory viewers are important tools for locating and correcting memory conflicts. Simply put, when you're using DOS, Windows, a third-party memory manager, and TSRs, you *will* run into problems. The streets are getting just too crowded, even though your memory manager is trying its level best to keep a good road map.

For instance, many older TSRs won't run in upper memory, while others require expanded memory in a way that confuses Windows. As still another example, some video drivers that switch between display modes (such as EGA and VGA) change their location in upper memory when they switch modes; this often causes a clash when Windows or your memory manager tries to use the same upper memory region.

We'll dig deeper into memory conflicts and troubleshooting techniques in Part 6. For now, just recognize that a good memory viewer can help you find and correct memory conflicts quickly. Your blood pressure might depend on it.

Advanced Features of QEMM-386 and 386MAX

QEMM-386 and 386MAX have several other memory optimization tricks up their respective shirt sleeves. Some of these features are built into the memory manager, while others only work if you give your manager the okay to use them.

In any case, most of the advanced features that we'll discuss in this section are useful if your memory is chronically in short supply and you need every spare byte. The options we'll explore don't represent an exhaustive list; we just want to give you a good introduction to the high-performance features of QEMM-386 and 386MAX.

Using Squeeze and FLEXFRAME

If you need to reserve a page frame for applications that use EMS, you can still use this page frame to load TSRs.

With QEMM-386, you can specify a page frame by including the RAM parameter in the DEVICE=EMM386 line of your CONFIG.SYS file. But when you do this, you lose 64K of valuable upper memory space that can't be used by your TSRs.

QEMM-386 and 386MAX both acknowledge this problem. Although these managers do have to keep the page frame available for your EMS applications, TSRs always load at startup time, when the page frame is still empty.

QEMM-386 and 386MAX take advantage of this space by letting a "shrinking TSR" use the page frame temporarily during its initialization phase. Then, when the TSR shrinks to its resident size, the page frame space is reclaimed. This procedure can be repeated for each shrinking TSR to be loaded into upper memory. After all TSRs have been loaded, the page frame becomes available for use by your EMS applications.

In QEMM-386, the option that provides page frame space temporarily to a TSR is called Squeeze. In 386MAX, this feature is called FLEXFRAME.

ROM Management

QEMM-386 and 386MAX are both smart enough to evaluate your system's ROM requirements and can often optimize the way ROM is used.

These two memory managers provide three basic ROM management techniques:

ROM Caching QEMM-386 and 386MAX test the ROM on your system's motherboard and other adapter cards. The idea is to determine whether ROM code can be transferred to the CPU faster if it is stored in RAM. If so, the memory manager caches the contents of ROM into RAM, usually in extended memory. This feature can speed up disk access and the speed at which your video driver redraws the screen.

Managing Shadow RAM Some computers have additional motherboard RAM that uses the 384K address range from 640K to 1024K. This RAM is used to move code from the slower ROM chips into fast RAM. This technique is called *shadow RAM.* QEMM-386 and 386MAX can either disable this shadow RAM and perform ROM caching instead, or can reclaim unused shadow RAM space.

Managing Shadow ROM On some systems, the ROM BIOS moves and remaps the ROM used by an EGA/VGA card. Specifically, the video ROM is moved from hex address C000 into the region starting at hex address E000. This procedure allows an expanded memory manager like EMM386 (which only searches for the range C000 to E000) greater flexibility in placing a page frame.

However, for a manager like QEMM-386 or 386MAX, which can search for and use all available upper memory, this creation of shadow ROM only serves to fragment the UMA and makes it more difficult to fit all TSRs and drivers into upper memory. QEMM-386 and 386MAX can turn off this shadow ROM capability, and then can relocate the video memory in an address space that does not result in fragmentation of upper memory.

QEMM-386 and 386MAX Support Standard Mode

If you try to start Windows in standard mode while EMM386 is running, you'll get an error message. You won't have this problem with QEMM-386 or 386MAX.

QEMM-386 version 6 and 386MAX version 6 support both standard and enhanced mode, so, you can start Windows in standard mode while QEMM-386 or 386MAX is running. You don't need to take any special steps to disable or modify QEMM-386 to support standard mode. Just type WIN/S at the DOS prompt.

386MAX Provides Extra Support for DOS Machines

386MAX lets you provide a separate upper memory environment for each DOS window.

386MAX provides virtual DOS machine enhancement through the use of two features: virtualized high DOS and instancing. The virtualized high DOS feature creates a unique copy of upper memory for each DOS machine. This feature allows you to load a TSR into high memory for only one DOS window; the TSR will not occupy space in the UMA for other DOS windows. Virtualized high DOS is useful if your upper memory space is cramped, because you can load different TSRs for different DOS windows.

Instancing adds a twist to the virtualized high DOS concept. Since 386MAX creates a unique upper memory area for each DOS machine, you can run the same TSRs in different DOS windows. In other words, you can run different instances of a single TSR. Many TSRs can normally run in only one window at a time. However, instancing removes this limitation.

Note: Virtualized high DOS and instancing are not available with QEMM-386.

HOT TIP

Use BlueMAX with PS/2 Machines

IBM PS/2 386 and 486 systems reserve 128K of upper memory for the ROM BIOS, while most other PCs reserve only 64K for the ROM BIOS. BlueMAX allows you to recover much of this memory.

Much of the PS/2 ROM BIOS space is reserved to support OS/2 and a few other rarely used devices. BlueMAX is a separate memory-management product designed specifically for PS/2s. BlueMAX can reclaim and use much of the unused ROM BIOS area of upper memory. If you have a PS/2, and are considering buying a third-party memory manager, we recommend you look into Qualitas' BlueMAX as an alternative to Qualitas' generic 386MAX or Quarterdeck's QEMM-386.

A Demonstration of 386MAX

In this section, we'll try to give you a better feel for the benefits of a third-party memory manager by demonstrating some capabilities of 386MAX. We've chosen to demonstrate 386MAX because other books—and even Microsoft—tend to favor QEMM-386. The only possible valid reason for this favoritism is that earlier versions of 386MAX operated unstably on some systems. However, the 6.0 version of 386MAX is just as clean and powerful as QEMM-386.

So, in the interest of fairness and in providing equal time for equivalent products, we'll use 386MAX for our demonstration. Do keep in mind, though, that most of the features we'll show in 386MAX are also available with QEMM-386.

Installing 386MAX

The 386MAX standard installation examines your upper memory area for unused addresses and copies the 386MAX files to your hard disk. The standard installation also deletes any existing memory manager references from your CONFIG.SYS and AUTOEXEC.BAT files.

After this standard installation has been completed, you are ready to reboot your computer and run the Maximize program. Maximize configures upper memory and determines the load order for your TSRs and device drivers. Then, Maximize modifies your AUTOEXEC.BAT and

CONFIG.SYS files so that the 386MAX UMB loader can load your TSRs and device drivers into memory.

HOT TIP

Do Not Install 386MAX's QCache Utility

When you install 386MAX, it will offer to install QCache, its cache utility. If you are running Windows 3.1, you do *not* need QCache; SMARTDrive 4.0 does a better job of caching code in extended memory. 386MAX will not be able to detect SMARTDrive 4.0, because it looks for SMARTDrive in CONFIG.SYS, not AUTOEXEC.BAT. (SMARTDrive 3.1 was a device driver, and was loaded from CONFIG.SYS; SMARTDrive 4.0 is a full-fledged TSR, and loads from AUTOEXEC.BAT).

You often cannot run two cache utilities at the same time. So, make sure you tell 386MAX not to use QCache. 386MAX will treat the SMARTDrive 4.0 line in AUTOEXEC.BAT just as it treats any other TSR. In other words, 386MAX will load SMARTDrive into upper memory. That's fine, but it's also unnecessary; SMARTDrive loads itself into upper memory automatically if it detects available upper memory blocks.

When you start Maximize from the DOS prompt, the screen shown in Figure 3.1 appears. You'll notice that this screen asks you whether to use the Quick or Full Maximize options.

Quick Maximize determines load order for you. Full Maximize allows you to specify where TSRs and devices should be stored in upper memory

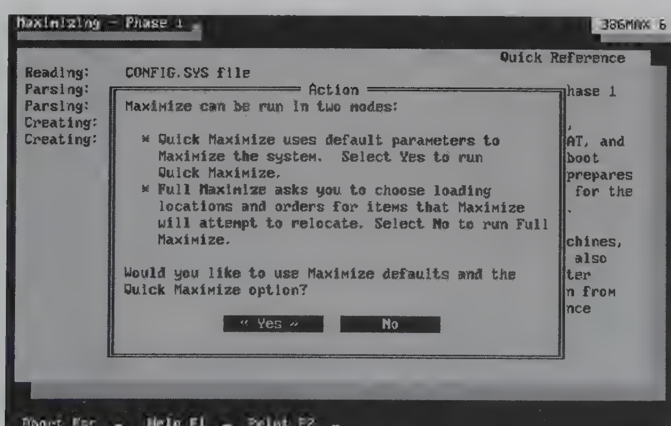


Figure 3.1 This 386MAX installation screen asks whether you want Maximize to determine load order or whether you want to control load order.

and in what order. Usually, you'll want to select the Quick Maximize option to let the Maximize program do all the work for you.

Before Maximize can determine load order, it needs to configure upper memory blocks currently reserved for system ROM. You'll see the screen shown in Figure 3.2, which asks you to confirm the ROM search function. Also note that you need to insert a diskette in a disk drive, so that Maximize can determine how your disk controller card's ROM is used.

After Maximize determines how to configure your upper memory, it reboots your computer so that the upper memory blocks can be created. You'll then see the screen shown in Figure 3.3. This screen lets you verify

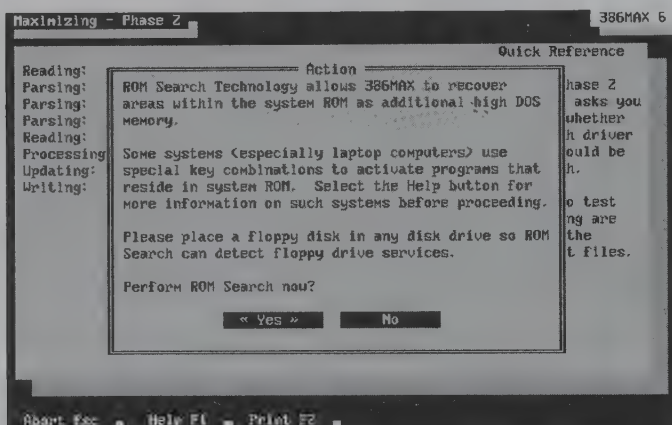


Figure 3.2 Maximize asks you to confirm the ROM search operation.

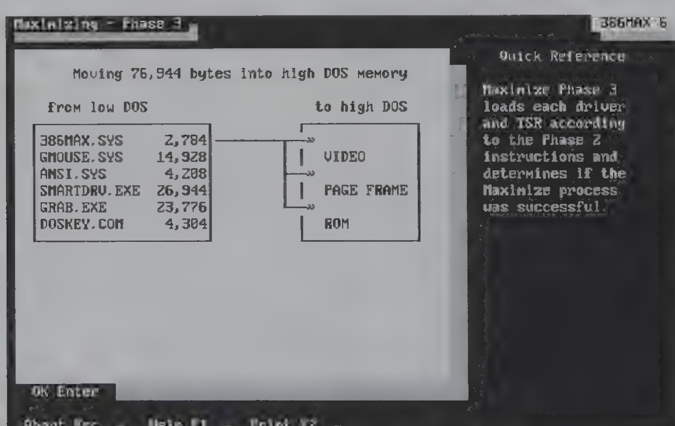


Figure 3.3 This screen identifies the TSRs and drivers that 386MAX will load into upper memory, and shows the general regions where they will be loaded.

the programs that the loader will place in upper memory. Figure 3.3 shows six programs that will be mapped into upper memory, including the 386MAX.SYS UMB provider itself.

When you press **Enter** to complete the Maximize procedure, 386MAX reboots your computer again, so that the designated programs can be loaded into upper memory. You can now start Windows.

Using ASQ

ASQ is the 386MAX memory viewer and analyzer (similar to the Windows-supplied MSD program and QEMM-386's Manifest program). Use this utility to examine your system's memory and to perform hardware diagnostics.

Type ASQ at the DOS prompt to start ASQ. When ASQ starts, it displays the "Load a Snapshot" screen shown in Figure 3.4. A snapshot is a file that stores a version of your system's memory use and hardware configuration. You can save different snapshots (files) to compare different memory and hardware configurations.

This feature is especially useful if you need to use trial-and-error troubleshooting techniques. For example, if you want to move hardware to different addresses to avoid memory conflicts, you might have to do some experimenting and rebooting to find a configuration that works. If you save each configuration to a snapshot file, you can keep track of configurations that you've already tested.

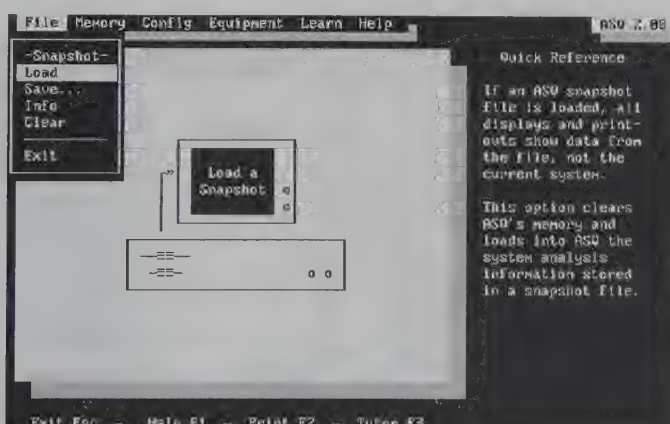


Figure 3.4 You can save your system's memory and hardware configuration to a snapshot file for diagnostic purposes.

Using ASQ to View Memory

The options on the ASQ Memory menu allow you to view all memory at once or to view only a particular type of memory (low DOS, high DOS, extended, expanded, and so on).

Figure 3.5 shows the display that appears if you choose High DOS from the Memory menu. The memory list provides the starting hex address for reserved areas and programs stored in upper memory, along with their total size in decimal (bytes). You'll also see PATH and other environment information for each program loaded in upper memory.

Using ASQ to View CONFIG.SYS and AUTOEXEC.BAT

You can use the Config menu to view the contents of your CONFIG.SYS and AUTOEXEC.BAT files.

Figure 3.6 shows how our AUTOEXEC.BAT file looks after we've used Maximize to organize upper memory. Notice the 386load size= setting for the three TSR programs that are loaded into upper memory. The value in this setting is the size required by the TSR to initialize itself (before it shrinks to its resident size).

You can only view CONFIG.SYS and AUTOEXEC.BAT; you can't edit these files from within the ASQ utility. Keep in mind that we haven't covered even half of the features that are available with a third-party memory manager like 386MAX or QEMM-386. For specific tips and troubleshooting issues regarding these two programs, see Chapter 17.

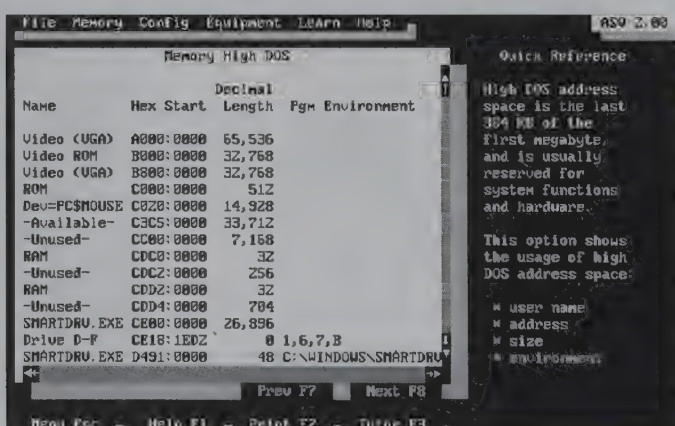


Figure 3.5 You can browse through all of your memory, or you can browse through each type of memory separately.

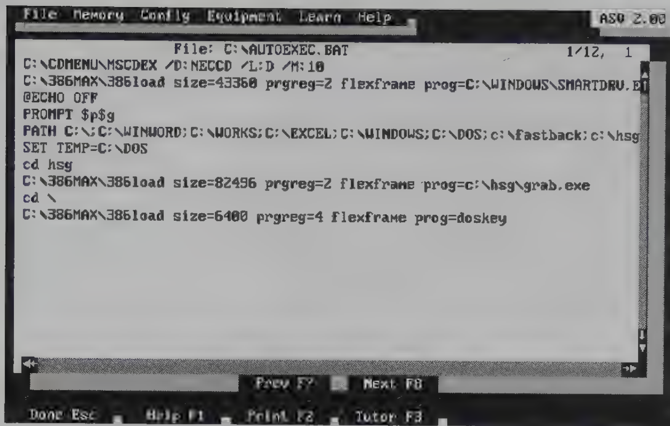


Figure 3.6 Use the Config menu to display the contents of your CONFIG.SYS and AUTOEXEC.BAT files.

CHAPTER

4

Windows Architecture and Optimization

This chapter is for those who aren't satisfied with simply knowing how to push the right keys and mouse buttons to make Windows work. If you're the type who opens the hood to your car and just *has to know* what that rectangular gray thing is behind the starter, then this chapter is designed for you. At any rate, we hope we've convinced you by now that a more thorough understanding of the way Windows works will help you get the most from the program.

In this chapter, we explain the software components that make Windows—well—Windows. We'll tell you the basic philosophy that underlies the operation of the Windows environment and we'll explain the files that turn this philosophy into an environment, that can manage virtually any application you start. Most important, we'll explain how to make the Windows operating environment work for you—with specific tips for optimizing Windows' performance.

Windows: An Event-Driven Environment

Thomas J. Watson, founder of IBM, was fond of telling his employees that, "Nothing happens around here until somebody sells something." Watson was trying to drive home the point that somebody has to *want* your product before you can set about to provide the product to people.

Watson was describing a sales-driven environment. You sell an idea to a buyer, then you set in motion all the forces required to manufacture the product that the buyer wants.

There's a parallel here for the way Windows works. Nothing happens in Windows until the user (you) does something. Windows is essentially a manufacturing environment waiting for a sales order to come in. The sales orders, of course, are initiated by you. Each time you press a key or click a mouse button, Windows kicks into gear and performs a multitude of processing steps to cater to your request.

Programmers call this an *event-driven* environment. (You'll sometimes hear events referred to as *messages*—different term, same meaning.) Each keypress and each click of the mouse triggers an event that causes Windows to respond in some way. In fact, the event-driven philosophy behind Windows is the key to allowing multiple programs to run safely at the same time.

Event-Driven Processing vs. Traditional DOS Processing

To explain event-driven processing, we need to compare this approach to the one used by DOS applications (like Lotus 1-2-3 or WordPerfect 5.1).

When you load an application like WordPerfect, the program essentially tells DOS to send all keypresses or mouse clicks directly to WordPerfect for its own use. If the program needs to use other devices—for example, to read a file from the hard disk or to print a file—the program requests one of the input/output services provided by DOS.

Under this operating philosophy, only one program can be "king of the hill" at a time. You can't easily run multiple programs simultaneously or have different programs share files if you're relying on this DOS-based approach. DOS wouldn't know what to do if, say, two programs requested printer services simultaneously, or if two programs tried to write to the same memory address.

Windows *does* allow this kind of multitasking by taking over many of the services that were once performed by DOS. Windows leaves some of the basic disk-management services to DOS, but provides its own routines for managing most other input/output events. And that's where the event-driven philosophy becomes important.

When Windows loads, it takes over the management of any other programs that you start and use while Windows is running. So, if you're currently running Word for Windows in the active window, each key that you press or each mouse movement that you make must first be passed to Windows as a separate event or message. Other Windows applications that are running, including Word for Windows, continually *poll* Windows to see if any events have been received for the application's use. In other words, each application continually asks Windows if it needs to respond to incoming messages (events). In our example, messages are passed to Word *only if they're intended for Word* (and not for Windows' exclusive use or for use by some other open application).

Since Windows might be polled by several different open applications, Windows itself—not the programs—determines how to handle each event. For example, when Word for Windows polls Windows to see if any keystrokes have been made, Windows passes the keystrokes to Word for its own use. Other open Windows applications "know" that they're not the active Window, so they won't poll Windows for keystrokes or mouse clicks. The difference between traditional processing and event-driven processing is illustrated in Figure 4.1. You might want to refer to this figure periodically as you read through the material in this chapter.

Some Events Do Not Get Passed to Applications

Message passing also lets you work directly with the Windows environment itself, since all keystrokes are first evaluated by

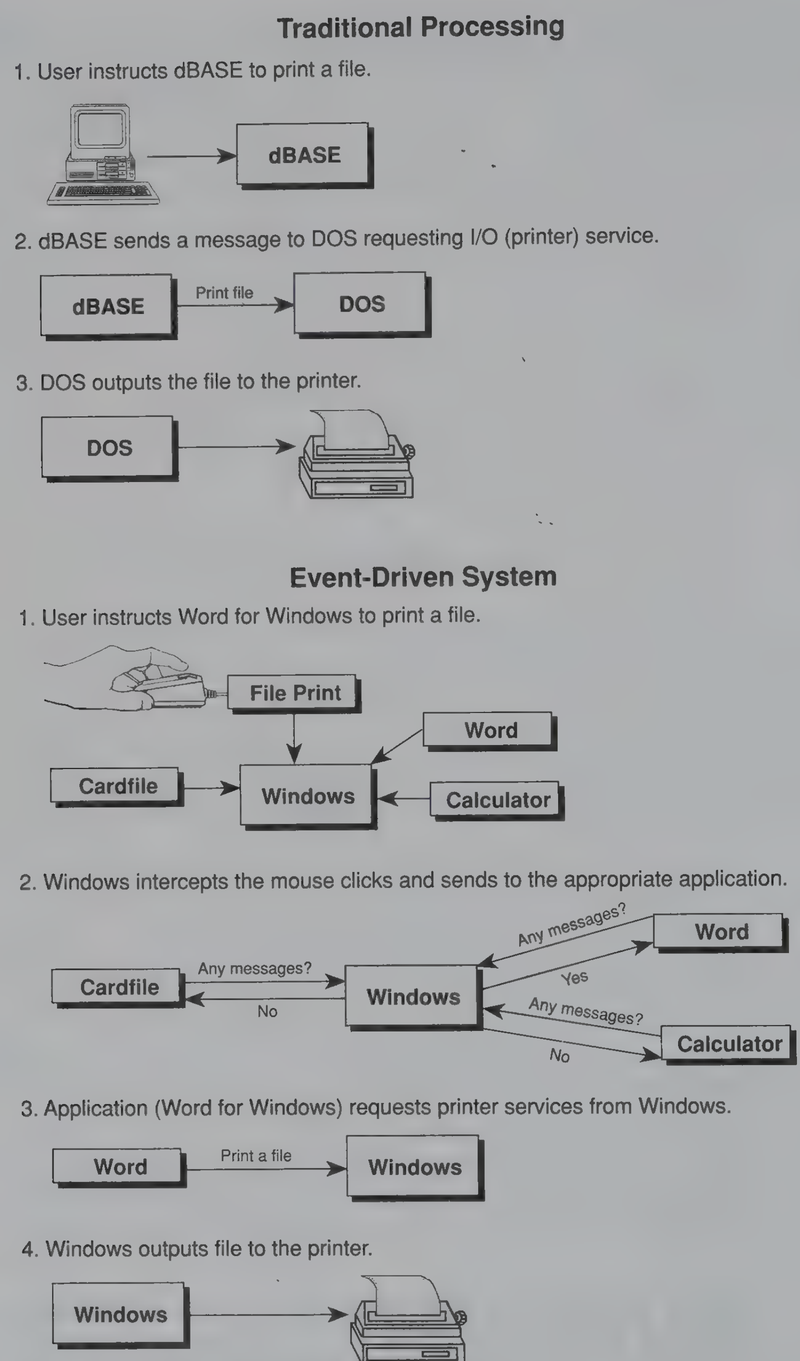


Figure 4.1 Event-driven processing is more complex than traditional (DOS-driven) processing, but allows several applications to run at the same time, without conflicting with each other.

Windows, not by the current application. Consider what would happen if you held down Alt and then pressed Tab.

Even though Word for Windows might be the active window, Windows won't pass the Alt+Tab keypress to Word. Windows gets the event first and recognizes that Alt+Tab is one of a set of keypresses reserved for its own use. When you press **Alt+Tab**, Windows responds by switching to a different application window, and doesn't pass the event to the active window.

In other words, Windows is capable of determining which events get passed to which applications, and can even determine whether events are meant for its own internal use or whether they get passed directly to DOS.

We'll provide some more examples of how events work later in this chapter. For now, it's important to keep in mind that Windows manages most, but not all, of the events that are triggered by your hardware devices. Windows still relies on DOS to perform some basic file-management services.

Windows Is a Bureaucracy That *Works*

Now that we've explained the basic concept of an event-driven environment, you might be wondering why you need to be concerned about all of this. Strictly speaking, you don't. You can do lots of tricks with Windows without ever recognizing how Windows manages events. But if you want to know how to optimize Windows as effectively as possible, you should understand not just Windows' event-driven techniques, but some of its underlying structure or architecture.

The Three Layers of Windows

You can think of Windows as a well-controlled bureaucracy that never passes the buck without a good reason. When Windows detects an event, it is essentially trying to determine how to avoid conflicts among different applications. Windows resolves potential conflicts by determining how to distribute the work (required to process the event) to other layers in its "bureaucracy."

Unlike Uncle Sam and his federal-government cronies, Windows is efficient because it is limited to three basic layers, and each layer has strict rules about when to pass an event to a different layer. True, Windows, has a few "sub-layers," but these additional layers are devoted specifically to the operation of higher-level layers. No particular sub-layer is an "agency" unto itself.

These are the three layers that Windows uses:

- Core
- Device-independence
- Applications

Figure 4.2 illustrates the relationships among the three layers of Windows and other parts of your system. You might want to refer to this figure from time to time as you read through the material in this chapter. Keep in mind that, even though we're going to describe the layers in Core, Device-independent, and Application orders, this hierarchical arrangement is just to bring organization to our discussions. The relationships and message passing among the different Windows components are actually much more complex.

The best way to uncloud the mystery of Windows and its operations is to explain the role of each Windows layer in the management of events. Along the way, we'll try to provide additional information about the specific capabilities and limitations of standard and enhanced modes.

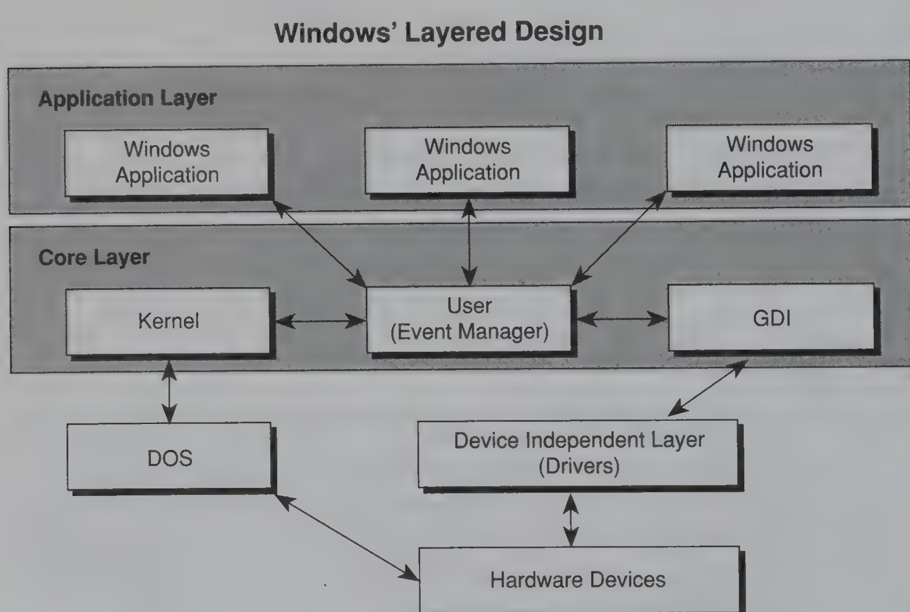


Figure 4.2 The relationships among different components of Windows, hardware devices, and applications are complex, but these relationships make it possible to run multiple applications at the same time.

Layer 1: The Core

As the name implies, the Windows *core* is a collection of files that control the basic central operations of Windows. The core files determine what to do with each of the millions of events that Windows might have to handle during the day. Three components make up the core layer of Windows: the kernel, the event manager (also called the user component), and the graphics device interface.

The Windows Kernel

In the realm of computers, a kernel is the part of an operating environment that controls all of the peripheral devices attached to the CPU.

In DOS, the kernel is the COMMAND.COM file, the ROM BIOS, plus a few hidden system files. You can't use your keyboard, monitor, printer, or any other device until the DOS kernel is loaded and ready to go.

The kernel used by Windows is different for standard and enhanced modes. The KRNL286.EXE file is the kernel when you are running Windows in standard mode, while KRNL386.EXE is the kernel when you are running Windows in enhanced mode.

Windows uses a file called WIN.COM to determine in which operating mode to start Windows. WIN.COM is the first file to load when you start Windows at the DOS prompt and is responsible for loading the rest of Windows. To make sure it does its job responsibly, WIN.COM determines what processor your system has and analyzes your basic memory configuration.

If you have a 286 processor or if you don't have enough extended memory to support enhanced mode, WIN.COM will load the KRNL286.EXE file. If you have a 386 or 486 processor, and enough extended memory to support enhanced mode, WIN.COM will load the KRNL386.EXE file. Of course, you can tell WIN.COM to start Windows in standard mode by typing **WIN /S** at the DOS prompt.

The KRNL286.EXE or KRNL386.EXE program file that WIN.COM loads next is responsible for allocating memory, starting and exiting applications, and scheduling the way different applications poll Windows and use multitasking features.

Only one kernel file can be loaded at a time. If you are running Windows in enhanced mode, for example, only KRNL386.EXE will be loaded into memory. In fact, if you never run Windows in standard mode, you could delete the KRNL286.EXE file from the Windows directory of your hard disk.

The Event Manager—USER.EXE

The file USER.EXE manages virtually all of the events that originate from you, the user. In other words, USER.EXE determines what to do whenever you press a key or click the mouse.

When you press a key, USER.EXE determines which application should be given the keypress. As you might guess, USER.EXE receives the polling requests made to Windows by open Windows applications.

USER.EXE also manages the way windows are sized and moved on the desktop, and tracks and controls the use of all icons. In addition, USER.EXE is responsible for responding to any mouse click made to any part of a window, such as the Control-menu box, the title bar, the minimize and maximize buttons, and the scroll bars. Whenever you make a mouse click or press a key that changes the way one or more windows will appear, USER.EXE passes this event to the file GDI.EXE.

The Graphics Device Interface—GDI.EXE

The GDI.EXE file is called the graphics device interface, and is responsible for drawing the more primitive portions of windows, icons, and other graphic images that appear on your screen. GDI.EXE responds to requests made by USER.EXE to modify the way your screen appears.

Suppose you want to resize a window or move an icon. The mouse clicks and mouse movements you make represent events that USER.EXE must examine and interpret. If USER.EXE determines that you want to resize a window border or reposition an icon, it passes the event to GDI.EXE so that the screen can be redrawn according to the instructions provided by your mouse movements. USER.EXE also sends data about the revised screen to the Windows kernel (KRNL286.EXE or KRNL386.EXE), so that the new graphical data (that describes the screen) can be stored in memory.

Managing System Resources

A major headache with Windows 3.0 was the inability for Windows to manage its system resources within the allocated 128K of

memory space. Windows 3.1 solves this problem by allocating an additional 64K for system-resource management.

USER.EXE and GDI.EXE do the real heavy-duty work required to manage menus, windows, icons, program groups, and other elements of the Windows environment. To support these two workhorses of the core layer, Windows 3.0 allocated 64K of memory space for each Windows component—64K for USER.EXE and another 64K of space for GDI.EXE.

Each 64K section of RAM is organized into a structure called a *heap*. Don't worry about the internal organization scheme used by a heap. Just keep in mind that the contents of the two Windows heaps represent the amount of *system resources* that Windows is currently using. In other words, system resources are the collections of information stored in the USER and GDI heaps, and are used to manage the screen, keyboard, and mouse I/O.

With Windows 3.0, Microsoft made a grave error in predicting the maximum amount of system-resource memory that users would require to manage open windows. Specifically, the USER heap grew too large when the user opened several group windows at a time or several applications at a time. The result was that Windows couldn't run because it was out of system-resource memory, even though several megabytes of physical memory might still be available for storing applications and data.

HOT TIP

Use the Program Manager to Determine Your System-Resource Usage

You can find the percentage of system-resource space that is still available to Windows by clicking on About Program Manager from the Program Manager Help menu. If Windows routinely reports that you have 25 percent or less of available system-resource space, you should consider reorganizing your program groups.

Microsoft has, for the most part, eliminated the shortage of heap space in Windows 3.1 by giving USER.EXE two heaps rather than one. So, in Windows 3.1, GDI.EXE has one 64K heap, while USER.EXE has two 64K heaps, for a total of 192K. This is a 64K improvement over Windows 3.0's heap space. You'll find that this additional 64K is usually more than enough to manage your Windows 3.1 environment.

HOT TIP

Reorganize Your Icons if You Run Out of System-Resource Memory

Even in Windows 3.1, there might be situations where you run out of system-resource memory. You shouldn't try to help Windows manage system-resource space by limiting the applications that you run: if you need an application, Windows should let you run it—provided you have enough physical and virtual memory to load the application.

However, you *can* control system-resource space by organizing your icons into several different program groups, and then keeping only one or two program groups open at a time. Here's why: for every icon in every open group, Windows has to store—in its heap space—information about the icon's position, color, appearance, and the font used for the icon's label. If all of your icons are located in one group, Windows has to manage all of these icons at once. By organizing the icons into different groups that can be opened selectively, Windows will need less system-resource space to manage your screen.

Layer 2: Device Independence

When you load an application from DOS (without using Windows), the application takes control of all devices attached to your computer system. In this sense, devices include your monitor, printer, mouse, network card, and communications ports. This hog-it-all approach won't work if you want to run two or more applications at the same time; to run multiple applications, your operating environment must be able to resolve any conflicts that might occur if different applications try to use a device—such as the printer or monitor—at the same time. By itself, DOS can't deal with the kinds of problems that can result when multiple programs try to use a device at the same time.

Windows provides a separate layer of files that support an approach called *device independence.* With this approach, devices attached to your system are controlled directly by Windows—and not by your applications. Windows applications can request the services of a particular device, but these requests have to be made to the Windows kernel.

Device independence works well with Windows applications because any program written specifically to run under Windows has to request screen, printer, and other device services directly from Windows. In other words, a Windows application can't request DOS to perform these screen management or printer services directly. In these situations, both DOS and Windows applications are essentially slaves to Windows.

As we'll explain a bit later, device independence becomes a bit of a problem when you run DOS applications under Windows, because these applications weren't designed to conform to Windows' device-independent rules. For instance, when you try to print a file within a DOS application, the program makes a request directly to DOS, bypassing Windows.

Device independence can be enforced for DOS applications running under Windows, but the technique is different depending on whether you are running in standard or enhanced mode. Again, we'll explain these differences later in the chapter.

Using Device-Independent Files

Windows accomplishes device independence through the use of device drivers. A different driver file controls each hardware device.

When you install Windows, the setup program identifies the type of keyboard, mouse, monitor, and network connected to your system, then installs the appropriate driver files for the devices. However, Windows can't determine which printer(s) you've installed; that's why you have to help out Windows by identifying the printer drivers that you want to install.

Device independence is convenient for you because all of your applications must use the same drivers. You don't need to set up different drivers for each application. For instance, if you've specified the HP LaserJet III as your active printer, then Excel, Word for Windows, Paradox for Windows, and all other Windows applications that you run have to use the HPPCL5A.DRV printer driver, which controls the operations of the HP LaserJet III. If you use the Control Panel to change the active printer, the new printer driver is used by all Windows applications.

Font Files

Although font files aren't really device drivers, they directly contribute to the device independence of Windows. Some Windows-supplied fonts are designed solely to support the display of text on particular monitors, while TrueType fonts support both screen and printer output. Other fonts not supplied with Windows

3.1 can by installed (downloaded) by font managers and other applications, or can be built into your printer's ROM. The point to keep in mind here is that Windows has to keep track of all fonts, regardless of whether they're supplied by Windows, by a font manager, or built into the printer.

Windows needs to track fonts so that it can coordinate the way text and graphics for different applications are displayed and printed. You tell Windows about downloadable fonts by using the Add Fonts dialog box, available from the Control Panel. (HP-supplied fonts use a separate Install Fonts dialog box within Windows.) You tell Windows about printer fonts when you install printer drivers, which tell Windows which fonts a given printer supports.

We won't spend a lot of time discussing fonts in this chapter. If you want to learn more about the way Windows uses fonts, refer to Chapters 11, 12, and 13.

HOT TIP

Identifying Your System's Display Fonts

You can determine which Windows display fonts are available for your monitor by starting the File Manager and clicking on Options Font. You will see a list of available display fonts and TrueType fonts. (The font that you choose here will be used to display information in all directory windows of the File Manager.)

DOS Application Support Files

Windows also installs files that help you run DOS applications under Windows. These files include mouse drivers for using your mouse with DOS applications in enhanced mode. More important, Windows provides specific drivers and fonts that allow you to run DOS applications under Windows.

The two basic files that support the display and operation of DOS applications under standard- and enhanced-mode Windows are listed below:

WinOldAp This shorthand name indicates a little bias on the part of Microsoft. Your Windows applications are "new apps," while any programs designed specifically for DOS are considered to be "old apps."

Windows provides two WinOldAp files: WINOLDAP.MOD and WINOA386.MOD. The WINOLDAP.MOD file controls the execution of DOS applications (old apps) when you are running Windows in standard mode; WINOA386.MOD controls the execution of DOS applications for enhanced mode. The WinOldAp file also works with *grabber* files to support the exchange of data between DOS applications and Windows applications.

HOT TIP

Not All DOS Applications Are Old "Apps"

Just because an application is designed to run under DOS doesn't mean the application won't run efficiently under Windows. Many DOS applications are written to be "Windows aware." That is, even though the application can run without Windows, the application does take advantage of Windows features when it detects that it is running under Windows.

Some developers create Windows awareness for their DOS applications by packaging a sophisticated PIF (Program Information File) with the application. Other DOS applications come with their own PIFs, but also contain built-in features that make use of some of Windows' capabilities. Central Point's PC Tools, version 7.1, is an example of a highly Windows-aware application. This version of PC Tools runs well under Windows *and* under DOS, and even includes utilities designed specifically for Windows 3.0 and Windows 3.1.

The moral: don't assume that an application is obsolete just because it is not a "Windows app." A Windows-aware DOS application can often be just as versatile—if not more versatile—than a comparable Windows application.

Grabber files Windows typically installs two grabber files—one to support standard-mode windows and—if you have a 386 or 486 machine—a second grabber to support enhanced-mode windows. Standard-mode grabber files have the extension 2GR, while enhanced-mode grabber files have the extension 3GR. The specific grabber files that Windows installs depends on the display driver in use (VGA, VGA Monochrome, EGA, Video 7, and so on).

In standard mode, the grabber works with the Clipboard to let you copy and paste text between a DOS application and a Windows application. The grabber also makes it possible to use the Print Screen key to make a "snapshot" of the current screen and copy the screen image to the Clipboard.

In enhanced mode, the grabber allows you to copy and paste text and use Print Screen, but also provides a few additional features. Specifically, the enhanced-mode grabber is responsible for displaying text in a DOS window, selecting a section of text in a DOS window for editing, and copying graphics from a DOS window to the Clipboard.

Identifying Device Drivers and Font Files

Windows stores most of its device driver and font files in the WINDOWS\SYSTEM directory. You can usually determine how these files are used by examining their file extension.

Why should you care which drivers and font files are installed on your system? Because Windows might have installed more drivers than you'll need. Often, you can reduce the amount of hard disk space that Windows claims by deleting unwanted drivers or fonts. Table 4.1 explains several common extensions for files used to support device independence.

Layer 3: Applications

The *application layer* isn't really built into Windows; instead, it consists of the actual applications (programs) that you run under Windows. But when you run programs under Windows—even if you're running DOS programs—the applications still are essentially controlled by Windows, so you should think of your applications as a separate layer of Windows operations.

Windows applications communicate directly with the USER.EXE portion of the Windows core by polling USER.EXE to see if you have initiated any events (messages) that the application needs to handle. Generally, all Windows applications are designed to conform to the Windows *application program interface (API)*. An API is a set of rules, or conventions, that a program must follow in order to request and receive help from an operating system. DOS has its own API, as you might guess. A network also typically has its own API.

As a user, you don't really need to know much about the Windows API. You'll see the term API used frequently in books and articles on Windows, though, so at least now you won't have to be puzzled about the meaning of this term.

The Windows API and DLLs

The Windows API is the set of rules that a program must follow to take advantage of the event-driven and device-independent fea-

Table 4.1 Common Windows File Extensions

File Type	*Common Extension(s)*	*Examples*
Printer Drivers	DRV, (WPD for PostScript definition files)	HPDSKJET.DRV HPIII522.WPD
Display Drivers	DR, DRV	VGA.DRV SUPERVGA.DRV
System Fonts	FON	COURF.FON VGASYS.FON
TrueType Fonts	FOT, TTF (each font family requires two files)	TIMES.FOT TIMES.TTF
286 Grabber	2GR	VGACOLOR.2GR VGAMONO.2GR
386 Grabber	3GR	VGA.3GR V7VGA.3GR
Multimedia Drivers	DRV	MCIWAVE.DRV MMSOUND.DRV
Network Drivers	DRV	LANMAN.DRV NETWARE.DRV

tures of Windows, as well as most of the graphical components of Windows. The Windows API features an extremely important capability called the dynamic link library (DLL).

A DLL is a collection of functions that sits on your hard disk until one of its functions is requested by an application. DLLs serve two major benefits (and a lot of minor ones):

1. A DLL doesn't take up memory space because it is never loaded unless a specific program requests one of the services provided in the DLL.
2. A DLL must conform to the Windows API conventions; therefore, the functions within a specific DLL can be used by several different applications. This feature provides additional memory and hard disk optimization. A program doesn't need to include a particular function within its own code if that function is provided somewhere else by a DLL. Instead, the program just calls the DLL when a particular function is needed.

An example of a DLL is the file MORICONS.DLL, which is a library of functions that can create more than a hundred icons designed specifically for DOS applications. You can use the Program Manager or some other application to call MORICONS.DLL if you need to create an icon for a newly installed DOS application.

HOT TIP

Call MORICONS.DLL from within the Program Manager

If you want to assign a particular icon to a newly installed DOS application, or if you want to change the icon for an application, you can use the icons stored in PROGRAM.EXE or MORICONS.DLL. Here's how to use both files to assign an icon to a particular application.

1. To change the icon for an application, click on the current icon for the application, then click on File Properties from the Program Manager.
2. Click on the Change Icon button.
3. In the Change Icon dialog box, enter **MORICONS.DLL** or **PROGRAM.EXE** in the File Name dialog box, then browse through the window of available icons. MORICONS.DLL contains more than one hundred icons designed specifically for various DOS applications.
4. Click on the desired icon, then click on OK to exit the Change Icon dialog box.

As you might suspect, DLLs are really more the concern of Windows programmers than the average user. But you've probably seen files with DLL extensions in your WINDOWS directory and in the directories of other Windows applications. So, now you know what these files do.

Opening Multiple Applications Is Not the Same as Multitasking

An open application can be active without doing any processing. Multitasking only occurs when two or more applications simultaneously use the CPU and RAM.

Suppose you have both Excel and Word for Windows open at the same time. However, you only work with one application at a time; you switch between these two windows whenever you want to use the application that's currently in the background. In this situation, you're not really multitasking, because only one application is doing any processing at a given time.

On the other hand, suppose you link an Excel worksheet with a particular Word for Windows document. Each time you make a change in the worksheet, Windows has to update the change in the Word document. So, when you recalculate the spreadsheet, Windows has to process Excel and Word simultaneously. This is true multitasking—two or more applications are actually doing processing (processing text or numbers, printing, or writing data to or from disk) at the same time.

Unless you're doing true multitasking, Windows applications operate pretty much the same in standard mode and enhanced mode. That's not so for DOS applications, which operate significantly differently in standard and enhanced mode. And if you *are* doing multitasking, even Windows applications operate differently for the two modes. In the next sections, we'll provide a detailed look at the way Windows manages applications in standard and enhanced modes.

Standard-Mode Implementation

In Chapter 1, we walked you through several of the components of standard-mode Windows. In that chapter, though, we were mostly concerned with the way Windows manages memory in standard mode. In this chapter, we'll try to give you a better feel for the way Windows manages applications in standard mode. Although we have to repeat a bit of the material that we presented in Chapter 1, we'll at least try to put a new twist to what you've already read.

Starting Windows in Standard Mode

As we mentioned earlier WIN.COM is the first file to load whenever you start Windows. One of WIN.COM's jobs is to determine whether to start Windows in standard or enhanced mode.

If you've got a 286 system, WIN.COM has to start in standard mode. If you've got a 386 system, WIN.COM will start standard-mode Windows only if you type **WIN /S** at the DOS prompt, or if WIN.COM can't locate enough extended memory required for enhanced mode.

In any case, when you start Windows in standard mode, WIN.COM first loads DOSX.EXE, which manages extended memory for standard-mode Windows. You might be scratching your scalp at this point, because we've already said that extended-memory management is the job of HIMEM.SYS. Right? Yes, in one sense, but no in another.

HIMEM.SYS is an extended memory manager for *DOS*. You have to use HIMEM.SYS to configure extended memory at the DOS level. And, of course, HIMEM.SYS in coordination with EMM386 is useful for making upper memory available to TSRs and device drivers on a 386 or 486 computer.

But HIMEM.SYS is deaf, dumb, and blind to the way Windows works. And since Windows wants to create its own rules for managing extended memory, it provides its own DOS extender—DOSX.EXE—that takes control when you launch Windows in standard mode.

Be warned, though, that you can't start Windows in standard or enhanced mode unless you've used a memory manager like HIMEM.SYS (or a third-party memory manager) to configure extended memory prior to loading Windows. If WIN.COM can't find any extended memory, it won't bother to continue; you'll just be kicked backed to the DOS prompt.

HOT TIP

The Minimum Standard-Mode Configuration

To start Windows in standard mode on a 286 or higher processor, you must satisfy these *minimum* memory requirements:

- At least 256K of conventional memory must be available
- At least 192K of extended memory must be available
- An XMS driver—such as HIMEM.SYS—must be loaded and must have already configured extended memory

Although you can start Windows in standard mode with as little as 192K of extended memory, you can't really use Windows efficiently unless you have at least 1MB of extended memory (in addition to the 1MB used as conventional memory).

DOSX.EXE evaluates extended memory and then loads KRNL.286, the kernel program for standard mode. KRNL286.EXE is responsible for loading the other components of Windows, including the device drivers required for standard mode and the other components of the Windows core.

As we explained in Chapter 1, the biggest drawback to running Windows in standard mode is the inability to run DOS applications and Windows applications simultaneously. Remember: DOS applications don't conform to the Windows API.

When you start a DOS application in standard mode, Windows prevents any memory conflicts from occurring by starting the Windows task switcher file, DSWAP.EXE. This file creates a DOS environment (an emulation of real mode) in extended memory and then puts the rest of

Windows "to sleep" so that DOS can stay in control. Any currently open Windows applications are stored temporarily in a swap file on your hard disk.

When you switch from a DOS application back to Windows, DSWAP.EXE stores the DOS application and its open files temporarily in a swap file on your hard disk. Then, DSWAP.EXE reactivates the Windows environment so that you can use your Windows applications. But, the bottom line is that, in standard mode, you can't run Windows applications while you're running a DOS application, and you can't run more than one DOS application at a time.

HOT TIP

When in Windows, Use the Windows Task Switcher, Not the DOS Task Switcher

If you use the MS-DOS 5.0 shell as a replacement for the File Manager under standard-mode Windows, you might already be aware that DOS 5 supplies its own task switcher. The DOS 5 task serves the same purpose as the Windows task switcher, allowing you to switch between open applications without exiting Windows or DOS each time you want to use a different application.

However, if you are running Windows in standard mode, the Windows task switcher is enabled automatically. So, if you also enable the DOS 5 task switcher, you'll just be wasting memory on a redundant feature. You should always disable the DOS 5 task switcher while you are running in Windows. To do so, click on Options from the DOS Shell menu bar, then deselect Enable Task Swapper if it is currently checked (turned on).

Enhanced-Mode Implementation

When you run Windows in enhanced mode, you're pushing the envelope of Windows' capabilities. Anything you can do in standard mode you can do in enhanced mode; but much of what you can do in enhanced mode isn't permitted in standard mode. Because Windows unleashes all of its power in enhanced mode, we'll spend the remainder of this chapter explaining different features and optimization techniques for enhanced-mode Windows.

The key to enhanced mode is actually the virtual-86 mode capabilities provided by the 80386 processor chip. Since we explained this mode in Chapter 1, we won't dig into it here. Just keep in mind that virtual-86 mode allows Windows to create virtual machines for DOS applications by defining an emulated real-mode environment for each DOS application. In virtual-86 mode, Windows can switch very rapidly between different virtual machines and Windows applications that are running in 386 protected mode (enhanced mode).

Each virtual machine is viewed as a separate computer by DOS, and all virtual machines are managed by Windows. With virtualization, Windows can run multiple DOS applications and multiple Windows applications—all at the same time. DOS is only a minor player in this scheme, and doesn't prohibit Windows from creating a separate window for each running DOS application. We'll explain how Windows takes advantage of this virtualization capability in the next sections.

Starting Windows in Enhanced Mode

In the previous section, we explained that, for standard mode, WIN.COM starts DOSX.EXE, the DOS extender for Windows. But DOSX.EXE can only create one emulated DOS environment, and therefore only allows one DOS application to run at a time. This won't do for the virtual-86 mode on 386 and 486 computers. Using DOSX.EXE in enhanced mode would be like asking Bobby Allison to race around the stock car tracks in a Yugo.

When WIN.COM detects that you want to start Windows in enhanced mode, it ignores DOSX.EXE and loads WIN386.EXE instead. The WIN386.EXE program file has built-in capabilities for virtualizing DOS applications and for managing extended memory under Windows. WIN386.EXE, in turn, loads the device drivers that support enhanced-mode windows and also sets up the *scheduling* environment that allows you to multitask.

Multitasking in Windows and the Timeslice Concept

In enhanced mode, Windows works its multitasking miracles by applying a concept called the timeslice. A timeslice is just the amount of CPU time, measured in milliseconds (thousandths of a second) that Windows allocates to an application before it gives CPU control to a different application. A typical timeslice is 20 milliseconds (.020 seconds).

Windows allocates timeslices differently for Windows applications and DOS applications. *All* open Windows applications share one timeslice.

This is efficient because Windows can assign processing priority based upon the events that need to be passed to the different Windows applications that are open. So, you don't have to worry about adjusting timeslices for Windows applications—unless you'll be running Windows applications concurrently with DOS applications.

Unlike Windows applications, *each* virtual DOS machine gets its own timeslice. Windows has to do this because DOS applications can't carry on the give-and-take type of message passing that is possible with applications designed specifically for Windows. If you've got several DOS applications open at a time, Windows switches quickly from one application to the next. Since this switching is based on timeslices that are measured in milliseconds, it will usually appear to you that Windows is processing all DOS applications simultaneously.

How Windows Schedules Tasks

Windows has to be told how to schedule, or manage, the different timeslices that are allocated to your applications.

As we've already mentioned, scheduling isn't a problem if you're only running Windows applications. USER.EXE can determine which Windows applications get processing priority based upon the number, frequency, and types of events that are passed to each application. But if you're running Windows applications and DOS applications simultaneously, scheduling gets more complex. The Windows kernel, in these cases, is responsible for scheduling.

The Windows kernel uses several built-in scheduling defaults that are designed to handle priorities and conflicts when Windows and non-Windows applications simultaneously need CPU time. And we should stress that, in most situations, the Windows defaults work just fine. But there *are* exceptions.

At any rate, you can do yourself a favor by learning how Windows schedules tasks. You might find that, with a little effort, you can modify scheduling defaults to better fit the way you use applications under Windows. At the very least, a knowledge of Windows scheduling capabilities can help you to identify performance problems and provide you with some ways to speed up a system that is performing sluggishly.

A Windows Scheduling Illustration

Scheduling is one of the most difficult concepts for Windows users to grasp, but it shouldn't be. We'll try to make the scheduling concept clear by using an analogy that's easy to follow.

The next few paragraphs require you to do a little number crunching. If you have a math phobia, don't sweat yet! We're going to make this information as painless as possible to digest. It's valuable information, so try to stick with us.

Windows assigns timeslices on a priority basis. Each application is given a *priority value* that is weighed against the total of priority values for all currently active applications. Yes, that's a confusing statement. But bear with us for a moment.

To understand timeslicing and scheduling, consider how this task-management approach compares with another commonly used approach, called a *queue*. A queue is just a waiting line, like the one you're often stuck in when you go to the supermarket. In the supermarket queue, people are served in order. Checkers don't adjust their service based upon the total number of items in your basket, nor do they take into account whether you have more or less items in your basket than the other customers in line. They just deal with each customer in the order in which customers are waiting.

In a timeslicing setting, though, you would get more attention if your basket was topped off with goodies. You might even get to move to the head of the line if you're really hauling a load.

Here's why. With timeslicing, the checker scans the line and totals the number of items for everybody currently waiting in line. (This is an *extremely* intelligent checker.) The total becomes the unit that the checker uses to calculate the percentage of time that each customer gets per minute. So, if eight people are waiting in line, and the total number of items in all eight baskets is 300, then the checker says, in effect, "Our best customers are the ones who are buying the most items; I'll figure out how to cater to them."

The checker now assigns a priority value based on the number of items in each customer's basket. Say you've got 120 items (you haven't shopped in a while), customer #2 has 20 items, customer #3 has 40 items, custom #4 has 12 items, customer #5 has 8 items, customer #6 has 30 items, customer #7 has 80 items—for a total of 300 items.

The checker now decides that, for each minute he spends servicing all eight customers, he'll give proportionately more time to the customers with higher priority values—those who have more items in their baskets. Since you've got 120 items (a priority value of 120) out of a total of 300 items, you get a whopping 40 percent of the checker's attention every minute. (120 divided by 300 equals .40); customer #7 has 80 items (a priority value of 80) out of a total of 300 items, so she gets about 26 percent of the checker's attention every minute (80 divided by 300 equals .26);

customer #3 has 40 items, so this customer gets about 13 percent of the checker's attention every minute; and so on.

In other words, each customer gets a percentage of the total amount of time allocated for all customers. And that's precisely how Windows allocates timeslices. Except Windows doesn't actually analyze the load of work that an application has to complete (which would equate to the load of items in each customer's shopping basket). Instead, processing priorities are set in advance for the Windows timeslice and for each DOS application.

We'll explain how to adjust priority values in a moment. For now, we want to make sure the concept of scheduling hits home. Suppose you're running four DOS applications simultaneously. Assume that the Windows applications are assigned a priority number of 100, while each DOS application is assigned a priority number of 50. So, 100 + 50 + 50 + 50 + 50 = 300. For every unit of CPU processing time, the Windows applications get about 33 percent of the CPU time (100 divided by 300 equals .33. Each DOS application gets about 16 percent of the CPU time (50 divided by 300 equals .16).

The precise amount of processing time that Windows allocates for each task is based on the timeslice value. But timeslicing is just a minimum. If you specify that each timeslice is 20 milliseconds, each task gets *at least* 20 milliseconds of CPU time before Windows turns the CPU over to another task. In our example, we have five tasks—Windows applications plus four DOS machines. If we allocate 20 milliseconds (.020 seconds) for each of five tasks (20 times 5), that means we have to distribute processing time across at least 100 milliseconds, or .100 (one-tenth) of a second. So the Windows applications get collectively about .33 seconds (.01 times 33) per timeslice, while each DOS machine should get .16 seconds (.01 times 16) per timeslice. Actually, the timeslices are a bit shorter than this, because Windows requires some of the total processing time just to switch between applications—and this time span varies depending on the clock speed of your computer. In any case, be aware that each timeslice can be considerably longer than the 20-millisecond minimum.

We've just explained how timeslicing works in theory. Unfortunately, Windows complicates matters by changing the rules when it puts timeslicing into practice. To see how and why Windows does this, read on.

The Actual "Time" for a Timeslice Can Be Elusive

Windows can adjust the timeslice for a task depending on whether the task is operating in the foreground or in the background.

The application that's in the currently active window (which is also the application that currently accepts input from the keyboard) is always in the foreground. All other open applications are considered to be in the background. In other words, you can have only one application at a time in the foreground, but you can have several applications running simultaneously in the background.

By default, Windows assigns a longer timeslice to the active window (the foreground application) than it does to other windows (background applications). Remember, one of Windows' goals is to make life easier for the user. Windows assumes (not always correctly) that the active window is currently getting your attention. By lengthening the timeslice for the foreground application, Windows is in a sense saying, "Hey, no one's paying attention to what I'm doing in the background, so I'll slow down the background applications and speed up the foreground application. That way, they'll think they're getting all my attention."

Adjusting Priority Values

Taken on its own merits, timeslicing for foreground/background applications is not too terribly difficult to grasp. (It's not too terribly easy to grasp, either.) But Windows gives you the ability to change the "balance of power" between your foreground and background applications—and that can lead to confusion.

As Figure 4.3 shows, you can use the 386 Enhanced dialog box to assign different priority values for the Windows task when it is in the foreground and when it is in the background. The default priority value is 100 when Windows is in the foreground, but this value changes to 50 when Windows is operating in the background. Remember our supermarket example. The values 100 and 50 are not percentages on they're own, they're just used to *determine* percentages. (If you have 100 grocery items in your cart, and the total number of items for all people waiting in line is 400, you get 25 percent of the checker's attention, not 100 percent.)

So, if you're running a DOS application in the active window, Windows will use the "Windows in Background" value of 50, rather than the "Windows in Foreground" value of 100. The point is that the actual timeslice for the Windows task can be adjusted depending on whether the task is running in the foreground or in the background. You can assign any value from 1 to 100000 in the Windows in Foreground and Windows in Background boxes.

Now take a look at the Exclusive in Foreground check box in Figure 4.3. If you select this box, Windows will ignore any DOS applications,

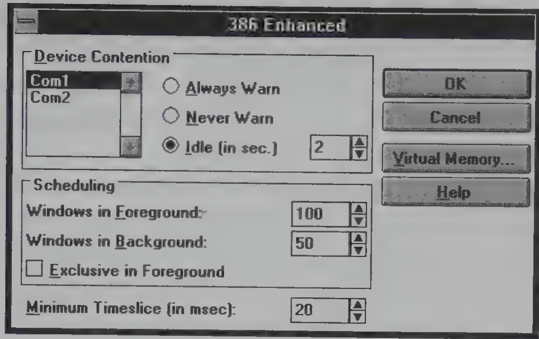


Figure 4.3 Use this dialog box to modify foreground and background settings for the Windows task.

while a Windows application is in the active window (foreground). In other words, Windows turns over the timeslices of all open DOS applications to the timeslice for Windows applications. No processing can take place in a DOS application while your Windows applications remain exclusively in the foreground. In fact, all DOS applications are swapped to disk if the Exclusive check box is selected. This setting is useful if you cannot use 32-bit disk access and must run all DOS applications in physical memory. (The 32-bit disk access feature is discussed later in this chapter.)

Running a DOS Application in the Background

You can also adjust priority scheduling for a given DOS application by modifying the application's PIF. For instance, you can determine whether to allow a particular DOS application to do any processing while it is running in the background.

As you might already know, a PIF contains memory management and other information that Windows uses to run a DOS application as stably and efficiently as possible. Windows stores PIF information for more than 200 DOS applications in the file APPS.INF.

When you use the Windows Setup to set up your DOS applications to run under Windows, the setup program creates a PIF for each DOS application by looking up the application's settings in the APPS.INF file. So, for most DOS applications, you don't need to create a PIF; Windows does it for you. However, you can view or modify a PIF's settings by starting the PIF Editor and then opening the PIF for a particular application. (We'll discuss PIFs in detail in Chapter 14 and Appendix B;

in this chapter, we'll just focus on the PIF settings that affect the way Windows manages the application in a multitasking environment.)

Figure 4.4 shows the PIF Editor dialog box for Microsoft Works 2.0. Take a look at the two Execution check boxes—Background and Exclusive.

When you select the Background check box, you tell Windows to let the application keep its timeslice when it is in the background. In Figure 4.4, the Background check box is selected for Microsoft Works. So, Windows will allow this application to do its processing even when it is not the active window (in other words, Works is not in the foreground).

If the Background check box is not selected, Windows will swap the DOS application to disk whenever you switch away from it. If you never do any processing in a particular DOS application when it is in the background, deselect this check box to increase the timeslices for other applications.

HOT TIP

The Windows Exclusive Check Box Overrides Any PIF's Background Setting

Windows will ignore the Background setting in a PIF file if you've also selected the Exclusive check box in the Windows 386 Enhanced dialog box. In other words, Windows won't let *any* DOS application run in the background when you're running Windows exclusively.

If you need to keep any of your DOS applications running in the background, make sure the Exclusive check box is deselected in the 386 Enhanced dialog box.

The Exclusive check box, also shown in Figure 4.4, is less useful than the Background check box, but it too can be used to manage timeslices. When you select the Exclusive check box in a PIF, Windows will give *almost* all processing time to this DOS application. Windows will still keep a timeslice for its own use so that it can manage memory and switch between the DOS application and the Windows environment. But your Windows applications and other open DOS applications won't be able to do any processing while this particular DOS application is running.

The benefit to this approach is that a DOS application requiring a lot of CPU time or memory can operate more efficiently. (What's "a lot?" you might ask; that depends on your CPU's speed, the amount of extended memory on your system, and the virtual memory settings that are in use.)

For example, if you're using dBASE IV to generate reports from some very large database files, you might want to speed up report generation

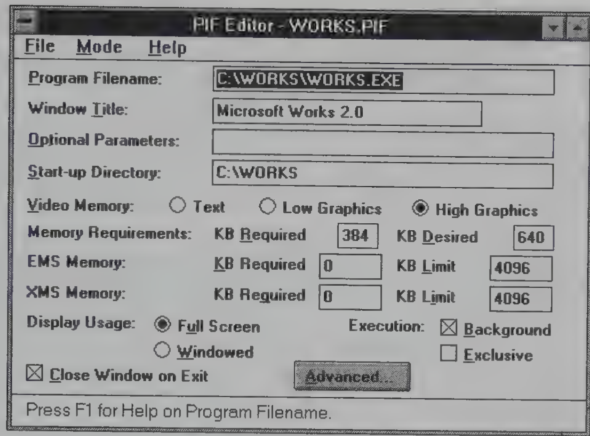


Figure 4.4 This PIF dialog box shows the default Execution (Background and Exclusive) settings for Microsoft Works 2.0.

by turning over most of the CPU resources to dBASE IV. If you need to, you can still interrupt the report-generation process by using **Alt+Tab** to switch back to the Windows environment. In most cases, though, you'll want to leave the Exclusive check box deselected in each DOS application's PIF.

Hot Tip

You Can Run a DOS Application Exclusively in a Window

In Windows 3.1, you can run a DOS application exclusively either in a window or in full screen. In Windows 3.0, however, you could only run a DOS application exclusively in full screen.

Adjusting Priority Values for DOS Applications

You can fine-tune scheduling for a DOS application by changing its priority values. The background and foreground priority values are available when you click on the Advanced button in the PIF Editor dialog box.

Figure 4.5 shows the Advanced Options (enhanced mode) dialog box that appears when you click on the Advanced button in a PIF dialog box. Notice the Background Priority and Foreground Priority boxes, which are similar to the boxes shown for Windows applications in Figure 4.3.

In fact, you use these settings much like you use the settings for Windows applications. In other words, you can tell Windows to give a

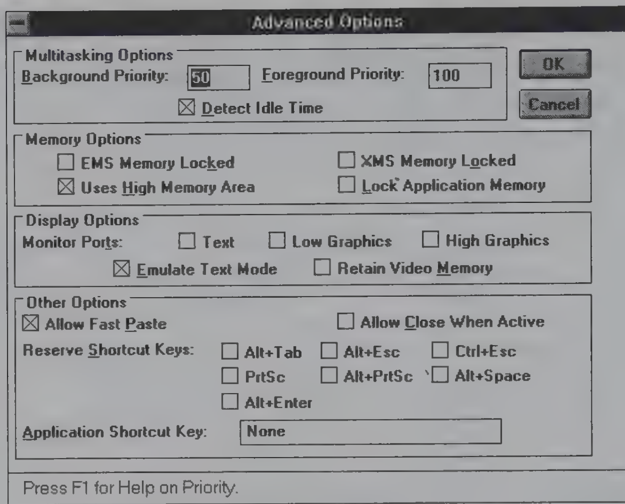


Figure 4.5 This Advanced Options dialog box shows the PIF settings for background and foreground priorities for Microsoft Works 2.0.

particular non-Windows application relatively more or relatively less processing time depending on whether it is running in the foreground or in the background. But remember: the Background Priority value is ignored if you've selected the Exclusive check box for Windows (available in the 386 Enhanced dialog box accessed from the Control Panel).

Using the Detect Idle Time Check Box

You can instruct Windows to give up a DOS application's timeslice if the application is not doing anything but waiting for characters from the keyboard. This feature is possible because Windows can detect when a DOS application is idle (not doing any processing).

Take a look at the Detect Idle Time check box in Figure 4.5. If you select this box in an application's PIF, Windows will swap the application to disk if it is not currently doing any processing. As soon as you send a keypress to the application, Windows will return the application's timeslice.

For instance, suppose you're using dBASE IV to print several reports in the background while you work on Excel in the foreground. The Detect Idle Time check box is selected in the PIF for dBASE IV. In this scenario, Windows will check to see whether dBASE IV is idle, so that the timeslice for dBASE IV can be turned over to some other application. Because dBASE IV is currently at work printing reports, Windows will let dBASE IV keep its timeslice.

Now suppose you start WordPerfect 5.1 (for DOS) to type a letter. Your phone rings and you interrupt your typing to take the call. If the Detect Idle Time check box is selected in the WordPerfect 5.1 PIF, Windows will notice that WordPerfect is just waiting for you to press a key. In response, Windows turns over the WordPerfect timeslice to dBASE IV so that your reports can be printed a little faster.

HOT TIP

Don't Use Detect Idle Time for a DOS Application That Is on a Timer

Some applications (especially calendars and other scheduling utilities) allow you to set a timer so that some sort of processing takes place at a particular time. For such an application, the DOS system time is examined periodically to determine when to begin processing. If the Detect Idle Time check box is selected in the application's PIF, the application won't be able to check the system clock. As a result, the timer won't work and the desired processing will never take place.

As an example, consider an option that's available in the MultiMate word processor. When you set up a MultiMate document to print, you can use the "Delay Print Until Time Is" option to schedule printing for a later time. If the Detect Idle Time check box is selected in MultiMate's PIF, this scheduling feature will only work if you're actively working in another MultiMate document at the time the other document is set to print. If MultiMate is just waiting for keyboard input, though, Windows will swap the application to disk; MultiMate won't be able to check the system clock to determine whether it's time to print a queued document.

To prevent this problem, deselect the Detect Idle Time check box for a DOS application that operates on a timer. If such an application comes with its own Windows 3.1-compatible PIF (or if Windows 3.1 was able to identify a PIF for the application during setup), chances are that the Detect Idle Time check box will be deselected by default. If you never use the application's timer features, you might want to select the Detect Idle Time check box.

Some Example Scenarios

To help you better understand priority scheduling, we've put together some scenarios that illustrate how priority values work under different processing situations.

At this point, we've described the settings that you can use to modify the way Windows schedules tasks during multitasking in enhanced mode. But the results that you'll get by changing background, foreground, and exclusive settings can be extremely difficult to determine. Unless you run the same set of DOS applications in the same way, every day, the priority values that you assign can produce widely different results in different situations.

Scenario 1 Let's say you've changed the Background setting in your dBASE IV PIF file to 500 and you've selected the Background check box. The idea is to allocate relatively more CPU time to dBASE IV to support some time-consuming querying that you do just about every day. You also changed the Background setting in your WordPerfect 5.1 PIF to 500 and selected the Background check box. The increased Background setting is intended to support lengthy print merging that you do at the end of each day. You use the defaults for all other multitasking settings—including the minimum timeslice value of 20 milliseconds and an "On" setting for Detect Idle Time in each DOS application's PIF.

If you are running WordPerfect and dBASE IV at the same time, in the background, Windows will have a Foreground Priority value of 100. Combine this value with the two Background Priority values of 500 for your two DOS applications and you have a total of 1100. Thus, your Windows environment gets about 9 percent of the CPU time, while each DOS application gets about 45 percent of the CPU time. In this situation, if you try to work with large files in a Windows application, you'll notice a lot of sluggishness.

Now, if WordPerfect isn't busy with its print merge operation, but it is still running in the background, Windows will detect that WordPerfect is idle and will release its timeslice. That means that the total priority is 600 (100 for foreground Windows and 500 for background dBASE IV). In this situation, Windows gets about 17 percent of the CPU time and dBASE IV gets about 83 percent.

If both WordPerfect and dBASE IV are idle, Windows gets all of the CPU time for its own use. But suppose dBASE IV is busy performing a query while you switch to the MS-DOS Prompt. Now you've got the DOS prompt in the foreground (the default Foreground Priority value is 100), and dBASE (Background Priority, 500) and Windows (Background Priority, 50) in the background.

In this environment, the DOS Prompt will get about 15 percent of CPU resources, Windows will get about 8 percent, and dBASE will get about

77 percent. If you start two more DOS applications under this configuration, dBASE IV's CPU time will drop to 62 percent.

HOT TIP

You'll Rarely Need to Assign a Background or Foreground Priority Value of Greater than 1000

Remember, a priority value is relative to the default values (100 foreground, 50 background) that Windows assigns to your PIFs. An excessively large value will leave you with almost no CPU time for your Windows applications.

For instance, if you're running only one DOS application in the background, with a Background Priority of 5000, Windows will give your DOS application 98 percent of CPU time when the DOS operation is not idle. The minuscule 2 percent allocated to Windows applications can slow your processing in Windows to a painful crawl. Even on a fast 386/33 or 486 machine with plenty of extended memory, you'll notice an appreciable difference in the ability of Windows to redraw a window or to scroll through a document. You will rarely find it effective to use a priority value larger than 500 or 1000.

Scenario 2 For this scenario, consider that the same priority and scheduling settings we used in Scenario 1 are still in effect for dBASE IV and WordPerfect 5.1.

Now suppose you enter a dBASE query and then walk away from your desk to get a cup of coffee. WordPerfect is idle in the background, along with one other DOS application and, of course, Windows. From previous experience, you expect dBASE IV to have found all the records you've requested by the time you return to your desk. But dBASE is still searching its database files when you return! What happened?

You forgot to switch dBASE to the background. Since dBASE is running in the foreground, Windows uses its Foreground Priority value of 100 rather than the desired Background Priority value of 500. The result is that dBASE gets 40 percent of CPU resources instead of the 76 percent it would have been assigned if you had switched dBASE IV to the background.

Scenario 3 The priority values and other scheduling settings we specified in Scenarios 1 and 2 are still in effect.

In this case, though, WordPerfect is idle, and you're using dBASE IV to print a low-priority report. The controller at the home office phones you with an emergency request for some budget information. Immediately, you start Excel, load a large worksheet, and try to perform some calculations. Excel responds, but does so at something approximating the pace of a box turtle walking backwards. The controller is getting impatient. By the time you've figured out what's wrong, he's lost his temper.

The problem, of course, is that dBASE was hogging all of your CPU resources on a report you could have printed anytime, while Excel was limited to the relative puny timeslice allocated for the foreground Windows environment.

Summary The point of these scenarios is that actual scheduling varies widely depending on the number of DOS applications you have open, the Background and Foreground Priority values for each open application, which application is in the foreground, and the types of processing your applications are engaged in at any given moment.

If your processing environment remains basically static from one day to the next, and if you use DOS applications frequently, you might benefit by assigning higher priority values for those specific DOS applications. Your Windows applications will run slowly, but if this doesn't bother you, the additional speed you get out of your DOS applications might be worth the effort.

On the other hand, if you're like many Windows users, you rely mostly on Windows applications and use DOS applications only occasionally. If this is true, you're better off using the default priority values that Windows assigns in your PIFs. Then, if you occasionally use DOS applications and want to adjust multitasking settings for only the current work session, you can temporarily adjust settings using the Control menu for each DOS window. These settings will override any multitasking settings stored in the PIFs for your DOS applications. Chapter 14 explains how to temporarily adjust multitasking settings from within a DOS window.

HOT TIP

Create Separate PIFs for an Application That You Use in Different Ways

Suppose you use dBASE every day, but usually all you need to do is look up or modify a record or two, or you occasionally perform a query that takes a few seconds to complete. But once a week, you use dBASE to generate a set of sales reports that typically take one hour or more to generate and print.

In this case, you might benefit by running dBASE under a customized PIF for the CPU-intensive processing situation only. In other words, you can use your existing dBASE PIF as a template for creating a second PIF that you use only for your weekly report-writing needs. Here's an example of the steps to follow to create a custom PIF for this type of situation:

1. Start the PIF Editor (which is available from the Main group).
2. Click on File Open, then double-click on the PIF that you want to use (for instance, DBASE.PIF).
3. Make sure the Background check box is selected if you want to be able to run the application in the background.
4. Click on the Advanced button.
5. Increase the Background Priority and/or the Foreground Priority value(s) according to your needs, then click on OK.
6. Click on File Save As and assign a unique name to the PIF (such as **DBREPORT.PIF**), then click on OK and exit the PIF Editor dialog box.
7. In the appropriate program group, click on the icon for the application you are customizing (such as the dBASE IV icon).
8. Click on File Copy to display the Copy Program Item dialog box.
9. In the To Group pull-down list, select the group you want to copy the icon to, then click on OK to make a duplicate of the icon.
10. Click on the duplicated icon you've just created, then click on File Properties.
11. In the Command Line, change the PIF name to the one you've created (for instance, **DBREPORT.PIF**).
12. In the Description line, change the icon name to something descriptive, such as **Weekly dBASE Reports**, then click on OK to save the changes.

In this example, you can now use the standard dBASE icon to load dBASE IV using the default PIF settings. But when you want to generate reports, start dBASE IV using the new icon you've created. The modified PIF settings will then take effect.

Using Swap Files

If you've read Chapters 1 and 2 (or if you're already familiar with the way virtual memory works), you know that enhanced-mode Windows virtualizes some of your hard disk space so that the space can be used just like extended memory.

As we explained in Chapter 1, the Windows virtual memory manager (VMM) uses a demand-paging approach to divide hard disk space into fixed-sized blocks called pages. Each page is 4K in size, and can be mapped to memory addresses that are recognizable by your applications. A pageswap device, built into the WIN386.EXE file, is used to swap pages between your hard disk and RAM on an as-needed basis.

When you install Windows 3.1 on a 386 or 486 system, the setup program suggests that you let it create a permanent swap file for storing virtual memory pages. Windows locates the largest amount of *contiguous* hard disk space (adjoining tracks and sectors), and then uses this amount to suggest a size for the permanent swap file. If you're upgrading to Windows 3.1 and you already have a permanent swap file, Windows will try to use this space plus any other contiguous tracks it can find. So, you don't have to worry about deleting a permanent swap file before you upgrade to Windows 3.1.

If you have a 386 or 486 computer, and if you agreed to the setup program's suggestions when you installed Windows, your system already has a permanent swap file. If you didn't allow the setup program to create a permanent swap file, Windows will create a temporary swap file whenever you start Windows in enhanced mode. The temporary swap file is created in the root directory of your hard disk, and is named WIN386.SWP. Windows deletes WIN386.SWP each time you exit Windows, so that the hard disk space can be used at the DOS level.

A temporary swap file is much less efficient than a permanent swap file, because it chains together noncontiguous hard disk space. That means the read/write heads on your hard disk have to work harder to locate parts of a file, often jumping from one track to another track at a distant location. The net result is that access time is much slower for a temporary swap file than for a permanent swap file. Also, the extra work required by the mechanical arms of your hard disk translate into extra wear and tear and a shorter operating life span for your hard drive.

When you create a permanent swap file, Windows creates a file called SPART.PAR in your hard drive's root directory. SPART.PAR is actually a very small read-only file that contains description information about the size and location of the swap file itself, which is a hidden file named 386PART.PAR. The read-only and hidden attributes of these files are included to prevent you from deleting them.

In any event, you should consider creating a permanent swap file if you do not already have one, and even if you do have one, you should think about creating a new permanent swap file—using the guidelines we'll supply in the following sections.

Using a Disk Optimization Program to Maximize Virtual Memory

Before you create a permanent swap file, you should defragment your hard disk by using a disk optimization utility program. (A disk optimization utility is sometimes called a defragmenter or a disk compaction utility.) Disk optimization utilities are available with many tool packages, including Central Point's PC Tools, and Symantek's Norton Utilities and Norton Desktop for Windows.

Why optimize your disk? Over time, as DOS tries to find space for the various programs and files that you have stored on your hard disk, the free space gets fragmented into several noncontiguous locations. The result is a hard disk that, from the point of view of DOS, looks like a slice of Swiss cheese, with parts of files scattered in distant cubbyholes of your hard disk, as illustrated in Figure 4.6.

A disk optimization program shuffles segments of your hard disk's files in an attempt to position most of your programs and data in contiguous locations. In other words, the major goal of a disk optimization utility is to cram all of the files on your hard disk as close together as possible to make the most efficient use of the mechanical arms and read/write heads

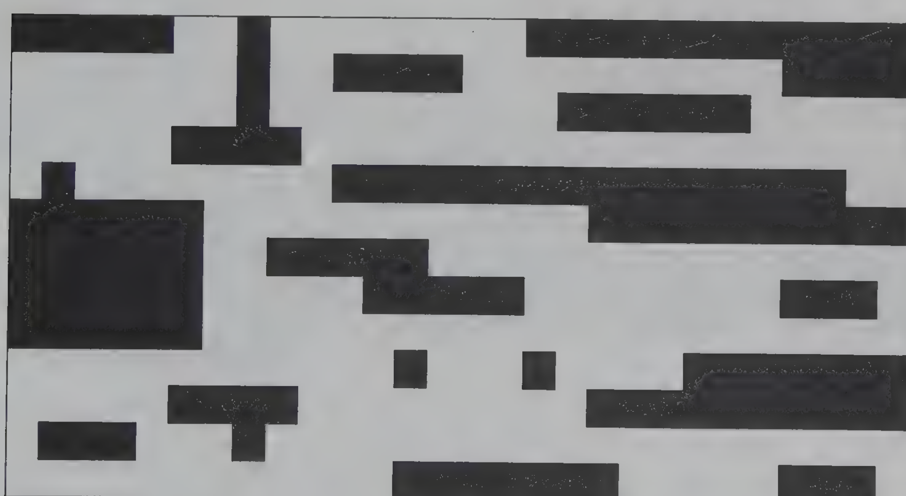


Figure 4.6 This conceptualized diagram of hard disk space shows how clusters of data can be fragmented across distant disk addresses.

of your disk drive. For this reason, even if you never create a permanent swap file, you should still defragment your hard disk periodically.

HOT TIP

Always Run the DOS CHKDSK Utility, Using the /F (Fix) Switch, before You Defragment Your Hard Disk

When you type CHKDSK /F at the DOS prompt, CHKDSK locates any *lost clusters* that might be lurking on your hard disk. A lost cluster is a portion of a file that DOS can't associate with a file name. Lost clusters sometimes result when your system crashes while an application is in the middle of some processing operation. Deleting lost clusters can increase the amount of free space on your hard disk.

When CHKDSK finishes examining your hard disk, it will offer to chain any lost clusters that it finds into files. In other words, DOS assigns file names for the clusters. That way, you can use a text editor to examine the lost data to see which file it originally belonged to. If you want to examine lost clusters, type **Y** when CHKDSK prompts you to convert the clusters to files. If you type **N** at this prompt, CHKDSK will go ahead and delete the clusters for you.

Warning: Never run CHKDSK /F while Windows is running. (When DOS 5 and Windows 3.1 both are running, you won't even be allowed to use CHKDSK with the /F switch.) Due to the way Windows manages swap file space, DOS will often think that two or more open Windows files are trying to occupy the same location on disk. CHKDSK responds to this "problem" by hacking your Windows files into smaller pieces and then moving the smaller files to separate disk locations. Needless to say, CHKDSK /F can create an absolute mess if you try to run it while Windows is running. This same problem will occur if you try to run CHKDSK in a DOS window (including the MS-DOS Prompt application). *Always exit Windows before you run CHKDSK /F.*

By compacting your hard disk files into contiguous tracks and sectors, it stands to reason that the remaining free (unoccupied) disk space will also be located mostly in a single contiguous block (see Figure 4.7). And that's the benefit of defragmenting your hard disk prior to creating a permanent swap file. Windows will ignore any fragmented space when it looks for free hard disk space to use in creating a permanent swap file. So, the maximum size of your permanent swap file can be no larger than the largest contiguous block of free space. Defragmenting your hard disk ensures that Windows will be able to create the largest possible permanent swap file.

ON DISK

Using DEFRAG

If you do not own a defragmentation utility, you can use the one supplied with the *Windows Insider Disks.* This easy-to-use utility, called DEFRAG, is stored in the DEFRAG directory.

Warning: Never run DEFRAG or any disk optimization utility while Windows is running.

HOT TIP

Delete Your Permanent Swap File before You Defragment Your Hard Disk

If you already have a permanent swap file, you should delete it before you defragment your hard disk. This step will allow your disk optimization program to use the swap file space in defragmenting your disk. Actually, you don't have to delete your permanent swap file each time you defragment your hard disk, but you should do so if you are defragmenting your hard disk specifically to increase the size of your permanent swap file.

Deleting a Permanent Swap File

Follow these steps to delete a permanent swap file prior to defragmenting your hard disk. After your hard disk has been defragmented, you can create a new (and often larger) permanent swap file.

Figure 4.7 This diagram of optimized hard disk space shows how files and available free space can be compacted into contiguous sections on the disk.

To create or delete a permanent swap file, you have to wade through a series of dialog boxes, starting with the 386 Enhanced dialog box accessed from the Control Panel. In the following steps, we'll explain how to delete your permanent swap file.

First, click on the 386 Enhanced icon from the Control Panel. When the 386 Enhanced dialog box appears, click on the Virtual Memory button to display the first Virtual Memory dialog box, shown in Figure 4.8. The figure shows that a permanent swap file is currently in effect. The file is 7,761K (7.7MB) in size.

Click on the Change button to display the second Virtual Memory dialog box, which is shown in Figure 4.9. We'll explain the virtual memory settings for this dialog box later. For now, we just want to show how to delete a permanent swap file. So, click inside the Type pull-down list box, then click on None to remove the Permanent swap file.

HOT TIP

Always Use the None Option if You Plan to Create a New Permanent Swap File Later

You could also select Temporary from the Type pull-down list box. But if you do this, you'll just have to delete the temporary swap file later before you re-create the permanent swap file. How come? If you try to create a permanent swap file when a temporary swap file is already present, Windows will incorrectly determine the amount of available hard disk space. The result will be a permanent swap file that is considerably smaller than the maximum possible swap file size. Always create a Permanent swap file when no swap file is in use (None is selected in the Type list box).

After you select None and click on OK, Windows will request that you reboot Windows to remove the permanent swap file. You *must* reboot Windows at this point for your changes to take effect. Click on Reboot Windows to continue.

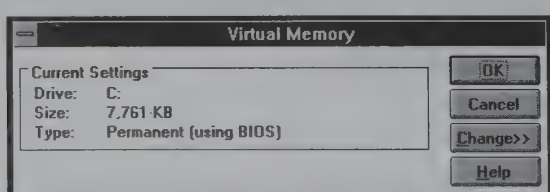


Figure 4.8 This Virtual Memory dialog box displays information about the type of swap file currently in use and its size.

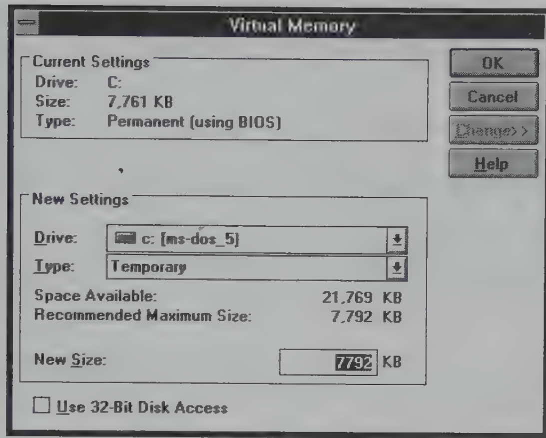


Figure 4.9 Use this Virtual Memory dialog box to change the type and size of your swap file.

HOT TIP

Without a Swap File, Your Startup Applications Might Not Load

When Windows reboots, you might get error messages if your extended memory is too insufficient to load all of the applications in your Startup group. Remember, at this point you have no virtual memory, so Windows is limited to physical memory for loading applications. Normally, these messages will do no harm; you just won't be able to load all of your startup applications. However, if you do have problems restarting Windows successfully, try either of the following, temporary workarounds:

1. Place a semicolon (;) in front of the line in your AUTOEXEC.BAT file that loads SMARTDrive, and then reboot the computer and load Windows. With SMARTDrive not installed, you can recover .5MB or more of extended memory. In many cases, this additional memory will be enough to load your startup applications.
2. Rename STARTUP.GRP (in your WINDOWS directory) to STARTUP.BAK. This will prevent Windows from loading the applications in your Startup group, therefore reducing the amount of extended memory required to start Windows. You can rename STARTUP.BAK to STARTUP.GRP after you've created your permanent swap file.

After you've rebooted Windows, return to the second Virtual Memory dialog box to make sure that no swap file is in use. If None is the swap file type, you're in good shape. Exit Windows and run your disk optimization routine.

Creating a New Permanent Swap File

After your hard disk has been defragmented, you are ready to create a new permanent swap file. This section explains how to create an effective permanent swap file.

If you've completed your disk optimization routine, restart Windows and return to the second Virtual Memory dialog box. Review the following three information lines in the New Settings box:

Space Available This is the total amount of free space on your hard disk.

Maximum Size This is the total amount of contiguous space on your hard disk. After you've defragmented your hard disk, the Maximum Size should be very close to the Space Available size.

Recommended Size This setting is deceptive. Although Windows tells you that this is just the recommended size to use for your new permanent swap file, the recommended size is actually the *maximum* permanent swap file size. Windows will allow you to create a permanent swap file up to the value specified in Maximum Size, but Windows will never use more virtual memory than the value in Recommended Size.

Because of this limitation, it usually doesn't make sense for you to create a larger swap file than the Recommended Size value; you would just be reserving hard disk space that cannot be used by Windows or any other application. However, there are some situations where you'll want to create a smaller permanent swap file than the Recommended Size.

Use the value in the New Size box if you want to create the largest possible permanent swap file. Enter a smaller value if you want to keep additional space available for storing new documents and programs on your hard disk.

Click OK after you have specified your permanent swap file size. Windows will request that you reboot, so that the new swap file can take effect. Click on Reboot Windows to complete the creation of the swap file.

Using SYSTEM.INI to Increase Your Maximum Swap File Size

By default, Windows rounds the total amount of physical memory on your system to the nearest 4MB, then multiplies this value by 4

to determine the total amount of memory (physical + virtual) Windows will create and use.

If your system has 5MB of extended memory, Windows adds your extended memory and conventional memory (6MB total) and then rounds the value up to the nearest 4MB value (in this case, 8MB). Windows now multiplies the new value by 4 to determine the total amount of memory Windows will manage—32MB in our example. Since 32 minus 6 equals 26, the maximum swap file size that Windows will recognize is 26MB.

You can increase the multiplier that Windows uses to calculate your total memory space, and consequently increase the maximum size for your permanent swap file. Again, the default multiplier is 4, and is specified in the PageOverCommit= line in the [386 Enh] section of SYSTEM.INI.

You can specify a value from 1 to 20 in the PageOverCommit= line. In our example, if we specify PageOverCommit=6, Windows will multiply 8MB by 6 to calculate a maximum memory limit of 48MB. This increases the maximum possible swap file size to 42MB (48 minus the 6 megabytes of physical memory).

You must reboot your system for the new PageOverCommit= setting to take effect. Also, keep in mind that, even though you can increase the size of your swap file, doing so will increase the paging required to support virtual memory. The increased paging can significantly slow down your system's performance. If you are in desperate need of increased memory, you should consider installing additional extended memory rather than increasing the PageOverCommit= value.

Note: Windows also determines your maximum swap file size based on the total amount of contiguous free space available on your hard disk. Increasing the PageOverCommit= value might have no effect if Windows determines that you do not have enough contiguous free space to support a larger permanent swap file.

HOT TIP

You Can Estimate Your Own "Recommended Size" by Creating a "Worst Case" Memory Situation

1. Create a "worst case" memory situation in Windows by loading as many DOS applications and Windows applications as you think you'll realistically ever need.
2. If you sometimes load multiple, large files with specific applications,

open several large files now to consume as much memory as you think you'll ever need for documents in any given Windows session.

3. Click on About Program Manager from the Program Manager Help menu. Note the amount of free memory. If you've still got several megabytes (four or five megabytes more) available after creating your worst-case memory situation, you can consider reducing the size of your permanent swap file.

You might not want to reduce your permanent swap file size right away. Just keep in mind that you have several megabytes of hard disk space available if you need them. Later, if your hard disk space becomes limited and you need to install a new application, or if you just want to make more space available for documents, you can probably reduce your swap file size safely without risking an out-of-memory condition. (Also consider compressing infrequently used files to recover additional hard disk space.)

Using 32-Bit Disk Access

If you've ever used Windows 3.0 or 3.1 to run several DOS applications at a time, you might have received an annoying "out of memory" message from Windows, even though you still have plenty of unused virtual memory. The problem involves some admittedly complex limitations arising from the basically unfriendly alliance between Windows and DOS applications. This same basic problem can make Windows hellishly slow when it needs to access a temporary or permanent swap file in enhanced mode.

Although the problem is a complex one and isn't all that easy to explain, the solution in Windows 3.1 is extremely simple. The key is to use a little-known but powerful feature of Windows 3.1 called *32-bit disk access*. Microsoft also uses the term *FastDisk* to refer to the collection of drivers that enable 32-bit disk access.

FastDisk is so simple to use that you can enable all of its features with one click of the mouse. Unfortunately, there are a few hardware situations where FastDisk will not operate at all or will operate unpredictably. For these reasons, we think it's best to give you some background into the technical realm of disk access with Windows. Then we'll explain the benefits of 32-bit disk access and we'll show you how to enable and disable this feature. In this way, if you try to use 32-bit disk access and have problems, you'll know why the problems exist and you'll be able to decide how best to correct them.

The Low-Level Life-Style of Hard Disks

We've made it a point throughout this book to stress that Windows performs many of the services normally handled by DOS—with some exceptions. Traditionally, disk I/O (input/output) is one of the services that Windows has left to the realm of DOS and the BIOS. The basic reason for providing these disk I/O services through DOS is the inability of Windows to work directly with different types of hard disk controller cards.

Applications communicate with disk devices by requesting "interrupt" services from DOS or from your ROM BIOS. Each physical disk in your system has a separate controller card that is responsible for operating the mechanical components of the disk drive. For instance, a hard disk controller activates the read/write heads on your hard disk whenever an application needs part of a program or data file stored on the disk.

Actually, the hard disk controller rarely communicates directly with your programs. Here's what happens instead. Your application makes a request to DOS for data or program instructions that aren't currently in memory. This request is called a *software interrupt.* The term *interrupt* is a little misleading, because your application doesn't really interrupt DOS to request a service, it just interrupts (pauses) its own processing long enough to make the request to DOS.

Each software interrupt is identified by a hexadecimal value. This value points to a location within conventional memory where DOS stores the code required to perform a particular I/O service routine. In the low-level assembly language that your CPU understands, a software interrupt is specified by an INT instruction.

For instance, if your application contains an INT 21H instruction, this "calls" DOS, telling DOS to use service 21 to perform some sort of low-level I/O processing. To complicate matters, a DOS service might need to make an additional call to your system's BIOS (the basic I/O system built into your computer's read-only memory).

Why is this so? Your system's BIOS normally contains the low-level instructions required to communicate with the various hardware devices attached to your computer—including your hard disk controller card. Your hard disk was probably installed by the same company that installed the BIOS for your computer, so it was possible to ensure that the BIOS could communicate successfully with your particular hard disk controller. That isn't possible with DOS, which you purchase independently of your hardware and cannot know in advance what type of hard disk controller card your system has.

Therefore, DOS often calls a separate BIOS service routine to handle the communications between the operating system and a particular hardware device, such as your hard disk.

Here's the bottom line: to read a file from your hard disk, your application might have to call DOS to request a particular I/O service. DOS, in turn, might have to make a separate interrupt call to a BIOS service routine, which can communicate directly with your hard disk controller. The hard disk controller then passes the address information back to the ROM BIOS, which in turn tells DOS where and how to find the information. DOS then reads the data and passes it to the application.

As you can see, there's a pronounced domino effect at work here. An application relies on DOS, which relies on the BIOS, which relies on the hard disk controller, and so on. To make matters worse, Windows has to switch between real mode (virtual 86 mode) and protected mode whenever a DOS application (virtual machine) requests data from your hard disk. This mode switching adds to the time required to access hard disk data.

DOS Can Handle Only One Interrupt Request at a Time

Although Windows can create many virtual DOS machines, only one virtual machine (including the Windows virtual machine) can request DOS or BIOS services at a time. This last sentence is so important, you should memorize it. In fact, this limitation is one of the most important keys to unlocking the mystery of enhanced-mode Windows.

Remember that enhanced-mode Windows uses a demand-paging system to swap data to and from your hard disk—stored within a temporary or permanent swap file—so that applications can run even when your system's physical memory is full. But consider the problems that this paging system can cause for a DOS application.

Each DOS application typically sets up a certain number of *buffers* that cache into memory the most recent data read from or written to disk. Essentially, a buffer is a mini-cache that DOS can use to retrieve recently used data from memory rather than from the hard disk. Thus, data buffers are designed to speed up disk access.

You set up the total number of data buffers that applications can use by including a BUFFERS= line in your CONFIG.SYS file. A buffer is normally 512 bytes in size—the same size as a sector on your hard disk. Extended-memory cache programs like SMARTDrive have largely made the BUFFERS statement unnecessary; however, DOS applications will still use these buffers if they are available.

We needed to explain data buffering only because it is this process that wreaks havoc on the demand-paging features of enhanced-mode Windows. Suppose Windows is managing a DOS application and swaps the application's data buffer to disk.

If the application requests that DOS move data from disk into the data buffer, DOS will call the BIOS to access the data. But the buffer isn't available because Windows has paged it to the swap file.

At this point, Windows recognizes that the controller is requesting buffer space that is currently paged to the swap file. Normally, Windows' demand-paging system can respond to an application's request for paged data by calling a DOS interrupt to page the data from the hard disk back into the application's data buffer. But in this scenario, that's not possible. DOS is already busy trying to read from the hard disk, and so is the ROM BIOS. So, when Windows requests DOS services, DOS is unable to respond. Windows can't call DOS, and DOS can't find the data it needs. The result? Your system hangs.

With Windows 3.0, Microsoft prevented this problem by not allowing an active DOS application to be paged to a swap file. If an application's data buffer is always in RAM, Windows and DOS will never get their signals crossed.

Under the Windows 3.0 environment, then, all active DOS applications had to be stored in physical RAM—conventional or extended memory. (An active application is any application that is running in the foreground, or that the PIF allows to run while it's in the background. An application that can't run in the background can be safely swapped to disk, since it won't try to request any DOS services.)

Under Windows 3.0, if you only had two or three megabytes of extended memory, Windows could quickly run out of storage space for active DOS applications, especially if SMARTDrive was taking up valuable extended memory space. So, Windows might have displayed an "out of memory" message when you tried to start a DOS application when your conventional and extended memory was full. This problem would occur even when you were running in enhanced mode, because Windows 3.0 won't allow active DOS applications to be paged to a swap file.

The Benefits of 32-Bit Disk Access

This same problem will occur in Windows 3.1 unless you turn on the 32-bit disk access feature. With 32-bit disk access turned on,

Windows bypasses DOS and handles disk I/O on its own. But your ability to use this feature depends on the capabilities of your hard disk controller.

The 32-bit disk access capability, or FastDisk for short, is a set of drivers that takes over the disk I/O services normally managed by DOS and by the ROM BIOS. These drivers allow Windows to remain in protected mode, swapping data to and from the hard disk 32 bits at a time, without being limited to the 16-bit disk access required in real (virtual 86) mode.

In other words, with the FastDisk capability enabled, Windows bypasses DOS and the ROM BIOS whenever a DOS application requests disk I/O service. Instead, the FastDisk drivers communicate directly with the hard disk controller, transferring 32 bits at a time between RAM and the hard disk, a transfer rate that is permissible only in enhanced mode.

Since FastDisk doesn't have to request DOS services, Windows can safely page DOS applications to a swap file. If an application requests data from a buffer while the buffer's data has been paged to disk, Windows can retrieve the data on its own—without having to make a call to DOS. There's no chance for DOS and Windows to pass each other coming and going.

This means you can open and keep active at one time as many DOS applications as will fit in physical *and* virtual memory (provided your system is running in enhanced mode, of course). Your DOS applications are no longer limited to physical memory. Also, data transfer between RAM and your hard disk is faster because Windows uses its own drivers to transfer data 32 bits at a time, rather than switching to real mode to use the 16-bit transfer required for DOS routines.

FastDisk Is Disabled by Default

Microsoft made FastDisk possible by writing a set of drivers that are compatible with a standard introduced by Western Digital for its 1003 series of disk controller cards. Most—but not all—hard disk controller cards available for PCs conform to this standard. Exceptions are disk controller cards that conform to the ESDI (Enhanced Small Device Interface) and SCSI (Small Computer System Interface) standards.

When you install Windows 3.1, the setup program tests your hard disk controller to see if it is compatible with the Western Digital 1003 (WD 1003 chip) standard. If so, Windows writes the following lines to the [386 enh] section of your SYSTEM.INI file:

```
32BitDiskAccess=off
device=*int13
device=*wdctrl
```

If these lines appear in your SYSTEM.INI file, Windows will make the Use 32-Bit Disk Access check box available within the Virtual Memory dialog box (available from the Control Panel). You use this check box to select the FastDisk feature. If Windows does not detect a WD 1003-compatible hard disk controller, this check box will not appear in the Virtual Memory dialog box.

Even if Windows detects the presence of a WD 1003 controller card, it will not actually turn on the FastDisk feature. Microsoft erred on the side of caution in this situation because it identified too many situations where 32-bit disk access might fail.

Specifically, the FastDisk device drivers sometimes recognize a hard disk controller as being WD 1003-compatible when in fact it is not. Also, most laptop and notebook computers use a battery-conservation technique designed to power down the hard disk when it is idle for a specific length of time. With this approach, the BIOS can power up the hard disk whenever an application makes a call to the BIOS for disk I/O services. Of course, since FastDisk bypasses the BIOS, Windows might try to access data from a dormant hard disk. As you might imagine, this situation is a disaster waiting to happen.

Because there are these few situations where FastDisk will not operate correctly, Windows elected to disable FastDisk by default to avoid complaints from users who are unsure how to disable the feature. But for more than 90 percent of the hard disks in use on 386 and 486 computers, FastDisk will work fine. *You should enable 32-bit disk access if you run DOS applications in enhanced mode.*

Enabling FastDisk

You can enable FastDisk easily by selecting a single check box.

To enable FastDisk (32-bit disk access), double-click on the 386 Enhanced icon from the Control Panel. Next, click on the Virtual Memory button to display the first Virtual Memory dialog box. Then, click on the Change button to display the second Virtual Memory dialog box.

Warning: You should back up your hard disk before you enable 32-bit disk access, in case Windows corrupts some of your hard disk files.

If Windows has detected a WD 1003-compatible hard disk controller on your system during setup, you will see the Use 32-Bit Disk Access button in the bottom-left corner of the dialog box. If this dialog box does not appear, you are out of luck; 32-bit disk access is not currently available for your system.

If you see the Use 32-Bit Disk Access check box, click on it to activate the FastDisk feature, then click on OK to exit the Virtual Memory dialog box. Windows will first display a standard dialog box asking if you are sure that you want to change virtual memory settings. Click Yes to continue.

At this point, Windows will respond with all the alarm of Chicken Little clucking about the falling sky. In other words, you'll see the dire warning shown in Figure 4.10.

The message in Figure 4.10 is Microsoft's disclaimer; it's essentially a "Don't blame us if this doesn't work" message. Don't worry. If FastDisk really doesn't work on your system, you can disable it easily. Click on the Yes button to bypass this dialog box. You will need to reboot your system for 32-bit disk access to take effect. Make sure you have closed any open DOS applications before you try to reboot your system, since Windows normally will refuse to reboot until all DOS applications have been closed.

HOT TIP

What if 32-Bit Disk Access Doesn't Work?

If you enable 32-bit disk access and your system hangs, reboot your computer. Next, start Windows by typing **WIN /D:F** at the DOS prompt. (The **/D** switch tells Windows you want to disable something, and the F parameter specifies that you want to disable 32-bit disk access for the current session.) Then, display the Virtual Memory dialog box and deselect the Use 32-Bit Disk Access check box.

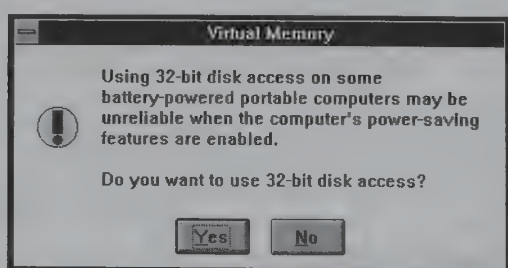


Figure 4.10 Windows displays this warning message when you try to enable 32-bit disk access. You need to heed this warning only if you are using a laptop or a notebook computer.

HOT TIP

If You Purchase a WD 1003 Controller after You've Installed Windows 3.1

If you add a WD 1003 (or compatible) controller to your system after you've installed Windows, you can easily enable 32-bit disk access by adding the following lines to the [386 Enhanced] section of your SYSTEM.INI file:

```
32BitDiskAccess=On
device=*int13
device=*wdctrl
```

The FastDisk drivers are built into WIN386.EXE, so you don't need to install additional drivers to enable 32-bit disk access. If the required lines appear in SYSTEM.INI, FastDisk will be enabled. However, you must reboot your system so that the new settings can take effect.

Warning: It's safest not to use 32-bit disk access if you are running Windows on a laptop or notebook computer. As we've already mentioned, the battery-saving features of most portable computers can cause 32-bit disk access to fail.

Part Overview

PART 2

Networked Windows

Chapters 5 through 7 will help you get your network and Windows to cooperate. We'll explain the factors involved in choosing how to install Windows and we'll show you how your network components relate to Windows 3.1.

In Chapter 5, you'll learn how to install Windows with your server; we'll provide tips and other helpful information for setting up diskless clients as well as for ensuring that PCs running Windows locally can interact reliably with the network. In Chapter 6, we'll explain factors involved in troubleshooting and tweaking Windows for optimal network operations. Finally, Chapter 7 we'll present some interesting and productive network-oriented shareware and freeware programs that add functionality and capabilities to Windows.

CHAPTER

5

Installing Windows on a Network

When Microsoft released version 3.0 of Windows, their sales and marketing people touted the ability of Windows to run peaceably alongside LAN (local area network) software. These claims often left network administrators scratching their heads in frustration or disbelief. True, Windows 3.0 supported the most widely used PC LANs, including the overwhelming favorite, Novell's NetWare. But Windows was difficult to install and often would not run on workstations unless you made significant hardware changes and/or added special drivers. (If you've had experience installing Windows 3.0 on a network, you're probably nodding your head right now.)

We've encountered more than a few system administrators who grumble whenever we merely mentioned the words "networking" and "Windows" in the same sentence. But users have been mercilessly insistent in their requests to run Windows on their networks, and they've forced a lot of administrators to travel the networked Windows learning curve.

Partly in response to the difficulties involved in installing and running Windows on a network, a number of shareware and third-party utility programs have become available to help you control your network from within Windows. And, of course, Windows 3.1 provides many improved network features over its 3.0 predecessor. Networking under Windows is no longer a nightmare for administrators, but it may still be the source of a few unpleasant dreams.

Specifically, network software and Windows no longer act like rivals under 3.1, but they don't always act like teammates, either. For instance, the options for installing and configuring Windows on your network are almost as numerous as the different network configurations available for LANs. And even with all the improvements and available utilities for installing and managing Windows on a network, there are still a number of network tasks that can still be accomplished only from the DOS prompt or server console.

This chapter details techniques for installing Windows with your network and for configuring the network components of Windows. You'll learn, with a little up-front work on your part, how to customize the installation process, thus controlling and limiting the installation of Windows options for individuals or groups on your network. We'll also pass along some helpful hints on configuring Windows options to work with your network.

Since this chapter is dedicated to helping you get Windows and your network working together, we've assumed that you'll be setting up more than a few network-connected PCs for Windows 3.1. Generally, the more PCs that you need to set up, the more important it will be to use the most efficient installation technique.

The Installation Process

Microsoft allows you to use two basic techniques for loading Windows onto your network PCs. You can use the installation diskettes to load Windows to the hard drive of each workstation (client PC), a process that will take you up to 30 minutes per system. Or, you can install Windows onto your server first—still a 30-minute task—and then elect to use one of two alternatives to install Windows onto all of the connected workstations; you can either copy the full version to each local hard disk, or you can have each workstation share the server copy of Windows, or you can use a combination of both. If you use a shared version of Windows, you will still need to copy several user files to each workstation's hard disk—unless the server is linked to diskless workstations.

Regardless of whether you copy all or just the minimum set of user files to each workstation, you can save yourself from 15 to 25 minutes of setup time per workstation by taking advantage of a few tricks and tips.

HOT TIP

Read NETWORKS.WRI First

Microsoft elected to place the information for installing Windows on a network in an on-line file called NETWORKS.WRI. (You won't find much help in the Windows 3.1 *User's Guide.*) Read this file before you install Windows on a network.

You can locate this uncompressed file on the installation disks and read it in Write or any ASCII editor. Also, Windows automatically copies NETWORKS.WRI into the WINDOWS directory during installation (regardless of whether it is a stand-alone or network installation), unless you specifically tell Setup not to copy this file.

NETWORKS.WRI contains most of the general information you'll need to install and run Windows 3.1 on a network and includes specific setup information for each of the Windows-supported networks. *Do not consider NETWORKS.WRI to be optional reading. The information you'll find here is essential to a successful network setup of Windows 3.1.*

Save Time by Installing on the Server First

By now, you've probably installed at least one stand-alone version of Windows 3.1, whether to verify program compatibility or to test new features. So, you already know that installing Windows from the installation diskettes isn't too terribly painful. But do you really want to repeat this process for, say, 40 or 50 workstations or more? It's a depressing thought, isn't it? It makes more sense to let your server do the installation work for you.

The easiest way to install Windows on multiple PCs is to first install to the server, even if you're not going to run Windows from the server. When you run Setup from the installation diskettes, Windows doesn't just copy files from the floppies to the indicated directory. Since most of the diskette files are compressed, Windows has to expand the files before they can be copied to a hard disk. The expansion process takes up most of the overall installation time. So, by installing Windows to a drive on the server, you only have to suffer through the expansion process one time. For each workstation, then, you can instruct Setup to copy the already uncompressed files to the appropriate directory.

Use the /A switch with Setup to install a version of Windows to your server so that this installation can in turn be used to install Windows on workstations. Log on to the network using an account that has create-directory privileges, insert the first Windows disk, change to the proper drive (either A or B), then enter the command:

```
SETUP /A
```

Windows will then copy and expand all of the files from the floppies into a directory on the server.

After the installation is complete, verify that all of the files in this directory are set to "read only"—that is, the read-only attribute is turned on. Normally, the Windows Setup program does this, but it's a good idea to check. By write-protecting the server files, you—or, more importantly, users—can't delete or change files needed for future installations.

This directory is now the starting point for all future installations of Windows onto networked PCs. It is also the directory from which shared versions of Windows will load most of their files.

Installing from the Network to a Local Hard Drive

If a significant number of network users want to run Windows at its full potential (opening several applications at a time, custom-

izing the Windows environment, multitasking, and so on), and if users' PCs have enough hard disk space, you should consider loading Windows to the local hard drives.

With most of the Windows files loaded locally, users can get optimal speed and performance from Windows. In this installation environment, users still have access to network services—including network printers and queues, multiple servers, and available network drivers—while enjoying the ability to load Windows and many applications directly from the hard disk.

There is an important drawback to loading a full-blown version of Windows on each local drive, but the drawback will mainly affect the system administrator, not users. If you're the system administrator, you'll have to update each individual hard disk whenever you update a new Windows version, or add drivers and options. In a network with even 20 workstations, hassling with local workstations can be time-consuming to the point of profound annoyance.

But if your chief concern is to provide maximum service to network users, a local Windows installation is not only the best approach, it's the only approach. Windows runs much faster locally than from a shared server version; users' productivity can suffer under the shared approach.

After you've expanded and installed all the Windows files on the server, you'll find it easy—if not downright boring—to install Windows on workstations. You'll need to log in on each workstation that needs 3.1 loaded, then change to the network directory where Windows is installed (N:\WINDOWS, for example), and run setup just as you would do if you were installing from the diskettes. In other words, just type:

```
SETUP
```

You don't need to specify a target directory, because Windows will prompt you for one. When Setup needs files normally found on an installation diskette, it won't prompt you to "Insert the diskette labeled Disk X into drive A," because Windows is able to find all files on the current network directory. And because Setup is copying uncompressed files to the local hard disk, the file expansion process is bypassed. Instead of waiting 30 minutes for Windows to install to each local hard disk, you'll only have to wait 15 minutes or so.

Sharing Windows on the Network

Running a shared version of Windows does have its advantages. In general, one of the major advantages of a network is the ability

for all network users to access a single copy of a program stored on the server—making it unnecessary to install and maintain separate copies of the program for each workstation. This same fact is, for the most part, true for Windows.

A shared Windows version makes sense if your users don't rely on a lot of I/O-intensive processing on a regular basis. For instance, if most users just keep a few applications open at a time and don't do any heavy multitasking, they probably won't mind dealing with the transfer rates of the file server.

Keep in mind, though, that Windows still requires you to install a subset of files on each workstation for desktop management. (With diskless workstations, these files can be copied into a personal directory on the network for each user.) So, with a shared version of Windows, users can still customize and arrange their desktops as they wish. All of the files associated with the desktop configuration, Control Panel settings, and configuration and .INI files for locally installed applications are loaded into the local directory, or at least into a personal directory on the network. But the majority of the Windows files and applications can reside in the shared directory on the network. In many (but not all) update or maintenance situations, most of the updates and changes can be made solely to the server; the local configuration files for each user often can remain unaffected.

The procedure you use to install a shared version of windows is identical to the procedure required to install to a local hard disk—with the addition of one switch. Specifically, you have to use the /N (for Network) switch with the SETUP command, like this:

```
SETUP /N
```

The /N switch tells Setup to copy only those files needed to control local desktop settings—such as .GRP (program group information) and .INI (initialization information) files—to the local hard disk or personal network directory.

In Enhanced Mode, Swap Pages to a Local Hard Drive

If workstations have a 386 or 486 processor, users will probably want to run Windows in enhanced mode. In enhanced-mode Windows, of course, each user can decide whether to use a

temporary swap file to support demand paging, or to use a permanent swap file. It's possible to page to the network, but it is much faster and less problematic to page to a local hard drive.

What's so difficult about swapping pages to the network? Regardless of whether you (the user) are trying to create a temporary or permanent swap file on the network, certain network conditions must exist for you to do this properly. First, you must have both write *and* create privileges for the drive where you want to store the permanent or temporary swap file. (And, of course, the directory cannot have a read-only attribute.) To compound the problem, a permanent swap file *must be placed in the root directory of the drive you specify.* Users often don't have write privileges to this directory. Therefore, if you want to set up a permanent swap file on a network drive, you might have to get help from the system administrator.

You can make life much easier by paging to your local hard drive. As we explained in Chapter 4, you use the Control Panel to access the Virtual Memory dialog box, which in turn allows you to create a permanent or temporary swap file. If your hard drive is too full to support a permanent or temporary swap file, consider using a program—like Stacker or SuperStor—that can compress unused applications and data files to a separate logical drive (a separate partition on your hard disk). By compressing files when they are not being used, you increase the available space on your hard drive for creating either a permanent or temporary swap file.

In any event, paging to a network drive is much slower than paging to a local hard disk—especially when network traffic is high. If you must page to the network, do so in a permanent swap file, not a temporary one. Because a permanent swap file is made up of contiguous tracks and sector space, disk access is much faster than with a temporary swap file. Again, Chapter 4 explains how to set the drive location and size for a permanent swap file.

Installing Windows on a Diskless Workstation

Setting up Windows on a diskless workstation is similar to setting up a shared copy on the network—with a few exceptions. Since there is no local hard drive, Windows must create the swap file on a network drive.

You must also set the TEMP variable in the AUTOEXEC.BAT file or login script to be a RAM drive or network drive. This is necessary because Windows uses the directory specified by the TEMP statement to spool print jobs. To add a TEMP directory to a network drive, you might want to add the following command to the login script:

```
SET TEMP=N:\USERDIR\TEMP
```

Each user must have his or her own TEMP directory and cannot share that directory.

Creating Customized Setups

When you install Windows, you have the opportunity to select either a Custom or Express setup. Although a Custom setup gives you more options in determining which files get copied to the server or to each workstation, this approach can be overly-time consuming; for each installation, you have to determine which files you want to install and which files you want Windows to ignore during the setup.

Although an Express setup requires you to do less hands-on work, it's still time-consuming, because Windows has to copy hundreds of files for each installation. Also, a full-blown Express setup can require as much as 16MB of hard disk space per installation. Fortunately, there's a better way. In fact, there are two better ways, which we'll discuss next.

Modifying SETUP.INF to Create a Reusable Custom Installation

For most business environments, it isn't necessary to copy all files to each workstation, or even to the server. For instance, do you really need to make Solitaire available to every network user? (Of course, your users will probably say, "Yes, yes, yes!") In fact, you can probably identify several megabytes worth of files that your users will never or rarely need.

You can prepare a customized installation for each workstation by changing the defaults that Setup uses to install Windows. This is possible (and easy to do) because the file SETUP.INF contains all the information used by Setup to configure Windows. SETUP.INF is not a compressed file on the installation disks, so it can be modified before or after the initial installation to the network.

In other words, if you want to prevent individual programs or groups of programs from being installed, you simply need to comment out certain lines of the SETUP.INF file in the shared Windows directory. In this way, you can do an express setup for each workstation, and be assured that Windows will copy only the programs that you've specified in SETUP.INF. You only need to make the changes to the SETUP.INF file once; the changes will affect each Windows installation from that time forward.

For instance, if you don't want Control Panel or the Games group to be loaded onto each workstation, you can edit SETUP.INF and comment out those lines that copy CONTROL.EXE, CONTROL.HLP, and GAMES.GRP to the local directory.

SETUP.INF is a large file—too large to be loaded into Notepad. But you can use Write to edit the file; just be sure that you save the changes as an ASCII file, not as a Write-formatted document.

Warning: Remember to make a backup of the SETUP.INF file before you edit it. That way, if you make a mistake or want to switch back to the generic setup, all you have to do is copy the backup version of the file back to the shared Windows directory.

Use the "Hands-off" Approach to Install Several Different Custom Versions of Windows

One of the most important improvements of Windows 3.1 over 3.0 is the level of network support that the newer version provides. One of the more useful of these improvements is the addition of the /H switch to Setup. The /H (for Hands-off) switch allows you, as a system administrator, to control the installation for individuals, groups, departments, or entire organizations.

Microsoft provides a shell file, SETUP.SHH, that you can copy and modify to create customized user-specific or group-specific setup instructions. You can use SETUP.SHH to specify the network type, printer and queue information, and directory defaults, and to select files to be ignored by Setup, such as the GAMES group or certain wallpaper files.

Before we describe the details of SETUP.SHH, we'll provide a brief example to illustrate how it works. To use SETUP.SHH, you should first make a copy of this file, assigning a different name to the copy, but keeping the .SHH extension. (The .SHH extension is mandatory.) If you don't make a separate copy of SETUP.SHH before you edit it, you'll have to remember to save the file under a different name; otherwise, you'll overwrite the default file. For instance, to set up all the users in your company's accounts-payable group with identical configurations, you could copy SETUP.SHH to SETUPAP.SHH.

Then, using Notepad, the MS-DOS Editor, or another ASCII editor, edit the file SETUPAP.SHH and add or delete comments as needed. Figure 5.1 illustrates what SETUPAP.SHH would look like if we remove the Games group from the installation process.

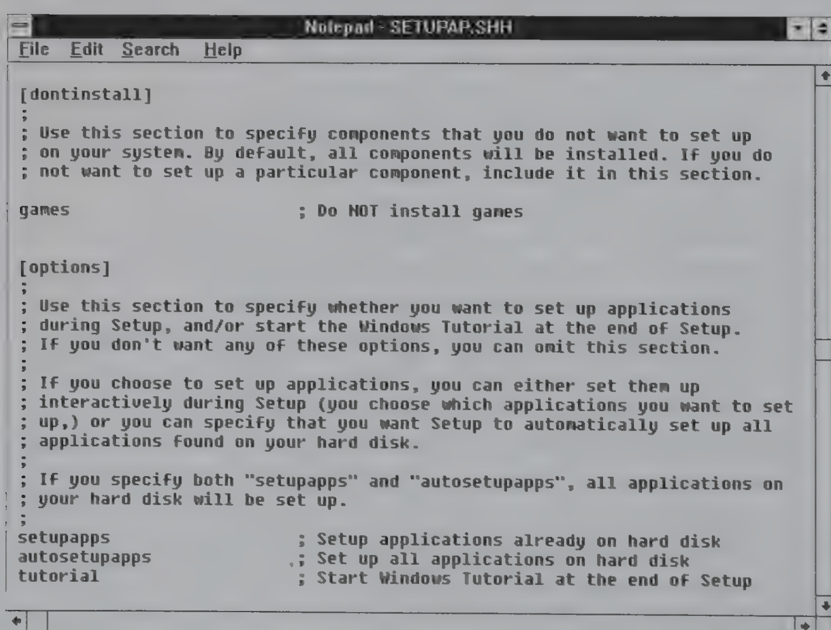
After you've edited and saved the file, you can return to the shared Windows network directory. Now, when you run Setup, you just add the /H switch and supply the SHH file that you want to use:

```
SETUP /H:SETUPAP.SHH
```

Setup will now use the information found in SETUPAP.SHH to install Windows. The result will be a complete installation of Windows—except for the Games group and the games programs, which will not be copied to the target directory.

The SETUP.SHH file has four main sections:

- System information
- User information
- Setup options
- End install options




```
[dontinstall]
;
; Use this section to specify components that you do not want to set up
; on your system. By default, all components will be installed. If you do
; not want to set up a particular component, include it in this section.

games                          ; Do NOT install games


[options]
;
; Use this section to specify whether you want to set up applications
; during Setup, and/or start the Windows Tutorial at the end of Setup.
; If you don't want any of these options, you can omit this section.
;
; If you choose to set up applications, you can either set them up
; interactively during Setup (you choose which applications you want to set
; up,) or you can specify that you want Setup to automatically set up all
; applications found on your hard disk.
;
; If you specify both "setupapps" and "autosetupapps", all applications on
; your hard disk will be set up.
;
setupapps                      ; Setup applications already on hard disk
autosetupapps                  ; Set up all applications on hard disk
tutorial                       ; Start Windows Tutorial at the end of Setup
```

Figure 5.1 In this SETUPAP.SHH file, the Games group has been added to the [dontinstall] section.

The system information section includes an option to override the hardware configuration that Setup detects on each workstation. This is especially useful if you are setting up diskless workstations or multiple workstations from the server console.

The user information section includes the user name and company name that Setup normally prompts for during the installation process. It also includes the directory where Windows will be installed.

The Setup section includes an area for components that you don't want to install (as in our example above). If you add a Windows component to this section, Setup won't install that component. This section also controls how Setup detects applications and whether the tutorial will be installed. Other options in this section include the ability to specify the printers and ports used by Windows. For example, to automatically install an HP LaserJet Series II printer on LPT1 and an Apple LaserWriter Plus on COM2, you would add the following lines to the printer section:

```
"HP LaserJet Series II",LPT1:
"Apple LaserWriter Plus",COM2:
```

The printer names must appear in quotation marks; all of the valid printer names that Windows can recognize and install drivers for can be found in the CONTROL.INF file. (See Chapter 13 for more information on CONTROL.INF.)

HOT TIP

Create a SETUP.SHH File for Each of the Different Configurations that You Want to Install

The major benefit in using SETUP.SHH is the ability to create multiple setup configurations. This is especially useful for diskless workstations, because you can then run Setup multiple times from one machine, using different SETUP.SHH files to install and configure Windows for multiple users.

An even more effective approach is to create separate SETUP.SHH and SETUP.INF files, and use them together to control not only how Setup installs Windows, but also how individual Windows components are installed. This technique is possible because Setup allows you to use multiple switches when installing Windows. To use SETUPAP.SHH and SETUPAP.INF (two customized files), your command line might look like the following:

```
SETUP /O:SETUPAP.INF /H:SETUPAP.SHH
```

Several other switches are available for you to control how Setup installs Windows. For a complete list of these options, type **SETUP /?** at the DOS prompt from within the Windows directory. Windows will display a list of all available switches along with an explanation telling you how each switch modifies the installation process.

HOT TIP

You Can Force Users to Install Windows Using the Shared Network Version

If your company lets individual users install Windows themselves from the shared network version, you can add the following line to the SETUP.INF file:

```
NetSetup=True
```

This will force users to do a SETUP /N even if they don't use the /N switch. In other words, the Setup program will copy only the required configuration and INI files to the local workstation, forcing users to run Windows from the shared network copy.

HOT TIP

Prevent Setup from Checking the Workstation Hardware

Some network adapter cards and drivers can cause workstations to lose their network connection during the Windows 3.1 installation process. This problem usually manifests itself after Windows displays the hardware configuration that it has detected on a particular workstation. Windows then begins to copy files and prompts you to enter a path for DISK1. You know that you are logged into the network, and that all of the needed installation files are on the network, but you can't continue because Setup has severed the network connection between the server and the workstation.

To solve this problem, use the /I switch when you run Setup. When you use this switch, Setup won't try to determine the hardware settings for the workstation. This switch will work if you've already specified hardware settings in SETUP.INF and /or in an SHH file.

Updating Drivers on a Novell Network

After you've installed Windows for each workstation, you still have to update the network drivers to support Windows 3.1. For a Novell network, that means updating all of your users' IPX.COM files. If you have a lot of users, just the *thought* of rebuilding new IPX.COM files will send you to that bottle of Maalox that system administrators tend to have stashed in their desks.

Fortunately, Microsoft has developed a utility called NEWIPX.EXE to help in this process. NEWIPX.EXE lets users update their own IPX.COM files. NEWIPX.EXE is available by downloading and expanding the file NOVELL.EXE from CompuServe. NOVELL.EXE actually contains several utilities, including NEWIPX.EXE, that simplify the job of updating network drivers on a Novell network.

To use NEWIPX.EXE, create a directory (on the network) that all users can access. Then copy (into the new directory) the IPX.OBJ file that comes with Windows 3.1, the NEWIPX.EXE file, the NetWare NLINK.EXE file, and finally the .OBJ files for the network cards used on your network. Now each user can update his or her own IPX.COM files by logging onto the network, changing to the directory that you've created above, and typing the following command:

```
NEWIPX C:\NET
```

Here, C:\NET refers to the location of the workstation's existing copy of IPX.COM. The NEWIPX utility will check to see if a new IPX.COM file needs to be created. If a new file is needed, the utility creates a new IPX.COM file that is 100-percent compatible with Windows 3.1. (You can also add a line to each user's existing login script to automatically check whether the user is running the latest version of IPX.COM.)

HOT TIP

Create This Batch File to Simplify the IPX.COM Update Process

Create the batch file listed below and place it in a public network directory to automate the update process for your users. You can also have users run this batch file whenever you get updated drivers from the manufacturer of a workstation's network card.

```
N:\PUBLIC\NEWIPX C:\NET
IF ERRORLEVEL 2 GOTO IPXERROR
IF ERRORLEVEL 1 GOTO NEWIPX
GOTO ALL_DONE
:IPXERROR
ECHO Your version of IPX could not be updated.
ECHO Please contact your system administrator.
```

```
PAUSE
GOTO ALL_DONE
:NEWIPX
ECHO Your network driver has been upgraded. Reboot your system
ECHO to take advantage of the new driver.
PAUSE
GOTO ALL_DONE
:ALL_DONE
```

For additional tips and warnings regarding Windows setup on various networks, please refer to Appendix A.

CHAPTER

6

Tweaking and Peaking

If you've already installed Windows for network use and if you've updated the network drivers, you're ready to optimize the network environment. This chapter provides several techniques and tips for controlling and enhancing the Windows 3.1 and network connections. Specifically, you'll see how to change network printing connections and network drive mappings and you'll learn how to control which applications users can access.

Controlling Windows and Network Connections

Even though you've installed Windows successfully on the server and on all workstations, you probably still need to modify the way Windows handles the network connections. In fact, Windows provides so many options for managing network connections, you might feel a little overwhelmed when you begin poking around through the network-related dialog boxes. Within Windows 3.1, for instance, you can control broadcast messages, drive mapping changes, multiple server connections, print spooler size and entry limits, printer connections and reconnections, and network printer queue user options like banner pages and form numbers.

The next few sections walk you through some useful techniques and tips for controlling these options. Windows 3.1 currently supports 11 different types of networks, and the dialog boxes and features for each network are different. Since we don't have the space to show you features for all 11 types of networks, we'll focus on the most widely used network software, Novell NetWare.

Using the Control Panel to Control the Network Connections

You configure basic network connections by using the Networks dialog box, which is available by choosing Networks from the Control Panel. The Networks icon will only appear if the Windows network drivers have been installed on the system and if Windows detects the presence of a network card.

Figure 6.1 shows the dialog box that appears for Novell networks. On other types of networks, some of the options shown in Figure 6.1 will not be available, and the dialog box that appears will be different. The Networks dialog box can be accessed from individual workstations,

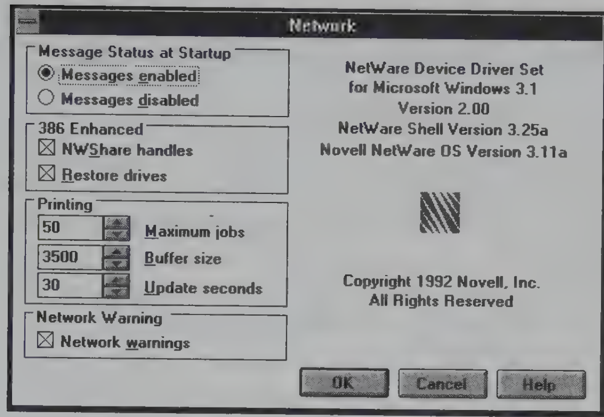


Figure 6.1 This Network dialog box appears if you are running Windows on a Novell network.

which means users can control many network options without getting help or permission from the system administrator.

Network Dialog Box: The Message Status at Startup Options

Use the Message Status at Startup option buttons to determine whether network messages—like net-mail alerts and messages sent from the server or from other users—will be displayed on the screen.

When you set up Windows to run on a Novell network, Windows automatically adds the broadcast-message utility NWPOPUP.EXE to the load= line of your WINI.INI file. So, NWPOPUP always runs in the background when you are connected to the network. If someone uses the NetWare SEND command to send a message to you, NWPOPUP will immediately display the message in your currently active window. If you turn on the "Messages disabled" button, you prevent incoming messages from appearing on your desktop. "Messages enabled" is equivalent to the NetWare CASTON (broadCAST ON) command, while "Messages disabled" is equivalent to CASTOF.

Networks Dialog Box: The 386 Enhanced Section

The 386 Enhanced section of the Networks dialog box allows you to control how Windows handles changes in drive mappings.

When Windows starts, it creates a table to keep track of the way network drives and directories are mapped to the workstation.

Normally, Windows uses the drive mappings that are made when you log into the network. When you turn on "NWShare handles," you can change drive mappings from within Windows. Simply open a DOS window (for instance, run the MS-DOS Prompt application in a Window), and change drive mappings the way you normally would at the DOS level. Windows will update its directory table, and will use the new drive mappings for each subsequent DOS window that you start.

If you select the Restore drives check box, Windows will not save the new drive mappings when you exit Windows. If you do not select this check box, Windows will save the new drive mappings for future Windows sessions. It's usually a good idea to deselect the Restore drives option. The "NWShare handles" and "Restore drives" check boxes will be dimmed (unavailable) when you run Windows in standard mode.

Network Dialog Box: The Printing Options

You can use the Printing options in the Network dialog box to improve the efficiency of the Print Manager in managing network print queues.

Notice that the Printing section in Figure 6.1 contains three different settings that you can control. The "Maximum jobs" option refers to the total number of print entries you can see at a time within the Print Manager window. The default is 50, but you can change this to a number from 1 to 250. A lower number will reduce the time it takes Windows to update Print Manager with the contents of the network print queue, while a higher number will increase the time required to update Print Manager.

The right number for you depends on the maximum number of jobs that tend to be in your network print queue at any given time. You might want to check the network print queue at several different times during the day, over a span of several days, to determine this information. (To check the contents of the network print queue, start the Print Manager, then click on View Selected Net Queue.)

If, for instance, you rarely see more than 10 or 12 documents in the network print queue, you could safely reduce the "Maximum jobs" value to 20 or 25. Changing this number does not change the number of actual print jobs that can be in the queue at any one time, it just changes the maximum number of entries that can be displayed in the Print Manager at a time. Be very careful with this setting, though. If you change the "Maximum jobs" value to 20 and the network queue is loaded with more than 20 jobs, your print jobs won't appear in the queue if they're, say, jobs 24 and 25. Again, your documents will still print; you just won't be able to view their location in the queue if the job numbers exceed the "Maximum jobs" value that you've specified. Of course, as jobs are completed and deleted from the queue, subsequent jobs will be "scrolled up" into the queue display.

The "Buffer size" is the total number of bytes that Windows allocates in memory for use in sending print jobs to the network printer. The default is 3500, but you can enter a value from 1 to 35,000. A larger buffer size will reduce the time required to spool your print jobs to the print queue. However, a large buffer size also means less RAM will be available to your applications—even when you are not sending a job to the print queue. The default value usually works well unless you frequently print large documents, in which case you should consider increasing the buffer size.

The "Update seconds" option is the frequency, in seconds, in which Windows updates Print Manager with information from the network print queue. The default value is 30, but you can specify a value from 1 to 65. In other words, with the default value, Windows will update Print Manager every 30 seconds with the status of the network print queue.

If you are in the habit of checking on your print jobs frequently, you might not have the patience to wait 30 seconds between updates. If that's the case, you can reduce the number. However, doing so can slow down other running applications because Windows will have to allocate a timeslice to the Print Manager more frequently.

HOT TIP

Increasing the Update Seconds Value

If you use a network printer that frequently experiences heavy traffic, you might find it necessary to increase the "Maximum jobs" value to 75 or 100. Although this will slow down the operations of Windows a bit, you can recover some of this time by increasing the "Update seconds" value. In this way, Windows has to update the Print Manager with relatively more information during each update, but has to update relatively less frequently.

For the sake of completeness, we should mention the Network warnings check box that appears at the bottom of the Network dialog box. When this check box is selected, Windows will warn you if you try to run

Windows with the wrong network drivers installed, or if your are not currently connected to the network. You might possibly deselect this check box if your network card has been removed for servicing or replacement, or if your system will be off of the network for several days, and you want to avoid the nuisance of seeing a warning each time you start Windows. In reality, though, you should rarely need to deselect this check box.

Controlling Printers and Queues

You use the Printers dialog box, available from the Control Panel, to install a printer for network use. In fact, installing a printer driver for a network printer is no different from the procedures you would follow to install a printer driver for a local printer. Since the steps for doing this are fairly elementary and are well covered in Chapter 6 of the Windows *User's Guide*, we won't cover that topic here. However, you do have to use a few additional features to route your printer port to a network printer, so we *will* explore these features.

Suppose you (or the system administrator) modified the SETUP.SHH file to include the HP LaserJet Series II and the Apple LaserWriter Plus printers, as we demonstrated in Chapter 5. The dialog box shown in Figure 6.2 would displays when you choose Printers from the Control Panel.

You determine which port to connect a printer to by choosing Connect from the Printers dialog box. Figure 6.3 shows the Connect dialog box that will appear. Even if you are using a network printer, you must connect the printer to a local port (LPT1, LPT2, COM1, COM2, and so forth).

Normally, a SPOOL, QUEUE, or CAPTURE command is present in each user's login script, so that the local printer port can be rerouted to the network cable and then to the appropriate network printer. For

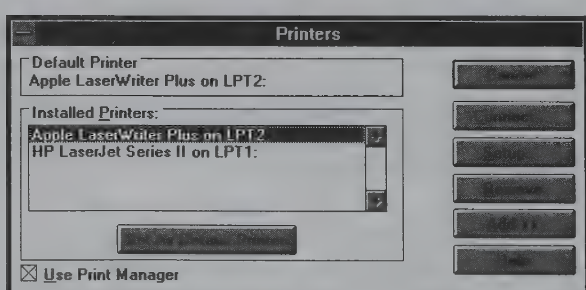


Figure 6.2 This dialog box shows that two printer drivers have been installed.

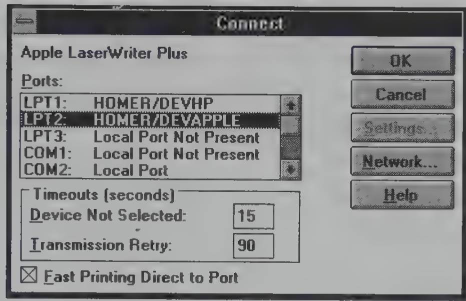


Figure 6.3 Use this dialog box to assign a local port to a printer, even if the printer will be a network printer.

instance, suppose the LPT1 port at the back of your computer has been rerouted to the network. Whenever an application tries to print to the LPT1 port, the print job will be sent instead to the network printer.

With Windows, you can remove routing commands from your login script and control the printer connections instead from within Windows. When you choose Network from the Connect dialog box, the available network queues and local ports will be displayed in the Network – Printer Connections dialog box, as shown in Figure 6.4.

You can then select the proper printer port and select the print queue where you want your print jobs sent. If your network has multiple printers, as is true for most LANs, you will see more than one network print queue listed. You might need to ask your system administrator which print queue name matches the printer closest to your workstation.

After you've connected your printer driver and port to a network print queue, it's a good idea to select the Permanent check box. Selecting this

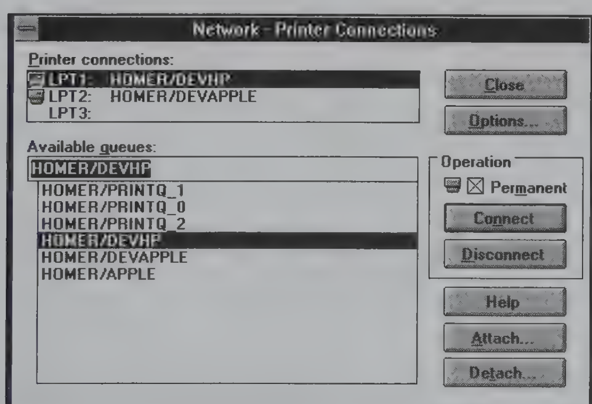


Figure 6.4 Use this dialog box to assign a printer driver and port to a network printer.

check box will cause Windows to reestablish the print queue connection every time you start Windows, eliminating the need to use a routing command in your login script. Since printers aren't usually moved from one location, selecting the Permanent check box usually makes sense.

You can still print to a different network printer at any time by changing the settings in the Network Printer Connections dialog box. You might want to make this kind of change if you're in a hurry to print a document, but your usual network printer is backed up with several large documents in the queue. You can view the contents of different network queues by starting the Print Manager and choosing View Other Queue. Then, pick the name of the queue that you want to examine. If you find a print queue that is empty or only has a few jobs, you can connect to this print queue temporarily.

HOT TIP

A Network User's Glossary

If you're new to networks or to networking in Windows, use this glossary of terms to familiarize yourself with basic network concepts.

bridge A software package used to determine which packets of information get passed between interconnected between two or more LANs

brouter A software package that combines the capabilities of a bridge and a router.

client A workstation (such as an individual PC) that requests and uses services from a network server.

diskless workstation A PC that is connected to a network server but does not have any disk storage capabilities of its own. Diskless workstations present special problems for Windows, because all files that Windows uses must be stored on the server.

file server A dedicated PC or other computer that includes mass storage facilities for storing applications and data shared by network users.

Internet Packet Exchange (IPX) Novell NetWare communications protocol used to control the transfer of information between different nodes of a Novell LAN.

local area network (LAN) A collection of computers and other devices that are physically linked via cable or other media, usually within a narrow geographic range (the same building, office complex, or campus); LAN services are managed by a network software package, such as NetWare.

NetWare The collection of programs that control and support a Novell local area network.

network A generic term referring to a collection of computers and hardware linked for communication. However, "network" also is frequently used to refer to specific brands of networking software—i.e., a Novell network, a Banyan VINES network, and so on.

network interface card (NIC) An adapter card that must be plugged into each PC in order to support physical network connections.

node Any workstation or other hardware device that can receive or send data across a network. Workstations, network printers, and file servers are all examples of nodes within a network.

Open Systems Interconnect (OSI) model A standard communications model developed by the International Standards Organization (ISO) that defines protocols for seven independent, yet hierarchical, communications layers, or network functions. Each communications layer enhances or builds upon the services provided by the layer beneath it.

packet A collection of data that is bundled and transmitted as a unit across network transmission lines.

password An authorization code that must be entered by a user in order to gain access to network services and to any of the nodes within a network.

peer-to-peer network Two or more computers that are physically linked, with communications controlled by network software, but that do not need to use network interface cards or a server.

print spooler A program that stores multiple print jobs temporarily while waiting for the services of a printer to become available. Networks typically use spoolers to establish and manage print queues for different network printers.

protocol A collection of standardized settings that must be used by all components of a network in order to ensure a common communications "language."

router A software package that determines which paths different packets take as they are passed between interconnected networks.

server (see **file server**.)

Transmission Control Protocol/Internet Protocol (TCP/IP) A set of protocols used widely among LANs to ensure accurate transmission and reception of data across network nodes.

workstation Typically, any personal computer that is linked to a network server.

Using the Printer Options Dialog Box

Under DOS, network printer commands, such as ejecting the current page before printing and printing an identifying banner at the start of each print job, had to be built into each network user's login script. You can now specify these options directly in Windows.

Windows provides an additional dialog box that is especially useful for managing the way your documents are printed on a network printer. To get to this dialog box, simply click on the Printer Options button in the Network Printer Connections dialog box. The Printer Options dialog box that you'll see next is shown in Figure 6.5. Most of the options in this dialog box replace switches that are normally added to the routing command used in your login script. We'll give you a brief tour of this dialog box next.

Notify If you select the Notify check box, Windows will display a pop-up message letting you know when one of your print jobs has finished printing. If you're doing a lot of printing, these messages can be a nuisance. On the other hand, if you let Windows notify you when your print jobs end, you can avoid having to run to the network printer periodically to see if your top-secret documents have finished printing.

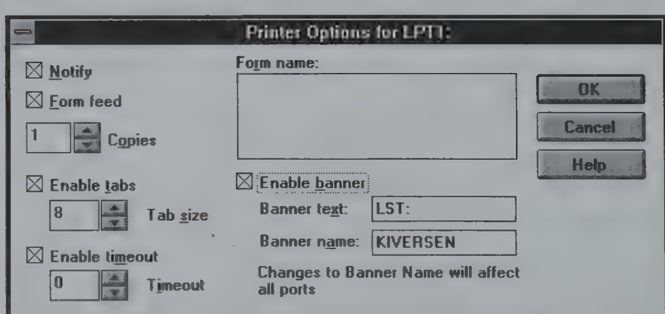


Figure 6.5 You can use this dialog box to determine the way Windows interacts with your network printers.

You'll often find Notify to be a useful feature, especially if your network printer is located quite a distance from your PC.

Form feed When you select the Form feed check box, Windows issues a page-eject command to the printer. This ensures that your print job starts at the top of a new page, and not on the last page of some other user's document. Form feed is useful if you're using a network printer that has continuous-feed (tractor-feed) paper. The Form feed option isn't of any value if your network printer uses a sheet feeder—as is the case with a laser printer—that ejects pages automatically after they are printed.

Copies This option indicates the number of copies to be printed for a given job. Normally, you'll want to leave this set at 1 and control the number of copies to be printed from within each application.

Enable tabs This option is a holdover from the days when some older DOS applications could not convert tab characters to blank space on paper during printing. You'll rarely—if ever—need to bother with this setting, since most applications can deal with tabs well enough on their own.

Enable timeout This setting is useful for applications that have to read to or write from a disk file prior to printing. If this happens, your printer might interpret the delay to mean that a print job that you've just started has finished, even though the application is just busy reading to or writing from disk before it can begin printing. For example, if you are using WordPerfect 5.1 (for DOS) to print a document that contains a graphic image, WordPerfect must load the graphic file into memory, from disk, before the document can be printed.

If you find that your documents are not printing, try selecting Enable timeout. The default value of 90 seconds tells Windows to wait 90 seconds—following the start of a print job—before the job is actually sent to the printer. You can enter a value from 1 second to 1000 seconds.

Enable banner A banner is simply a cover page that can be printed at the start of each print job. If you select the Enable banner check box, you can create banner text and specify your name for the cover page. You should use a banner if you want to prevent other users from mistakenly picking up your print jobs from the network printer.

All of the options in the Printer Options dialog box affect only the currently selected port, with the exception of the banner settings, which affect all ports.

Using the Network Printer Connections to Attach to Other Servers

The Network Printer Connections dialog box includes two buttons—Attach and Detach—that you normally wouldn't associate with printing. The Attach and Detach options allow you to connect to multiple servers on the network.

Figure 6.6 shows the Attach File Server dialog box that appears when you click on the Attach button from within the Network Printer Connections dialog box. You can use this dialog box to select a server to attach to, and to enter your user ID and password (if necessary). If you choose the Detach button from the Network Printer Connections dialog box, Windows will let you detach from any of the servers you are currently attached to—even the server you are currently logged onto.

Controlling Drive Mappings

On most networks, the drives and directories that you can use are controlled from the system login or user login script. In Windows 3.1, however, you can use the File Manager to control how the drives and directories on your network appear.

To set your drive mapping, start the File Manager, then choose Disk Network Connections. The Network-Drive Connections dialog box, shown in Figure 6.7, appears.

With this dialog box, you can easily select a drive and change the mapping of the directory for that drive. If you want to make the new drive mapping permanent, just select the Permanent check box. When you make a drive mapping permanent, Windows updates the [Network] section of WIN.INI with the mapping information, so that every time you start Windows, the drive mappings that you made permanent are restored. Figure 6.8 illustrates what a sample WIN.INI file would look like with the PUBLIC and NETBAT directories mapped to the F and N drives, respectively.

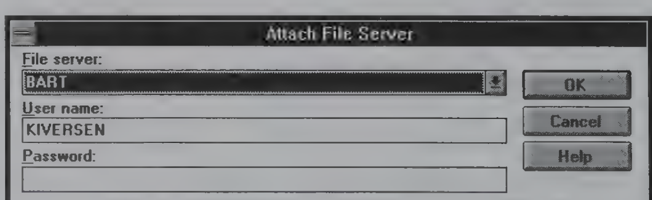


Figure 6.6 If your network has multiple file servers, you can use this dialog box to attach to a different server.

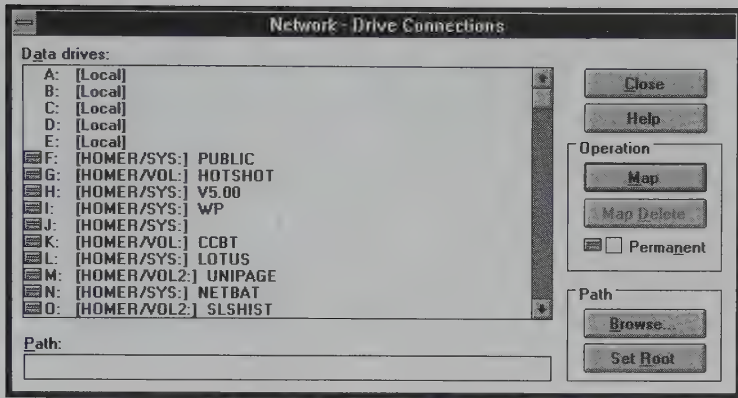
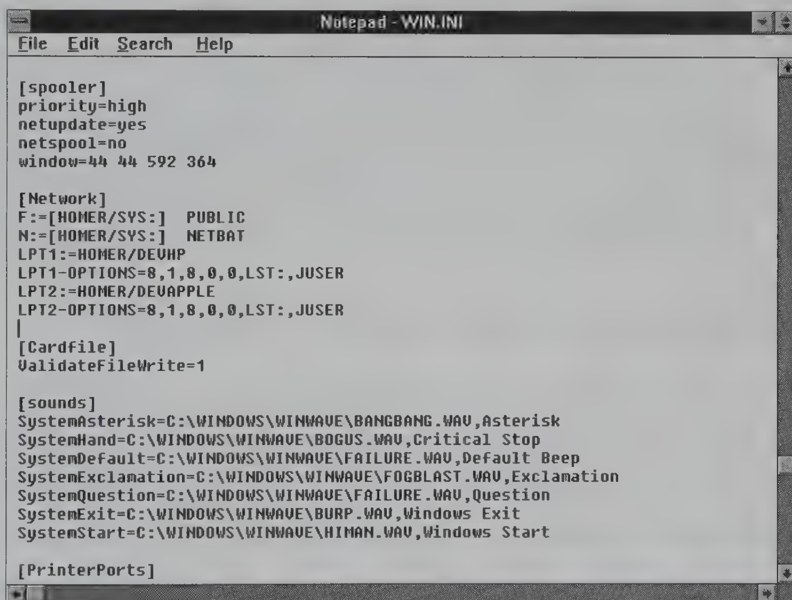


Figure 6.7 Use this dialog box to change the mapping for selected network drives.


Notepad - WIN.INI

File Edit Search Help

```
[spooler]
priority=high
netupdate=yes
netspool=no
window=44 44 592 364

[Network]
F:=[HOMER/SYS:]  PUBLIC
N:=[HOMER/SYS:]  NETBAT
LPT1:=HOMER/DEVHP
LPT1-OPTIONS=8,1,8,0,0,LST:,JUSER
LPT2:=HOMER/DEVAPPLE
LPT2-OPTIONS=8,1,8,0,0,LST:,JUSER

[Cardfile]
ValidateFileWrite=1

[sounds]
SystemAsterisk=C:\WINDOWS\WINWAVE\BANGBANG.WAV,Asterisk
SystemHand=C:\WINDOWS\WINWAVE\BOGUS.WAV,Critical Stop
SystemDefault=C:\WINDOWS\WINWAVE\FAILURE.WAV,Default Beep
SystemExclamation=C:\WINDOWS\WINWAVE\FOGBLAST.WAV,Exclamation
SystemQuestion=C:\WINDOWS\WINWAVE\FAILURE.WAV,Question
SystemExit=C:\WINDOWS\WINWAVE\BURP.WAV,Windows Exit
SystemStart=C:\WINDOWS\WINWAVE\HIMAN.WAV,Windows Start

[PrinterPorts]
```

Figure 6.8 The [Network] section of WIN.INI contains drive mappings for the PUBLIC and NETBAT network directories.

Hot Tip

Create Drive Mappings for Different Login Groups

You can create separate files for different login groups, so that each group has its own customized mappings for all network drives and directories. By doing so, you can eliminate the tedious process of mapping multiple drives and directories on multiple workstations.

Here's how to use this technique: Open Notepad and open WIN.INI. Highlight the section that contains the current drive mappings and choose Edit Copy. Next, choose File New, then choose Edit Paste to copy the entire block of text to a separate file. Chose File Save, then save the file to a public directory or—better yet—to the network Windows directory. You can append other drive mapping combinations to this file to support different user groups. Users can then open this "clip art" file to copy the appropriate drive mapping section into their own WIN.INI files, and can even change their WIN.INI files occasionally if they need to use a different set of drive mappings.

Note: Make this file read-only, so that users can't accidentally cut sections from the file.

HOT TIP

The Windows "Permanency" Features Let You Reduce the Size of Login Scripts

If you select the Permanent check box in the Network Printer Connections dialog box, you can remove the spool or capture commands from your existing login script, thus reducing the time it takes to log into the network. Likewise, if you map drives in the File Manager, you can remove the map commands from your user and system login scripts.

Controlling User Access to Other Programs

As users gain experience in using the network, they tend to develop an uncontrollable urge to experiment, often switching to various network drives to see what other programs they can access. You can control the environment in which users work by removing options in the Program Manager File menu. You can also remove the File menu altogether to prevent users from creating, deleting, and modifying program groups. You can even force your users to remain in Windows by disabling their ability to Close from Windows. Figure 6.9 illustrates the user-limiting commands that you can place in the Settings section of PROGMAN.INI.

Here's an explanation of the three settings that appear in Figure 6.9:

NoRun=1 removes the Run command from the File menu, restricting users from running applications that are not present as icons on their desktop. Users will still be able to add new applications to program groups, but with the **NoSaveSettings=1** command, any new programs added during a session will not be saved for future sessions.

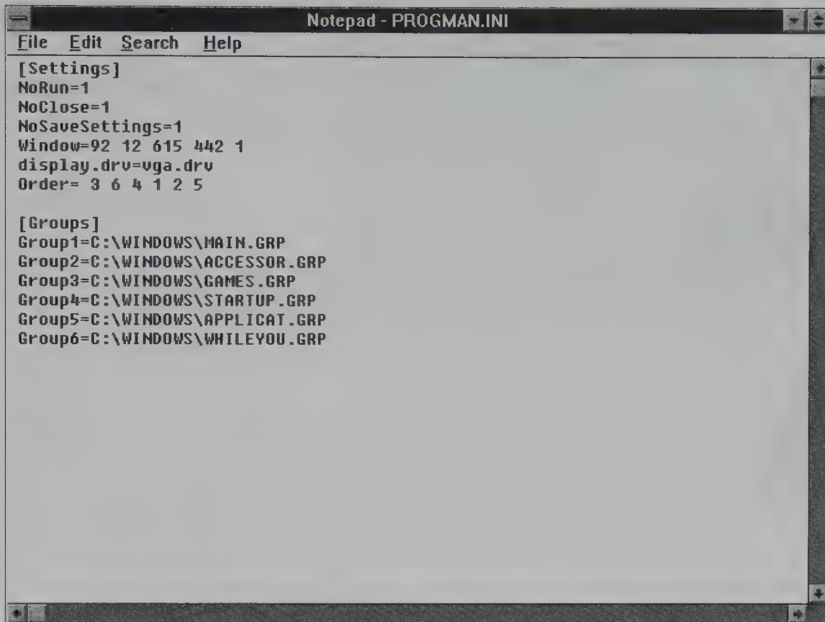


Figure 6.9 This sample PROGMAN.INI File shows how you can restrict options for certain network user groups.

NoClose=1 will not allow users to exit Windows—unless they want to reboot their workstations.

You can also add the following lines to PROGMAN.INI to further control what programs users can add to their Windows desktops:

NoFileMenu=1 removes the File menu altogether.

EditLevel=x (where x equals 1, 2, 3, or 4) determines which File menu options are available to users.

When x equals 1, users cannot access the File New, Move, Copy, and Delete commands when a group is selected.

When x equals 2, the File New, Move, Copy, and Delete commands are never available.

When x equals 3, text in the Command Line text box of the File Properties dialog box can't be changed, in addition to the restrictions that apply when x equals 2. This restriction level prevents users from adding an application by changing the properties of another application. Users can still change the other text boxes in the File Properties dialog box.

When x equals 4, all of the restrictions of x equals 3 are in effect, *plus* users will no longer be able to change any of the text boxes in the File

Properties dialog box. This prevents users from changing the name of applications, working directories, and shortcut keys.

For more information on potential problem areas in running Windows on various networks, please refer to Chapter 16.

CHAPTER 7

Network Utilities

This chapter introduces some of the shareware and freeware utilities and programs that can make your network more user-friendly and can make your job much easier. These utilities include a phone messaging system, a network applications installer, a Novell MHS alert program, and an E-mail program.

While You Were Out

While You Were Out (WYWO) from Caliente International acts as an electronic phone-message-slip system, and can perform all its tricks using only a few very well designed windows. You can download WYWO from the WINADV forum on CompuServe. WYWO includes a great little monitor program that acts as a message center "receptionist" for all of your network users. The shareware version of the program is for evaluation purposes only. If you are switching all of your network users to Windows, you might give some serious thought to purchasing While You Were Out. It's a great time-saver.

Figure 7.1 illustrates the default "message slip." You can use this window to send messages to other users, either individually or as group members.

WYWO protects your phone messages from unauthorized viewing through the use of passwords. The Monitor program acts as a central

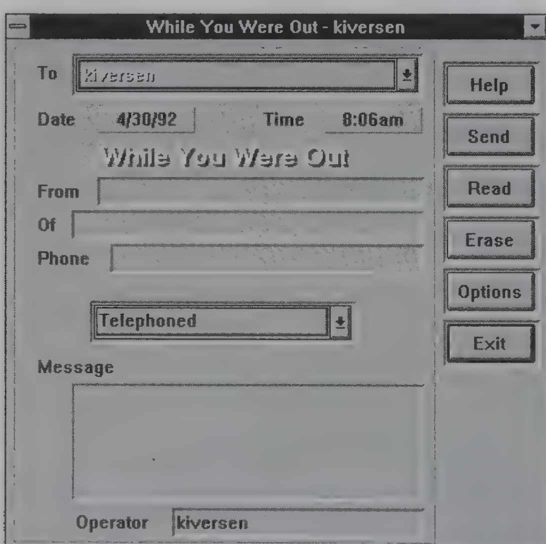


Figure 7.1 This is the default While You Were Out message slip.

message center, allowing any network user to check whether any messages are waiting in his or her message slot.

Installing the program is easy. Download the program and unzip the file. Next, choose File Run from the Program Manager and select the INSTALL.EXE program, then choose OK. The install program asks for a directory to copy the WYWO files into, then proceeds with the installation after you've supplied a directory name.

The install program creates a program group that contains three icons. It then displays a Notepad README file that contains instructions for using WYWO, as well as information for purchasing the software from Caliente. When you exit Notepad, the install program returns you to your desktop.

The three icons in the WYWO group all access different programs. Use the WYWO icon to access the phone messaging system, which allows you to send and read messages. The WYWO Setup icon loads the program that you need to use to set up different users and user groups. You can also use WYWO Setup to change and customize the types of messages that users are allowed to send. For example, you can add a message such as "Phoned, see them immediately" to the generic list of message types. The third icon, WYWO Monitor, displays the message rack for all users and groups.

Helping Users Add Their Own Windows Applications

If you want to give even your shakiest Windows users some help in adding applications to their desktops, try installing the Network Application Installer (N.A.I.), from Aleph Systems.

The N.A.I. utility, available in the WINADV forum on CompuServe, assists users in adding and removing programs, and is just about the most idiot-proof Windows utility available for network users. NAI11.ZIP contains all the programs and information you'll need to install and run the Network Application Installer—with one exception. You'll also need the VBRUN100.DLL file, which is also available in the WINADV forum on CompuServe.

Figure 7.2 shows the easy-to-use N.A.I. screen that helps you add applications to and delete applications from the Program Manager.

N.A.I. comes with configuration files for 10 Windows applications, several of which—unfortunately—are not really widely used. However, you can run NAIMAINT.EXE to add applications to or remove applications from this list—in effect customizing the list of available applications

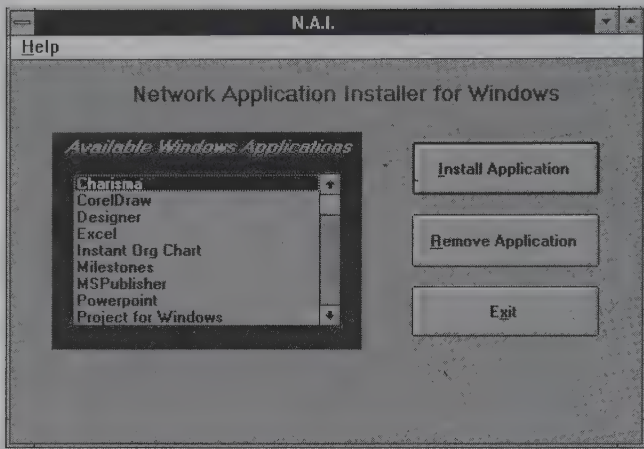


Figure 7.2 Your network users can add or delete programs with this simple, customizable Network Application Installer window.

to match the needs of your users, or to match the available applications stored on the file server.

Basically, the N.A.I. program will copy any needed files from the server to the local disk, update WIN.INI, and install the appropriate icon on the local desktop. To remove a program, N.A.I. shifts into reverse, removing any files from the local system, removing unneeded entries in WIN.INI, and removing appropriate icons from the desktop. Of course, N.A.I. won't actually delete any applications from the server.

NAI is well documented, both in the help system and through a set of README files stored in Write format. If you want users to be more responsible for the programs they access, this utility will help them to be more independent.

ON DISK

Keep VBRUN100.DLL Available for Applications That Need It

Many of the utilities available as shareware or freeware require that the file VBRUN100.DLL be located in the WINDOWS directory. This file is needed for programs developed in Visual Basic. You can download this file from the WINADV forum on CompuServe. Look for the file VBRUN.ZIP. This file is approximately 170KB, so the download is going to take time. But you'll probably agree that it's time well spent once you see and use all the applications that require this file.

ON DISK

The VBRUN.ZIP File

This file (explained in the previous Tip), is available in the NETWORKS directory of the *Windows Insider Disks*. To use this file, unzip its contents and copy it into your Windows directory.

Keeping Track of Incoming and Outgoing MHS Messages

With most E-mail systems, users have to start up the mail program whenever they want to check for messages. You can save yourself and your users some time by installing a neat little utility called MHS Mail Alert, by Chris A. Means. This shareware utility is available on the WINADV forum of CompuServe.

MHS Mail Alert has settings for most of the common MHS mail systems—including Coordinator, Beyond Mail, and DaVinci eMAIL. By adding MHSALERT.EXE to their Startup group, users will be alerted immediately whenever they have new mail. The program requires that the DOS variable USR be set to the user name in each login script. You can set this variable in the AUTOEXEC.BAT file by entering the following command:

```
SET USR=JUSER
```

Here, JUSER is the user's login name. The program then appears on the desktop as an icon that looks like a postal carrier (a male postal carrier, actually). This icon actually takes two rather clever forms: only the mailman appears when no new mail is present; if the program detects that you've received new mail, the icon changes to a mailman with mail coming out of his mail bag. There is also a counter under the icon to let you know how many new mail messages waiting for you.

Figure 7.3 illustrates the Settings dialog box that appears when you click on the mailman icon. You'll find that most of the options are self-explanatory, although a Help button is provided if you want additional explanations for the available settings.

One of the more interesting features of this program is its ability to put the right mouse button to use. Yes, that all-too-often-ignored button actually has a purpose in MHS Alert. Either the left *or* right mouse button will launch the E-mail program when you click on the mailman icon.

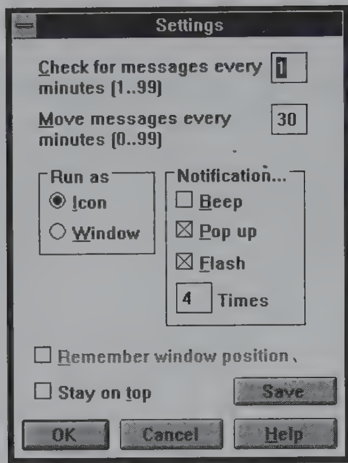


Figure 7.3 Use this dialog box to modify default settings for MHS Alert.

MHS Alert can be configured so that users don't even need to have the E-mail program icon on their desktops. The program can also be used by remote users of your LAN (for instance, sales reps who are out in the field, but have their laptops handy). Remote users can check their messages whenever they dial up and log onto the network.

Using DFREE to Keep Track of Disk Space on the Server

One of the biggest nightmares for network administrators is, of course, trying to keep track of available disk space on the server.

DFREE for Windows, a shareware utility from NEOCOM Microspecialists, Inc., allows you to graphically track individual drive capacities, as well as all drives on your network. DFREE.ZIP is available in the WINADV forum on CompuServe, and requires that VBRUN100.DLL be stored in the Windows directory. DFREE.ZIP contains two files—the program DFREEWIN.EXE and the help file DFREEWIN.HLP. The program is very easy to use and can display drive capacity and free space as either a pie chart or bar chart.

A Free E-mail System

For many smaller networks, like those running NetWare Lite, E-mail is a luxury. Well, thanks to Ivory Tower Software, even the smallest of networks can afford some simple E-mail capabilities. EMAIL2.ZIP is available on CompuServe in the WINADV forum. Ivory Tower Software's Email program is freeware, so you won't even be obliged to pay a registration fee. The program does require that VBRUN100.DLL be available. This program suggests, however, that VBRUN100.DLL be placed in the same directory as the EMAIL.EXE program. It isn't mandatory to store VBRUN100.DLL in the same directory as EMAIL.EXE, but the program runs much faster if you let it open the DLL from the same directory.

The freeware version includes the ability to send text messages, provide pop-up notification of new mail, use distribution lists, and attach documents, programs, and pictures to messages. The program can even archive messages to files. A more powerful commercial version is also available. However, you should give the freeware version a try first. You might even find that your users are satisfied with the options available with the freeware version.

Figure 7.4 illustrates a sample message screen for this E-mail system. The program lets you identify users by their full name if you want. You can use the login user ID for user names, but the full-name features adds a nice touch of personality to the system. The program also provides a new-mail indicator: the icon for Email changes when new mail arrives. All things considered, this is a good E-mail program to use as a starter for your network.

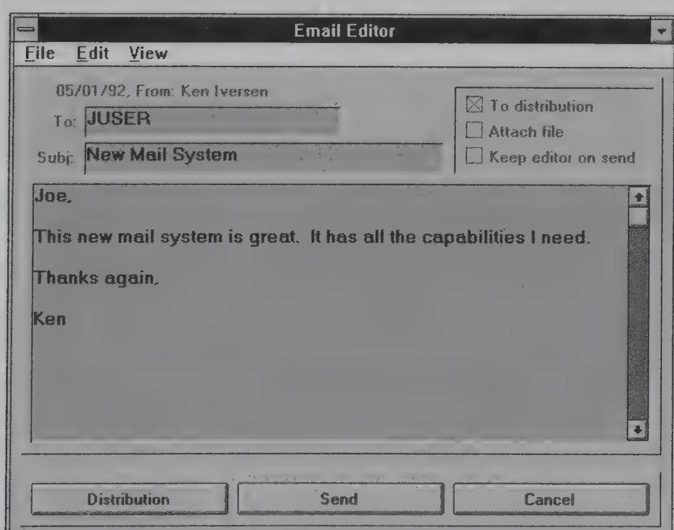


Figure 7.4 Here is an example of Ivory Tower Software's Email message screen.

Part Overview

PART 3

Managing Your Files and Disks

By design, the title for this part of the book is not "Using the File Manager." The File Manager works great for some file- and disk-management operations, but it's not always the best tool for these kinds of tasks.

In this part of the book, you'll discover how and when to use the various file- and disk- management tools that are available for use with Windows. Chapter 8 focuses on many of the little-known power features of the File Manager and offers some tips for using .INI and .INF files to customize your Windows system. Chapter 9 explains how to manage your files using DOS commands in coordination with Windows. Chapter 10 describes and demonstrates some effective utilities that are available for enhancing and customizing Windows file and disk management.

CHAPTER

8

Managing Your Desktop and Hard Disk Space

In the simple era of DOS programs, maintaining your hard disk was pretty easy. You used the old standby commands—like ERASE, COPY, and RENAME—to get rid of or to back up old files and programs periodically and to add new applications to your hard disk. And managing your display environment wasn't even an issue. After you finished using one program, you just exited to the DOS prompt and started another. You really didn't have a lot of control over how to display programs and files.

In Windows, though, computing life is more complex. A GUI (graphical user interface) like Windows might be easy to *use,* but the number of options available to you can be overwhelming. You can view multiple programs in different windows at one time, and you can use several different techniques to move, copy, delete, and print files. As you add program items, program groups, applications, and files, your desktop can take on some of the characteristics and confusion of a Salvador Dali painting.

Another important factor to consider: when you install Windows or a typical Windows application, you might find that the files copied onto your hard disk take up several megabytes. It's customary for a Windows application's setup program to ask you whether you want to install different categories of files. But this can be a Catch-22: you often don't know whether you need certain files until you've learned how to use the application's various features. The end result is that you probably err on the side of safety and copy more files to your hard disk than you need. Over time, as you install different applications, your hard disk can become cluttered with dozens of megabytes' worth of unused and unnecessary files.

In this chapter, we'll show you how to use the File Manager and Program Manager to optimize both your desktop and your hard disk space so that you can work as efficiently as possible. Our tips for customizing the Program Manager and File Manager environments can be especially useful in your quest for optimal desktop management. We'll also explain how to rid your hard disk of Windows files that you don't need.

For complex file- and disk-management jobs, you might find that the right tool for the job is the too-often-ignored macro Recorder. When the macro Recorder is applied correctly, frequently used file-management routines can be automated to save you literally hours of work each month.

Windows Is Only as Efficient as You Are

How organized and efficient are you *really*? Do you know precisely how far your car can go on a gallon of gasoline? Did you make your bed this morning? Does a sink filled with dirty dishes look to you like a landfill in miniature? And, most important, do you feel that coworkers who keep their desks messy are engaging in behavior that's deeply immoral?

If you answered "No" to all four of the above questions (or even to two or three of the questions), then you need this chapter. You might be grumbling now—"a clean desk is a sign of a sick mind" is probably your motto—but you'll thank us later. Organizing your desktop and hard disk resources in Windows is not only essential, it actually can be enjoyable.

And even if you answered "Yes" to one or more of these questions, you'll appreciate the flexibility that Windows offers you in managing your computing resources. This chapter will show you some of the less obvious but powerful Windows desktop- and disk-management tricks and techniques.

We'll begin by explaining some Program Manager features that you might not realize are available to you.

Tiling Is More Powerful Than You Might Think

You've probably experimented with the Windows Tile and Cascade features for arranging different windows on the desktop. Frankly, cascading—which displays all open Windows so that their title bars are visible—isn't really very useful. If you want to identify which applications are currently open (from the Program Manager), it's easier to just click on Window (from the Program Manager menu bar) and examine the list of applications that appears there.

On the other hand, tiling, when used in conjunction with other Windows desktop management features, provides a lot of flexibility in the way you organize icons, windows, applications, and even individual files. So, it's a good idea to learn to use every tiling technique available.

As you might already know, tiling from the Program Manager (clicking on Window Tile) displays open windows as a set of "tiles" so that you can view the contents—at least in part—of each open window. When you tile from the Task List, Windows tiles all currently open applications.

ON DISK

Using Big Desk

One of our favorite shareware utilities, Big Desk, creates a virtual desktop map for all your open applications. The virtual desktop makes it easy to find and switch to different applications—without using the Task List. Big Desk is stored in the UKBIGDSK directory on the *Windows Insider Disks.*

Windows Tiles on a Most Recently Used Basis

When you tile—either from the Program Manager or from the Task List—Windows always places the active window in the first position; the order for other Windows is determined by the order in which they were previously made active.

This feature is important if you want to control the order in which the windows appear. When you tile three windows, for instance, the windows are displayed from left to right, with the currently active window appearing in the left-most position, the next most recently used window appearing in the center position, and the least recently used window appearing in the right-most position. When you tile four or more windows, the windows are displayed from top to bottom first and then from left to right; in this case, the currently active window appears in the top-left position, and other windows appear—in order of their most recent use—from top to bottom and from left to right. The "four or more" arrangement is a bit counter intuitive, though, since you might expect Windows to be tiled from right to left first, and *then* from top to bottom. Figure 8.1 shows the tile order for two, three, and four windows

Why is this feature important? Consider a few examples. Suppose you've organized your Program Manager so that your Main, Applications, and Accessories group windows always appear open when you boot Windows, as shown in Figure 8.2. Notice that the Main window appears in the left-most position, while the Accessories group appears in the middle and the Applications group appears to the right.

It would make sense to organize your tiled Windows from left to right (to the extend that Windows allows you to do this), in the order in which they are most frequently used. Due to the way Windows creates tiles, the left (or top-left) window will always be visible when you tile. If this is your most frequently used window, then this is the result you want.

Over time, suppose you realize that you use the Applications group most frequently, the Accessories group less frequently, and the Main

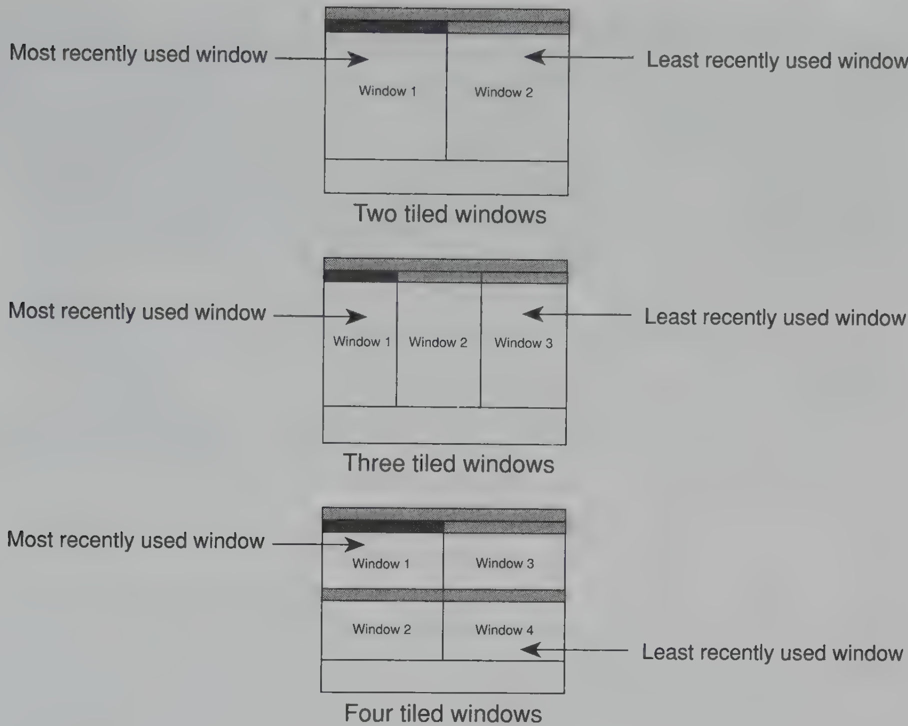


Figure 8.1 For two and three open windows, tile order is from left to right; for four or more open windows, tile order is from top to bottom *first* and then from left to right.

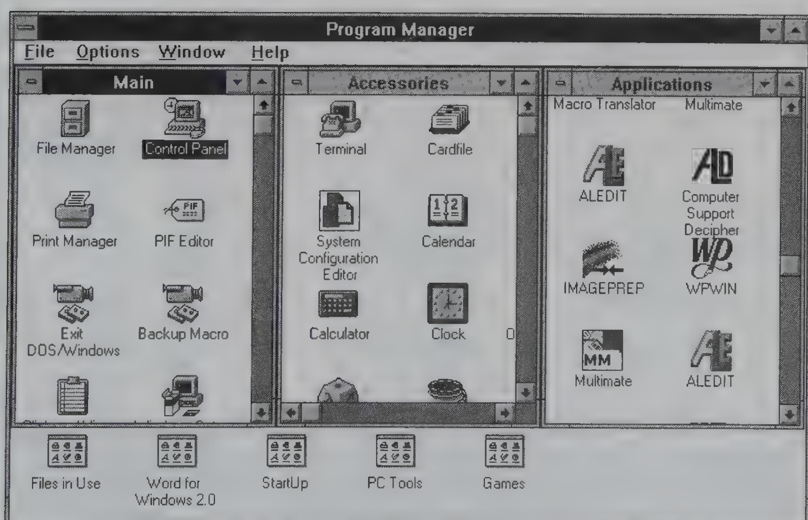


Figure 8.2 This is the order of group windows before they are retiled.

group least frequently. To reflect this pattern of use, you want your Applications group to appear to the left, your Accessories group in the center, and your Main group in the right-most position, as shown in Figure 8.3.

To make this change, you might be tempted to drag the Windows to their new positions. But this is more work than you need to do, and you might also have to do some resizing to make the windows fit their new positions snugly. If you realize that Windows always puts the active window in the first tile position, you can accomplish this kind of change in a few seconds. Just click on each window in the *reverse* order that you want them to appear, then tile the windows. Here are the steps to follow:

1. Click anywhere inside the Accessories window to make it active.
2. Click anywhere inside the Applications window to make it active.
3. Choose Window Tile to reorder the positions of the windows.

In this example, Windows realizes that the Applications window is current and the Accessories window was the previously active window; the windows were then reordered on this most recently used basis. This principle seems so basic that you might not consider it to be all that important, but it can be valuable when you quickly need to tile windows in a particular order. Take a look at our next example for further evidence of this principle's value.

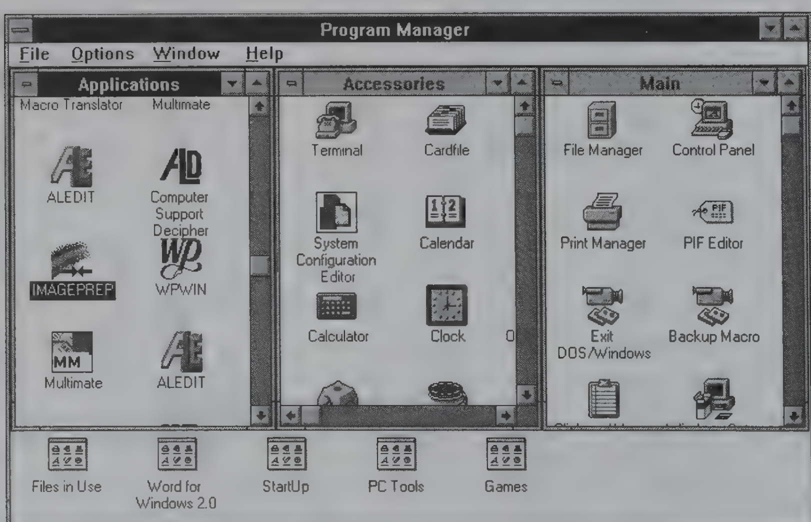


Figure 8.3 This is the order of group windows after they have been retiled.

HOT TIP

You Can Tile in Three Different Ways

Keep in mind that tiling can be performed from three different places in Windows. You can tile in the Program Manager to tile all open group windows; you can tile in the File Manager to tile all open directory windows; and you can tile from the Task List to tile all open applications.

Using Drag-and-Drop to Create Document Icons

In Windows 3.1, you can create an icon for a document simply by dragging the document from the File Manager into a group within the Program Manager.

Because Windows 3.1 makes it so easy to create icons for documents, it's a good idea to create a separate group for documents that you use frequently. If an application has its own program group, you can also drag and drop document icons into this group. In this way, you can easily find your frequently used documents, and you can launch a document and an application simultaneously from the Program Manager.

Notice in Figure 8.4, that we've created a program group called Files in Use and have made it an open, tiled Window. We use this group window to store document icons for files that we're currently working on or that we use frequently. Of course, you can give this group any name

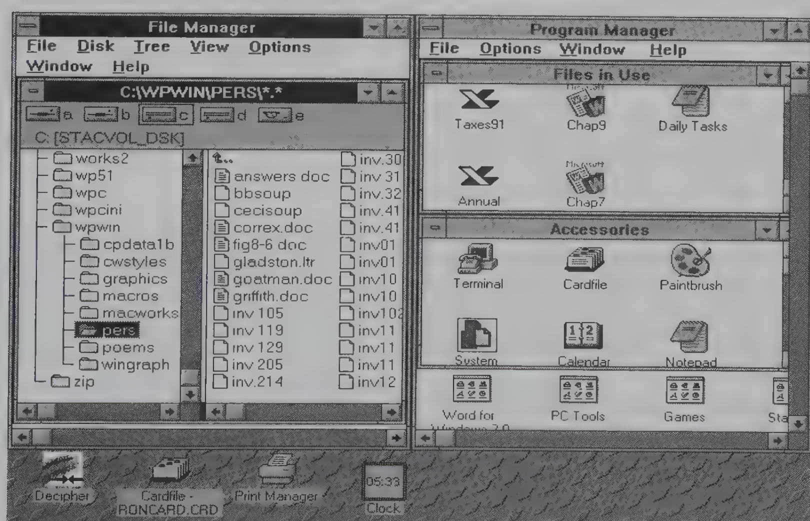


Figure 8.4 In this Program Manager arrangement, we have created a separate group window for storing frequently used files.

that you choose; the point is that by creating a group window for frequently used files, we make it easier to open them directly from the Program Manager.

In Figure 8.4 we've also used the Task List to tile the File Manager and Program Manager. Here's how you could do the same to easily add a document icon to this group:

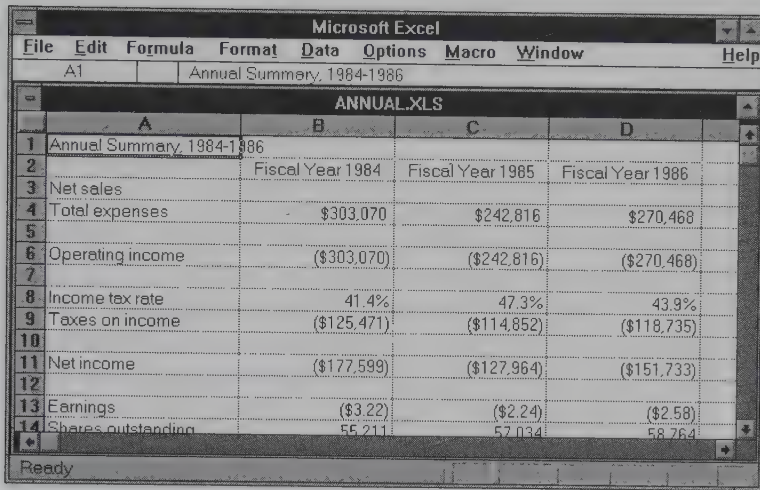
1. Open the File Manager and double-click on the directory that contains the file you want to place in the Files in Use group window (or other appropriately named group window).
2. Press **Ctrl+Esc** to display the Task List window, then click on the Tile button to tile the Program Manager and File Manager windows.
3. Drag the appropriate file from the File Manager into the desired group window in the Program Manager.

That's all there is to it. Windows automatically creates an icon for the document when you copy it to the Program Manager—assuming that an *association* exists for the application. (If an association—a link between a document and an application—doesn't already exist, you'll need to create one. We'll show you how to do this later in the chapter.) Windows assigns to the document the same icon that is used by the application that created the document. In other words, in Figure 8.4, the document called Annual is an Excel document because it has the Excel icon, the Daily Tasks document is a Notepad document because it has the Notepad icon, the Chap9 icon is a Word document because it has the Word icon, and so on.

Now, when we double-click on an icon in the Files in Use group window, Windows will start the application that the document is associated with, and then will automatically load the document. For instance, when we double-click on Annual, Windows responds by displaying the window shown in Figure 8.5.

Did you notice that the order in which the Files in Use window is tiled has some significance? If you'll take a look at Figure 8.4, you'll see that the Applications and Main windows aren't in view (even though they were tiled in Program Manager); only the Files in Use window can be seen in its entirety.

Suppose we had tiled the File Manager and Program Manager in reverse order. For instance, if the File Manager was already open and the Program Manager was currently the active window, then Task List would have tiled these windows in the reverse order. But even in this case, the



	A	B	C	D
1	Annual Summary, 1984-1986			
2		Fiscal Year 1984	Fiscal Year 1985	Fiscal Year 1986
3	Net sales			
4	Total expenses	$303,070	$242,816	$270,468
5				
6	Operating income	($303,070)	($242,816)	($270,468)
7				
8	Income tax rate	41.4%	47.3%	43.9%
9	Taxes on income	($125,471)	($114,852)	($118,735)
10				
11	Net income	($177,599)	($127,964)	($151,733)
12				
13	Earnings	($3.22)	($2.24)	($2.58)
14	Shares outstanding	55,211	57,034	58,764

Figure 8.5 This Excel display appears when we choose the Annual document icon from the Files in Use group.

Files in Use window would be totally visible within the Program Manager because it is the first (top-left) tiled window (out of a total of four tiles).

You can see that the order in which different windows are tiled isn't always a trivial matter. If the Files in Use window was tiled in either the third or fourth positions, we couldn't have simply dragged a document from the File Manager into the Files in Use group. We would have had to retile the Program Manager or done some resizing to make both the File Manager and the Files in Use windows visible on the desktop simultaneously. That kind of desktop reorganization can be a real annoyance when you're in a hurry.

What's the moral of this scenario? Tile your program groups in an order that best fits the way you use Windows to manage your files and applications.

HOT TIP

Double-Click on the Desktop to Pop-Up the Task List

You can also display the Task List by double-clicking on an empty area of the desktop. Of course, if you have Program Manager or some other application maximized (covering the entire desktop), you'll need to reduce the size of the maximized window to make the empty portion of the desktop visible.

Associating Documents with an Application

Associating documents with an application isn't really a new or difficult feature in Windows 3.1. If you've been working with Windows for any significant length of time, you've very likely already figured out how to associate a document with an application.

But if you've never ventured into this area, creating associations can seem formidable. An *association* is just a link that Windows creates between a document extension and an application. For instance, if you associate all files that have the extension LTR with WordPerfect for Windows, you can create icons for all WordPerfect documents that have the LTR extension, and you can launch these documents directly from the File Manager or the Program Manager. The association tells Windows that every LTR file on your hard disk is used with WordPerfect for Windows.

Even if you already know how to create associations between documents and applications, we're willing to wager that you've overlooked some important techniques that are available to you in this area. At any rate, we've included the next few sections to help you exploit the Windows association feature to its fullest.

Creating an Association

What's the point of creating an association? When you associate a document with an application, you can launch both the document and its application simultaneously—from the File Manager or from the Program Manager.

In the File Manager, any document that has an association is assigned an icon that looks like a piece of paper with lines written on it. A document that has no association appears within the File Manager as a blank document. Figure 8.6 shows this difference.

Most Windows applications assign a default extension to files. For instance, Word for Windows files are given the extension DOC by default, while Excel worksheet files are given the extension XLS by default. But you can assign any extension to virtually any Windows document. When you create these "custom" extensions, Windows is unable to link the document with its application unless you create an association for that extension. Here's how to create an association:

1. Use the File Manager to display any directory that contains a file with the extension you want to associate.

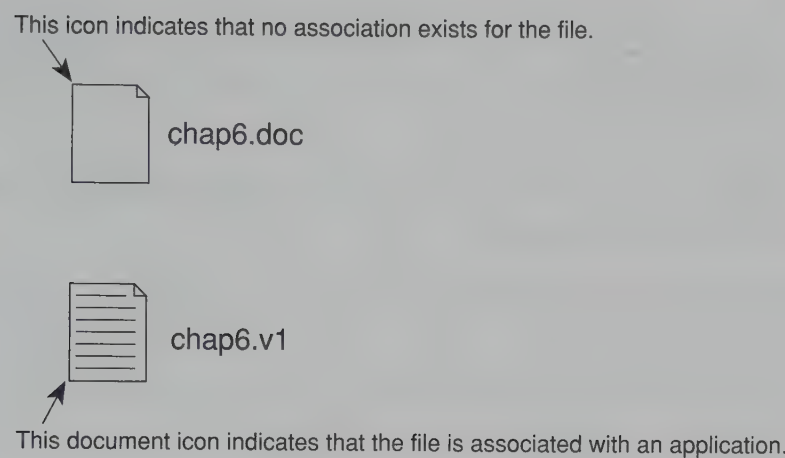


Figure 8.6 In the File Manager, files that have an association appear to have "text" in their icons, while files that have no association appear as blank pages.

2. In the File Manager directory window, click on any file that has the extension for which you're creating the association.
3. Click on File Associate to display the Associate dialog box, shown in Figure 8.7. Note that the extension that you are creating an association for appears in the Files with Extension text box.
4. Scroll through the list of registered applications, and click on the application you want the extension to be linked (associated) with. If the application doesn't appear in the list, enter the path and name of the application's EXE file in the Associate With text box. For instance, if you want to associate all LTR documents with WordPerfect for Windows, and if the program is stored in the C:\WPWIN directory, you would enter the following in the Associate With box:

   ```
   C:\WPWIN\WPWIN.EXE
   ```

   If you're not sure about the path and EXE file name for your application, you can click on the Browse button to display a list of EXE, PIF, COM, and BAT files stored on your hard disk.
5. Click on the OK button to store the association.

Now, whenever you click on any file in the File Manager that has the extension LTR (in any directory), Windows will load WordPerfect for Windows and open the file that you've selected.

If you drag an LTR file from the File Manager into your Files in Use (or similarly named) group, Windows will create an icon for the file and will allow you to launch both the file and its application simply by clicking on the icon.

Register Non-Microsoft Applications to Simplify the Association Process

When you set up applications to run under Windows, only Microsoft applications designed to run under Windows will be registered. This means only Microsoft's applications will appear in the Associate With list in the Associate dialog box. However, you can add other applications to the Associate With list by using the Registration Editor.

Windows 3.1 is somewhat arrogant in the way it treats associations. By default, the Associate dialog box only lists applications that have been *registered* in the Windows registration database. Microsoft created this tool for Windows 3.1 to help the File Manager recognize file associations and to track the DDE (dynamic data exchange) and OLE (object linking and embedding) capabilities of different Windows applications.

Unfortunately, the Windows Setup program only registers Windows applications created by Microsoft. Windows does allow non-Microsoft applications to register themselves; but any non-Microsoft Windows applications that were designed prior to Windows 3.1 (such as Lotus for Windows and WordPerfect for Windows version 1.0) don't know about the registration editor and therefore can't "sign themselves in."

In creating file associations for non-Microsoft applications, you'll probably find this limitation to be annoying, if not aggravating. For instance, if your word processor is WordPerfect for Windows and you want to associate several file types, you'll have to enter the full path for

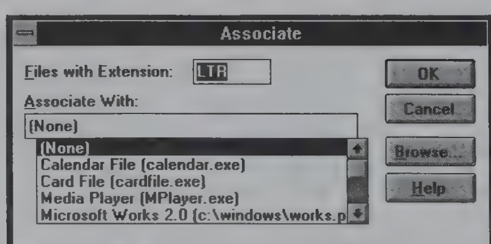


Figure 8.7 Use this dialog box to link all files having a particular extension with an application.

the WordPerfect for Windows EXE file each time you want to associate a new file type.

You can get around this problem by registering an application yourself. It's actually very easy to do, although you won't find the procedure documented anywhere in the Microsoft Windows *User's Guide* or in most books on using Windows 3.1. To illustrate the procedure, we'll register WordPerfect for Windows. But keep in mind that you can follow similar steps to register any application:

1. From either the Program Manager or File Manager, click on File Run to display the Run dialog box.
2. Type **Regedit**, then press **Enter** to display the Registration Info Editor dialog box, shown in Figure 8.8.
3. Click on Edit Add File Type to display the Add File Type dialog box, shown in Figure 8.9.

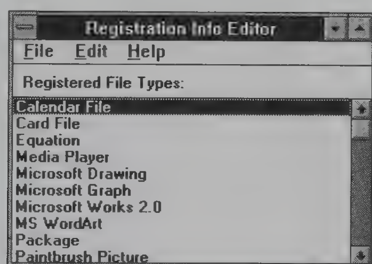


Figure 8.8 This window appears when you start the Registration Editor, and lists all currently registered applications.

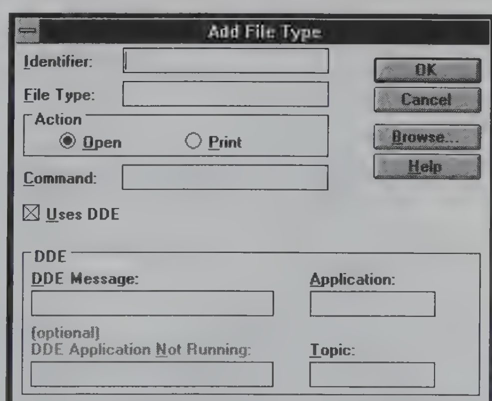


Figure 8.9 Use this dialog box to register an application for use by the File Manager and to support DDE/OLE functions.

This dialog box contains several options and text boxes that you don't need to worry about when you create a new registration entry. You only need to concern yourself with the following entries:

Identifier This is a unique word that the registration database uses to link file types with the application. The identifier can contain up to 63 characters, although you should limit this entry to only a few characters. The space and backslash (\) characters are not permitted in an identifier.

File Type This is the description that will appear in the Associate dialog box in the File Manager.

Command Use this text box to enter the path and executable file name for the application you are registering.

The following steps demonstrate how you might register WordPerfect for Windows.

4. In the Identifier text box, type **WPWIN** or any other identifier that you want to use.
5. In the File Type text box, type **WordPerfect for Windows** or the appropriate description that you want to appear in the Associate dialog box.
6. In the Command text box, type **C:\WPWIN\WPWIN.EXE** or the appropriate path and EXE file name for the actual application you are registering.
7. Click on OK to save the registration information. The new description will now appear in the Registration Info Editor window.
8. Close the Registration Info Editor.

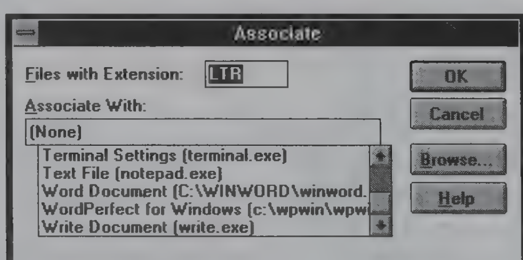


Figure 8.10 This Associate dialog box shows the addition of WordPerfect for Windows, which is now registered.

Now, if we start the File Manager and then click on File Associate, the entry "WordPerfect for Windows (c:\wpwin\wpwin.exe)" will appear in the Associate With list box, as shown in Figure 8.10. To create an association for WordPerfect for Windows, you need only to click on the entry in the Associate With list; you won't have to enter the path for WordPerfect for Windows each time you create a new association.

HOT TIP

Create an Association for a Non-Windows Application

You can associate a file type with most non-Windows applications. You just need to make sure that you associate a document with its appropriate PIF, not with the application's EXE file. For instance, if you want to associate WKS documents with Microsoft Works (for DOS), you would enter the following in the Associate With text box:

```
C:\WINDOWS\WORKS.PIF
```

All PIF files are stored in the WINDOWS directory—unless you have moved them to a different location. If you don't specify the application's PIF when you create the association, Windows will start the application using the _DEFAULT.PIF file, which probably won't contain the ideal settings for the application.

HOT TIP

Register a Non-Windows Application

You can easily register a non-Windows application so that it appears in the Associate With list. Just follow the steps that we've already explained for adding a file type to the registration database. However, remember to enter the path and name of the application's PIF (not its EXE file) in the Command box of the Add File Type dialog box.

Customizing and Managing the Desktop

Have you ever wondered what happens when you tell the Program Manager to save your desktop settings when you exit? Basically, Windows just updates the settings in your GRP (group) files. Your PROGRAM.INI file also gets updated, but this file is largely responsible for identifying the GRP files that are on your system and for storing the size and position of the Program Manager window itself.

Most of the desktop settings, including the size and location of group windows, are stored within the GRP files, so it's a good idea to use these files wisely.

Just about all other changes you make to your Windows environment are stored in the various INI (initialization) files stored in your WINDOWS directory. Most of the INI files are updated when you use the Control Panel to change settings. In a few cases, you need to manipulate INI files directly; but usually you will only need to work from the Control Panel to change settings.

In other words, there are two basic ways to change your desktop environment: by saving GRP settings while using the Save Settings on Exit option, and by using the Control Panel. If you understand these two general concepts, you'll be well on your way to optimizing and customizing your desktop and Windows environment.

In the following sections, we'll provide some specific techniques and tips to manage your GRP files, along with the INI files that are involved in creating your desktop environment.

HOT TIP

Save New Program Manager Settings without Exiting Windows

If you've ever created a new Program Manager (desktop) environment that you wanted to make permanent, you probably know what a pain this can be. Here's the drill. After you've rearranged and adjusted all of your program groups and other startup windows, you have to turn on the Save Settings on the Exit option and then reboot your computer for the new settings to take effect. But then you have to remember to *turn off* the Save Settings on the Exit option when you restart Windows. If you forget to do this, and you make a temporary change to your desktop, Windows will save the new arrangement the next time you exit Windows.

It's enough to make you wonder sometimes why people say this program is so easy to use. In fact, there *is* a much easier way to save Program Manager settings:

1. Arrange program groups, program-item icons, startup applications, and other desktop objects to appear the way you want them to look permanently whenever you start Windows.
2. Hold down **Shift**, then click on File Exit and release the Shift key.

And *that's it.* Your new desktop setup is permanently saved to disk. Windows won't display any message to this effect, but you'll know the process worked because Windows is still displayed. In other words, you've saved the new settings without having to exit the Program Manager.

You can use this identical procedure to save changes to the File Manager without having to exit the File Manager.

Warning: Even if you use this technique, make sure the Save Settings on Exit option is turned off before you exit Windows; otherwise, your new "permanent" settings will be overwritten the next time you change your desktop appearance and exit Windows.

Making a Backup Copy of Your Windows Environment

As you add new applications and customization features to Windows, you increase the likelihood that Windows will become unstable. You should make a backup copy of all INI and GRP files so that you can easily recover from problems created by newly installed applications.

The setup programs for different Windows applications and for "Windows-aware" applications and utilities often directly modify your CONFIG.SYS, AUTOEXEC.BAT, and other INI files. These changes can lead to memory conflicts or can obscure Windows' ability to determine how to load files for different applications.

If Windows runs reliably for you now, don't wait another second: immediately make a backup copy of all .INI and .GRP files. Then make backups for your CONFIG.SYS and AUTOEXEC.BAT files.

In this way, you can restore your complete Windows environment if a newly installed application or utility has created problems for Windows. Troubleshooting the problem might take hours or even days. But if you have a backup copy of the files that drive the Windows environment, you can use these files to restore Windows immediately to its previous "life." You can have Windows up and running within a few minutes. You won't have to wait until you've solved the problem that caused Windows to fail or to behave erratically. The following steps are an easy way to back up your Windows environment:

1. Open the File Manager and create a BACKUP directory underneath the WINDOWS directory.

2. In the directory tree, click on the WINDOWS directory icon, then click on View By File Type.
3. In the Name box, type ***.GRP**, then press **Enter** to display all GRP files.
4. Make sure the first file name is highlighted in the directory, then scroll to the last file name in the directory. Press **Shift** and click on the last file name to highlight all GRP files.
5. Hold down **Ctrl**, then drag the files onto the BACKUP directory icon in the directory tree to copy all GRP files to the BACKUP directory.
6. Modify and repeat Steps 2 through 5 to view and copy all INI files into the BACKUP directory.
7. In the directory tree, click on the root icon (the top c:\ icon).
8. Click on View By File Type, type *.*, then press **Enter** to select all files for viewing.
9. Copy the files AUTOEXEC.BAT and CONFIG.SYS into the WINDOWS\BACKUP directory.

   If your available hard disk space is low, you might want to back up these files onto a floppy diskette. (All of the files should fit on a double-density 5-1/4" floppy.)

If you need to restore your Windows environment, you should first rename your current INI and GRP files, then copy the backup versions of the INI and GRP files to the WINDOWS directory and reboot your system.

If you still get error messages in Windows, you can feel confident that your INI and GRP files are not to blame. You can restore the more current versions of these files with some confidence.

Now rename your current AUTOEXEC.BAT and CONFIG.SYS files, then copy the backup versions of these files into your root directory and reboot your system. If you do not receive any error messages in Windows, you'll know that the modified version of either AUTOEXEC.BAT or CONFIG.SYS is to blame. If you still receive error messages, you should refer to Chapters 16 and 17, which describe specific hardware and software problems that can prevent Windows from running successfully.

HOT TIP

Compress Your Backed Up Windows Environment Files

If you have a file-compression program, like PKZIP, compress all files in your WINDOWS\BACKUP directory (or whatever directory you've used to store your Windows environment files) into a single compressed file. Then, if you need to recover your Windows environment, you can just uncompress the contents of your backup directory into your WINDOWS directory, then reboot your system. (We explain how to use PKZIP and PKUNZIP in Chapter 9.)

Of course, under this approach, your AUTOEXEC.BAT and CONFIG.SYS backup files will get copied to your WINDOWS directory rather than to the root directory of your hard drive. But this might be insignificant if your Windows problem stems from a change made to an INI or GRP file. And even if you do need to restore your AUTOEXEC.BAT and/or CONFIG.SYS files, you can do so easily by copying the backup versions of these files to the root directory.

Preventing Other Users from Tampering with Your Desktop Setup

If other people in your office also use your computer, you might want to prevent them from changing your desktop settings. To protect your desktop, simply write-protect all of your GRP files.

Remember, the GRP (group) files stored in your WINDOWS directory contain most of the settings that Windows uses to create your desktop. So, you can safeguard your desktop environment by using the File Manager to make each GRP file read-only. Here's an easy way to write-protect all GRP files:

1. Start the File Manager and select the WINDOWS directory icon in the directory tree.
2. Choose File Select Files to display the Select Files dialog box.
3. In the File(s) text box, type ***.GRP**, then press **Enter** and click on the Close button.
4. Click on File Properties to display the Properties dialog box.
5. Turn on the Read Only check box, then click on OK. Your GRP files are now write-protected. If you have one or more groups that you've created for personal use, you might just want to protect these groups, rather than all GRP files.

In any case, write-protected GRP files cannot be altered unless the Read Only attribute is turned off. This means that the positions of the groups can't be changed, and icons can't be added to or deleted from the protected groups. You also can't drag and drop files from the File Manager into a write-protected group.

Note: If you want to add a new icon to or delete an icon from a protected group, you'll need to turn off the Read Only attribute first.

HOT TIP

"Quick Maximize" Any Window by Double-Clicking Anywhere on Its Title Bar

You don't always need to position the cursor on the Maximize button to maximize the current window. If you double-click on the title bar, the window will be maximized. If you double-click on the title bar again, the window will be restored to its previous size.

Use the StartUp Group Sensibly

Keep in mind that each application you add to your StartUp group has to be loaded into memory every time Windows boots. If you overload your StartUp group, you increase the time required to start Windows, and you increase the amount of memory that is occupied before you even begin your Windows session.

Limit your StartUp group to applications that you almost always use during every Windows session. This is just common sense, of course. But we've seen more than a few people abuse the StartUp group just because "it's there," waiting to be trampled.

Some users drag half a dozen or more applications into the StartUp group and then (for each application) click on the Run Minimized check box from the Properties dialog box, believing that a minimized application isn't fully loaded into memory. That's wrong! Every application in your StartUp group is occupying memory, regardless of whether it starts in a window or as a minimized icon.

Unless you have plenty of extended memory to spare, you just slow down your system by loading applications unnecessarily when Windows boots, especially if you load multiple DOS applications. Why? With all those applications occupying what limited extended memory you *do* have, new applications that you start might have to be swapped to your temporary or permanent swap file. Applications run much slower when they are swapped to disk rather than loaded entirely in RAM.

HOT TIP

Drag Your Working Document into the StartUp Group Before You Exit

Suppose you've been working on a complex Excel worksheet for several hours today. At the end of the day, you save the spreadsheet and quit Excel. Tomorrow morning, you plan to take up where you left off—working in this same spreadsheet.

If you closed the spreadsheet, closed Excel, and exited Windows in the manner we've just indicated, you'll probably have to restart Excel after Windows boots and then load the worksheet file. Although this process really doesn't take a lot of time, you can still save yourself at least a little time by dragging the document into the StartUp group before you exit Windows.

By dragging your Excel document into the StartUp group, you are instructing Windows to automatically load Excel and the spreadsheet document (or whatever application and document you've been working on) the next time Windows starts up. And you don't need to save your desktop settings for this change to take effect. Whenever you add an icon to a group, Windows immediately updates the GRP file. So, if you drag a document to your StartUp group, Windows automatically updates the STARTUP.GRP file; your document will open automatically the next time you start Windows.

This procedure is especially useful if you plan on working on a particular document for several days—such as a complex report or spreadsheet. For it to work, of course, the document must have an extension that has already been associated with a registered application. Just remember to remove the document icon from the StartUp group when you've finished working on the document.

Making the Clock a Permanent Part of Your Desktop

Most users know that the Windows Clock will continue to display the time—in the bottom of the Program Manager—when the Clock application is minimized. However, you probably aren't aware of how easily the Clock can be customized to display the time from within any application, at all times.

Unfortunately, Microsoft has made the procedure for customizing the Clock somewhat awkward—which means that most users don't know how to use the Clock window effectively. Actually, the inner workings of the Clock are complex only because its many features are, for the most part, undocumented. But if you learn how the Clock works, customizing it is simple.

Figure 8.11 shows a digital Clock display that appears within an empty area of the title bar of any active window. (Actually, this is *usually* true, but not always. If you're not running an application maximized, the Clock might even appear outside of the application. In any event, it's up to you to find the most unobtrusive Clock location on your desktop.)

So how did we manage this trick? We just used these simple steps:

1. Drag the Clock icon from the Accessories group to the StartUp group, then double-click on the Clock icon to run the Clock.
2. Click on Settings, then select Digital and deselect the Date option.
3. Click on the Minimize button to reduce the Clock to an icon.
4. Click on the Clock icon to display its Control menu, then select the Always on Top option to ensure that the Clock window always displays on top of the currently active window.
5. If necessary, click outside of the Control menu to close it.
6. Double-click on the Clock icon to display the Clock in a window.
7. Click on Settings, then select the No Title option. When you turn on No Title, the title bar is removed from the clock display.
8. Move the clock to an unoccupied part of the currently active window's title bar, then size the Clock until it fits within the title bar, as shown in Figure 8.11.
9. In the Program Manager, hold down **Shift**, then click on File Exit to save the new Clock settings.

Later, if you need to modify the Clock settings, just double-click on the Clock icon and the title bar will appear. Then, click on the Maximize button to display the Settings menu bar option and select the appropriate options from the menu. Unfortunately, if you want to save any new Clock settings, you'll need to follow the steps above to remove the title bar, restore the clock to its desired size and position, then save the settings. (Remember, though, you set the time itself from within the Control Panel, not from within the Clock.)

Warning: If you frequently use the Task List to tile multiple applications—such as the File Manager and the Program Manager—the Clock setup we've used when created can behave like an ill-tempered

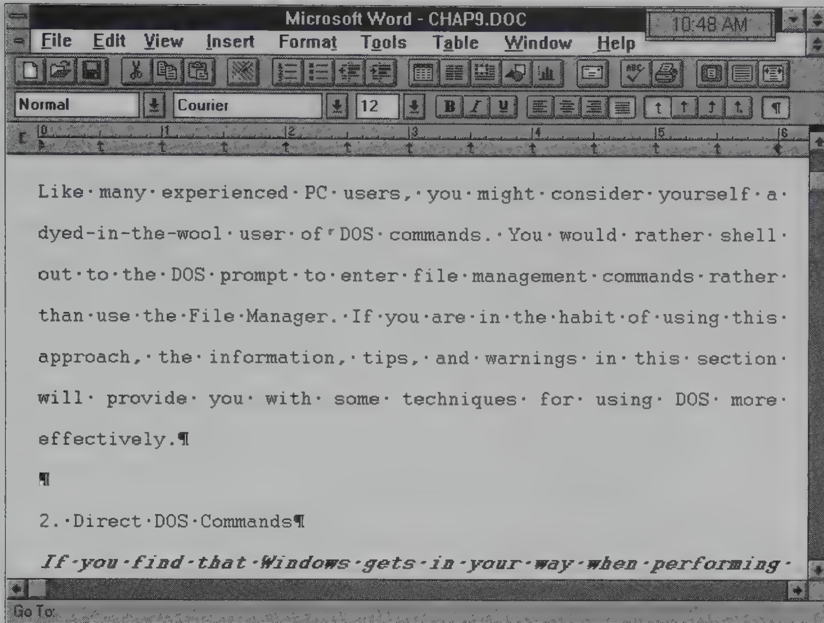


Figure 8.11 In this Word window, the Clock appears in the title bar.

gremlin. The problem is that, even though we've made the Clock display as small as possible, the Clock is not minimized.

So, when you use the Task List to tile open applications, the Clock will be tiled as well. You can't easily return the Clock to its desired size and position without performing several steps to remove the Clock's title line again and to resize the clock manually. There's no easy way out of this problem. If you use the Task List to tile only on rare occasions, you can just minimize the Clock for the remainder of your Windows session—if you don't feel like resizing and repositioning it. But if you use the Task List frequently to tile applications, you're probably better off not displaying the Clock "Always on Top."

Using the Windows Calendar to Set Alarms

Although the Windows Calendar works great for noting appointments, most users prefer to rely on printed schedulers or planners. But don't overlook the Calendar's ability to set "reminder" alarms throughout the day.

We've met a few people who actually use on-line schedulers, like the Windows Calendar. But you probably don't count yourself among this minority. After all, on-line planners like the Windows Calendar just require you to duplicate what you write into your printed planner.

Suppose you're called to a meeting by, say, the vice president of Sales/Marketing. During the meeting, Francine—the VP—asks you to form a task force that will meet next Tuesday at 10:00 A.M. You can't realistically ask Francine to excuse you for a few minutes while you jog down the hall to your PC and enter the meeting information into your Windows Calendar. No, you're going to pencil the meeting into the planner book that you carry with you from meeting to meeting. The Windows Calendar gets ignored. But the Calendar does deserve some of your attention.

Specifically, you can use the Calendar to set alarms that remind you of appointments, phone calls you need to make, or other events that you have to attend during the day. When you start the Calendar, Windows automatically displays the Calendar for the current day. So you can easily and quickly set an alarm (a reminder). Suppose you've promised to call Roswell Snyder at 2:00 P.M. You can remind yourself of this by following these simple steps:

1. In the Accessories window, open the Calendar.
2. Position the cursor on the time (2:00 P.M., in this example) for which you want to sound an alarm.
3. If you want, enter a reminder message.
4. Click on Alarm, then click on Set.

Windows will now sound an alarm at the appointed time to remind you of the event that you've noted in the Calendar. If you entered a description for the alarm, you can click on the Calendar icon to view the reminder description.

The Calendar application is fairly large, requiring more than 90K to load. For this reason, we don't recommend adding the Calendar to your StartUp group unless you use the Calendar routinely—almost every day.

Note: An alarm will not sound if it is stored in a closed Calendar file; the Calendar file must be open at the time the alarm is supposed to sound. So, if you want to set alarms for a date other than today, you'll need to name and assign a Calendar file to your StartUp Group to ensure that alarms sound on the desired date.

Porting Your Colors to a Different System

If you've created a custom color scheme for your desktop, and you then have to move to a new computer, you can usually take your color scheme with you to your new system.

Windows users seem to become easily fascinated by the array of colors that can be assigned to different parts of the desktop. Unless you're color blind, you're probably no exception. Although Windows provides 23 preset color schemes—from the sober yet eye-pleasing Windows Default to the outrageously loud fluorescent scheme and the punkish Black Leather Jacket—many users prefer to create their own custom color schemes.

If you've created and saved one or more unique color schemes for your system, you might have spent up to an hour or even longer experimenting with different color combinations to get the scheme that you liked best. You might even have created a few custom colors for different parts of the desktop.

Now, if you buy a new computer or if you have to move to a different computer elsewhere in your office, you probably don't want to spend still more time trying to re-create your desktop masterpiece.

If your current system uses the same or a compatible display driver as your new computer system, you can easily copy your custom color scheme (or multiple custom color schemes if you've created more than one), along with any custom-created colors, to the new computer system.

Windows stores color settings in two places: in your CONTROL.INI file and in your WIN.INI file. CONTROL.INI contains many of your Control Panel settings, including the color settings you've created using the Colors dialog box. Figure 8.12 illustrates the partial contents of a typical CONTROL.INI file. The sections involved in setting and managing colors are [current], [color schemes], and [Custom Colors].

The [current] section contains the name of the color scheme currently in use (Arizona in Figure 8.12).

The [color schemes] section stores the color settings for all 23 preset Windows color schemes and for any custom color schemes you've created. Figure 8.12 shows settings for two custom schemes—roncolr and fancyclr. The other schemes—Arizona, Black Leather Jacket, and so on—are the settings for the Windows-supplied schemes.

Each line in the color schemes section contains a set of hexadecimal codes for the 21 desktop components—Desktop, Application Workspace, Menu Bar, Window Background, and so on. Each hex code is made up of three parts (reading from left to right), with two hex characters for each part: two characters for the intensity of blue in each picture element (pixel) in the desktop component, two characters for the intensity of green in each pixel of the component, and two characters for the intensity of red in each pixel of the component. In other words, read the hex code (for example, FFFF80) as though a slash or dash has been inserted after each two characters, like this: FF/FF/80 or FF-FF-80.

```
Notepad - CONTROL.INI
File  Edit  Search  Help
[current]
color schemes=Arizona

[color schemes]
roncolr=C0C0C0,E8FFFF,FFFFFF,0,FFFFFF,0,80,FFFFFF,FFFFFF,0,C0C0C0,
fancyclr=8080FF,FFFF,FFFFFF,0,808000,0,80,FFFFFF,FFFFFF,0,C0C0C0,E
Arizona=804000,FFFFFF,FFFFFF,0,FFFFFF,0,808040,C0C0C0,FFFFFF,4080F
Black Leather Jacket=0,C0C0C0,FFFFFF,0,C0C0C0,0,800040,808080,FFFF
Bordeaux=400080,C0C0C0,FFFFFF,0,FFFFFF,0,800080,C0C0C0,FFFFFF,FF00
Cinnamon=404080,C0C0C0,FFFFFF,0,FFFFFF,0,80,C0C0C0,FFFFFF,80,C0C0C
Designer=7C7C3F,C0C0C0,FFFFFF,0,FFFFFF,0,808000,C0C0C0,FFFFFF,C0C0
Emerald City=404000,C0C0C0,FFFFFF,0,C0C0C0,0,408000,808040,FFFFFF,
Fluorescent=0,FFFFFF,FFFFFF,0,FF00,0,FF00FF,C0C0C0,0,FF80,C0C0C0,0
Hotdog Stand=FFFF,FFFF,FF,FFFFFF,FFFFFF,0,0,FF,FFFFFF,FF,FF,0,C0C0
LCD Default Screen Settings=808080,C0C0C0,C0C0C0,0,C0C0C0,0,800000
LCD Reversed - Dark=0,80,80,FFFFFF,8080,0,8080,800000,0,8080,80000
LCD Reversed - Light=800000,FFFFFF,FFFFFF,0,FFFFFF,0,808040,FFFFFF
Mahogany=404040,C0C0C0,FFFFFF,0,FFFFFF,0,40,C0C0C0,FFFFFF,C0C0C0,C
Monochrome=C0C0C0,FFFFFF,FFFFFF,0,FFFFFF,0,0,C0C0C0,FFFFFF,C0C0C0,
Ocean=808000,408000,FFFFFF,0,FFFFFF,0,804000,C0C0C0,FFFFFF,C0C0C0,
Pastel=C0FF82,80FFFF,FFFFFF,0,FFFFFF,0,FFFF80,FFFFFF,0,C080FF,FFFF
Patchwork=9544BB,C1FBFA,FFFFFF,0,FFFFFF,0,FFFF80,FFFFFF,0,64B14E,F
Plasma Power Saver=0,FF0000,0,FFFFFF,FF00FF,0,800000,C0C0C0,0,80,F
Rugby=C0C0C0,80FFFF,FFFFFF,0,FFFFFF,0,800000,FFFFFF,FFFFFF,80,FFFF
The Blues=804000,C0C0C0,FFFFFF,0,FFFFFF,0,800000,C0C0C0,FFFFFF,C0C
Tweed=6A619E,C0C0C0,FFFFFF,0,FFFFFF,0,408080,C0C0C0,FFFFFF,404080,
```

Figure 8.12 This portion of CONTROL.INI shows the contents of the [current] and [color schemes] sections, including the addition of two custom color schemes.

Each two-character portion of the hex code represents a decimal value from 0 (minimum intensity) to 255 (maximum intensity). Remember that you can use the Windows Scientific Calculator to convert hex values to decimal values.

The relative intensities of the three colors can be combined to produce more than 16 million different colors (256 x 256 x 256 = 16,777,216).

For instance, the hex value FFFF80 (the active title bar color for the Pastel and Patchwork schemes) can be separated and interpreted this way:

FF Hex = 255 Decimal (high-intensity blue)
FF Hex = 255 Decimal (high-intensity green)
80 Hex = 128 Decimal (medium-intensity red)

The precise color that these three intensity combinations create will depend on the monitor type and the resolution and color display capabilities of the display driver. On a VGA monitor using the standard VGA display driver (VGA.DRV), this color looks like a light aqua.

The [Custom Colors] section stores the intensity settings for up to 16 colors that you create using the Custom Color Selector (which you access by clicking on the Define Custom Colors button in the Colors dialog box). Note in Figure 8.13 that we've created these two custom colors:

ColorA=FFAEAE (255 blue, 174 green, and 174 red)
ColorB=E945D5 (233 blue, 69 green, and 213 red)

When you use the Control Panel to specify a particular color scheme, Windows enters the name of the scheme in the [current] section of CONTROL.INI, and then looks in the [color schemes] section for the set of hex codes for the current scheme. These settings are then converted to their decimal values, which are in turn copied into the [colors] section of the WIN.INI file.

Figure 8.14 shows how the WIN.INI [colors] section looks after we select Patchwork as the current color scheme. If you study this section closely, you'll note that the three decimal values for each Windows component are displayed—from left to right—in *red, green, blue* order. For instance, notice the following line:

```
ActiveTitle=128 255 255
```

But hold on. What's going on here? You'll recall that the hex code for this setting in CONTROL.INI appeared in this order: FF (255 blue), FF (255 green), 80 (128 red). In other words, the color intensity values in WIN.INI appear in red, green, blue order, while the same values in CONTROL.INI appear in the hex code in *blue, green, red* (the reverse) order.

Actually, this piece of trivia plus 75 cents will get you a cup of coffee and not much else. Just remember that the key to porting colors from one

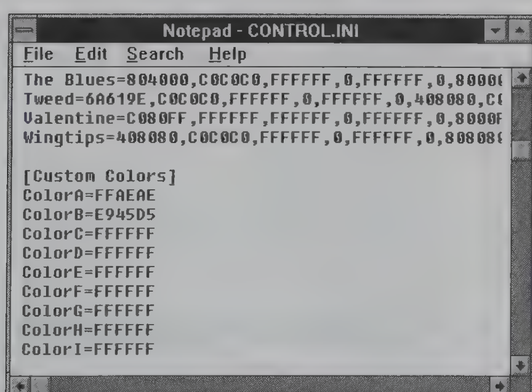


Figure 8.13 This portion of CONTROL.INI shows that two custom colors have been created; they have the hex codes FFAEAE and E945D5.

system to another lies in copying the color sections from your current system's CONTROL.INI into the CONTROL.INI file for your new computer system.

Warning: Always back up CONTROL.INI before you change it.

In fact, all you have to do is follow these steps:

1. Start the Notepad and open CONTROL.INI, which is stored in the WINDOWS directory.
2. Highlight all of the entries, including the section headers, for the [current], [color schemes], and [Custom Colors] sections in CONTROL.INI.
3. Choose Edit Copy to copy the highlighted entries to the Clipboard.
4. Choose File New to create a new Notepad file, then choose Edit Paste to copy the color settings into the new file.
5. Save the file under a unique name, such as COLORS.TXT.
6. Copy the new file to a floppy disk (or to a network directory if both your old and new computers are linked to the same network server).
7. Bring the floppy disk with you to your new computer and start Windows, then start the Notepad.

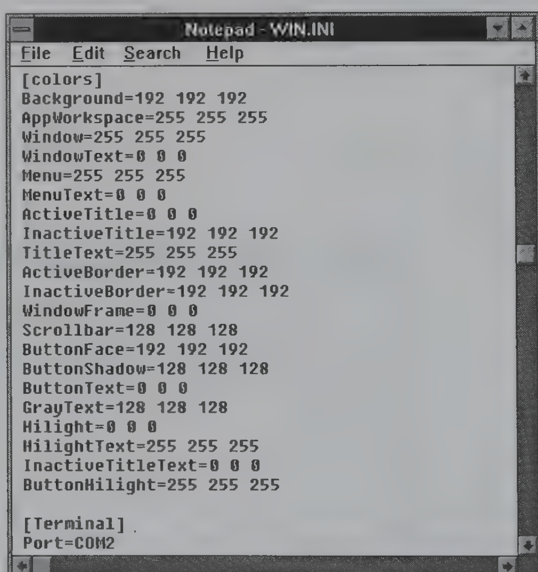


Figure 8.14 The [colors] section of WIN.INI contains the intensity values for the currently selected color scheme, for all 21 desktop components.

8. Open the file that contains your custom colors (such as COLORS.TXT, stored on the floppy disk).
9. Copy the contents of the file to the Clipboard, then open CONTROL.INI on your new computer system.
10. Highlight the [current], [color schemes], and [Custom Colors] sections of CONTROL.INI, then press Delete to remove the current settings. (Don't use Edit Cut.)
11. Choose Edit Paste to paste your color schemes and custom colors into the CONTROL.INI file.

Now reboot your computer so that the new CONTROL.INI file will take effect. Then use the Colors dialog box—from the Control Panel—to select the desired color scheme; you'll notice that your custom color schemes and custom colors will be available.

When you select a color scheme, Control Panel updates the color settings in your WIN.INI file. Your custom color scheme is then permanently installed on the new system, unless you later use the Control Panel to select a different color scheme.

Again, you might notice that your custom color scheme looks a little different on the new monitor. But the difference will be slight if the two systems use the same display driver. If the systems use different display drivers, the colors will probably look *a lot* different. For this reason, porting your colors from one system to another usually only works when the two systems use the same display driver.

Never Try to Port Your Entire Desktop

You could conceivably port your entire desktop environment from one computer to another by copying all INI and GRP files from one system to the other, but in reality this almost never works, and can actually be dangerous.

For you to port your entire desktop safely to a different computer, the new computer system would have to share the same CPU and use the same peripherals, the same device drivers—including the mouse and printer drivers, the same memory configuration, and the same disk drive types and disk partitions.

Actually, we're just getting warmed up. The Windows Setup program examines and creates settings for thousands of hardware and software variables. The bottom line is that if you're thinking about moving your Windows environment from one computer to another: *Don't do it.*

If Windows loads at all on the new system, it's a sure bet you'll see multiple error messages on startup, and you'll be lucky if you can operate for more than a few minutes without your system locking. Always let the Windows Setup program run its diagnostics on a new system, even if this means you'll later have to set up new program groups, icons, printers, and so on.

Customizing the File Manager

The File Manager is probably the most beleaguered and maligned of all the Windows applications. In Windows 3.0, the File Manager's poor reputation was probably deserved. The Windows 3.0 File Manager was not only slow, it was difficult to use. But the File Manager in Windows 3.1 is a much-revamped version of its predecessor. The Windows 3.1 File Manager is not only fast, it can perform file management feats that up to this point have been available only with alternative file management software.

We won't cover the File Manager's basic features here. But there are some little-known guidelines that can help you use the File Manager efficiently, plus a few tips on configuring the File Manager for your custom needs. We'll cover these techniques and tips in the following sections.

Creating Directory-Window Icons for Frequently Used Directories

If you find yourself frequently using the File Manager to display a particular directory, you should create a permanent minimized icon for the directory window. This technique allows you to display the contents of the directory without having to search the directory tree to locate it.

Windows allows you to open a directory window that displays only the files in a particular directory; the directory tree will not appear. You can combine this feature with a few other basic File Manager techniques to create a permanent set of frequently used directory-window icons.

This technique has several benefits. First and most obvious is the fact that the icons are always visible. You can easily display the contents of a directory simply by double-clicking on the icon; you don't need to scroll through the directory tree to locate the directory that you want to view.

Second, if your hard drive is partitioned into two or more logical drives, you've probably grown tired of having to click on the drive icon every time you want to switch from, say, drive C to drive D. To get around this step, you can create minimized directory icons for frequently used directories located in different drives. Then, when you want to open that directory, you simply double-click on the icon. You don't have to switch to the desired drive and search the directory tree for the directory that you want to view. You can use this same approach to create directory icons for different directories stored on network drives.

A third benefit is the ability to immediately display the contents of a directory that is stored several levels deep in the directory tree. With this approach, you can avoid expanding different levels and searching through different branches of the tree.

Take a look at Figure 8.15, which illustrates all three of these benefits. In this File Manager arrangement, we've created five minimized directory-window icons.

The first two icons display the contents of the WINDOWS and PCTOOLS directories stored on drive C. Since the File Manager sorts the directory tree alphabetically, these two directories don't appear in the tree that displays when we first start the File Manager. So, for instance, if we want to display the contents of the WINDOWS directory, we just need to double-click on the icon that appears at the bottom of the File Manager window; we don't have to scroll through the tree to find the WINDOWS directory.

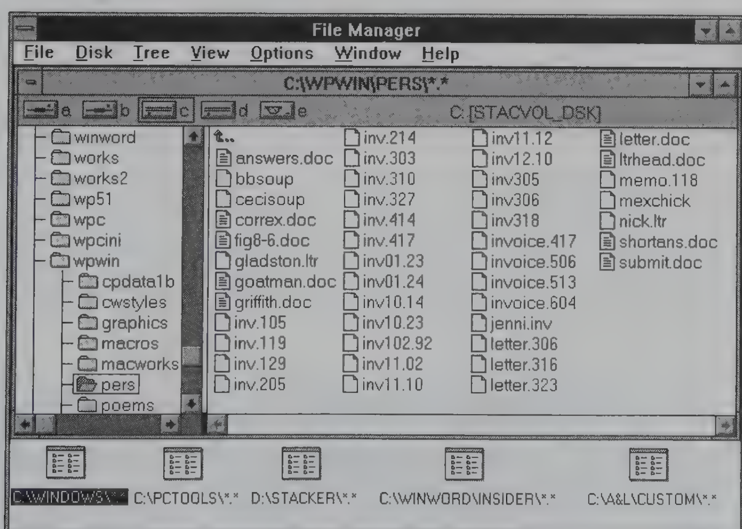


Figure 8.15 This File Manager display shows five permanent icons created for five frequently used directories.

The third icon displays the contents of the STACKER directory stored on drive D. By double-clicking on this icon, we can display the contents of the STACKER directory without having to switch to the D drive.

The fourth and fifth icons display the contents of directories that are two levels deep under the root directory. So, for example, by double-clicking on the C:\WINWORD\INSIDER icon, we can display the contents of this subdirectory instantly; we don't have to scroll down to the WINWORD directory, expand it to display the INSIDER subdirectory, then double-click on the INSIDER directory icon to display its directory window. We've replaced that tedious process with two clicks of the mouse.

To create a directory-window icon for a frequently used directory, follow these steps:

1. Start the File Manager.
2. In the directory tree, locate the directory that you want to create a minimized directory-window icon for.
3. Hold down **Shift**, then double-click on the directory to display the contents of the directory in a separate directory window.
4. In the new directory window, click on the Minimize button to reduce the window to an icon.
5. Hold down **Shift**, then click on File Exit to save the new File Manager setting.

As you create minimized icons, you'll notice that *instance numbers* appear at the end of the directory name. For example, if you minimize the DOS directory, you might see the icon name C:\DOS*.* 1 under the icon. The number 1 indicates that this is the first instance of the directory to have been opened. If, say, the WINDOWS directory window is already displayed, and you then open and minimize another version of the WINDOWS directory, you'll see the icon name C:\WINDOWS*.* 2.

Don't be concerned about instance numbers. They only appear in the current File Manager session. If you exit File Manager and then restart it, you'll notice that the instance numbers have disappeared from the icon names.

HOT TIP

Change the Relative Size of the Directory Tree

By default, Windows splits the directory tree and directory window portions of the File Manager in half. With a font size of 11 points, this means you have room for almost one dozen directory levels within the directory tree. You'll probably never have anywhere near this number of directory levels. (In fact, if you have more than four or five directory levels, you should consider reorganizing your directory structure.)

It's a good idea to resize the window split, so that the directory tree is relatively small, while the directory-contents side is relatively large. For instance, in Figure 8.15, notice that our directory-contents portion is four times wider than the directory-tree side of the File Manager. This allows us to display three levels of the directory at a time, which is fine because very few of our directories have more than three levels.

On the other hand, we've made it possible to display more files at a time in the directory-contents side. This expanded view is useful because the contents of a directory often contains several dozen files.

To change the size of the File Manager window split, follow these steps:

1. Position the pointer on the vertical line that separates the two window halves. A new pointer should appear, displaying a double-headed arrow.
2. Drag the vertical line to the right or left until the two sides of the window are split in the proportion that you want.
3. Hold down **Shift**, then click on File Exit to save the new File Manager settings.

Selecting Files for a Directory Icon

You can select a subset of files that you want to appear in a particular directory-window icon. This technique is very useful if you regularly need to view files with a particular extension.

If you want to create a permanent directory icon that lists only those files that have a particular extension, use the File Manager's View By File Type commands. Take a look at Figure 8.16. Here, we've created a directory-window icon named C:\WINDOWS*.INI. Since all INI files are associated with the Notepad application, we can view the contents of any INI file quickly, simply by opening the C:\WINDOWS*.INI window and double-clicking on the name of the INI file that we want to view.

Here's how we created this icon:

1. Open the File Manager and display the directory tree.
2. Hold down **Shift**, then double-click on the WINDOWS directory icon

in the directory tree to create a separate window for the contents of the WINDOWS directory.

3. Choose View By File Type.
4. In the Name text box, type ***.INI**, then press **Enter** to display all INI files stored in the WINDOWS directory.
5. In the C:\WINDOWS*.INI directory window, click on the minimize button to create an icon for the window.
6. Position the new icon where you want it to appear permanently in the File Manager window, hold down **Shift**, then choose File Exit to save the new icon.

ON DISK

Using DirNotes

The *Windows Insider Disks* include a utility called DirNotes, which provides a File Manager-like display that lets you add up to a 38-character description for any file. DirNotes is not a replacement for the File Manager, but can be useful if you want some extra help identifying files. DirNotes is stored in the DIRNOTES directory of the *Windows Insider Disks.*

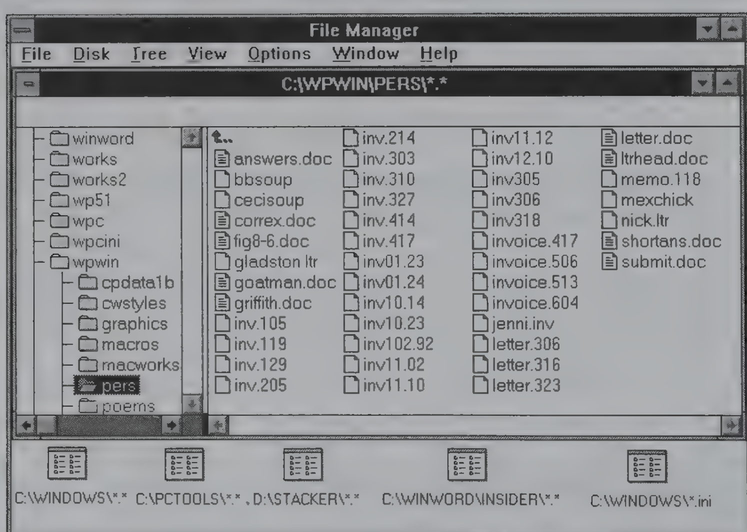


Figure 8.16 Notice the C:\WINDOWS*.INI icon that appears in the bottom right of this File Manager setup; C:\WINDOWS*.INI, when restored, will display all .INI files stored in the WINDOWS directory.

Sizing the File Manager to Facilitate Drag and Drop

If you use drag-and-drop techniques regularly to create program-item or document icons, or to drag files to the Print Manager, consider arranging your File Manager so that it can be viewed side by side with the Program Manager.

Figure 8.17 provides a good example of placing your File Manager and Program Manager side by side. In this File Manager setup, we've sized the File Manager window so that it occupies less than half the desktop vertically. As you can see, we've aligned all our directory-window icons vertically along the right side of the File Manager window so that they're always visible. When we created each directory window, we sized it to overlap the default directory tree/directory window. In this way, each directory window can be viewed within the File Manager space that we've created.

Figure 8.18 shows how easily we can use this File Manager setup to drag an application or a document into the Program Manager. In this Figure, we've first tiled the File Manager and Program Manager. Note that tiling hasn't really disrupted our default File Manager setup, because we arranged our File Manager specifically to support our drag-and-drop activities. Now we only need to open the C:\WINWORD\INSIDER *.* directory icon (which is shown already open in Figure 8.18), then drag the file we want into the Files In Use group window in the Program Manager. The entire process takes about five seconds to complete.

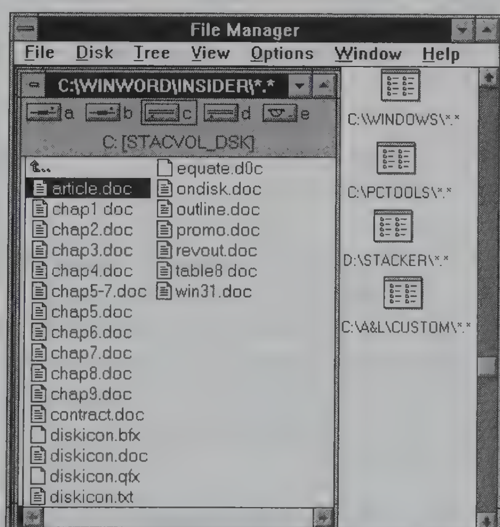


Figure 8.17 This File Manager setup is useful if you frequently use the Task List to tile and move files between the File Manager and Program Manager.

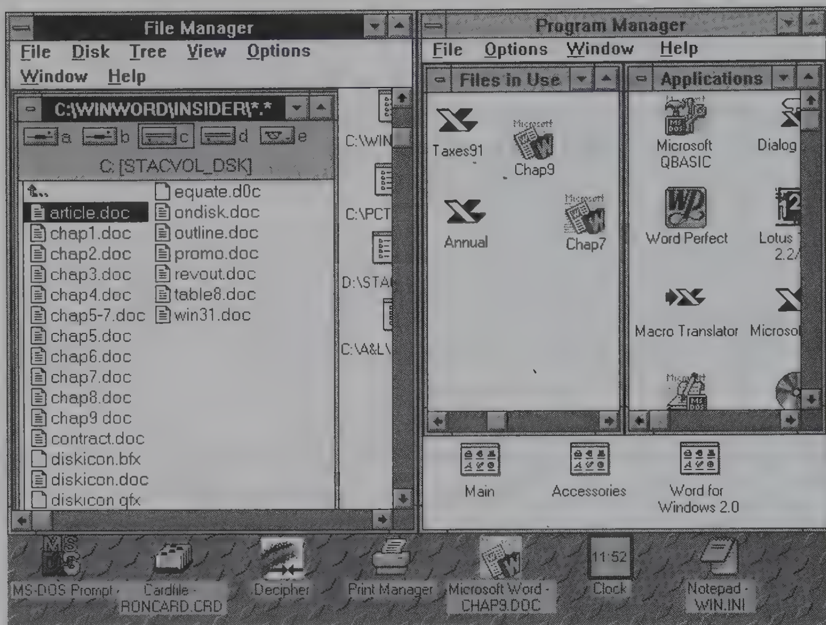


Figure 8.18 When File Manager and Program Manager are tiled under this setup, the current directory appears, while all other directory icons are still visible for selection. Files can be easily copied from any frequently used directory into a Program Manager group.

Using the File Manager's File Selection Options Effectively

Even the most experienced Windows 3.1 users often find it difficult to determine whether to use the View By File Type, the File Select, and the File Search options to display selected files. You, too, probably can benefit from a set of criteria for using these seemingly similar features.

We mentioned earlier that most of the features of the File Manager are self-explanatory or are fairly easy to understand and use. One exception lies in the three File Manager options for selecting files, all of which may seem to serve the same purpose. However, each of these features has its own benefits. If you want to work with the File Manager in the most efficient manner possible, you'll want to know when and when not to use these three options.

Viewing Files by Extension Name If you want to view only those files in a given directory that have a specific extension, such as GRP, DOC, or

INI, use View By File Type. Specifically, display the desired directory window, then choose View By File Type and specify the extension in the Name text box. The By File Type dialog box lets you further filter your selection by limiting the display to directories, programs, documents, and other files. Usually, though, you'll want to keep all four File Type check boxes selected. An example might be to display all INI files in the WINDOWS directory.

Note: After you've finished viewing the files by file type, you should choose File By File Type, then enter *.* in the Name text box. Otherwise, the File Manager will only display files of the indicated type when you switch to other directories.

If you want to locate one or more files that have a particular extension, but you're not sure where the files are stored, use the Search command from the File menu. The Search command allows you to search all subdirectories in the directory tree, starting with the currently selected point in the tree.

To use this search technique, select the directory in the directory tree where you want to begin the search. (Often, this will be the root directory of the current drive, such as the C:\ directory.) Then, choose File Search to display the Search dialog box. In the Search For text box, enter either the name of the file or the extension for files that you want to locate.

Although the Search dialog box also allows you to change the starting location for the search, and choose whether you want to search all subdirectories, you'll normally want to use the defaults. If you don't want to begin the search from the designated location or if you don't need to search subdirectories, you probably should have used the View By File Type commands instead.

Preparing to Copy, Delete, or Move all Files That Share an Extension If you want to copy, delete, or move all files in a directory that share an extension, use the Select Files command from the File menu. Select Files will highlight all files in the current directory for the extension that you specify in the Select Files dialog box.

This feature is more effective than using the File Search commands, because the selection is automatically limited to the currently displayed directory. After you've selected the appropriate files, you can delete them, or you can drag them elsewhere to copy or move them.

To select files in this way, display the directory that contains the files you want to select, then choose Files Select. When the Select Files dialog box appears, type the appropriate wildcard(s) and extension (such as *.DOC) in the File(s) text box, then click on the Select button. You'll then need to choose the Close button to remove the Select Files dialog box from view. At this point, you can easily view, delete, copy, or move the selected files.

HOT TIP

In Windows 3.1, You Can Drag and Drop a File to the Print Manager

This tip isn't really a secret, but we've talked to many Windows users who don't know they can print a file by dragging its document icon from the File Manager into the Print Manager, or onto the minimized Print Manager icon. This procedure is useful chiefly if you are already printing other documents, because Windows will have automatically started and minimized the Print Manager. If the Print Manager isn't open, drag-and-drop isn't really a time-saver, because you can simply select a document, and then click on File Print from the File Manager menu.

In any case, to use this procedure, make sure the Print Manager is open (either in a window or as a minimized icon), and make sure the document you are printing is associated with a Windows application.

When you drag a file from the File Manager to the Print Manager, Windows automatically opens the application, loads the document, sends the document to the Print Manager for printing, then closes the document and the application for you.

Note: You can only drag one file at a time to the Print Manager. If you highlight multiple files in the File Manager and then drag them to the Print Manager for printing, Windows will display an error message. Also, because Windows has to know how to close the document's application, as well as how to print from the application, this drag-and-drop technique only works with applications that conform to the Windows API—in other words, Windows applications only. If you try to drag a document associated with a non-Windows application to the Print Manager, Windows will display an error message—even if you've created a valid association for the document's file type.

Creating a Macro Icon

Macro-icon creation is another little-known and little-used Windows capability, but it's quite a powerful one. By creating an icon for a macro, you can avoid having to open a Recorder file each time you want to run a macro—and you can place the icon in any

program group. Since the macro is considered to be a program item, you can place it in the StartUp group to make it available as a minimized icon throughout your Windows session.

We're not going to give you a primer on using the Recorder here. That's pretty basic stuff, although we know that most people ignore the Recorder, due to some of its admittedly awkward features. One bit of awkwardness is the need to open a macro file before you can run a macro. But you can get around this problem by creating an icon for a particular macro.

If you're willing to learn this technique, we think you'll find yourself using the Recorder more frequently—well, at least *using it.* And the Recorder can be genuinely useful for performing repetitive File Management tasks, such as backing up several directories at a time and on a regular basis, or for searching for and copying and moving selected files.

The trick in creating a macro icon is to understand that you are actually setting up file and shortcut key parameters for the Recorder. You can't create a macro icon unless you've specified a shortcut key combination for the macro.

The command to start the Recorder is RECORDER.EXE. So, the program item icon that you create will need to do three things:

1. Start the Recorder (RECORDER.EXE).
2. Load a Recorder file.
3. Self-execute a shortcut key combination for a particular macro in the Recorder file.

Specifying a shortcut key is the only problem-child part of the process. Basically, you create a macro icon the way you create any other icon—by using the Program Item Properties dialog box.

However, with most applications, you specify a shortcut key sequence in the Shortcut Key text box. With a macro, you want Windows to execute the shortcut key. Otherwise, the icon will just open the appropriate .REC file, and you'll have to press the appropriate shortcut key combination or select the macro from the list in the Recorder dialog box. This would defeat the purpose of creating a macro icon!

To get around this problem, you must signal a shortcut key combination in the Command Line for the icon by typing **-H**. Then, since you can't actually use shortcut keys in the Command Line, you represent them with the following symbols:

Key	*Symbol*
Shift	+
Ctrl	^
Alt	%

So, if you've got, say, a backup macro that's stored in a macro file named MACRO1.REC, and has been assigned the shortcut key sequence Ctrl+Alt+B, the Command Line for the macro icon would look like this:

```
RECORDER.EXE -H ^%B C:\WINDOWS\MACRO1.REC
```

Macro files are stored in the Windows directory by default, although you can save a macro file to any directory.

This procedure might sound complex, but it's pretty simple once you get the hang of it. The steps below demonstrate how to create an icon for the backup macro. Assume that you've assigned Ctrl+Alt+B as the keyboard shortcut for the macro, and the macro is stored in the Recorder file MACRO1.REC. Here are the steps for creating the icon:

1. Save the backup macro to a file, if you haven't already done so (again, we'll save to the file MACRO1.REC). To do so, choose File Save As from the Recorder window, then specify a file name. (If the Macro Recorder icon says Untitled, you haven't saved your macro to a file yet.)
2. Decide which group window you want to store the macro icon in, then make this the active group window.
3. From the Program Manager menu bar, choose File New, make sure Program Item is turned on, and then click on the OK button.
4. In the Description text box, type **Backup Macro** or another description of your choice.
5. In the Command Line text box, type

   ```
   RECORDER.EXE -H ^%B C:\WINDOWS\MACRO1.REC
   ```

   and then press **Enter** to create the icon.

Your macro will now have the same "Camcorder" icon as the Recorder application itself, unless you assign the macro a different icon from the Program Item Properties dialog box. At this point, you can double-click on the icon to run the macro.

Viewing the Contents of a Macro

Although it's true that you can't edit a Macro that you've created with the Recorder, you can view each keystroke and mouse click that you've included in a particular macro.

This technique is useful for finding errors in a macro, although it might take some practice reading the contents of your macro file. To display the events (keystrokes and mouse clicks) in a given macro, follow these steps:

1. Start the Recorder and open the macro file that contains the macro that you want to view.
2. Highlight the desired macro.
3. Hold down **Shift**, then click on Macro Properties. You'll see the Macro Events dialog box, similar to the one shown in Figure 8.19.

    Unfortunately, you can't edit this file, although if you learn how to read its contents you can often quickly trace the source of a macro error. In general, your macros should consist entirely—or almost entirely—of keystrokes; using the mouse within a Macro can create problems if the positions of your icons, applications, and windows change each time you try to issue the macro.

    In a keystroke macro, the contents of the Macro Events dialog box are divided into six entries per event, with each entry separated by a comma. From left to right, the entries in a keystroke event are:

1. Event number (0001, 0002, and so on).
2. Event type (Key Up, Key Down, and so on).
3. Key pressed (b, B, Alt, and so on).

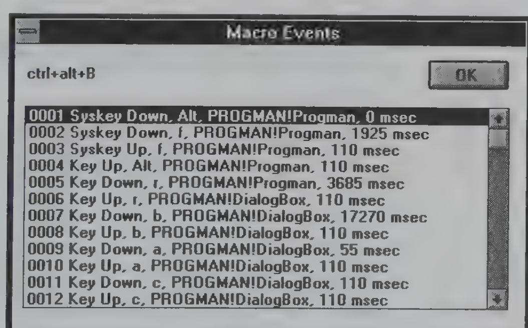


Figure 8.19 This Macro Events dialog box lists each keystroke event in the Ctrl+Alt+B macro.

4. Location of event (PROGMAN!Progman, PROGMAN!Dialog Box).
5. Duration of event in milliseconds (156 msec, 3645 msec, and so on).

For mouse events, the entries are comparable; however, event types equate to down-and-up clicks on the mouse button—or movements left or right, and instead of indicating the key pressed, the screen coordinates for the mouse are shown. Figure 8.20 shows mouse events for a macro.

Optimizing Your Hard Disk Space

One of your most valuable hardware resources is your hard disk. Its space is limited and you'll find that, over time, you'll run out of hard disk space if you don't take some steps to manage it. That's the purpose of the last section of this chapter. We'll let you know which Windows files you need to keep on your hard disk, and we'll show you how to scrunch your occupied hard disk space into a much smaller space, thereby making it possible to store more programs and files on your hard disk than you would have otherwise thought possible.

Using the Windows Setup Program to Remove Unwanted Applications

You probably already know how to use the Windows Setup program to set up new applications to run under Windows or to change hardware drivers. However, Windows Setup can be just as useful in removing many applications from your hard disk.

One of the quickest ways to restore some disk space is to remove Windows files and applications that you never use. Figure 8.21 shows the

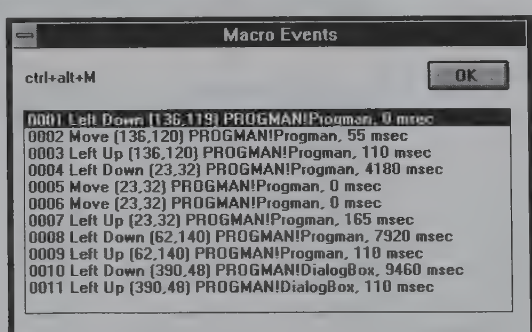


Figure 8.20 This dialog box demonstrates the display of mouse events used in a macro.

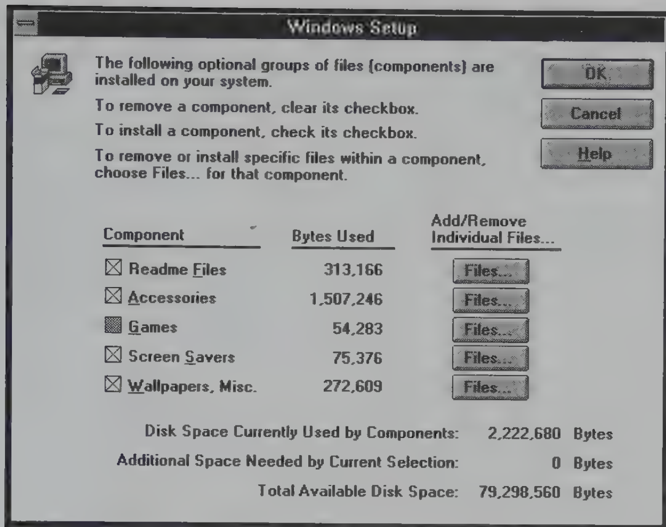


Figure 8.21 Use this dialog box to remove many unwanted categories of Windows files.

dialog box that appears when you choose Options Add/Remove Windows Components from the Windows Setup window. (You access Windows Setup by clicking on its icon in the Main group window.)

As you can see, this dialog box identifies five categories that are the most likely candidates for deletion: README files, accessories, games, screen savers, and wallpaper (and sound) files. You can probably identify at least half-a-megabyte's worth of files here that you don't use.

Also, notice that you can clear all files in any category, or you can select individual files to delete for a particular category. For instance, if you know you'll never use either of the games supplied by Windows 3.1, you can clear the Games check box, then click on the OK button to remove all of the games-related files. By the way, this technique is much easier than using File Manager to delete files, because Windows Setup will automatically delete any HLP (Help) and DLL (Dynamic Link Library) files for an application that you choose to remove. You don't have to try to figure out yourself which files to delete.

To delete selected files in a particular category, just choose the Files button for the category. Figure 8.22 shows the dialog box that appears when you click on the Files button for The Readme Files category. In the text box on the right, Windows lists a description for each README file followed by its file size.

To remove one or more of these files, just click on the file description to highlight each file that you want to delete, then click on the Remove

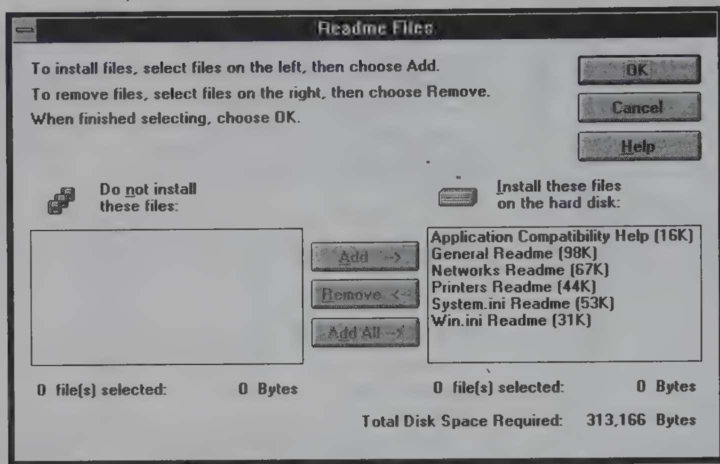


Figure 8.22 You can use this dialog box to remove selected files within a category.

button. Click on the OK button to permanently delete your selections from your hard disk.

HOT TIP

Print a Copy of all README Files, Then Delete Them from Your Disk

If you want to keep a reference copy of each README (WRI) file supplied by Windows, use Write to print a copy of these files. Then, use Windows Setup to remove them from your hard disk. Note that Windows Setup only removes README files that are supplied by Windows itself. If other applications install their own README files, you'll have to use DOS or the File Manager to locate and delete these files.

HOT TIP

Windows Setup Has Its Own README File, Which You Can Delete

Windows copies a Notepad (ASCII) file called SETUP.TXT onto your hard disk when you install Windows. This file describes problems that might occur during the Windows Setup procedure for many different hardware and software configurations.

Although we can understand why Microsoft wanted to copy this file automatically during Windows' installation, SETUP.TXT is one of the most absurd, yet valuable, files in the Windows arsenal. Think about it. If you've successfully installed SETUP.TXT, you probably don't need this file. If you haven't been able to install Windows on your system, then you won't be able to read SETUP.TXT from within Windows.

Actually, SETUP.TXT is valuable, because it is stored on the Windows installation disks in an uncompressed ASCII format. So, if you are having trouble installing Windows on your system, you can use the MS-DOS Editor—or any other ASCII text editor—to load and read SETUP.TXT to find out about any special setup procedures you need to perform with your hardware and software.

Also, SETUP.TXT describes problems that can occur when you try to install or run Windows after some troublesome TSRs have already been loaded. SETUP.TXT includes a listing of TSRs that might create problems for Windows.

If you've already successfully installed Windows on your system, we recommend using Notepad to print SETUP.TXT, then deleting this file from your system. If you have to install Windows on another system and have problems, you can refer to the printed version of this file to do your troubleshooting.

Removing Other Unwanted Windows Files

At this point, we've begun to swim in murky waters. There are many, many possible categories of files stored on your hard disk that you might be able to delete without affecting the performance of your system. We'll try to do our best in helping you detect which files on your hard disk can be deleted. But ultimately, it's your responsibility to determine which files should and shouldn't be deleted from your hard disk.

Before we begin, we need to issue an important disclaimer, if only to cover our own collective backsides: always back up the contents of your hard disk before you remove any files. In the next few pages, we're going to mention a host of files that you *might* want to remove from your hard disk. As authors, we humbly admit our infallibility.

So, please don't ask us to take the blame if you delete a file that you find out later is important to the operation of your system. It's *your* hard disk, not ours. If you issue a Delete command to remove a file, let it be—with all due respect—on your head. But for your own protection, please, please, please back up your hard disk before you begin a wholesale massacre of your supposedly "unwanted" files. And remember that the DOS 5 UNDELETE command is available for recovering deleted files.

Perhaps the best way for us to show which files you might want to remove from your hard disk is through a table that lists and describes the purpose of various Windows-related files. Please use this table carefully. If you remove files that you haven't already made backup copies for, you might not be able to restore them should you later find them to be necessary to the way you use Windows.

All of the files listed in Table 8.1 can be found either in the WINDOWS directory or in the WINDOWS\SYSTEM subdirectory.

HOT TIP

Prune WIN.INI Periodically

Windows reads WIN.INI *every time* you start Windows. The larger your WIN.INI file, the longer it will take for Windows to boot. When you delete applications and fonts, their "footprints" often get left behind in your WIN.INI file. It's a good idea to review your WIN.INI file periodically, looking for deleted fonts, drivers, and other entries related to long-since-deleted applications.

HOT TIP

Search For and Delete ~ or TMP Files

Many Windows applications create temporary files that start with the ~ (tilde) character and end with the extension TMP. These files are used only for one particular Windows session, but might remain on your hard disk long after they are required—which often happens if you reboot Windows or an application due to a system lockup. Periodically, use File Manager to search for all files that begin with a tilde and end with TMP (type **~*.TMP** in the Search dialog box, and begin the search from the root directory of your hard drive). Highlight and delete all of these files.

Chapters 9 and 10 include some tips for using file compression programs. These programs "squeeze" or compress a file into about half its original size, so that the file requires less disk space to store. Since most disk compression programs work at the DOS level, we've put this discussion in the next chapter. (Chapter 10 provides a demonstration of one of the most popular run-time file-compression programs, which is called Stacker.)

Table 8.1 Windows System Files

File(s)	*Purpose*	*When to Delete*
MSD.EXE MSD.INI	The Microsoft Diagnostics files. You need these to assess your hardware and software, including memory mapping; if you call Microsoft Technical Support for help, you might be asked to run MSD.	Remove if you have a third-party application (such as 386MAX, QEMM-386, PC-Tools, or Norton Desktop for Windws) that can diagnose hardware and software and display memory information.
TASKMAN.EXE	This is the Windows Task Manager.	Remove if you never use the Windows Task Manager or if you have installed a third-party task manager to replace TASKMAN.EXE.
CLIPBRD.EXE	This is the Clipboard Viewer Program.	If you never use Clipboard Reviewer, you can remove this file.
DOSX.EXE	The DOS extender for standard mode.	Remove the following six files if you never run Windows in standard mode.
DSWAP.EXE	The utility for swapping DOS applications to disk in standard mode.	
KRNL286.EXE	The kernel program required for all 286 PCs and for 386 PCs that have less than 2MB RAM.	
WINOLDAP.MOD	Standard mode suport for non-Windows applications.	
WSWAP.EXE	The utility for swapping Windows applications to disk in standard mode.	
*.2GR	The grabber file for standard mode. The precise file name depends on the resolution of the installed monitor.	
CONFIG.* AUTOEXEC.*	CONFIG.SYS is required, but other, unnecessary CONFIG files might reside on your hard disk. AUTOEXEC.BAT is required, but other, unnecessary AUTOEXEC files might reside on your hard disk.	Whenever you install a new application, the application's INSTALL or SETUP program often tries to modify your AUTOEXEC.BAT and CONFIG.SYS files, and then creates a backup version of these files in case you need to restore the previous versions. If your system is operating without problems, it is probably safe to delete any backup AUTOEXEC or CONFIG files.
BOOTLOG.TXT	This file stores information about the success or failure in loading various applications when Windows boots.	BOOTLOG.TXT stores bootup information for several previous iterations of Windows, so you can print this text file as a diagnostic tool to help detect problems involved when you start Windows after installing new applications.

Table 8.1 Windows System Files (Continued)

File(s)	*Purpose*	*When to Delete*
EMM386.EXE	The expanded memory manager for DOS 5.	If you are not loading EMM386 in CONFIG.SYS to provide upper memory support for TSRs, or if you are using a third-party memory manager to configure upper memory, you can delete this file.
RAMDRIVE.SYS	The RAM drive program for Windows.	RAMDRIVE.SYS sets up a portion of RAM to emulate a disk drive so that disk files can be stored in memory. If you are using SMARTDrive 4.0 (and you should be), you probably won't need RAMDRIVE.SYS.
CONTROL.HLP	The help file for the Control Panel.	If you feel that you're an experienced user, you might want to delete CONTROL.HLP and SETUP.HLP. If you're already well-versed in using the Control Panel and the Windows Setup Program, these help-file omissions won't bother you.
SETUP.HLP	The help file for the Windows Setup Application.	
WINHELP.EXE	The Windows help engine.	Do note that if you delete the file WINHELP.EXE, you will disable help for *all* Windows applications. You should keep WNHELP.EXE available, unless you *never* use the Windows help facility.
*.PIF	Information required by Windows to manage a non-Windows application.	Delete PIF files if you never use their DOS applications.
APPS.INF	The file containing all of the information needed to create PIFs for more than 100 DOS applications.	Delete APPS.INF *only* if you are sure you will never need to install additional DOS applications to run under Windows.
CGA40WOA.FON CGA80WOA.FON EGA40WOA.FON EGA80WOA.FON	Font files used to support the display of non-Windows applications.	These files are used to display DOS applications in a window in enhnaced mode. If you only run DOS applications full screen in enhanced mode you can delete these files.
*.DRV	Drivers for various hardware devices and software.	Search for DRV files and delete those that belong to devices that you no longer have (such as a discarded monitor, mouse, or printer).

Table 8.1 Windows System Files (Continued)

File(s)	*Purpose*	*When to Delete*
.FOT/.TTF	The pairs of files for TrueType fonts.	Each TrueType font has two files: an FOT file and a TTF file. Windows lets you install up to 1,170; however, unless you do some heavy desktop publishing, you probably don't need more than a dozen or so TrueType files. Consider deleting TrueType fonts (or any fonts, for that matter) that you never use.

HOT TIP

Create a Backup Log

Many of the files that are mentioned in Table 8.1 might be useful to you, but only on rare occasions. For instance, you might want to delete all of the files that support standard-mode Windows if you never run in standard mode. However, it's a good idea to back up these files to a diskette first. Then, if enhanced mode fails for some reason, you can at least run in standard mode temporarily by copying the required files back to your hard disk. We recommend using Notepad to create a log of all files that you've deleted from your hard disk but backed up to diskettes. Include the purpose for each file and the hard-disk directory where each file was originally located. If you use Notepad to create this log, the file will be readable at the DOS prompt or from within any ASCII text editor.

CHAPTER 9

Alternate File-Management Techniques

You've probably heard the expression, "Use the right tool for the job." Obviously, you don't want to use a socket wrench to hammer sixteen-penny nails into a two-by-four. In a similar vein, there are times when the File Manager might not be the right tool to use in performing file and disk management. And even when the File Manager might not necessarily be the *wrong* tool for the job, it still might not be the *best* tool.

In some cases, you are better served by using DOS commands or other tools to manage your files and disks. This chapter presents techniques for managing files without using the File Manager—or at least without using *just* the File Manager. You'll discover how to access DOS commands directly from within Windows, and you'll learn about several other DOS-based shortcuts. Along the way, we'll offer some tips for using Windows to back up your files. And we'll provide an extensive review of DOS commands and their effectiveness under Windows at the end of this chapter.

Techniques for Using DOS Commands Directly

Like many experienced PC users, you might consider yourself a dyed-in-the-wool user of DOS commands. You would rather shell out to the DOS prompt to enter file management commands than use the File Manager. (You can return to the DOS COMMAND.COM level by choosing the MS-DOS Prompt icon from within Windows; by doing so, you run DOS as a shell application without having to exit Windows.) If you are in the habit of using this approach, the information, tips, and warnings in this section will provide you with some techniques for using DOS more effectively.

Direct DOS Commands

If you find that Windows gets in your way when performing file and disk maintenance commands, you can easily run many DOS commands without shelling out to DOS.

You might think that every time you want to use a DOS command, you have to exit to the DOS prompt, enter your DOS command, wait for it to finish executing, then type EXIT to return to Windows. However, you can enter and execute many DOS commands directly from within Windows by using the Run command in the File menu (from either the Program Manager or the File Manager).

Keep in mind that when you choose Run, Windows prepares itself to search for an EXE, COM, BAT, or PIF file that matches the name of the file that you enter in the Run text box. Now consider that most external DOS commands have either an EXE or COM extension. So, here's what will happen if you enter the following command in the Run text box:

```
XCOPY C:\WINWORD\MEMOS C:\WINWORD\BACKUP
```

Assuming that the directory to your DOS files is included in your system's PATH command, Windows will find the file XCOPY.EXE in the DOS directory, then shell out to DOS, execute the command (copying the files in the specified directory), and then return automatically to Windows.

Remember, this capability is available only for external DOS commands. Internal commands like COPY, DEL, and DIR cannot be used in this way (although there are tricks for using internal commands from within Windows, and you'll learn how to do so in this chapter). Windows will display an error message if you try to execute internal DOS commands in the Run dialog box. Table 9.1 at the end of the chapter lists DOS internal and external commands and their usefulness within Windows.

HOT TIP

Using External DOS Commands Effectively

External DOS commands that display information following completion of their routine—such as CHKDSK, FC, FIND, MEM, and TREE—should not be executed from within Windows. If you try to execute one of these commands by using the Run dialog box, Windows will display the information that you need for only a fraction of a second, and then will return to the Windows environment. The information will not stay on-screen long enough for you to review it.

Also, the redirect symbol (>) and the MORE filter will not work with an external DOS command that is executed from within Windows. For example, if you want to use the CHKDSK command and instruct DOS to route the output to your printer, you would normally enter a command similar to this one:

```
CHKDSK > LPT1
```

When you execute this command from within Windows, DOS will not recognize the ASCII code for the > symbol and will fail to execute your command. However, you can execute, from within Windows, a batch file that uses redirect and filter symbols. We'll explain how to create DOS batch files a bit later in the chapter.

HOT TIP

Getting Help for DOS Commands

You can view a complete list of DOS internal and external commands, along with a brief description for each command, by typing HELP in the Run dialog box. This technique is great when you need a quick reminder of the purpose of one or more DOS commands. After you scroll through the list, you will be returned automatically to Windows.

Creating Batch Files to Use Internal DOS Commands

You can issue an internal DOS command from the File Run dialog box if you first create a batch file for the command. To use most internal DOS commands effectively, you'll need to include replaceable parameters in the batch file.

Although you can't issue an internal DOS command from the File Run dialog box, you *can* start a batch file. So, if you write a batch file that executes an internal DOS command, you'll get the same effect. The key is to include *replaceable parameters* within your batch file.

A replaceable parameter is a percent (%) symbol followed by a number from one through nine. (You can include up to nine *replaceable* parameters in a batch file.) The word "replaceable" just means that DOS replaces the %*n* (where *n* is a number from one to nine) that appears in a batch file with any text that you type when you issue the command to start the batch file. Each parameter is separated from other parameters by a period. A parameter is any string of text that can be understood by DOS, such as a file name (CHAPTER1.TXT), a drive designation (D:), a path (WINDOWS\SYSTEM), a DOS switch (/P, /D, and so on), or a combination of these (C:\WINDOWS\TXTFILES\CHAPTER1.TXT).

Here's a simple example. This is the contents of a batch file (called TIME.BAT) that lets us change the date and time quickly—even from within Windows:

```
TIME %1
```

We used the MS-DOS Editor to create the batch file by typing **EDIT TIME.BAT** in the File Run dialog box. The WINDOWS directory contains a PIF file, called EDIT.PIF, which loads the MS-DOS Editor automatically when you type **EDIT**. Of course, you could also double-click on the MS-DOS Editor from the Applications window in the Program Manager.

Now when we type **TIME 3:42** in the File Run dialog and press **Enter**, Windows loads the file TIME.BAT and replaces %1 in the batch file with the 3:42 parameter that you specified when you executed the batch file. DOS automatically resets the system clock and returns you to Windows.

HOT TIP

Use a Batch File Name That Matches the Desired Name of the Internal DOS Command

If you're creating a batch file for the COPY command, name the file COPY.BAT; if you're creating a batch file for the DIR command, use the name DIR.BAT. If you use this approach, you won't have to remember the names of your batch files. Just issue the commands from the Run dialog box as you normally would from the DOS prompt.

Take a look at another example. This one uses multiple replaceable parameters to create a DIR.BAT file. Here's what the one-line batch file looks like:

```
DIR %1 %2 %3 %4
```

Here, we've included four replaceable parameters. The first (%1) parameter allows us to insert the path that we want to search. The second, third, and fourth parameters (%2, %3, and %4) let us use up to three switches to control the directory display.

Suppose we type the following command in the Run dialog box:

```
DIR C:\WINDOWS /O:N /W /P
```

This command tells DOS to display the contents of the C:\WINDOWS directory (parameter 1), organize the files alphabetically by file name (/O:N, parameter 2), display the files in wide format (/W, parameter 3), and pause whenever the screen fills (/P, parameter 4). Figure 9.1 shows how this output might appear.

Note: The Organize (/O:) switch—which allows you to display a directory in order by file size, most recently modified, by name, and by extension—is not available in versions prior to DOS 5.

You don't need to use every parameter that you've specified in a batch file. For instance, we could have typed **DIR C:\WINDOWS /P**. Windows

```
 Volume in drive C is STACVOL_DSK
 Directory of C:\WINDOWS

[.]             [..]            123.PIF         256COLOR.BMP    ACCESSOR.GRP
APPLICAT.GRP    APPS.HLP        APPS.IL         ARCADE.BMP      ARCHES.BMP
ARGYLE.BMP      [BACKUP]        BANGBANG.WAV    BARTBUTT.WAV    BEAMUP2.WAV
BEBACK.WAV      BEEF.WAV        BENICE.WAV      BING.WAV        BLEED.WAV
BLOWS.WAV       BOGUS.WAV       BOOP.WAV        BOOTLOG.TXT     BORIS1.WAV
BUGS1.WAV       BULLWNK1.WAV    BURP.WAV        BUTHED.WAV      CALC.HLP
CALC.EXE        CALENDAR.EXE    CALENDAR.HLP    CALENDAR.ANN    CANYON.MID
CARDFILE.HLP    CARDFILE.EXE    CARS.BMP        CASTLE.BMP      CHARMAP.HLP
CHARMAP.EXE     CHIMES.WAV      CHITZ.BMP       CHORD.WAV       CLIPBRD.EXE
CLIPBRD.HLP     CLOCK.INI       CLOCK.EXE       CONTROL.EXE     CONTROL.INI
CONTROL.HLP     CONTROL.ANN     CPBACKUP.PIF    [CPDATA1]       [CPDATA2]
CPSDOS.GRP      CPSICONS.DLL    DATAMON.PIF     DESKTOP.PIF     DING.WAV
DIR.PIF         DISKFIX.PIF     DM.PIF          DOSAPP.INI      DOSPRMPT.PIF
DRWATSON.EXE    DRWATSON.LOG    ECMDDOS.PIF     EDIT.PIF        EEPREFS.PRE
EGYPT.BMP       EMM386.EXE      ENCPDLL.DLL     EXCEL4.INI      EXPAND.EXE
FAILURE.WAV     FAXMNG.EXE      FF.PIF          FILEFIX.PIF     FILES.GRP
FIREALL.WAV     FIX2.BAT        FLOCK.BMP       FOGBLAST.WAV    FORCE.WAV
FREEMEM.INI     FS4.PIF         GAMES.GRP       GLOSSARY.HLP    HAL.WAV
HALMIND.WAV     HAN.WAV         HELLOCMP.WAV    HELPHK.DLL      HIMAN.WAV
Press any key to continue . . .
```

Figure 9.1 This DIR display is controlled by a batch file that contains four replaceable parameters.

would have inserted these two parameters in the %1 and %2 replaceable positions in the DIR command, and would have ignored the %3 and %4 parameters.

Use the following steps to create a batch file for an internal DOS command. We'll use the REN (Rename) command as an example.

1. From the File Manager or Program Manager, click on File Run.
2. Type **EDIT REN.BAT**, then press **Enter**. When you specify a file name in starting the MS-DOS Editor, DOS creates an empty file named REN.BAT and allows you to make entries into the file.
3. Type **REN %1 %2.** You only need two replaceable parameters for the REN command—one for the source path and file(s) to rename, and a second parameter for the target path and file name(s).
4. Click on File Save, then click on File Exit.

You can now issue the REN command from the RUN dialog box, just as you would do from the DOS Prompt.

HOT TIP

Run Internal DOS Commands in Windows without Using Batch Files

You Can Run an Internal DOS Command Directly from the Run Dialog Box. Yes, we said you can't do this, but you can actually start many internal DOS commands if you're willing to accept some unusually lengthy command strings.

COMMAND.COM accepts a /C switch that lets you run a particular internal DOS command. Although this approach is a little "wordy," you'll sometimes find it useful in a pinch because you can run just about any internal DOS command from the Run dialog box.

For instance, if you want a listing of files in the current directory, pausing after each screen is full, you can enter this command in the Run dialog box:

```
COMMAND.COM /C DIR /P
```

Shelling Out to the DOS Prompt

If you do need to shell out to the DOS prompt to execute a DOS command, you can do so without having to select the DOS Prompt icon from the Main group.

Using DOSPRMPT.PIF

If your Main group window is not always open when you are in the Program Manager, the DOS Prompt icon will not be instantly accessible. And if you are in the File Manager, you cannot select the DOS Prompt icon without first switching to the Program Manager. Try the following approach instead.

Keep in mind that every icon has a command line that Windows uses whenever you select the icon. This same command line can be entered in the Run dialog box to execute a program without having to locate and select the corresponding icon. The command line for the DOS Prompt icon is DOSPRMPT.PIF.

So, you can exit to the DOS prompt at any time simply by entering **DOSPRMPT** in the Run dialog box.

Setting the Date and Time

If only out of force of habit, you might be tempted to reset the system date or time from the DOS prompt. However, if you are working in a Windows application, it can be faster to change the system date and time from the Control Panel.

The Run command from many Windows applications' Control menu allows you to switch immediately to the Control Panel. This approach makes it easy to change the system date and time without shelling out to DOS.

Figure 9.2 shows the Run command that appears from the Control menu for Word for Windows. Figure 9.3 shows the dialog box that pops up when you choose Run. If you turn on the Control Panel option button and then click the OK button, Windows will switch to the Control Panel, allowing you to use the Date/Time application. When you are finished setting the date or time, Windows will return you to your application (in this case, Word for Windows). This feature is neither available for DOS applications nor for some Windows applications.

Running the DOS Prompt Minimized

You can speed up access to the DOS prompt by running the DOS Prompt as a minimized icon whenever you start Windows.

To start the DOS Prompt as a minimized icon when you start Windows, simply add the MS-DOS Prompt to your StartUp group and select the Run

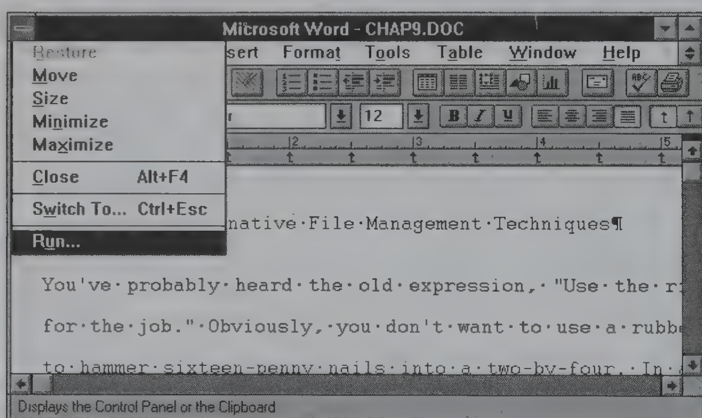


Figure 9.2 The Control menu includes the Run command to start the Clipboard or the Control Panel.

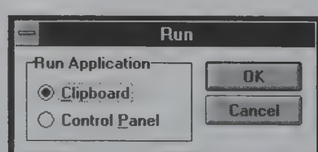


Figure 9.3 Use this dialog box to switch to the Control Panel.

Minimized check box in the Program Item Properties dialog box. The major benefit to this approach is that the DOS Prompt is always available to you—from within any Windows or non-Windows application.

To switch to the DOS Prompt application, simply press **Alt+Tab** until the MS-DOS Prompt displays. Then use DOS as you would under any other situation. When you are done, press **Alt+Tab** to switch back to your application.

HOT TIP

Using the DOS Prompt as a Minimized Icon

If you are running the MS-DOS Prompt application as a minimized icon, *do not* type EXIT when you are done using the prompt. Doing so will close the MS-DOS Prompt application, and it will no longer be available to you as a running application. If you want the DOS prompt to remain available, always use Alt+Tab to switch from the MS-DOS Prompt to other applications.

HOT TIP

Starting Applications from the DOS Prompt

You might think it's possible to start a Windows application from the DOS prompt if Windows is already running. Unfortunately, this is not possible (although you can start DOS applications in this way).

Windows creates a virtual machine for DOS to prevent the two environments from conflicting. In other words, DOS acts as though it is on a separate computer and doesn't "know" that Windows is running. As a result, when you try to start a Windows application from the MS-DOS Prompt application, you will receive the following message:

```
This program requires Microsoft Windows.
```

You must instead start the Windows application using the Program Manager or the File Manager.

Running DOS as a Pop-Up Window

You might already know that you can display DOS in a window by pressing **Alt+Enter** after you've started the MS-DOS Prompt application. However, you can also make DOS behave like a pop-up window by making a few adjustments to the way the MS-DOS Prompt application appears. This approach is useful because you can keep the DOS Prompt displayed at all times while you are in the Windows environment, and you can pop up the DOS prompt from within any Windows application. You get the best of both the Windows and DOS worlds—always literally at your fingertips.

Creating a DOS Window

Use the PIF for the MS-DOS icon to create a permanent DOS window. Figure 9.4 shows an example of an MS-DOS Prompt window displayed from the Program Manager. Notice that this configuration provides room for the Program Manager window, three tiled group windows, and three minimized icons.

The following steps explain how to produce this MS-DOS Prompt display, making it a permanent part of your Windows environment. (Of course, you can adjust the size of the window differently to meet your own needs.)

1. Copy the MS-DOS Prompt to the StartUp group, if it isn't already there.
2. Start the PIF Editor (from the Main group window).

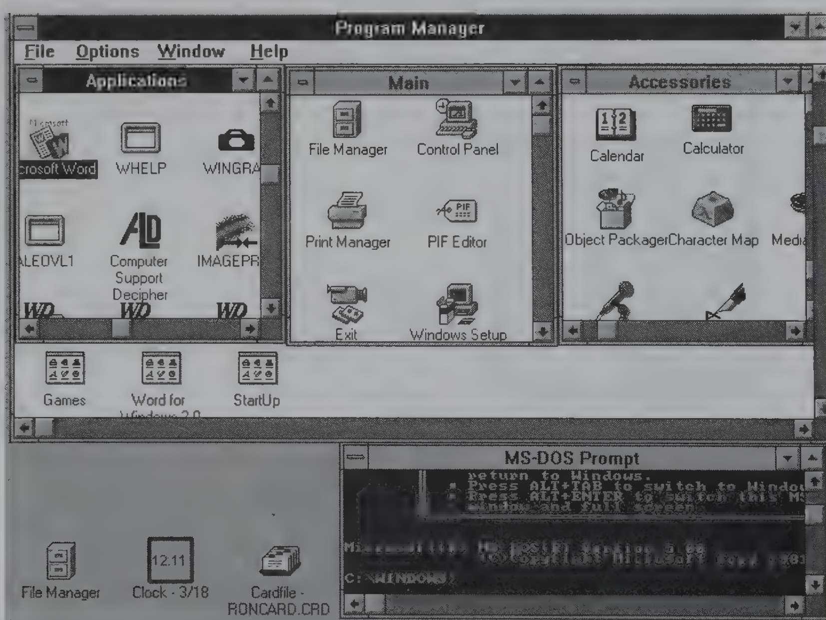


Figure 9.4 You can create a DOS window and make it a permanent part of your Windows environment.

3. Choose File Open.
4. In the File Name list box, choose dosprmpt.pif to display the program information file for the MS-DOS Prompt application.
5. In the Display Usage section of the PIF Editor window, turn on the Windowed option button. This setting tells Windows to display the DOS Prompt in a window by default, rather than in full-screen mode.
6. Choose File Save, then exit the PIF Editor.
7. Start the MS-DOS Prompt application, if necessary.
8. Size the DOS window to the desired height and width, then drag it to the desired location (bottom left in Figure 9.4). Size and tile any other windows, as desired.

   If you want, you can also change the font display size to see more information in the DOS window at one time. To change the font size, display the Control menu for the MS-DOS Prompt application, then choose Fonts. The Font selection dialog box, shown in Figure 9.5, appears. Notice that you can adjust both the width and height of characters. Select an appropriate width and height in the Font list box, then choose OK. (Use this same technique to change the font display in any DOS application window.)
9. Save your Program Manager settings to make the DOS window permanent. (Hold down **Shift**, then click on File Exit to save settings without exiting Windows.)

With the DOS prompt now displayed in a small window, you can pop it up from within any Windows application by pressing **Alt+Tab** until the

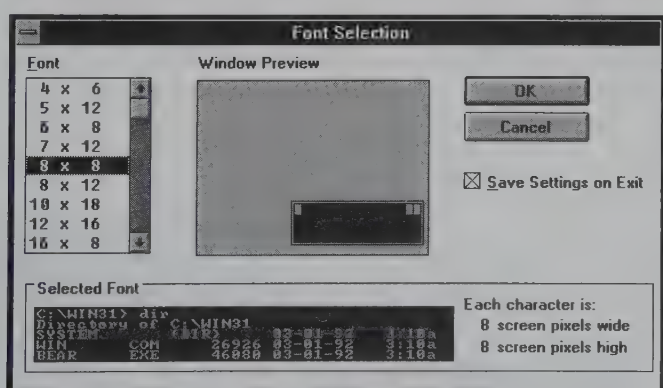


Figure 9.5 Use the Font Selection dialog box to adjust the display font for DOS application windows.

MS-DOS Prompt application name appears. Figure 9.6 shows how the prompt you created in the steps above will appear from within a Word for Windows document. A drawback to this approach is that the DOS window is too small to display the contents of directories (with the DIR and TREE commands, for instance). However, this small size works great for issuing single-line DOS commands (like COPY, DEL, MD, PRINT, and so on). And, of course, you can maximize the DOS window at any time.

HOT TIP

MS-DOS Prompt Loads Last

Windows always loads the MS-DOS Prompt last—assuming that you've added it to the StartUp group. So, any batch files that you've included in your StartUp group will run before the DOS window opens up. Normally, this is not a problem. However, don't expect to see these batch files executing in the MS-DOS Prompt window.

Also, because the MS-DOS Prompt window loads last, it will automatically pop up over another Windows application that loads from the StartUp group. If you don't want this effect, you'll have to set the DOS prompt to run minimized at startup time.

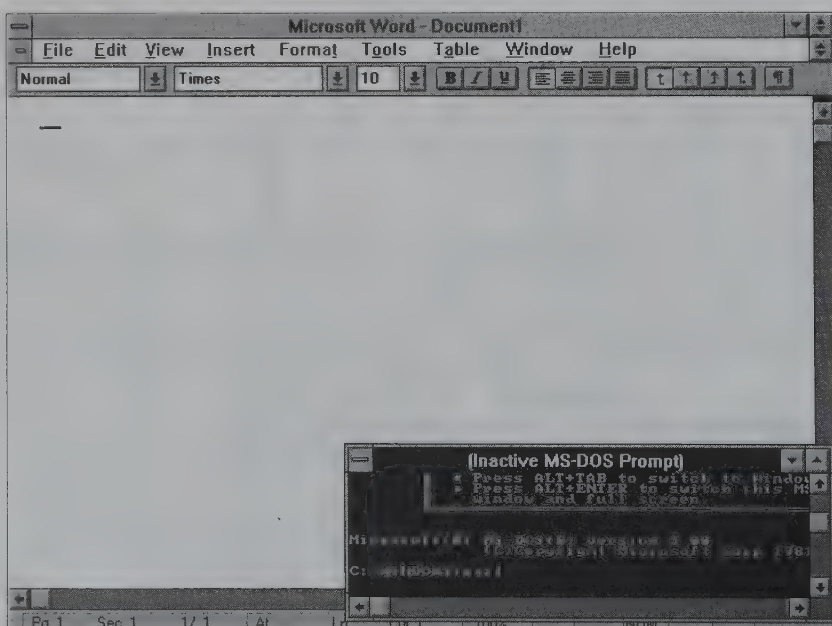


Figure 9.6 A DOS window, popped up from within Word for Windows.

HOT TIP

Change the DOSPRMPT.PIF File So That You Can Exit Windows

You can't exit Windows until you've exited all DOS applications, and you usually can't use Windows to exit a DOS application. Windows won't close the application for you because many DOS applications (such as WordPerfect, Word for DOS, and Lotus 1-2-3), save configuration information before they exit. If Windows were to exit DOS applications for you, this might create lost clusters on your hard disk, or worse, might cause the DOS application to behave erratically the next time you open it.

Although this safeguard makes sense, you'll find it annoying when you have to exit harmless DOS applications—such as the MS-DOS Prompt—by typing **EXIT** every time you're ready to quit Windows.

If you're sure a particular DOS application doesn't write to any files before it closes, you can tell Windows to close the application for you. The MS-DOS Prompt is a case-in-point. The MS-DOS Prompt application is really just a separate instance of the COMMAND.COM file. COMMAND.COM can respond to interrupt requests to perform low-level file I/O for other programs, but it never writes to files on its own. So, if you like to keep the MS-DOS Prompt open during your window session, but you don't like having to open the DOS window and type **EXIT** every time you prepare to exit Windows, follow these steps:

1. Start the PIF Editor, then choose File Open.
2. In the Files list box, click on dosprmpt.pif.
3. In the PIF Editor dialog box, click on the Advanced button to display the Advanced Options dialog box.
4. Select the Allow Close When Active check box. Windows will respond with a dire warning about the potential for losing data or damaging your hard disk.
5. Click OK to verify that you really want to perform this act of "gross irresponsibility." (You really won't do any harm if you let Windows close the DOS Prompt for you.)
6. Click OK to exit the Advanced Options dialog box, then choose File Save to save the revised version of the PIF.
7. Exit the PIF Editor dialog box.

When you exit Windows, you'll still see a warning dialog box telling you that your DOS Prompt application is still open. But now you just click on OK when this dialog box appears and Windows will close. With this approach, you can exit Windows solely by using the mouse; your hands don't have to touch the keyboard.

Using DOS Commands and Other Tools to Manage Files

The File Manager can do many tasks that DOS cannot (such as deleting directories that contain files, and renaming or moving directories). But it's also true that DOS 5 has a few tricks up its sleeve where Windows is left short-sleeved.

This section describes a few DOS utilities and tips that you might find helpful—or even necessary—at one time or another. Since an extensive description for all of these commands can be found in your DOS 5 User's Manual or in any competent book on DOS 5, we won't describe them in great depth. Instead, we'll just let you know what's available and provide enough of a description to get you started. All of the utilities described below—even though they're part of DOS—can be run easily from within Windows.

At the end of this chapter, we've included a table that lists most DOS commands—both internal and external—and explains how to use the commands in Windows. The table also includes warnings for DOS commands that you should not run from within Windows.

Adding SHARE to Your AUTOEXEC.BAT File

SHARE is a DOS utility that allows multiple applications to share the same files. SHARE tracks information about open files and will allow an application to use a particular file only if it is not open and in use by a different application.

SHARE does not use much RAM and can protect your data files from being corrupted inadvertently. If you're working on a stand-alone machine (rather than a network), it's probably unlikely that one application will try to access a data file currently in use by another application; but it can happen.

More importantly, some applications recommend or require that SHARE be installed before they run. For instance, one application might try to use another application's DLLs (dynamic link libraries). To prevent read and write conflicts, an application that "leeches" off of another program's DLL environment probably often will not run unless you have installed SHARE as a TSR in your AUTOEXEC.BAT file.

Another example: Word for Windows 2.0 recommends that you install SHARE, but does not require it unless you try to run two or more instances of Word for Windows at the same time.

SHARE can take two parameters, in this format:

```
SHARE /F:<file space> /L:<Maximum number of locked files>
```

The /F switch designates the amount of RAM that SHARE can reserve for tracking file information; the /L switch determines the total number of files that SHARE can lock, or prevent from being accessed while their in use by another application.

The recommended SHARE setting for a stand-alone system is 2K for file space and 20 locked files. Here's the setting that would appear in AUTOEXEC.BAT:

```
SHARE /F:2048 /L:20
```

Recovering Lost Files

Windows has no provision for recovering deleted files. If you've got Norton Desktop for Windows or another file-recovery utility, this Windows shortcoming won't bother you. If you haven't purchased a file-recovery utility, and you think that you've erased some important files from your hard disk or from a diskette, it's still not time to panic. As long as you're running DOS 5, you can use the UNDELETE utility program to recover erased files.

After you realize you've accidentally erased files, follow these basic steps *as soon as possible*:

1. Choose File Run from the Program Manager or File Manager.
2. Type **UNDELETE** and the path, then press **Enter**. The UNDELETE utility uses the current DOS directory, *not* the current directory displayed in the File Manager, so, you must enter the path to your deleted file(s). For instance, if you've deleted a file in the BUDGET directory under your EXCEL directory, type the following in the Run dialog box:

```
UNDELETE C:\EXCEL\BUDGET
```

Windows shells out to DOS and displays a list of any deleted files that can be recovered. DOS actually deletes only the first letter of a file name, which allows the file allocation table (FAT) to assign that disk space to data for another file. If you haven't done a lot of work since the time you realized you've erased files accidentally, the chances are excellent that all the data in the files still exists.

3. To recover a deleted file, enter **Y** to the Undelete (Y/N)? prompt and enter a letter for the first character in the file name. (This does not have to be the letter that was used originally.) Repeat this procedure for all other files that you want to recover from this directory.

HOT TIP

Using Multiple Virtual DOS Machines

Windows can create virtual machines for two or more DOS applications, including DOS itself. So, you can safely issue DOS commands from the Run dialog box, even if you have a version of MS-DOS Prompt already running. You might want to do this if you are working in the File Manager or Program Manager and your minimized MS-DOS Prompt icon or your DOS window is not currently in view.

HOT TIP

Add the DOS MIRROR Command to Your AUTOEXEC.BAT File

The MIRROR command, new in DOS 5, is essential for unformatting a diskette. It also can assist in recovering files that you've accidentally deleted. You should add the MIRROR command to your AUTOEXEC.BAT file so that it maintains a continuous history of your hard disk.

Whenever you run MIRROR (ideally, ever time you start your computer), it creates a hidden file named MIRROR.FIL that is stored in the root directory. MIRROR writes the contents of the drive's file allocation table (FAT), root directory, and bootup information into the MIRROR.FIL file.

The MIRROR.FIL remains on disk *even if you accidentally reformat your hard disk.* For this reason, you can usually use the UNFORMAT command successfully if you accidentally erase the contents of all or part of your hard disk.

MIRROR also uses an optional /T (delete tracking) switch that stores backup information about any file that you delete. If you add the /T switch to your MIRROR command in AUTOEXEC.BAT, DOS can do a better job when you use the UNDELETE command to restore accidentally deleted files.

Unless you tell it otherwise, MIRROR tracks deletion information only for the default drive. If you have more than one logical drive (such as a D or an E hard drive), you need to add a separate switch for each logical drive. Here's an example:

```
MIRROR /T D:/T E:/T
```

The first /T switch tells MIRROR to track deletion information on the default drive; D:/T forces MIRROR to track deletion information on the D drive; and E:/T forces MIRROR to track deletion information on the E drive.

Unformatting a Disk

Suppose you've just finished formatting a diskette that previously contained data. You thought you no longer needed the files that were stored on this diskette; that's why you reformatted it. But now you realize you've made a grave mistake. If you're running DOS 5, the chances are good that you can unformat the diskette and restore the original files and directories. You can even unformat your hard disk in many situations.

In DOS 5, if you inadvertently try to format your default hard drive (probably because you forgot to specify a diskette drive when you issued the format command), DOS will warn you. But if you're not paying attention, you could conceivably bypass this message and allow DOS to reformat your hard drive.

By default, DOS will create a MIRROR.FIL file if you let it format your hard disk or a floppy disk. If DOS can't create a MIRROR.FIL file, it will tell you so. This problem typically occurs when you reformat a floppy diskette using a format that differs from its previous format (such as formatting a high-density diskette in 360K or 720K format).

If a current MIRROR.FIL file exists for your hard drive, you can probably recover most of your programs and files successfully. Follow these basic steps *as soon as possible* after you realize you've accidentally formatted your hard disk:

1. Insert your backup DOS diskette in drive A or B and reboot your system from the diskette.

2. Type **UNFORMAT C:**, then press **Enter**.
3. Respond to the prompts that DOS displays until your hard drive has been rebuilt. This may take 30 minutes or longer.

If you accidentally reformatted a floppy diskette, you can use these steps from within Windows:

1. Choose File Run from the Program Manager or File Manager.
2. Type **UNFORMAT** and the drive, then press **Enter**. The UNFORMAT command has other optional switches you can use. Refer to your DOS User's Manual for more information.
3. Windows shells out to DOS and prompts you to insert a diskette in the appropriate drive. After you have done this, DOS searches for a MIRROR image file that it uses to reconstruct your diskette. If this file is found, DOS will ask whether you want to rebuild the system area of the diskette. Type **Y** to continue.

Using the System Configuration Editor

If you're a longtime DOS user, you've probably had plenty of experience modifying your CONFIG.SYS and AUTOEXEC.BAT files—probably more than you would care to recall. Fortunately, you no longer need to use EDLIN to edit these files. Instead, you can use the MS-DOS Editor (from the DOS prompt or from within Windows) or Notepad (from within Windows). Although these text editors do provide a lot of convenience, there's still a better way.

Windows provides an undocumented and little-known feature called the System Configuration Editor, which you start by typing **SYSEDIT** in the Run dialog box. This editor is actually a version of Notepad with a built-in, cascaded display of four widely used system configuration files: SYSTEM.INI, WIN.INI, CONFIG.SYS, and AUTOEXEC.BAT. The System Configuration Editor is shown in Figure 9.7.

The System Configuration Editor is extremely valuable because you'll often find it necessary to work with more than one of these configuration files at the same time. And even experienced DOS users occasionally forget which environment variables are in CONFIG.SYS and which are in AUTOEXEC.BAT. When you use the System Configuration Editor, you don't have to remember these details.

Again, just type **SYSEDIT** in the Run dialog box to load the System Configuration Editor. Since this editor is really the Notepad in disguise,

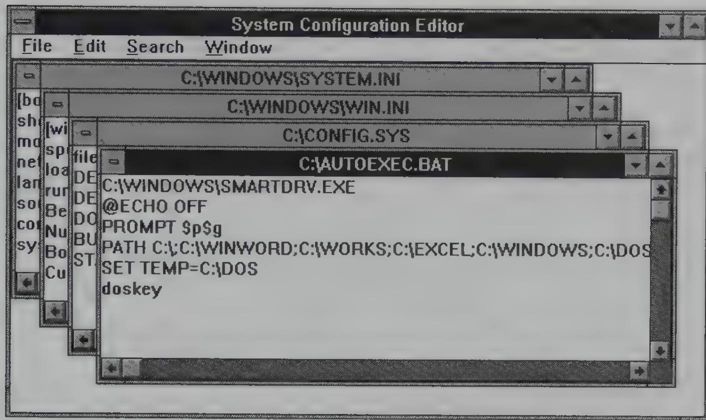


Figure 9.7 Use the System Configuration Editor to edit CONFIG.SYS and AUTOEXEC.BAT.

you use it just like you would the Notepad, except that you will need to switch between the four document windows.

HOT TIP

Create an Icon for the System Configuration Editor

Windows doesn't automatically create an icon for the System Configuration Editor when you install Windows. But if you use SYSEDIT with any regularity, you'll probably want this program available as an icon. Just follow these steps to create a SYSEDIT icon:

1. Open the group window where you want to store the SYSEDIT icon.
2. Click on File New, then click on OK to display the Properties dialog box.
3. In the Command text box, type **SYSEDIT.EXE**, then press **Tab**.
4. In the Description text box, type **System Configuration Editor.**
5. Click on the OK button. SYSEDIT.EXE has its own icon, which will be assigned to the program automatically when you click on OK.

Using File-Compression Software

Windows and Windows applications are notorious for hoarding your hard disk space. So, you'll want to optimize available hard disk space by compressing files when possible. Most of these programs work at the DOS level.

The term *file compression* refers to several techniques for reducing the amount of disk space required to store files. Most file-compression programs compress files at about a 2:1 ratio. In other words, the program can compress every two bytes in a file into about one byte. The result is that the amount of space required to store a particular file is cut in half.

The DOS BACKUP and RESTORE utilities are some of the oldest examples of PC file-compression programs. When you run the BACKUP program, all files in specified directories are copied onto floppy diskettes, but in a compressed format. Applications can't read files that are stored in a compressed format, so you need to use the RESTORE program if you want to uncompress your backed up files.

File compression became popular with the proliferation of electronic bulletin boards and information services like CompuServe. (Since compressed files have been reduced to approximately half their original size, the time required to download files through telephone lines is also cut in half.) File compression is also used in commercially available backup programs like Norton Utilities and PCTools.

Today, though, compression programs have increased in value due in large part to the popularity of Windows. A typical Windows application, such as Excel or Word for Windows, can require as much as 10MB of hard disk space—and sometimes more. This kind of disk gluttony can reduce a 120MB hard disk to a stuffed pig in a matter of months.

The problem becomes even more critical with laptop and notebook computers. Hard disks are big, heavy devices. To keep down the size and weight of laptops and notebook computers, manufacturers have had to keep hard disk sizes down as well. So, file compression is just about essential if you use a portable computer.

The most popular file-compression utilities are shareware programs made available by PKWARE, Inc., called PKZIP and PKUNZIP. PKZIP compresses files, while PKUNZIP restores compressed files to their original format. You can recognize files that have been compressed, or "zipped," with PKZIP because they'll have the file extension ZIP. An older version of PKZIP stored files in a different, less efficient compressed format that can be identified with the ARC (for "archive") extension.

Because PKZIP is so popular, many other file-uncompression programs recognize the ZIP (and sometimes the older ARC) compression format.

ON DISK

Using PKZIP and PKUNZIP

PKZIP and PKUNZIP are stored in the PKZIP directory of the *Windows Insider Disks.* A second, Windows-based set of compression/

uncompression programs is stored in the WINZIP directory of the *Windows Insider Disks.* You can use either PKUNZIP or WINZIP to uncompress other programs on the *Windows Insider Disks.*

To use PKZIP, you simply type **PKZIP** at the DOS prompt, along with the appropriate file parameters. Here's the general format:

```
PKZIP [options] [destination path for compressed file] [files/
directories to be compressed]
```

For instance, suppose we want to compress all files in our WordPerfect for Windows directory (WPWIN) and store them in a file called WPERF.ZIP, in the directory WPBACK. Both the source and destination directories are located on the same hard disk. Here's the command we would use:

```
PKZIP C:\WPBACK\WPWIN C:\WPWIN\*.*
```

Unfortunately, this command doesn't delete the original, uncompressed files. So, you'll actually have *less* free disk space until you delete the original files.

Installing a Run-time File-Compression Program

Run-time file-compression programs compress almost all of your hard-disk's contents when you install the compression program. Then, whenever you boot your system, the file-compression program loads a TSR that intervenes whenever an application tries to read from or write to your hard disk. This TSR uncompresses programs and data files whenever you try to access them, and compresses files whenever you store them to disk.

Run-time compression programs are especially useful for portable laptops and notebook computers, since these systems often have relatively small hard disks to conserve the size and weight of the system. But if you use several Windows programs, you'll be surprised at how easily even a large hard disk (100MB or more) can become full. A program like Stacker or SuperStor effectively doubles the capacity of your hard disk without disrupting performance.

If space on your hard disk is limited, and you plan on installing additional applications, you should consider installing a program like Stacker or SuperStor. Chapter 10 includes a demonstration of Stacker.

The two most popular, commercially available run-time disk compression programs are Stacker and SuperStor. Both programs work with Windows 3.1, although some of SuperStor's features must be disabled or modified to successfully run under Windows. Stacker has become the industry standard, because its file-compression format is used by the backup utilities in Norton Utilities, Norton Desktop for Windows, and Central Point Backup.

A run-time compression program is more useful than a compression utility like PKZIP because the compression/uncompression software operates automatically in the background. After you install a program like Stacker or SuperStor, you don't have to take any additional steps to compress and uncompress files.

Backing Up Files

There are probably as many file backup utility programs as there are theories on how and when to back up files on your hard disk. In any case, with the current proliferation of PC viruses, you'll definitely want to have some backup system in place.

If you already use a backup utility, you might not feel you need to read this section although—at the very least—you'll find some useful batch file techniques here.

In fact, we highly suggest you purchase a separate Backup utility to safeguard the files on your hard disk. (Even better, purchase a high-speed tape backup system.) Chapter 10 includes a review of some software products that allow you to back up and restore files both from within Windows and at the DOS level. If you don't have a separate backup utility, you can use the DOS BACKUP command to back up your hard disk periodically. The BACKUP command has a reputation for being relatively slow, unwieldy, and out-of-date. Much of this reputation is deserved. However, you can tweak the DOS BACKUP utility by using it in conjunction with a batch file. In fact, the backup approach we provide below is relatively painless and is easy to set up and use.

The BACKUP command copies files from your hard disk to diskettes, in a compressed format. If you need to return any of these compressed files to their original format, you must use the RESTORE command. Refer to your DOS User's Manual for information on using RESTORE.

The general format of the BACKUP command is:

```
BACKUP <path> <target drive>
```

So, to back up all files in a directory called BUDGET, located under the EXCEL directory, you would enter the following in the Run dialog box:

```
BACKUP C:\EXCEL\BUDGET A:
```

If you don't want to back up your entire hard disk (and you probably won't), you'll need to enter a different BACKUP string for each directory you want to back up.

Chances are that you use a half dozen or more data directories on a regular basis. For example, you might store all your spreadsheets in one or two directories, your word processor files in one or two directories, your database files in one or two directories, and so on. Obviously, it can be quite a pain to issue the BACKUP command for each directory that you want to back up—every time you back up your files. If you back up your files weekly, this process can be positively nerve-frazzling.

We suggest you create a BACKUP batch file that you can launch from within Windows. Simply add a separate BACKUP line for each directory that you want to back up. Here's an example of a BACKUP.BAT file that we created using the MS-DOS Editor:

```
BACKUP C:\WINDOWS
BACKUP C:\WINDOWS\SYSTEM
BACKUP C:\WINWORD\INSIDER /M
BACKUP C:\WINWORD\PERS /M
BACKUP C:\EXCEL\INWORK /M
BACKUP C:\PARADOX\SALES /M
BACKUP C:\PARADOX\CUSTOMER /M
```

The result is this: BACKUP.BAT will execute the first BACKUP line in the file; after all files in this directory have been backed up onto floppies, BACKUP.BAT will execute the second BACKUP line in the file, and so on, until all seven BACKUP commands have been executed.

The first time we use this file, DOS will probably spend an interminably long time backing up the entire contents of all seven of these directories, and you'll probably need several boxes of diskettes. However, notice that, in each directory, we've included the /M switch. This switch tells the BACKUP program to copy only those files that have been modified since the previous backup.

So, for each subsequent time that we run BACKUP.BAT, only files that we've changed will be backed up to floppies. This significantly reduces the time required to back up files and reduces the number of diskettes you'll need.

One other benefit of our BACKUP.BAT file is the ability to edit it easily. Suppose you create a new directory under EXCEL for storing a certain category of spreadsheet files. Or suppose you've deleted one of the directories in BACKUP.BAT. You can use the MS-DOS Editor to open BACKUP.BAT and quickly add new directories, delete unwanted directories, or change the contents of an existing BACKUP line.

HOT TIP

Use a Network Drive to Back Up Your Files

You can avoid the use of backup diskettes altogether if you are running Windows on a network and have a personal network directory that you can use to back up files. Create the backup macro as described above, except specify your network drive and personal directory as the target for the backups.

Then, if something happens to the network file server, your files are safely stored on your hard disk. If something happens to your hard disk, your files are stored safely on the network file server. There's an additional advantage to this approach: your organization's network is probably backed up nightly, providing you with a second level of protection.

Purging Your Backup Diskettes

If you back up files on a weekly basis and use high-density diskettes to make backups, your BACKUP routine can quickly become a costly affair. However, you probably will need to save only two to four "generations" of backup diskettes.

As the files on older generations of backup diskettes become obsolete, you can purge the files by following this procedure:

1. Insert the diskette to be purged into the disk drive.
2. Display the Run dialog box, then type **ATTRIB A:*.* -R**. This turns off the read-only attribute, allowing you to delete files. (If your backup files are stored on a different drive, specify the appropriate drive letter in place of A:.)
3. Use the Delete command from the File Manager's File menu to delete the contents of the diskette.

Warning: If you use the /M switch when you make backups, always maintain a copy of the first generation of backup diskettes you created.

Here's why: the first time you backed up your hard disk or the selected directories, DOS copied *all files* in the indicated directories onto floppy diskettes.

In subsequent backups, DOS only copied files that had been modified. Some files, especially program files, are rarely if ever modified. Backup versions of these files will only exist on the first set of backup diskettes you created. If you want to restore programs as well as data, chances are you'll need this first set to restore all program files.

ON DISK

Using FLEXIBAK

The *Windows Insider Disks* include a menu-driven program called FLEXIBAK that runs at the DOS level and allows you to schedule backups and to select files and directories that you want to back up. This program is stored in the FLEXIBAK directory.

Using DOS Programs with Windows: A Review

The following table lists most of the files that are available with DOS 5, and indicates when and how to use them under Windows. We haven't included some of the more obsolete or little-used commands, such as EDLIN and DEBUG. However, all other file and disk-management commands are included in Table 9.1.

Table 9.1 DOS Commands and Their Relationship to Windows

Command	*Command Type*	*Usage*
ANSI.SYS	Device Driver	Include in CONFIG.SYS to provide additional keyboard and screen support features.
ASSIGN	External Command	Include in AUTOEXEC.BAT, or start from Windows Run dialog box or execute from the DOS Prompt application; assigns a different drive letter to an existing drive.
ATTRIB	External Command	Start from Windows Run dialog box if you want to reset DOS attributes; run from DOS Prompt window if you want to view existing attributes (by typing ATTRIB without any parameters).
BACKUP	External Command	Start from Windows Run dialog box, DOS Prompt window, or from within a batch file to back up multiple directories.

Table 9.1 DOS Commands and Their Relationship to Windows (Continued)

Command	*Command Type*	*Usage*
CD	Internal Command	Use COMMAND.COM to run from Run dialog box, run from DOS Prompt, or create a batch file; changes the default directory.
CHKDSK	External Command	Run from DOS Prompt window. **Warning:** never run CHKDSK with the /F switch from within Windows. Exit Windows first.
COMMAND.COM	DOS Environment	Can be started from the Windows Run dialog box, along with the /C switch, to run internal DOS commands (such as DIR, REN, COPY, and DEL).
COPY	Internal Command	Use COMMAND.COM or create a batch file to run COPY from the Run dialog box; or use COPY from the DOS Prompt window.
DIR	Internal Command	Use COMMAND.COM or create a batch file to run DIR from the Run dialog box; or use DIR from the DOS Prompt window.
DOSKEY	External Command	Add in AUTOEXEC.BAT or start from Run dialog box or DOS Prompt to maintain a history of previously used DOS commands.
EDIT	External Command	Loads MS-DOS Editor. Type **EDIT** in Run dialog box to load the editor.
FC	External Command	Run from MS-DOS Prompt to compare two files or two sets of files.
FIND	External Command	Run from MS-DOS Prompt to find a text string in a file or collection of files.
HELP	External Command	Start from Run dialog box or from DOS Prompt window to display a help list for DOS commands.
MD	Internal Command	Use COMMAND.COM to run from Run dialog box, run from DOS Prompt, or create a batch file; creates a directory.
MIRROR	External Command	Place in AUTOEXEC.BAT file or start from Run dialog box or DOS Prompt window.
PRINT	External Command	Start from Run dialog box or from DOS Prompt window to print an ASCII file.
RD	Internal Command	Use COMMAND.COM to run from Run dialog box, run from DOS Prompt, or create a batch file; removes a directory.

Table 9.1 DOS Commands and Their Relationship to Windows (Continued)

Command	*Command Type*	*Usage*
REN	Internal Command	Use COMMAND.COM to run from Run dialog box, run from DOS Prompt, or create a batch file; renames one or more files.
RECOVER	External Command	**Warning:** Always exit Windows before running RECOVER, which tries to restore data located in bad sectors of a disk.
RESTORE	External Command	Start from Run dialog or from DOS Prompt window, or create a batch file; restores files that have been backed up to diskette using BACKUP.
SUBST	External Command	Start from AUTOEXEC.BAT or execute from RUN dialog box to assign a shorter path to a lengthy set of subdirectories. **Warning:** SUBST can confuse the Windows Setup program and the File Manager; avoid using SUBST when running Windows.
UNDELETE	External Command	Start from Run dialog box or from MS-MS-DOS Prompt; allows you to restore files that have been deleted accidentally.
UNFORMAT	External Command	Start from Run dialog or DOS Prompt window *only* if you are unformatting a diskette. Exit Windows before unformatting a hard drive.
XCOPY	External Command	Start from Run dialog box or from DOS Prompt window; copies files and subdirectories.

CHAPTER

10

Third-Party File-Management Tools

If you've tried sorting through magazine advertisements for Windows add-on products, you might have wondered whether Windows is really all that great a program. If Windows can do everything that Microsoft brags it can do, why the need for all these dozens—hundreds, actually—of Windows utilities?.

The truth is that the Windows API makes it easy to enhance the built-in capabilities of Windows through additional software products. And isn't that the way the computing world *should* be? Imagine if it was that easy to add 40 horsepower to your car's engine or to snap on an additional bedroom to your house or apartment.

The key is to figure out what features you need that Windows doesn't already supply. And if your individual tastes in software incline you toward a file manager, task manager, or desktop manager approach that differs from the ones you'll find in Windows, you might want to consider some alternative products.

That's what this chapter is all about. We want to provide you with a preview of some of the additional or alternative features available through some leading commercial Windows utilities.

Categorizing Third-Party Utilities

To keep the size of the chapter from mushrooming out of control, we're going to limit our discussions to five basic areas:

- Disk optimization
- File backup
- Run-time compression/uncompression
- Desktop customization and automation
- File and task management

After we've provided a brief explanation for each of these five areas, we'll give you a tour of four products that support features for one or more of these areas:

- PC Tools and Central Point Backup for Windows
- Stacker
- NewWave
- Norton Desktop for Windows

Most of these products can be purchased for about $100 (each) or less.

Of course, we're being somewhat arbitrary in the way we're limiting our lists. For instance, we're not covering shareware products here (there are just too many), nor are we going to discuss features that come under the heading of bells and whistles—such as interesting or humorous screen savers and sound drivers.

And there are other laudable products that we just don't have space to discuss. Our intent is to give you a good feel for the basic features and categories of products that you can buy off the shelf. We're not trying to be exhaustive in our reviews. So keep in mind that every product we describe in this chapter has at least one—and usually more than one—major competitor. We're going to play the role of tour guide and leave it up to you to play the role of smart shopper.

Disk Optimization

We covered hard disk optimization briefly in Chapter 4. A disk optimization program (also called a disk compaction program or a defragmenter) moves different parts of each file on your disk into sequential, adjoining locations. This rearrangement of files makes it easier for the read/write heads of your hard drive to find and retrieve a file, since the heads don't have to hop around from one sector to another in search of different parts of a single file.

A disk optimization program makes it possible for you to increase the size of your permanent swap file, which can only contain contiguous free space on your hard disk.

You should optimize your hard disk at least every six months because, over time, DOS will fragment your files into far-flung locations—even if you've already optimized your disk in the past. Before you optimize your hard disk, you should remove your permanent swap file, if you have one. You can then re-create the swap file—usually a larger one—after you've optimized your disk. Chapter 4 explains how to disable and create a permanent swap file. Disk optimization is available in PC Tools, Norton Desktop for Windows, and Stacker.

Warning: Never optimize your hard disk while Windows is running; always exit Windows before you attempt to use any disk optimization program.

File Backup

Although you can use the Windows File Manager to copy files and entire directories easily onto floppy disks for backup purposes,

there are several problems with this approach—all of which suggest the need for a more specialized backup system.

First, you can't copy files across multiple diskettes. For instance, if you're trying to copy the contents of an entire directory, and the diskette you're copying to becomes full, Windows will halt the copy operation and prompt you for another diskette. If you're copying large files, you might waste quite a lot of diskette space under this approach.

Second, when you use the File Manager to copy files and directories, Windows does a byte-for-byte copy. That is, the size of each copied file on the floppy diskette is the same as the file's size on the hard disk. No file compression takes place.

Third, you can't use Windows to automate the scheduling of backups. Of course, you could enter alarms into a Calendar file to signal when to backup files, but even then you have to select and copy all files yourself.

Fortunately, there are several backup programs available that work well with Windows, and do their work at much higher speeds than the DOS BACKUP and RESTORE utilities. Third-party backup programs allow you to span files across as many diskettes as you need to complete a backup operation. These programs also store the backup versions of files on diskettes in a compressed format to save space.

Some backup programs also allow you to schedule backups so that they are performed automatically, on a regular basis; backups can even be performed unattended if you are backing up your hard disk to a network, to a tape drive, or to a separate location on your hard disk.

Norton Desktop for Windows and PC Tools both include backup programs that can be used under Windows 3.1. However, it's important that any backup program you use also allows you to restore backed up files from DOS as well as from within Windows. Suppose you accidentally erase your entire hard disk, or your hard disk crashes. In these types of situations, you probably won't be able to run Windows. So, it's essential that you be able to restore Windows and all other programs and data files at the DOS level. Both Norton Desktop for Windows 2.0 and PC Tools 7.1 support backup and restore features from within Windows as well as from the DOS prompt.

Run-Time Compression/Uncompression

We introduced run-time compression/uncompression in Chapter 9. The term run-time in this context refers to a program that runs in the background, storing files in a compressed format whenever you save them to disk, and uncompressing files whenever you need to read them.

Run-time compression software is invaluable if your hard disk is bloated with programs and data files. Windows, especially when you run in enhanced mode, doesn't like a full hard disk because it becomes difficult— if not impossible—to create temporary and permanent swap files. And, of course, if your hard disk is stuffed to the gills with programs, you can't add new applications.

A run-time program like Stacker or SuperStor can compress the contents of your hard disk in half—effectively doubling the capacity of your hard disk. This feature is almost a must-have for laptop and notebook computers, which usually have hard drives that store no more than 60MB. Both Stacker (which we'll explain in this chapter) and SuperStor are Windows-aware and can coexist with Windows swap files. A Windows-aware program either recognizes the way Windows uses memory and system resources and stays out of the way or uses a set of custom PIFs that tell Windows how to allocate memory and system resources for the application's use. Both Stacker and SuperStor supply their Windows awareness internally in their program code. You don't need separate PIFs for these utilities.

Desktop Customization and Automation

If you've always relied on DOS and have never used a Macintosh or a software environment like GEM or any other GUI, you might think Windows is heaven-sent. But if you have previous experience with the Macintosh GUI or some other powerful graphical user interface, you might feel somewhat restricted by the desktop management limitations of Windows. If so, you should know about some alternatives that are available.

One of the biggest problems with Windows is its need to divide system management into two separate shells—the Program Manager and the File Manager. The Macintosh computer, by contrast, lets you organize groups, programs, and files in a more seamless way—all under a single graphical shell.

If you're happy with the performance of the Program Manager, you might not be interested in installing a desktop management system like Norton Desktop for Windows or NewWave. These systems make it easier to combine file management and program management under one roof. For instance, with Norton Desktop for Windows, you can create program groups within program groups, and you can drag files to a wastecan (much like the Trashcan on the Macintosh) for easy deletion. NewWave also provides a unique organization scheme that makes it unnecessary to search through directories, trees, or program groups in order to launch an application or view a file.

Norton Desktop for Windows and NewWave both serve as a replacement for the Program Manager and the File Manager. These packages provide some great ways to customize the desktop to make it easy to launch applications and to automate frequently performed tasks (without struggling with the Windows Recorder).

File and Desktop (Task) Management

With programs like Norton Desktop for Windows and NewWave, file management and desktop management blend into a single, seamless environment. Other systems, like PC Tools and Synergy, let you use the Program Manager but provide an alternative file management shell.

If you've got multiple file managers running on your system, a program like hDC Power Launcher lets you switch between them frequently so that you can choose the most effective file manager for the task at hand.

File management systems like PC Tools, Norton Desktop for Windows, and NewWave all provide simplified techniques for finding and launching applications quickly.

The four packages we'll describe next all provide features in at least two of the five areas we've just reviewed—and with varying degrees of success. That's the purpose of describing the software; you'll be able to get a sense for which packages provide which features, and how well. We're not going to try to sell you on any of these Windows add-ons. But if you'll take a close look at these popular applications, you're likely to come away with a strong understanding of the additional or enhanced features that are available.

Armed with this information, you can determine for yourself which add-ons you might want to buy, and you'll probably find it easier to assess the features promoted in the dozens of other Windows utilities and accessories available to Windows users like yourself.

Central Point's PC Tools and Backup for Windows

The ever-popular PC Tools has been around for almost a decade, and gained fame chiefly by providing a menu-driven file management shell that made it unnecessary to use the more cryptic DOS commands, like

COPY, DEL, REN, and so on. Other major features of previous versions of PC Tools have included a fast, menu-driven backup program, and diagnostic routines for evaluating memory and hardware and for optimizing hard disks.

PC Tools 7.1 includes all of these features and provides additional support for Windows 3.0 and Windows 3.1. (Central Point Backup 7.2 is built into PC Tools 7.1, but is also available as a separate product.) In the following sections, we'll focus on the features that are most useful to Windows users.

PC Tools Provides a Group Window for Launching Its Utilities from within Windows

When you install PC Tools 7.1, the installation program detects the presence of Windows and creates a program group for PC Tools. Most of the utilities available with PC Tools 7.1 can be launched by clicking on the appropriate icon in the PC Tools group window.

Figure 10.1 is an example of the PC Tools group window. Your PC Tools group window might contain more or fewer program icons, depending on which utilities you elect to install.

This group is a bit deceptive, because only a few of the icons launch true Windows programs. Most of the icons launch DOS programs, all of

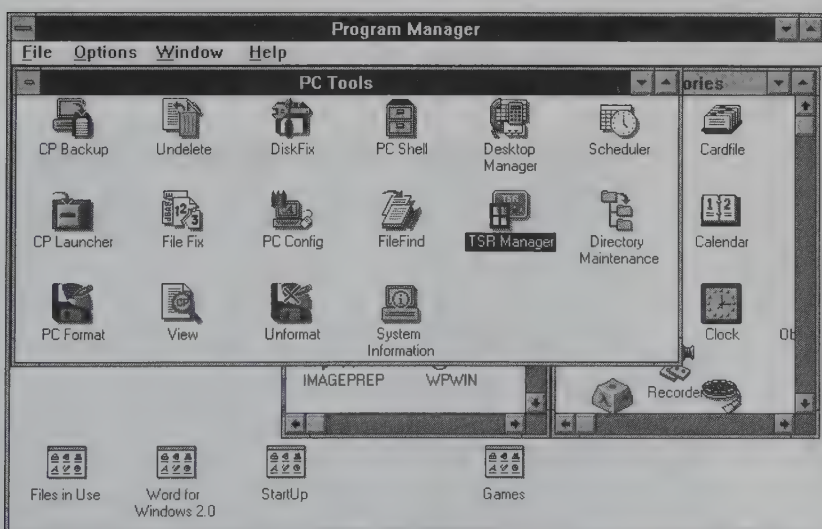


Figure 10.1 Use this group window to select different utilities available with PC Tools.

which have their own PIFs. Of the sixteen programs shown in Figure 10.1, only CP Backup, Undelete, Scheduler, and CP Launcher are Windows programs. All other programs will run under Windows, but they are in reality DOS programs.

Using CP Backup

PC Tools offers a superior Windows backup program and also a version that you can run under DOS, if necessary. Central Point Backup is included within PC Tools, but you can also purchase it separately, for less than half the cost of the full PC Tools package.

Figure 10.2 shows the Central Point Backup dialog box that appears when you choose the CP Backup icon. Choose the Backup button to set up options for backing up selected directories and files.

You can use the Restore button to restore files if you are able to run Windows following a hard-disk problem. The Compare button allows you to double-check that the files you've backed up have not been corrupted due to some file transfer problem or due to bad sectors on the floppy diskette. You can verify your backups by comparing the backed up files with the original versions.

Figure 10.3 shows the main Central Point Backup dialog box that appears when you click on the Backup button. You can set up all backup options from there. In Figure 10.3, we've set up the C:\WINWORD\INSIDER directory to be backed up to 1.44MB floppies on drive B.

Figure 10.4 shows the Modify Backup Schedule Item dialog box, which you can use to schedule weekly backups that are performed by

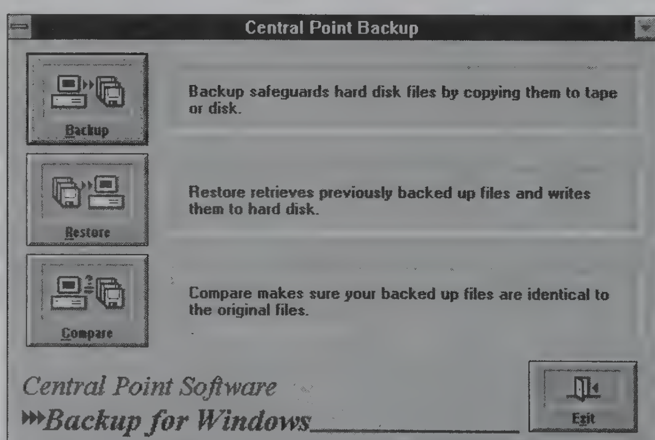


Figure 10.2 Use this dialog box to specify whether you want to use Central Point Backup's backup, retrieve, or compare options.

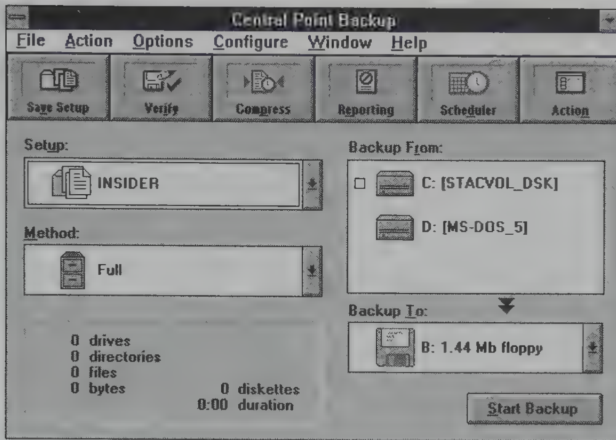


Figure 10.3 The full range of backup options is available from this dialog box.

PC Tools automatically. Figure 10.4 shows that we've scheduled the directory C:\WINDOWS\INSIDER to be backed up each Friday at 5:30 P.M. To ensure that backups are performed automatically, simply drag the Scheduler icon from the PC Tools group into the StartUp group and run it minimized.

Using the Undelete Utility to Restore Accidentally Deleted Files

If you accidentally delete a file, either from within a Windows application, from the Windows File Manager, or from DOS, you can use the PC Tools Undelete feature to restore the file, without leaving Windows.

Figure 10.5 shows the dialog box that appears when you choose the Undelete icon from the PC Tools program group. This dialog box shows two files in the C:\WINDOWS directory that have been recently deleted. DOS removes a file from its file allocation table by removing the first

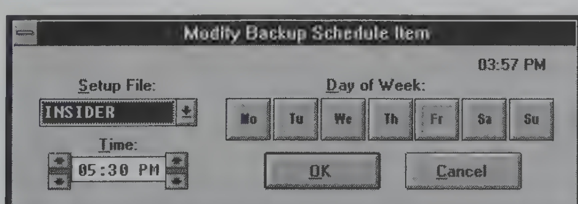


Figure 10.4 Use this PC Tools dialog box to schedule regular backups for a drive or selected set of directories or files.

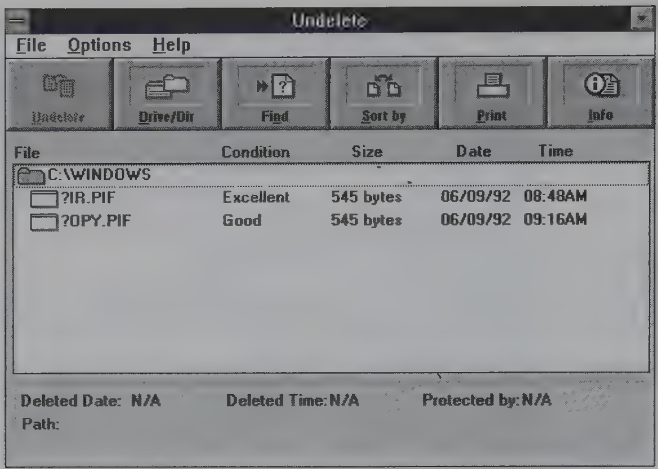


Figure 10.5 Use the PC Tools Undelete dialog box to restore files that you have deleted accidentally.

character of the file name. So, if the file space used by a deleted file has not yet been claimed by some other file, you can undelete the file by restoring the first letter of the file name.

The Undelete dialog box lets you view and select files that you want to delete and lets you quickly switch to different directories to view other deleted files. You can also search for the location of a specific file that you want to delete.

Using the PC Tools Shell as an Alternative to the File Manager

The PC Tools Shell isn't a true Windows application, but if you run it in a window (in enhanced mode) you can use it in a manner that's almost identical to the Windows 3.1 File Manager. In fact, the PC Tools Shell offers all the file management capabilities as the File Manager, plus many other features.

The Microsoft Windows 3.1 File Manager looks and operates a lot like the shell used in Central Point's PC Tools. So, when you display the PC Tools Shell, you'll probably feel comfortable with it immediately.

One significant drawback is that the PC Tools Shell is not a true Windows program. You can display it in a window in enhanced mode, although you can't exit Windows until you've closed the Shell. Figure 10.6 shows the PC Tools Shell displayed in a window. You'll probably immediately recognize the resemblance between this file management shell and the Windows File Manager.

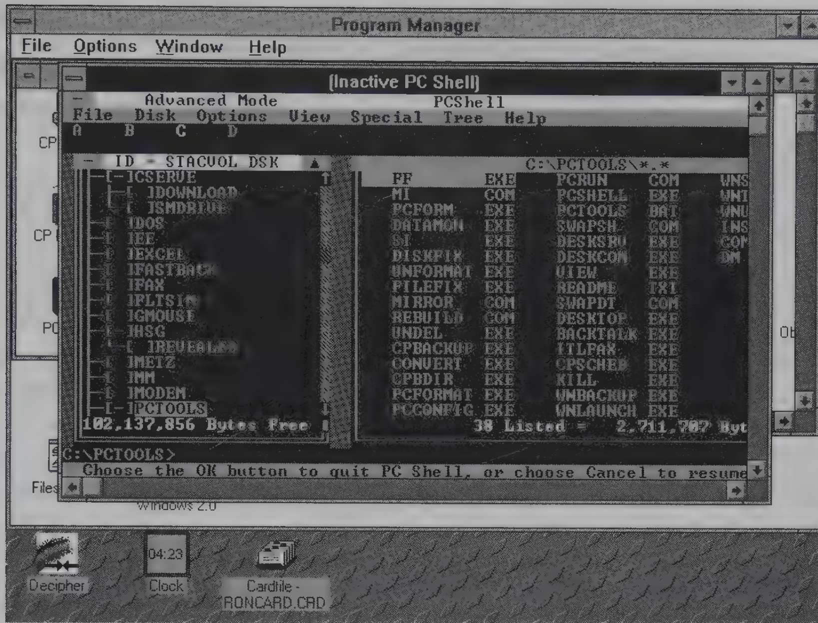


Figure 10.6 The PC Tools Shell resembles the File Manager, but offers additional features.

But the PC Tools Shell offers many features that aren't available from within the File Manager. For instance, you can view files in ASCII mode, view files from within any directory, prune selected portions of a directory tree, encrypt files so that they can't be accessed by unauthorized users, compare two versions of a file, and much more.

HOT TIP

Turn Off the PC Tools Graphics Font if You Run the Shell in a Window

When you first try to display the PC Tools Shell within a window (in enhanced mode), you'll see a lot of garbage characters. This happens because Windows doesn't recognize the graphics font used by PC Tools. But you can easily disable this font by adding the /NF switch to the Command Line dialog box for the PC Shell icon. (The Command Line should read PCSHELL.PIF /NF.)

You can also use the PIF Editor to add the /NF parameter in the Optional Parameters box for the PCSHELL.PIF file. (This solution is more permanent, since the fix will remain in effect even if you accidentally delete the PC Shell icon and then later re-create it without including the switch.)

Using the CP Launcher to Load an Application from within Any Other Application

The CP Launcher lets you start any application without the need to switch to the Program Manager.

If you drag the CP Launcher icon (shown earlier in Figure 10.1) into your StartUp group, PC Tools will add the CP Launcher to every application's Control menu. You can then use CP Launcher to start an application from within any other application. In Figure 10.7, we first clicked on the Control-menu icon in the Word for Windows title bar, then we clicked on CP Launcher to display the cascaded menu. (Note that you can use Launcher to run CP Backup from within any application.)

Now, when we click on the Run option, Launcher displays the Run dialog box shown in Figure 10.8. This dialog box pops up directly within Word (or whatever application you're currently in). One slightly annoying feature of CP Launcher is that the Files list box displays only .EXE files for the current directory, which allows you to start only Windows applications. It would be more helpful if Launcher also listed PIF files, which would allow you to start non-Windows applications as well.

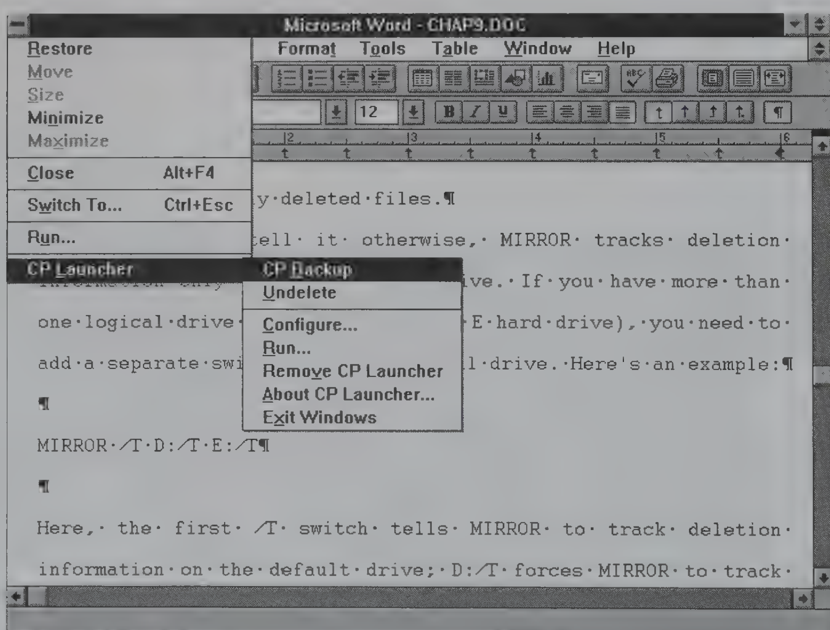


Figure 10.7 When you run CP Launcher, it is available from the Control-menu of any open application.

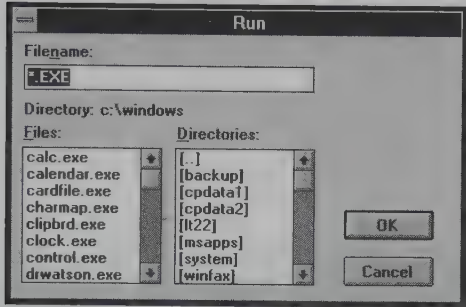


Figure 10.8 CP Launcher's Run dialog box lets you conveniently launch Windows applications from directly within another application.

However, this is a minor complaint; you can easily change the default extension (to .PIF, .BAT, or .COM) in the Filename box. Or if you know the name of the PIF for the DOS application you want to start, you can enter it directly within the Filename text box. The Launcher is especially useful if you routinely use a Windows accessory—like the Calendar, Cardfile, or Calculator—and want to start it quickly, without having to switch to the Program Manager.

PC Tools Includes a Disk Optimizer

Use the PC Tools COMPRESS program to optimize your hard disk periodically. Remember, disable your permanent swap file before you run COMPRESS, because the COMPRESS program cannot manipulate the hard disk space claimed by a permanent swap file.

The Disk Optimization utility that's built into PC Tools, called COMPRESS, doesn't appear in the PC Tools group window because you can't run it from within Windows. In fact, you should never run any disk optimizer from within Windows. Even if you try to run the PC Tools disk optimizer by using the Run dialog box, PC Tools will display a message telling you that the program can't be run from within a multitasking environment.

To run the PC Tools disk optimization program, you exit Windows, then type **COMPRESS** at the DOS prompt. The COMPRESS program displays a map representing your hard disk usage, and indicates the percentage of your hard disk that has fragmented files. If more than one percent of your hard disk has been fragmented, COMPRESS will recommend that you optimize it.

Warning: Never run the PC Tools COMPRESS program on a disk drive that has been compressed using Stacker, SuperStor, or some other run-time disk-compression program. If you want to optimize a drive that has been compressed with one of these programs, you must use the disk optimization program provided with the compression software.

Using Stacker

Programs like PKZIP and d'Compress for Windows (formerly called ZIP Manager) work great when you want to tuck away infrequently used programs or directories in order to reclaim more of your hard disk space. These file-compression programs let you compress files into about half the original space required to store the files. However, if you want to reuse a compressed file, set of files, or set of directories, you'll need to take time to uncompress them.

Run-time compression programs, like Stacker and SuperStor, perform their compression and uncompression routines automatically whenever you read from or write to a file, including program files. Stacker has become the industry standard for run-time compression software, so we'll describe its use in the following sections.

Stacker Installation Is Easy, but Not Always Error Free

Stacker's installation program is menu-driven and very easy to use, for the most part. In most installation situations, you won't have any problem restarting your system. However, in a few cases, you might have to take some simple steps to get Stacker to operate properly. The key to doing this lies in understanding how Stacker affects your hard disk.

When Stacker installs itself on your hard disk, it examines your CONFIG.SYS file to determine which drivers you've included there. Stacker then creates a separate logical *unstacked* (uncompressed) drive on your hard disk (usually drive D or E). This uncompressed drive is where Stacker stores its own STACKER directory and files, your DOS directory, your CONFIG.SYS and AUTOEXEC.BAT files, and any driver files that need to be loaded by CONFIG.SYS.

So, what's the purpose of the unstacked drive? Simple. Your computer can't start if your entire hard disk contains only compressed files. Even Stacker wouldn't be able to start. The unstacked drive provides a place to store files, in an uncompressed format, that are required to start and configure your computer system.

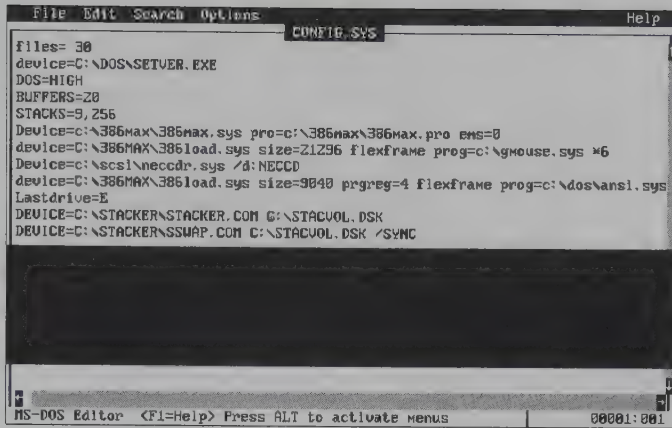


Figure 10.9 This CONFIG.SYS includes the lines required to run Stacker and to configure a hard disk into a stacked and unstacked drive.

Figure 10.9 shows an example CONFIG.SYS file that results after Stacker has been installed. Stacker copies this new file onto your unstacked drive. This drive (probably drive D) is actually treated by DOS as drive C until the Stacker drivers are loaded from within CONFIG.SYS.

You'll notice that the Stacker device drivers, STACKER.COM and SSWAP.COM, appear at the end of CONFIG.SYS. This is done because a few, very few, drivers will not operate correctly if they are installed after Stacker. The Norton cache driver is one such example.

The Location of Stacker's Device Drivers in CONFIG.SYS Is Important

The location of the Stacker drivers in CONFIG.SYS can be critical to the ability of your system to boot correctly. We'll demonstrate this point with an example.

Figure 10.10 shows the uncompressed D drive that Stacker created when it installed itself on one of our systems. You'll note that Stacker copied the contents of the 386MAX directory onto the D drive, because it detected that the 386MAX memory-management drivers need to be installed at startup time. In fact, Stacker copies any files that it finds in CONFIG.SYS onto the unstacked drive to ensure that all drivers will install correctly. So far, so good.

However, a problem can arise if one of your drivers directly calls some other file, typically a COM file. For instance, in our example, the 386MAX loader needs to use a file named XLAT.COM. But this file doesn't appear

```
D:\>dir

 Volume in drive D is MS-DOS_5
 Volume Serial Number is 17ZA-75EE
 Directory of D:\

SCSI         <DIR>      03-30-92    6:12p
DOS          <DIR>      05-10-91    2:47p
STACKER      <DIR>      06-04-92    2:49p
386MAX       <DIR>      05-06-92    9:21a
COMMAND  COM      47845 04-09-91    5:00a
GMOUSE   SYS      21296 03-10-91    3:33p
STACKER  LOG       2206 06-04-92    3:36p
AUTOEXEC BAT        494 05-15-92    8:18p
AUTOEXEC STC        482 05-15-92    8:18p
CONFIG   SYS        433 06-11-92    8:00p
CONFIG   STC        368 06-04-92    2:50p
CONFIG   _SB        433 06-04-92    4:02p
       12 file(s)      73557 bytes
                     4481024 bytes free

D:\>
```

Figure 10.10 This is the contents of an unstacked drive, which contains all files required to start the computer system and the Stacker program.

in the CONFIG.SYS file, so Stacker neglects to copy it to the unstacked drive during installation.

As a result, the file is stored on the compressed drive. However, because the Stacker DEVICE= lines don't actually appear until after the DEVICE= lines that install the 386MAX drivers, 386MAX won't be able to find XLAT.COM. The XLAT.COM file is compressed, and Stacker can't uncompress it until the Stacker drivers have been loaded. But these drivers don't get loaded until *after* 386MAX has already tried to get itself up and running—unsuccessfully.

So, what's the solution to this circus act? There are two possible solutions, actually. If you discover that Stacker has overlooked a required driver file, you can simply copy it to the unstacked drive and then restart your system. Or, you can move the DEVICE= lines for Stacker near the top of CONFIG.SYS, directly under the FILES= and BUFFERS= lines.

We've illustrated this problem because it is typical of the driver problems you can encounter with Stacker. Most of these problems are easily solved—as long as you recognize how Stacker works. This issue becomes especially important when you install a new device after you've already installed Stacker.

If you place a DEVICE= line *after* the Stacker DEVICE= lines in CONFIG.SYS, the driver file(s) for this device must appear on the stacked drive—which typically becomes the C drive as soon as DOS installs the Stacker drivers.

But if you place a DEVICE=line *before* the Stacker DEVICE= lines, then the driver files(s) for this device must appear on the unstacked drive, so that DOS can find them in uncompressed format prior to Stacker's installation.

You Cannot Create a Permanent Swap File on a Stacked Drive

Stacker can trap any attempt by Windows to read to or write from disk—even when you're using 32-bit disk access in enhanced mode. For this reason, Windows can create and use application and temporary swap files on a stacked drive. However, when Windows tries to create a permanent swap file, it won't recognize hard disk space on a stacked drive; you must create a permanent swap file on a stacked drive.

The Stacker installation program is smart enough to recognize whether you've already created a permanent swap file for Windows' use. If Stacker detects a permanent swap file, the setup program will then ask you to allow it to remove the file so that it can optimize your hard disk. Stacker displays a message letting you know that it will reserve space on the unstacked drive so that you can re-create the permanent swap file later.

Unless you very recently (perhaps a few weeks ago) optimized your hard disk, you should let Stacker remove your swap file so that it can defragment your hard disk as effectively as possible. Unfortunately, when Stacker removes your permanent swap file, it just notes the size of the current swap file and allocates *only* this much space on the unstacked drive for the creation of a new, permanent swap file.

This allocation approach presents a problem, because Windows 3.1 determines the size of a permanent swap file based on the total amount of contiguous hard drive space available. If the amount of contiguous free space is limited—20MB or less—Windows will allow you to create a swap file that is no more than half the size of your total available contiguous space. Windows enforces this restriction to ensure that sufficient space is available for storing data files and for installing new applications.

So, if you have a permanent swap file that is, say, 10MB in size before you install Stacker, you'll probably only be able to create a new permanent swap file that is about 5MB or less in size because Stacker will only give Windows 10MB of uncompressed, contiguous free disk space.

Increasing the Size of Your Unstacked Drive to Enlarge Your Permanent Swap File under Stacker

If you want Windows to recognize more optimized, free disk space for use in creating a permanent swap file, you can increase the size of your unstacked drive. You can do this easily, by running Stacker's SDEFRAG utility with the /G switch. You should

run SDEFRAG /G immediately (or soon) after Stacker has been installed successfully.

The SDEFRAG utility optimizes your hard disk. When you include the /G switch, Stacker lets you increase the size of your uncompressed drive after the optimization routine has been completed.

Warning: Disable SMARTDrive before you run the SDEFRAG (or any disk optimization utility). SMARTDrive might try to cache disk contents while SDEFRAG is running, which can cause SDEFRAG to optimize the disk incorrectly. To disable SMARTDrive, add a semicolon or REM at the start of the line in AUTOEXEC.BAT that loads SMARTDrive, then reboot your computer.

If you run SDEFRAG /G immediately after you've installed Stacker, your hard disk will be fully optimized and Stacker will display, almost immediately, the screen shown in Figure 10.11. (If you run SDEFRAG several weeks or more after you previously optimized your hard disk—especially if you've deleted or added several dozen files or more—you might have to wait 30 minutes or longer before Stacker allows you to add unstacked space.)

When the screen shown in Figure 10.11 appears, highlight the "Make more uncompressed space available" option and press **Enter** to tell Stacker that you want to enlarge the size of the unstacked drive. Stacker will then display the screen shown in Figure 10.12. This screen shows the current size of the unstacked drive and the maximum size you can create for the unstacked drive. If possible, you should specify a number twice as large as the size you want to use for your permanent swap file.

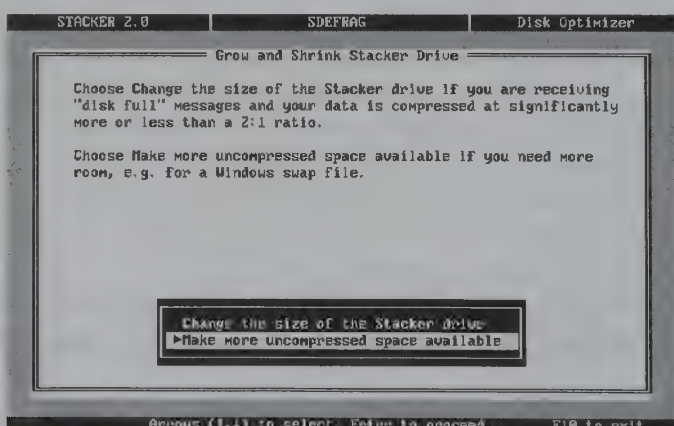


Figure 10.11 Use this menu to tell Stacker to increase the size of your uncompressed drive and, consequently, the size of your permanent swap file.

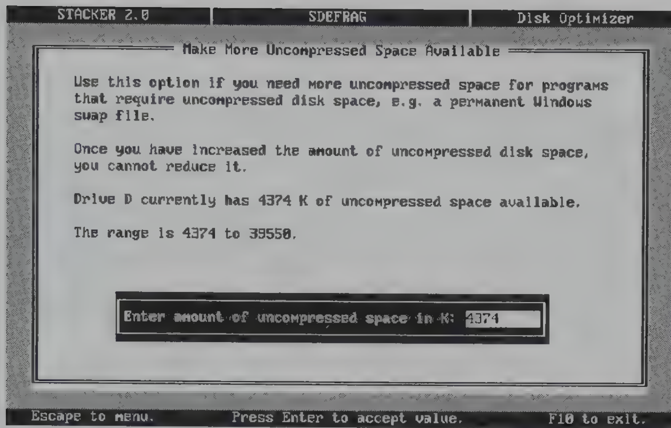


Figure 10.12 This screen lets you specify the size of your unstacked drive.

After you've increased the size of your unstacked drive, follow the procedures we've described in Chapter 4 to increase the size of your permanent swap file. *Make sure you create the swap file on the uncompressed drive* (probably D or E while Stacker is running).

The drawback to this technique is that Stacker will not allow you to decrease the size of your unstacked drive after you've increased it. But in many cases, the benefits that you gain in increasing the size of your swap file will outweigh this limitation. And, with extra space available on your unstacked drive, you'll easily be able to add new drivers to your unstacked drive.

Stacker Needs to Track Any Changes You Make to AUTOEXEC.BAT and CONFIG.SYS

Any time you make a change to AUTOEXEC.BAT or CONFIG.SYS (and that includes changes that a newly installed application makes to these files), Stacker will want to update its copy of this file. Always let Stacker make this update.

The SSWAP.COM driver, which is installed in your CONFIG.SYS file, compares the versions of CONFIG.SYS and AUTOEXEC.BAT stored on both the stacked and unstacked drives—each time you boot your system. If SSWAP.COM recognizes that you or some other program has made a change to CONFIG.SYS or AUTOEXEC.BAT, it will ask you whether it can update the change to the other version of the file. *Always let Stacker perform this update.* This ensures that your system boots correctly regardless of the point in CONFIG.SYS at which the Stacker drivers load.

Otherwise, confusion and disaster can result if drivers and TSRs have been added to different CONFIG.SYS or AUTOEXEC.BAT files.

HOT TIP

How to Load STACKER.COM

STACKER.COM is the controller program for Stacker, and remains resident in memory to trap all disk I/O requests. STACKER.COM is essentially the traffic cop that makes sure files get uncompressed when an application tries to read from the files, and compresses files when they are written back to disk.

STACKER.COM can be loaded into upper memory. If you are using DOS 5 to load drivers into the UMA, you can use the DEVICEHIGH command, as in this example:

```
DEVICEHIGH=C:\STACKER\STACKER.COM C:\STACVOL.DSK
```

If you are using QEMM 386, 386MAX, or some other memory manager, use the appropriate load command to load STACKER.COM into upper memory.

The SSWAP.COM program, which is also loaded from CONFIG.SYS, should never be loaded into the UMA. SSWAP.COM simply compares the CONFIG.SYS and AUTOEXEC.BAT files on the stacked and unstacked drivers at startup; after SSWAP.COM has checked to make sure these files are the same on both the stacked and unstacked drives, it removes itself from memory. Since SSWAP.COM does not remain resident, you don't need to load it into the UMA.

Stacker Works with Most Cache Programs

Stacker does not conflict with most of the popular disk cache programs, including SMARTDrive 4.0.

Stacker works well with most cache programs, including the SMARTDrive 4.0 TSR included with Windows 3.1. In fact, you can effectively double the capacity of SMARTDrive when you run it under Windows.

A cache utility like SMARTDrive operates by using memory to emulate the way a disk drive is mapped. Since SMARTDrive caches data on a compressed drive, you effectively double the capacity of the cache, because Stacker will cache data to SMARTDrive in compressed format and will only uncompress the data when it is read into RAM or written back to the hard disk. (However, you need to make sure SMARTDrive caches the correct hard drive. See Chapter 16 for more information.)

Stacker Cooperates with Most Backup Utilities

Most of the popular file backup utilities—including DOS BACKUP, Fastback, Norton Backup, and CP Backup—work seamlessly with Stacker.

Stacker uncompresses data whenever it is read, even if the data is read by a backup utility. So, virtually all backup software will work with Stacker; the Stacker software uncompresses files when the backup program reads them, then lets the backup program manage the compression of files to the target disk. By contrast, when you restore files to a stacked disk, Stacker writes the files to disk in its own compressed format.

HOT TIP

If You Have a Stacked Drive, Always Use SDEFRAG to Optimize Your Hard Disk

If you try to optimize a stacked hard disk using the disk optimization programs available with PC Tools or Norton Desktop for Windows, you'll probably receive a warning of some kind. Heed this warning. If you have installed Stacker, SuperStor, or any other run-time compression software, you *must* use its own, built-in optimization program if you want to defragment your hard disk.

For instance, with Stacker, you must use the SDEFRAG utility to optimize your stacked hard disk. If you accidentally run the disk optimization routine available with Norton Desktop for Windows or PC Tools, you should immediately run CHKDSK, followed by Stacker's SCHECK utility. If either routine uncovers lost clusters or other problems, run CHKDSK and SCHECK, in that order, using the /F option for both programs.

Warning: Never run CHKDSK /F or SCHECK from within Windows.

Stacker Is Compatible with Most Disk Repair Utilities

Stacker is compatible with most of the popular utilities that recover files and data from damaged sections of your hard disk.

Stacker works seamlessly with Norton Disk Doctor, PC Tools' DiskFix utility, and SpinRite. These utilities detect problems on hard and floppy disks and allow you to recover files that might in part be located within bad disk sectors. If the problem files are located on a stacked drive, the disk recovery utilities in widespread use will still work effectively.

NewWave 4.0

NewWave 4.0, by Hewlett Packard, is a desktop environment that you can use in place of both the Windows Program Manager and File Manager. Keep in mind, though, that NewWave is a Windows product and *requires* Windows in order to run. We've talked to many users who believe that NewWave is a replacement or alternative to Windows. That's not right. If you're not running Windows, you can't run NewWave.

NewWave is ideal for you if the way you want to use your PC fits into one or more of the following categories:

- You have a Macintosh background and want a more "Mac-like" interface.
- You can identify many tedious tasks that you do repeatedly, and that you would like to automate more easily and more effectively than you could do with the Windows Recorder.

NewWave is very powerful and, in our view, is the big sleeper in the Windows utilities arena (although Norton Desktop for Windows 2.0 runs a close second). If you're used to the Macintosh GUI and you've always wanted to emulate this environment on your PC, you'll feel right at home with NewWave. You'll probably be tempted to say, "At last, somebody got it right!"

Curiously, few Windows users seem to know about NewWave, and that's a shame. The NewWave environment fills just about every desktop and file management gap in Windows, and adds several valuable optimization features.

Chiefly, NewWave lets you automate a task that you perform repeatedly and usually grudgingly. Sounds like the Recorder, you say. Not by a long shot. NewWave uses an application called Agent, which is responsible for organizing and tracking all tasks that you automate. Agent makes it easy for you to perform tasks from several different places within NewWave and within Windows. Also, you can edit any automated task that you create in NewWave. That's not possible with the Windows Recorder.

We should also mention that the NewWave Agent lets you easily create automated tasks that open, use, and switch between multiple applications. The bottom line is that, virtually any repetitive task you perform within Windows can be automated (and edited) using the NewWave Agent.

In the following sections, we'll show you some of the major benefits and features of NewWave. Do keep in mind that NewWave has many, many features that we don't have space to discuss.

Installing NewWave

By far, the biggest learning curve with NewWave involves installation. Although the NewWave installation is very intelligent and just asks you to confirm a few of its proposed actions, you'll undoubtedly be surprised by some of the limitations that NewWave imposes.

In our view, the most significant limitation is the inability of NewWave to recognize more than one extension for documents created and used by a particular application. You'll find this problem to be especially annoying if you routinely use different extensions to categorize your documents for a particular application.

This file management brick wall shouldn't represent a dead end, though. If you're willing to cooperate with the way NewWave insists on organizing applications and files, you can always rename your application's file extensions to the extension that NewWave expects. You can enforce the organization scheme you want by creating separate *folders* for storing documents that normally would have different extensions.

When you install NewWave, the installation program will offer to set up, for NewWave's use, all applications that are in existing Program Manager groups. NewWave uses all of the existing setup and configuration information available in Windows to set up the NewWave environment. So, you should let NewWave set up all applications for you during its installation.

Note: NewWave requires that you have at least 10MB of available hard disk space prior to installation. Also, NewWave can take as much time to install as Windows itself—often 20 to 30 minutes or longer.

NewWave Is a Powerful Object-Oriented Environment

Although the term "object" means different things to different computer users, in a NewWave context an object is any application, document, picture, or any combination of these that can be represented as a single icon.

In NewWave, the icon is totally mobile. You don't have to worry about whether a NewWave icon can be moved from the File Manager to the

Program Manager, from the Program Manager to the File Manager, from one program group to another, from one location on the desktop to another, and so on. In NewWave, your motto is *have icon, will travel—anywhere.*

Although NewWave is loaded with features, the mobility and versatility of icons (remember, they're *objects* in NewWave) is the underlying concept behind NewWave. It's also the feature that makes NewWave so easy to use—easier than Windows, and almost as easy as the Macintosh.

HOT TIP

After You've Installed NewWave, Drag the NewWave Icon into Your StartUp Group

Since NewWave is a replacement for the Windows Program Manager and File Manager, you'll want to load NewWave whenever you start Windows. To do so, just drag the NewWave icon from the NewWave group window into the StartUp group.

Using the NewWave Office to Organize Your Work

NewWave recognizes two basic types of objects: "tools" and "objects." Basically, we're saying that NewWave includes a category of objects that is actually called "object." A tool is just an object that doesn't typically open and manipulate document files. The Windows Clock, Control Manager, and Calculator are examples of tools. Objects (in other words, programs that do open, close, and manipulate data files and are actually labeled as "objects" in NewWave) include most applications, like Word for Windows, Paradox for Windows, Excel, and so on.

Figure 10.13 shows a typical window that appears when you start NewWave. Hewlett Packard calls its interface the NewWave Office, although this term probably won't mean much to you unless you need to spend time using the NewWave User's Guide to learn different features. The default NewWave window shows icons for all tools that NewWave has identified. Notice the third icon in the top row. You can use this Waste Basket to delete any object—program icons, folders, documents, macros (called agent tasks), and so on—simply by dragging the object onto the Waste Basket icon.

Applications are typically organized within the File Drawer, which we'll show and demonstrate next.

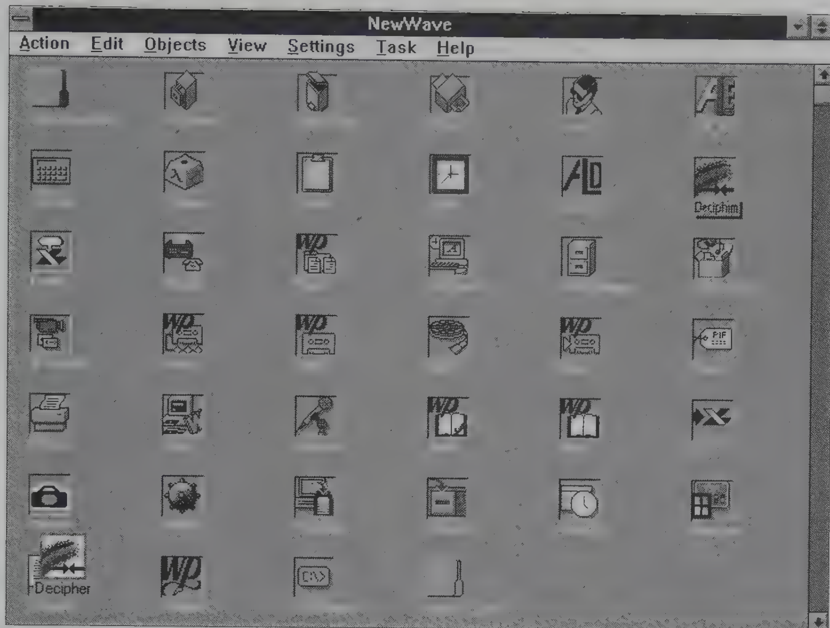


Figure 10.13 This NewWave window contains icons for all of the tools that you use with Windows.

Using NewWave to Organize Your Applications and Files

You won't see any directory trees in NewWave. Instead, NewWave lets you organize your applications and documents within a File Drawer that can contain any number of folders, including folders inside other folders. In other words, the File Drawer serves a similar purpose as the Windows File Manager; however, the File Drawer stores files within folders, rather than directories.

You'll notice that many of the familiar Windows icons appear in Figure 10.13, including the Control Panel, File Manager, and many of the Windows accessories. Double-clicking on any of these icons launches its applications, just as you would expect if you did so from within the Program Manager. All of these icons are considered to be *tools* rather than applications. In NewWave, a tool is a program that does not have a file extension associated with it.

Take a look at the second icon on the top line in Figure 10.13, File Drawer. This icon is the gateway to the file organization system used by NewWave. The File Drawer organizes files and directories within a series of folders. Your applications and documents are all typically stored within the File Drawer. You can store folders within folders, making it easy to zoom in on different collections of files, starting from a general level and proceeding to successively more specific levels.

Figure 10.14 shows the contents of our File Drawer window. We've created three folders in the File Drawer; however, if you eventually use NewWave and fully use its features, your File Drawer will contain dozens of folders. In our example, we've created a separate folder for three applications—Word for Windows, Excel, and WordPerfect for Windows.

Figure 10.15 shows the window that displays when we open the Word for Windows folder. We've added the Word program to the folder so that we can launch Word directly from here, and we've created separate folders for each project. Each folder is equivalent to a directory within the File Manager.

We'll take this just one step further to demonstrate how easy and effective it can be to embed folders. In Figure 10.16, we've opened the Insider folder that was shown in Figure 10.15. The Insider folder contains all of the documents used to create the book you're now reading. Of course, we could continue to embed other folders. For instance, we could create a Backup folder within the Insider folder to store backup copies of each Insider document.

Figure 10.14 Use the NewWave File Drawer to organize applications and documents within folders.

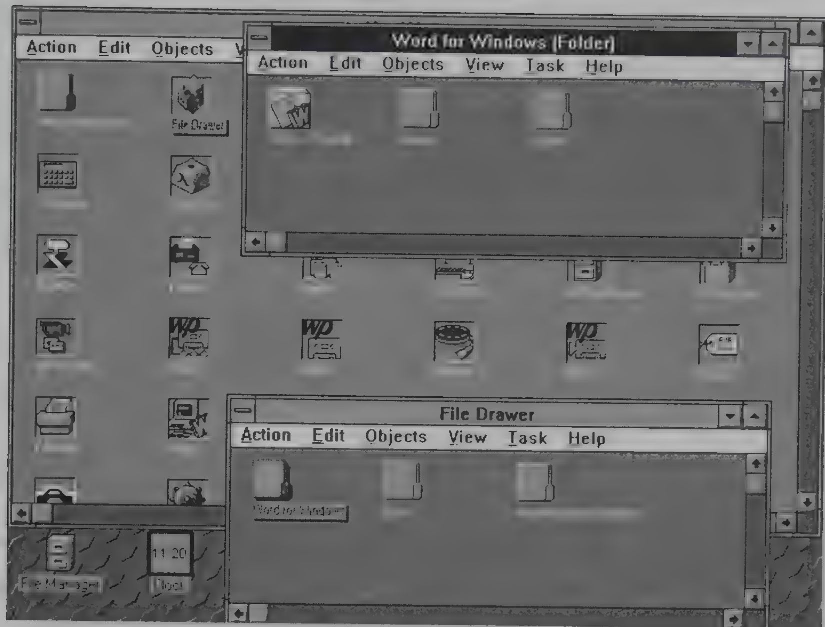


Figure 10.15 A folder can create programs, documents, and other folders. This folder contains a program and two additional folders.

Figure 10.16 This folder, embedded three layers deep within the File Drawer, contains documents only.

Using the Agent to Automate Tasks in NewWave

In NewWave, you can use a feature called the Agent to create automated tasks. The Agent is similar to the Windows Recorder only in the sense that you record tasks by making selections from the keyboard and from the mouse. However, the similarities between these two tools end there. The Agent makes the Recorder look stiff and feeble by comparison.

The Agent is one of the most powerful components of NewWave. If you routinely perform the same, complex tasks, you might consider purchasing NewWave solely for the Agent's time-saving capabilities.

The major difference between NewWave's Agent and the Windows Recorder is the ability to edit tasks that you create with Agent. If you make a mistake when you create an Agent task, you can review and modify the contents of the task, which consists of easy-to-understand statements.

Figure 10.17 shows the language statements for a task that we created to select and configure a particular printer. Keep in mind that we didn't enter any of the text that you see in Figure 10.17. We just used point-and-click techniques to indicate what steps we wanted to include in the task; NewWave did the rest. To play back the task, we simply drag the task's icon onto the Agent icon.

To illustrate how easy it is to create an Agent task, we'll provide the steps we used to create the Configure NEC Printer task:

1. From the Objects menu, choose Create a New to display the Create a New window.

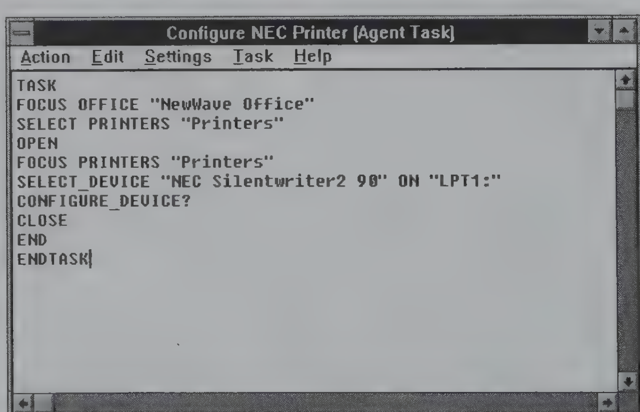


Figure 10.17 This Agent task automatically selects the NewWave printer icon to open the Printers dialog box, selects a printer and displays its configuration dialog box, waits for input from the user, and then closes the Printers dialog box.

2. Scroll to and choose the Agent Task icon, enter a name for the task (in this case, **Configure NEC Printer**) in the dialog box that appears, then press **Enter**. NewWave will add the Configure NEC Printer Agent task icon to the NewWave office desktop.
3. Choose the Configure NEC Printer Agent task icon to open it.
4. Choose Action Start Recording to begin recording steps for the task.
5. Choose the Printers icon to display the Printers window.
6. Scroll to and select the appropriate printer to configure (in this case, the NEC SilentWriter 2 90).
7. Choose Device Configure Device to display the configuration dialog box for the selected printer.
8. Select any desired configuration options, then click on the OK button to exit the dialog box.
9. Close the Printers window, then press **Ctrl+S** to stop recording.
10. In the Configure NEC Printer agent task window, choose Action Compile to convert the statements into executable code.

The task is now complete; you can play it back at any time by dragging the task's icon onto the Agent icon (which usually appears in the top line of the NewWave desktop).

When we play back this task, it automatically opens the Printers window, selects the NEC printer, displays the configuration dialog box for the printer, and pauses for user input. After you've made any configuration changes and have clicked on the OK button, the task closes the Printers window. (By the way, mouse actions are much more reliable within a NewWave task than they are in a Windows macro. In fact, if you include any mouse movements in a task that don't seem to work or that you later decide you don't want, you can easily delete the mouse-movement statements from the task.)

Building on Existing Tasks to Create More Complex Tasks

The task we created in the preceding section is pretty simple; however, you can add new steps to a task at any time. You never need to rerecord steps that already exist in an existing task.

The same features that make it possible for you to edit a NewWave task allow you to add new steps to it. (And, of course, you can remove steps if you want.) To add new steps to a task, you simply open the task's window, position the insertion point where you want the new steps to begin, then start the recording process.

Figure 10.18 shows several steps that we've added to the Configure NEC Printer task. Specifically, after the task has saved the printer configuration changes, it opens the file drawer, opens the Word for Windows folder, and then opens the Insider folder. These additional steps let us immediately zoom in on the folder that contains the document we want to print using the new printer configuration.

Note the "Recording restarted" comment line that appears after the CLOSE statement in Figure 10.18. We positioned the insertion point after the CLOSE statement so that the new steps would begin after the Printers window is closed. NewWave inserts the "Recording restarted" comment to let you know which statements are new—in case you need to make corrections.

We've only scratched the surface of NewWave's features—especially its automation features. NewWave includes excellent help windows that contain formats and explanations for every statement in the task language. You can use these help windows to learn the language, if you want, or simply to uncover task errors that stem from improperly used statements. The NewWave documentation is also easy to read and understand.

NewWave is worth considering if you want to create a Windows environment that allows you to do some heavy automation without requiring you to learn a complex macro language.

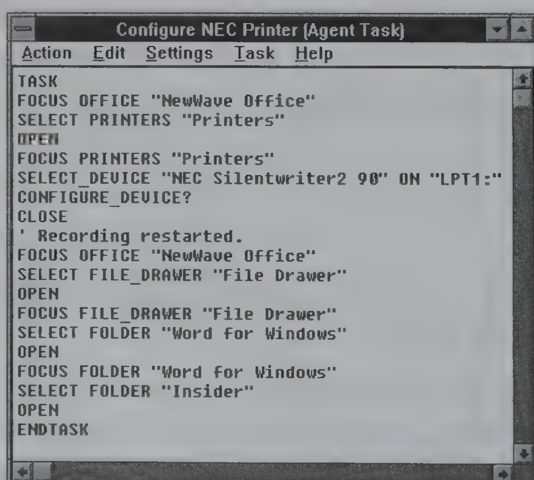


Figure 10.18 Here, we've added several steps to our Configure NEC Printer task, starting after the CLOSE statement.

Norton Desktop for Windows 2.0

We've saved the best for last. The Norton Utilities package has been available in various versions and forms for almost a decade and has gained a well-earned reputation for providing the most reliable and comprehensive disk diagnostic and recovery utilities available for the PC.

However, Symantec, manufacturer of the Norton Utilities, has had difficulty keeping up with Windows. In the past, various versions of Norton have run with some unpredictability within Windows 3.0 and Windows 3.1.

But with the release of version 2.0 of Norton Desktop for Windows, Symantec has—without a doubt—got it right. The desktop environment for this version works almost faultlessly with Windows 3.1, and is one of the most intuitive, easy-to-use desktops we've seen.

We created the desktop shown in Figure 10.19 to show and demonstrate many of the features of Norton Desktop for Windows. You might guess—and rightly so—from Figure 10.19 that the Norton Desktop environment is highly customizable. We've organized the desktop in this way both to demonstrate Norton's features and to suit the way we like to work. You can do the same on your system.

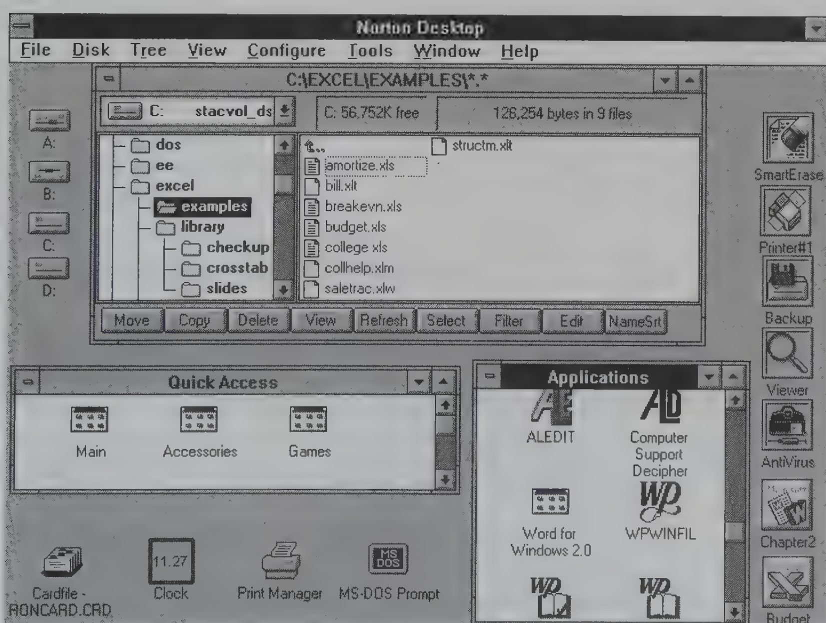


Figure 10.19 This arrangement shows many of the objects that Norton allows you to place and arrange on a single desktop.

Of course, we're just going to touch on some key features in our demonstration of the software. Norton Desktop has many capabilities that we just don't have space to cover here. In touring Norton, we'll start at the top-right corner of Figure 10.19 and work our way clockwise.

Undeleting Files with SmartErase

SmartErase is the cornerstone of the Norton package; in fact, it was largely Peter Norton's file and disk recovery software that brought him to prominence among end users. The SmartErase tool is a more refined Windows version of Norton's earlier file recovery utilities.

SmartErase appears in Figure 10.19 as the eraser icon at the top of the icon strip along the right edge of the desktop. SmartErase uses at least two other utilities to protect your files from permanent erasure: SmartCan and Image. Both SmartCan and Image are normally loaded when you boot your computer. SmartCan protects files by temporarily (for five days, typically) moving deleted files to a hidden directory so that they can't be overwritten. Image works much like the MIRROR.FIL file in DOS 5, by creating a table containing disk storage information that can be used to find and rename deleted files.

You can use SmartErase to recover files that you accidentally deleted, from any directory and from any disk drive. Always run SmartErase as soon as possible if you realize you've accidentally deleted one or more files.

Figure 10.20 shows the window that appears when you choose the SmartErase icon. This window shows three deleted files that can be recovered from the current drive and directory (the name of each file begins with a ?). You can move through the directory tree just like you

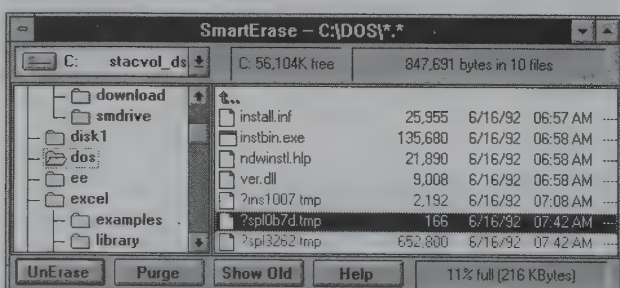


Figure 10.20 SmartErase includes an easy-to-use file management window that lists both existing and deleted files in a directory.

would do in the File Manager to search for other deleted files that can be recovered.

HOT TIP

You Can Configure SmartErase to Ignore TMP, SWP, and INI Files

Norton recommends that you configure SmartErase so that the Image utility ignores TMP, SWP, and INI files when it updates its hidden directory. Since Windows frequently updates files that have these extensions, you can slow down the performance of your system if Image has to continually track these files. However, you might want to have Image track your INI files just to be safe. If the performance of your system seems to be suffering, you can easily reconfigure SmartErase to ignore the INI files.

Using the Printer Tool to Configure the Active Printer and to Print Files

The Printer tool icon is located directly beneath the SmartErase icon in Figure 10.19. You can use it to print any document directly from the desktop.

When you click on the SmartErase icon, the Print dialog box that appears will display the name of the document that you last worked with. However, you can enter the name of any document in this dialog box. You can also print by dragging a file from the Norton file manager onto the Printer icon.

Norton Includes a Powerful Backup Utility

Norton's Backup utility rivals the Central Point Backup for Windows utility, although it does not provide support for most tape drive formats. However, the Norton Backup utility fully supports Bernoulli boxes and network backup. You can restore files either from the DOS prompt or from within Windows.

The Backup tool icon is shown third from the top (under the Printer icon) in Figure 10.19. When you choose this icon, a Norton Backup window appears, similar to the one shown in Figure 10.21. You use this window to configure your backup environment, specify and name selected groups of files and directories for backup, and restore backed up files.

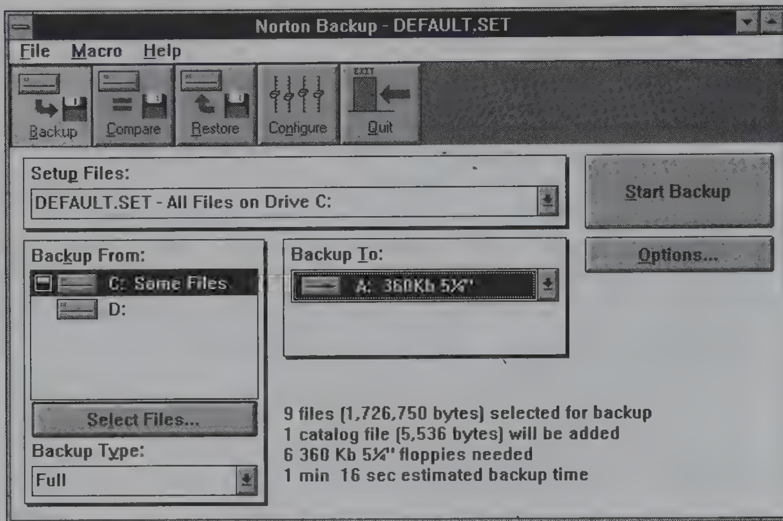


Figure 10.21 You use the Backup window to set up most backup options in Norton.

In Figure 10.21, we've selected a set of files from drive C for backup onto floppies in drive A. We haven't saved this configuration to a file, although this is easy to do by using the File menu.

The Backup tool reports on the progress of your backup, unless you tell it to operate in the background. Figure 10.22 shows the Backup Progress dialog box that appears. Notice that Norton reports on compression and size information, as well as percentage completed and number of disks required.

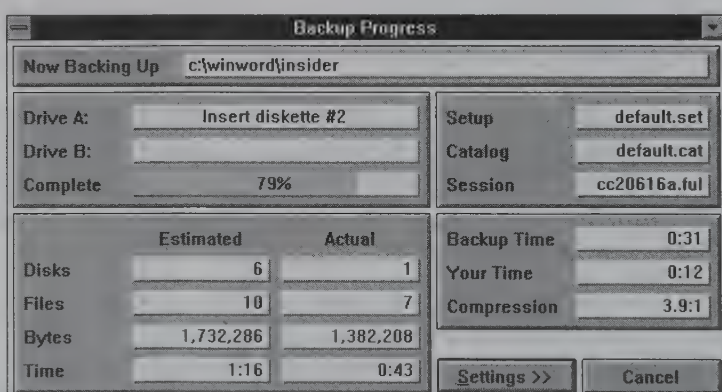


Figure 10.22 Norton Backup reports on the status of a backup in progress, using this dialog box.

Scheduling Backups for Several Different Backup Configurations

The Norton Scheduler lets you specify scheduled backups for each backup configuration that you've saved to a setup file.

Figure 10.23 shows the Scheduler dialog box that you use to specify regular backups. We've already created an INSIDER.SET file using the Backup utility shown in Figure 10.21. In Figure 10.23, we've used the Scheduler's Edit dialog box to schedule a weekly backup for these files, at 5:30 P.M. on Fridays. Note the Load With Windows option box. If you select this box, you ensure that the Scheduler is enabled at the time the backup is supposed to take place.

Using the Viewer to Quickly Review Files

The Viewer tool lets you view virtually any file. This tool is useful if you want to review a file briefly without starting the application that normally opens it.

The Viewer icon (the magnifying glass) appears below the Backup icon in Figure 10.19. You can use the Viewer to display files either in ASCII format or as a hex dump (for program files).

The Norton AntiVirus Examines Your Hard Disk Each Time You Boot

The Norton AntiVirus scans your hard disk for viruses before any TSRs and drivers are loaded into memory.

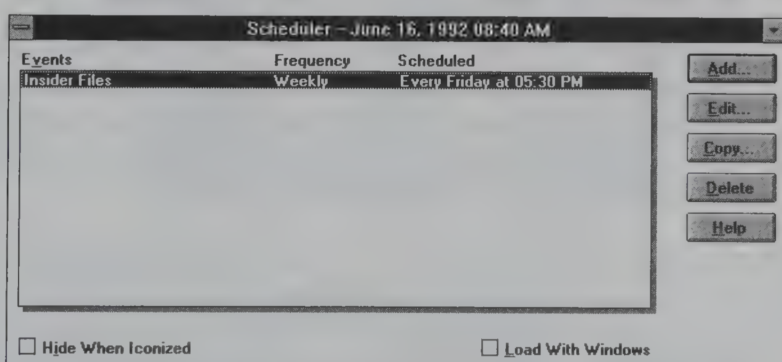


Figure 10.23 The Scheduler dialog box contains buttons that display additional dialog boxes for setting backup frequency and time information for selected groups of files.

The AntiVirus program is installed automatically when you install the Norton Desktop for Windows. You can remove it if you find that the virus detection routine overly prolongs your boot time; however, viruses have become more and more common these days—we recommend that you keep Norton AntiVirus installed at all times. The Norton AntiVirus utility also is sold as a separate program.

Placing Desktop Icons Just About Anywhere on the Desktop

The easily customizable desktop is one of the most attractive features of Norton Desktop for Windows. You can move documents, windows, drive icons, and groups anywhere. You can even place groups within existing groups.

In Figure 10.19, you'll notice that we've placed a Word and Excel document in the bottom-right corner of the desktop. These aren't minimized applications. We simply dragged the icons from the file manager and placed them in a corner of the desktop for easy access.

Similarly, we've dragged our most frequently used group window (Applications) to the bottom of the desktop so that we can keep it open and in view while we're working on the desktop.

Notice in Figure 10.19 that we've copied the Word for Windows group window into the Applications group. This technique is forbidden fruit in Windows, but you can use Norton to "nest" groups within groups simply by dragging a group icon onto another group icon or group window.

The Norton File Manager Provides Access to All Tools as Well as Providing File Management

In Windows, the File Manager is segregated from the Program Manager, making it difficult, in a sense, for you to see both the desktop and file management environments at one time. By contrast, the Norton file manager works seamlessly with the other parts of the desktop environment.

Notice in Figure 10.19 that, even though the Norton file manager appears as a window on the desktop, just as you might do with the Windows File Manager, the menu bar is part of the desktop itself. In other words, file management is fully integrated within the desktop environment. Even if you close or minimize the Norton file manager, you can still select, move, copy, search for, delete, and perform all other file management operations.

You also have direct access to every Norton tool and feature from the menu bar.

Norton Can Diagnose and Repair Disk Problems

Norton Desktop for Windows includes a utility called Disk Doctor, which you can use to recover data from damaged disks.

Figure 10.24 shows the Norton Disk Doctor screen. As this figure shows, Disk Doctor examines your hard disk's partition table, boot record, FAT, directory structure, file structure, and can look for and recover lost clusters—all from within Windows. (You can also diagnose diskettes.)

Disk Doctor produces a report identifying any problems and suggests how you can let Disk Doctor make necessary repairs. A special Fix-It disk is also provided—in case you aren't able to run Windows or Norton from your hard drive. The Fix-It disk runs the diagnostic and repair routines from the floppy.

Norton Provides an Extensive Set of Options for Examining Your System's Hardware

The System Info option, available from the Tools menu, provides access to information about your hardware and memory

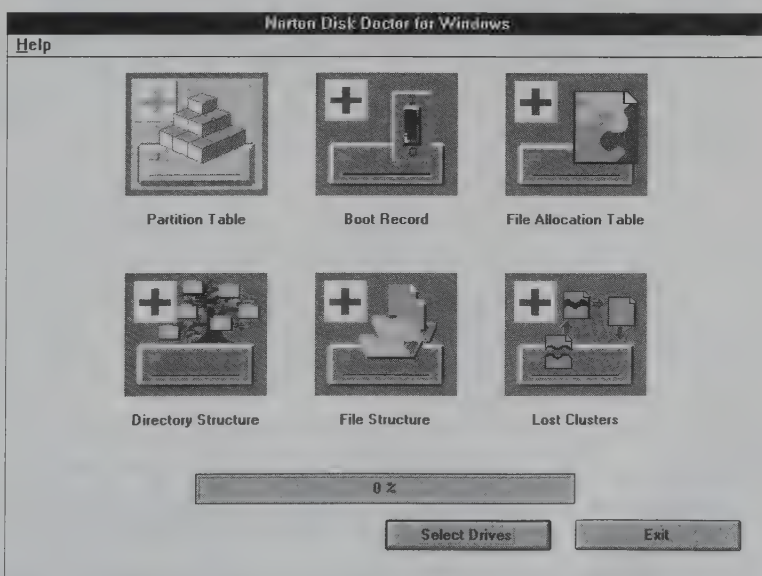


Figure 10.24 Disk Doctor examines your hard disk (or a floppy) for damage and other problems, and suggests repair procedures.

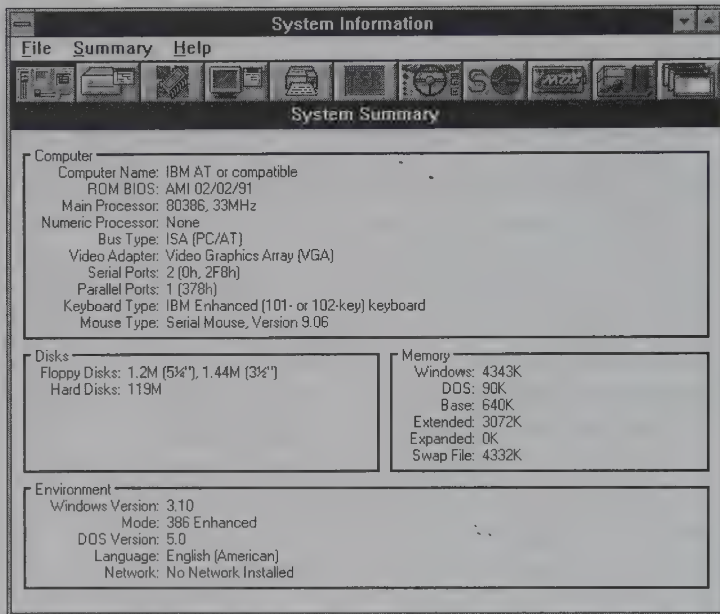


Figure 10.25 This screen summarizes your hardware and memory configuration.

configuration, including reports on TSR and device driver usage. All of the remaining figures in this chapter show system information that you can access from the System Information window.

Figure 10.25 shows the default System Summary that appears when you first display the System Information. Other options are available to you by selecting buttons on the button bar, which appears along the top of the window.

Figure 10.26 shows the Windows Memory display, which graphically portrays how Windows is using your memory. The stacked bar charts are broken down into discardable and non-discardable code segments.

The display in Figure 10.27 summarizes usage for current TSRs, including their hex address, decimal size, and software interrupt hooks. Figure 10.28 lists all DOS device drivers currently in use, including their base address and offset.

Figure 10.29 shows a unique and highly useful Norton feature; you'll notice that this display of configuration files looks nearly identical to the System Configuration Editor display available from within Windows.

However, Norton added an additional set of bells and whistles here. When you choose Edit! from the System Information menu bar, Norton displays the currently active window in its text editor so that you can edit

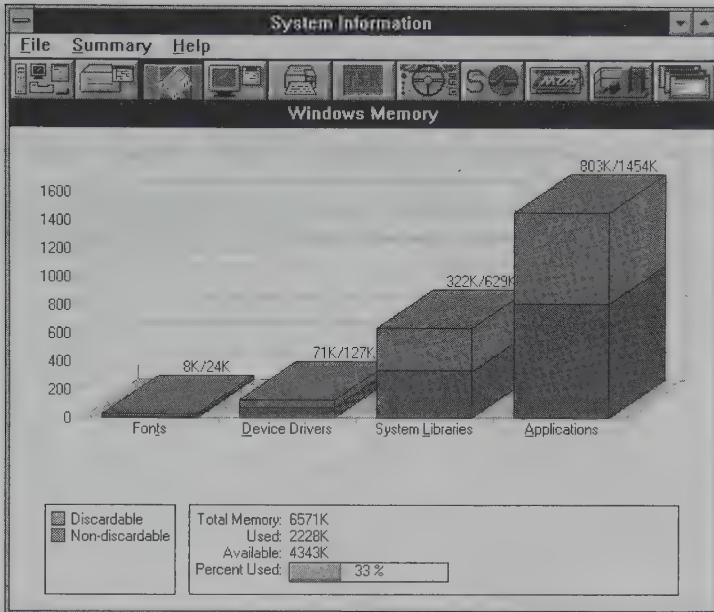


Figure 10.26 This display shows how Windows is using your system's memory.

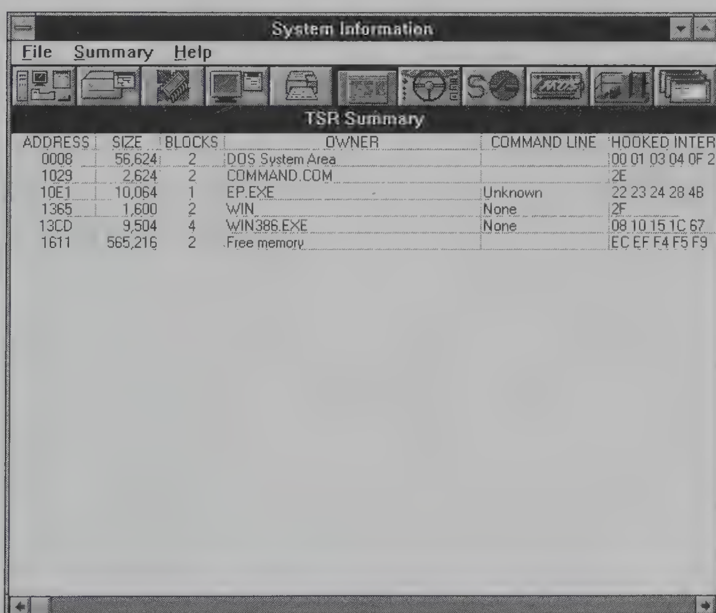
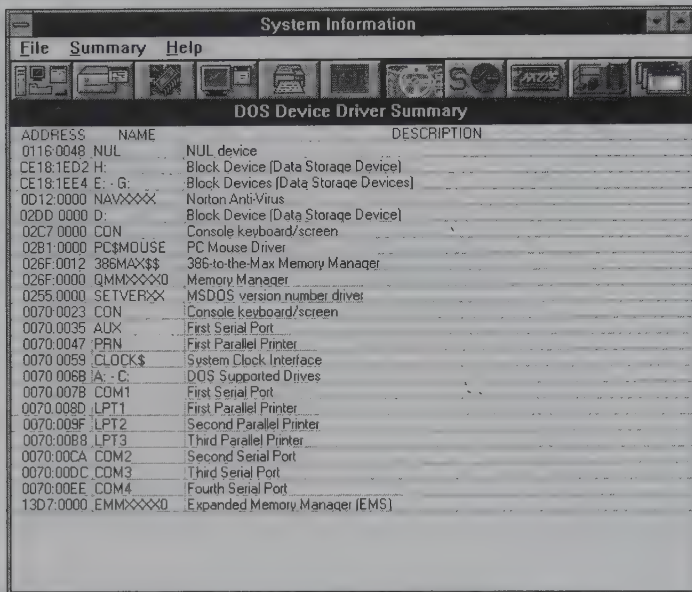


Figure 10.27 Use this display to examine the locations and sizes of currently loaded terminate-and-stay resident software.

it. In this way, you can focus on one file at a time for editing purposes, but you can view five files within one window. (Norton's NDW.INI file also appears in this window; it's in the background in Figure 10.29.)



ADDRESS	NAME	DESCRIPTION
0116:0048	NUL	NUL device
CE18:1ED2	H:	Block Device (Data Storage Device)
CE18:1EE4	E: - G:	Block Devices (Data Storage Devices)
0D12:0000	NAVXXXX	Norton Anti-Virus
02DD 0000	D:	Block Device (Data Storage Device)
02C7 0000	CON	Console keyboard/screen
02B1:0000	PC$MOUSE	PC Mouse Driver
026F:0012	386MAX$$	386-to-the-Max Memory Manager
026F:0000	QMMXXXX0	Memory Manager
0255.0000	SETVERXX	MSDOS version number driver
0070:0023	CON	Console keyboard/screen
0070.0035	AUX	First Serial Port
0070:0047	PRN	First Parallel Printer
0070 0059	CLOCK$	System Clock Interface
0070 006B	A: - C:	DOS Supported Drives
0070 007B	COM1	First Serial Port
0070.008D	LPT1	First Parallel Printer
0070:009F	LPT2	Second Parallel Printer
0070:00B8	LPT3	Third Parallel Printer
0070:00CA	COM2	Second Serial Port
0070:00DC	COM3	Third Serial Port
0070:00EE	COM4	Fourth Serial Port
13D7:0000	EMMXXXX0	Expanded Memory Manager (EMS)

Figure 10.28 This display provides a listing of device drivers currently in use.

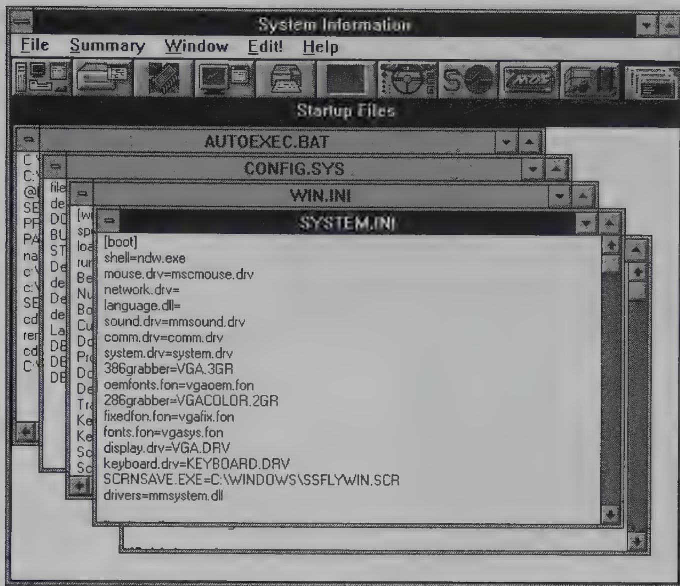


Figure 10.29 The System Information window is similar to the Windows System Configuration Editor, but provides additional options for editing files.

Part Overview

PART 4

Working with Fonts and Printers

Chapters 11, 12, and 13 are designed to give you the expertise you need to control the way your documents look—both on screen and when they're printed. Chapter 11 deals with the way Windows creates and uses fonts to both display and print your documents. In an attempt to make life as simple as possible for you, we'll limit our discussions in this chapter to generic information about fonts and their relationship to screen and printing issues. In Chapter 12, we'll explore every detail involved in using Windows 3.1's new TrueType engine and TrueType fonts. In Chapter 13, we'll dive into the murky waters involved with specific printers and printing issues.

You probably won't need to read every section in these chapters, just the ones that apply to your situation. However, we do recommend that—if your knowledge of font technology is sketchy—you read the opening general discussions in Chapter 11 dealing with typology and the way Windows generates and manipulates type.

CHAPTER

11

Windows and Fonts

It's a familiar sight: Judy, a wary and somewhat weary computer user, stands over one of the office laser printers, clutching her Styrofoam cup of coffee, waiting. The printer has been named "Damien" by a few of the more cynical members of the company, and Judy now knows why. She's trying for the umpteenth time to get her document to print the way it looks on screen. When Damien ejects the first page, she scans it, looking for problems. She sighs and then groans. Characters are printing on top of each other, lines are breaking in bizarre locations, the page length is too short, and the coffee is awful. Life is not good today.

It's probably easy for you to identify with Judy—as easy as it is to immediately despise Damien, a mere inanimate object. That's because printers have a universal reputation for being unruly, unpredictable, and ill-mannered devices. And the reputation is partially justified, but also partially unfair.

Judy is up against several problems—all imposed on her at once. She doesn't know if the source of Damien's evil powers actually comes from the fonts she is using, from Windows, from her word-processing application, from the screen driver, from the printer driver, or from the printer itself. What Judy needs—and what you need—is enough information to protect against the many pitfalls that font and printer technology can present.

Let's be clear and honest: Windows' font and printing capabilities present outstanding opportunities for creating professional-looking, even awesome documents. As is true with any technology that offers great flexibility and power for the user, font management and printing can be complex areas. And most users, frankly, haven't taken the time to learn about the intricacies involved in creating complex printed documents. True, computers *should* make your life more simple, and they can. It's just that you need to know how to harness the many options and features that are available to you. In this chapter, we'll explain how to understand, use, display, and print fonts in Windows 3.1.

Fonts, Personal Computers, and Windows 3.1

Until about 20 years ago, font technology (we'll define "font" in a moment) was limited to ink-and-paper methods. If it wasn't printed, then it wasn't type. So, if you understood how fonts were used to create characters on paper, you pretty much knew all you needed to know about fonts and printing.

However, two major and relatively recent technological advancements have made font technology both more powerful and more difficult to understand: the personal computer and desktop publishing.

Your PC uses both a screen *and* a printer to provide output. The problem here is that, if you connect any given printer to any given computer system, it's almost a sure bet that the *resolutions* of your screen and printer won't match. This difference in resolution, which is just the density of dots used to create each character, makes it difficult for your applications to provide fonts that will look the same on both your screen and your printer.

This leads us into the second problem. Thanks to the relatively low cost of PCs and high-quality printing devices, just about anybody, armed with just a little publishing knowledge, can create documents and full publications that were once limited to the domain of professional designers, typesetters, and printing companies. Unfortunately, since several hundred kinds of printers are available for use in desktop publishing, Windows and other software must be able to accommodate all of these devices. Each category, brand, and model of printer has its own particular requirements and quirks. Printers tend to want to use only a few fonts at a time, and sometimes even allow you to use only built-in fonts. Some printers allow you to supply additional fonts, provided they are supplied in the correct way, and from an "appropriate" source.

In any case, Windows 3.1 has taken a giant step forward in simplifying font use and printing by including TrueType technology. But even TrueType has its limitations and quirks. In the following sections we'll walk you through the basics of fonts, font creation, and printing with Windows 3.1. The information that we provide in this chapter should help uncloud much of the mystery of Windows related to fonts and printing, and should help you solve printer-related problems more easily.

The Components of Type

When you're dealing with text, either on the screen or as printed characters, you're dealing with *type,* in other words, characters. In the printing business, the study of type and its many characteristics is called *typography.* Since your computer system handles most of the details involved in managing type, you don't need to have an in-depth knowledge of typography or its history. What you do need to know about are the many options that Windows, applications, and your screen and printer offer for creating and managing related characters of type, called fonts.

What's in a Font?

The term "font" has a different meaning in general typography and in Windows. Unless you aspire to a career in the publishing industry, the definition you need to understand is the one used by Windows.

In Windows, a *font* is just a name for a collection of characters—all the letters of the alphabet, numbers, and other special characters—that share the same appearance characteristics.

In any given font, characters might also share ornamental features, such as serifs or decorative curls. Examples of fonts are Bookman, Bookman Heavy, Helvetica, Helvetica Narrow, Times Roman, Arial, Gothic, and so on. (In Windows applications, italics and boldface characteristics are often—but not always—considered to be independent of a font because they're added by you, the user. However, Windows stores a separate font file for each italics and boldface character set for each font name.

Figure 11.1 shows many of the terms and components involved in typography. You'll notice that most of these terms apply to different areas of *letters* of the alphabet, rather than numbers or other symbols. We'll briefly review these components next.

x-height The typical height of many lowercase letters, including the letters x, a, c, o, and e.

Cap height The height of all capital letters and some lowercase letters; the maximum height of any character in a font.

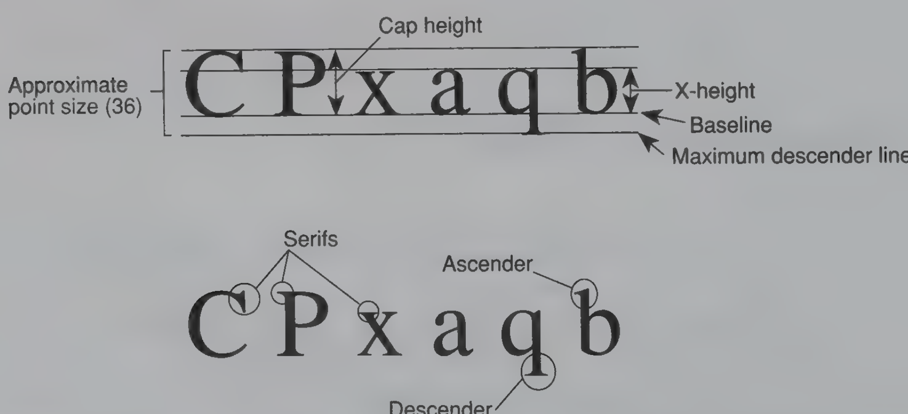


Figure 11.1 This diagram shows many of the terms used to describe parts of characters and fonts.

Baseline Marks the bottom of the x-height and cap height, and is used to position characters horizontally along a line.

Ascender A portion of a lowercase letter that extends above the x-height. The letters t, b, d, f, and h are examples of characters that have extenders.

Descender A portion of a lowercase letter that extends below the baseline and x-height. The letters g, j, p, and q all have descenders.

Serif A decorative projection that extends from the tips of various parts of a character. A font can either be serif or sans serif (without serifs). Serifs make it easier for you to read characters when a lot of type has to fit in a relatively small amount of space. The text in most books and magazines is set in serif type. Sans serif fonts are frequently used for headlines, captions, labels, and other type that can be set in large sizes or that runs for no more than a few lines. Times Roman is an example of a serif font; Arial and Helvetica are examples of sans serif fonts.

Point size The height of characters is measured in points, with one point equal to 1/72 inches. So, characters that are 12 points are about 1/6 inches in height. The point size for a given font is usually determined by the distance from the top of the cap height to the maximum descender line. Many characters, including most lowercase characters in the font, will actually be smaller than the font's point size.

Vertical and Horizontal Spacing Can Vary for Different Fonts

Just because two fonts share the same point size does not mean that all characters in the two fonts will align identically—either vertically or horizontally. Point size is an approximation of the height of characters, but it's not an exact measurement. A size of 12 points in one font can actually be larger than 12-point type in a different font. And the variations in horizontal spacing (the space between two characters) can vary in several ways.

Suppose you're working in a Windows word processor and want to change the font for a section of text—from Times Roman 10-point, to Courier 10-point. To add some horror to this scenario, let's say that the way lines wrap as well as the way lines fit on each page are important factors to this text. In fact, you stayed up late last night with a half-gallon container of coffee, working out the justification scheme, the hyphenations, paragraph breaks, and page breaks for pages in a particular publication you are preparing.

Now, after you highlight the desired section of text and change the font to Courier, you notice that your carefully designed pages have been transformed into a jumble as confusing as Times Square traffic at midday. What happened?

Take a look at Figure 11.2, which shows two pairs of words—with both pairs set in the same size, 48 points, but in different fonts. In the left half of the figure, the words "pills" and "lucky" have been set in a Times Roman font, while on the right, the same two words have been set in a Courier font. You'll notice that the height of each character in the Times Roman font is greater than the height of each corresponding character in Courier. This difference is largely a matter of license taken by the person or company that designed the font. But the effect in terms of your page layouts is important. You can fit more lines on a page (but not necessarily more text) with Courier than with Times Roman.

You'll also notice that the horizontal spacing between characters is drastically different for the two fonts. Characters in Times Roman are *proportionally spaced.* That is, the space between characters is adjusted relative to the actual width of characters. For instance, the entire five-letter word "pills" uses about the same amount of space as the first four letters in the word "lucky".

By contrast, Courier uses *fixed-width* spacing; each character occupies the same amount of horizontal space. So, the words "pills" and "lucky" (in Courier) are equal in width because they each contain five characters. Thanks to the difference between proportional and fixed-width spacing, an identical paragraph set in both Times Roman 12-point and Courier 12-point will wrap in different locations. In fact, you can usually fit several more characters on a line with Times Roman than with Courier.

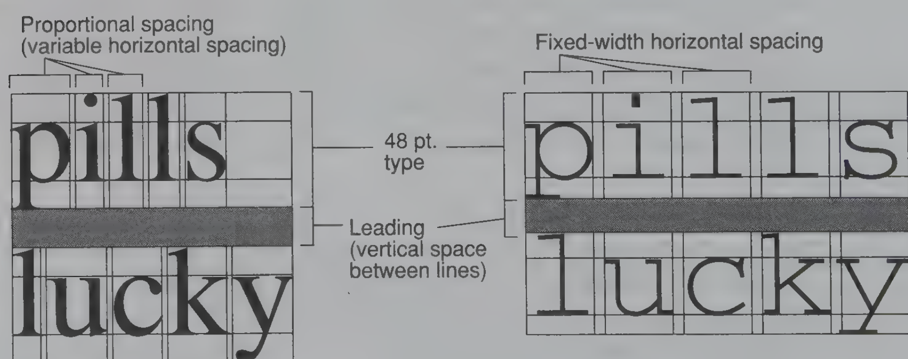


Figure 11.2 These two sets of words demonstrate how spacing can be much different for different fonts of the same size.

Actually, our use of Courier was a bit of a radical example. We used it mainly because many printers have some form of Courier available as a built-in font and because we needed to illustrate fixed-width spacing. But in reality, most fonts are proportionally spaced. But even with proportionally spaced fonts, the widths of characters can differ, as shown in Figure 11.2.

Most Fonts Are Available in Three Widths

In many Windows applications, you have three options for specifying the width of characters: normal, expanded, and condensed. Normal width, of course, is the default width for each character in a font. Expanded characters are slightly wider (elongated horizontally) than normal-width characters. Condensed characters are slightly compressed horizontally.

Expanded and condensed characters have their own unique appearance that works well for certain documents. But you might also find it useful from time to time to expand or condense characters in order to fill out or compress a line of text to fit within a given space. This approach is useful if you're using an application that doesn't allow you to *kern* (adjust spacing between individual characters).

Some applications can expand or condense characters in any font available in Windows. For instance, Figure 11.3 shows the Character dialog box in Word for Windows, which allows you to use the Spacing list box to change the width of characters to expanded or condensed type. The word "Times" that appears in the Sample text area is an example of expanded type.

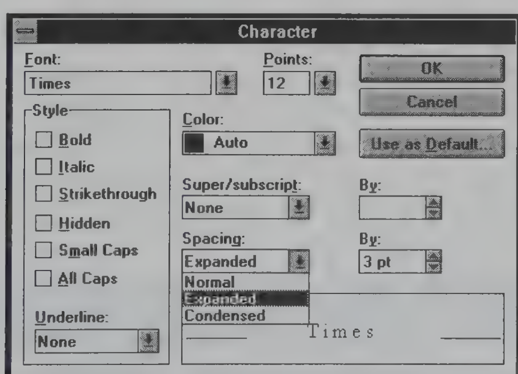


Figure 11.3 In Word, as in many Windows applications, you can change horizontal character spacing to expanded or condensed.

Installing Fonts

It would be ideal if Windows provided every font that you'll ever need at the time Windows itself is installed. In fact, Windows does install many of the fonts that you use. But depending on your printing needs, you might need many other, non-Windows-supplied fonts as well. So, before we discuss specific categories of fonts and how they're generated, we should first explain where fonts come from and how they're installed.

If you have a printer, your fonts are supplied from at least two sources: from Windows itself and from the printer. Depending on the printer you own, you can also install fonts from two additional sources: from cartridges and from companies that supply downloadable fonts.

Windows 3.1's Three Categories of Fonts

All Windows-supplied fonts are installed on your hard disk automatically when you install Windows: they are stored in the WINDOWS\SYSTEM directory. Windows doesn't let you determine which of these files are installed on your system, although you can remove specific font files after Windows has been installed.

Windows-supplied fonts can be divided into three categories, according to the way they are used for display and/or for printing:

Raster fonts A raster font is also called a bitmap because it is stored on disk as a collection of bits that can be mapped to specific pixels (picture elements) on screen. The term *raster* refers to the display surface of your screen. A program called a *rasterizer* maps the bits for each character, which are stored in a font file on disk, to specific pixels on your screen. Because a raster font is essentially an organized collection of pixels and nothing more, there is no way to effectively scale one of these fonts to larger or smaller sizes. For this reason, each available point size for each raster font must be stored on disk as a separate file.

Windows uses raster fonts chiefly for displaying text on screen, although some printers can also print these fonts. All raster fonts have the extension FON. In fact, all FON files stored in the WINDOWS\SYSTEM directory are raster fonts, except for ROMAN.FON, SCRIPT.FON, and MODERN.FON (which are vector font files).

Vector fonts If you can visualize a font that creates characters solely by using a "connect-the-dots" approach, you would have a good understanding of how vector fonts work. Each character in a vector font is essentially a collection of points that are connected by lines to create the character outline. On screen, pixels that fall within the character outline are typically activated by your monitor's electron guns to fill in the outline with black or some other color.

Vector fonts can be easily scaled by changing the proportions between the dots, which in turn changes the length of the lines that connect the dots. However, vector fonts typically don't display well at small sizes. Figure 11.4 shows why. The left-most letter A in the figure shows what happens when a vector font draws a character at a relatively small size—in this case, 12 points. The outline activates a fairly tightly packed and relatively small number of pixels; the result is a character that only roughly approximates the desired character. This effect is much worse on low-resolution monitors.

As Figure 11.4 shows, vector fonts tend to improve in appearance at larger sizes, since the outline activates a relatively large number of pixels as the size of characters increases. Figure 11.4 displays the letter A, using the Roman vector font in sizes 12, 24, 36, 48, 72, 96, and 124 points.

Since vector fonts can be hard to read on screen at smaller sizes (at 18 points or smaller on a typical VGA screen), it's usually better to use

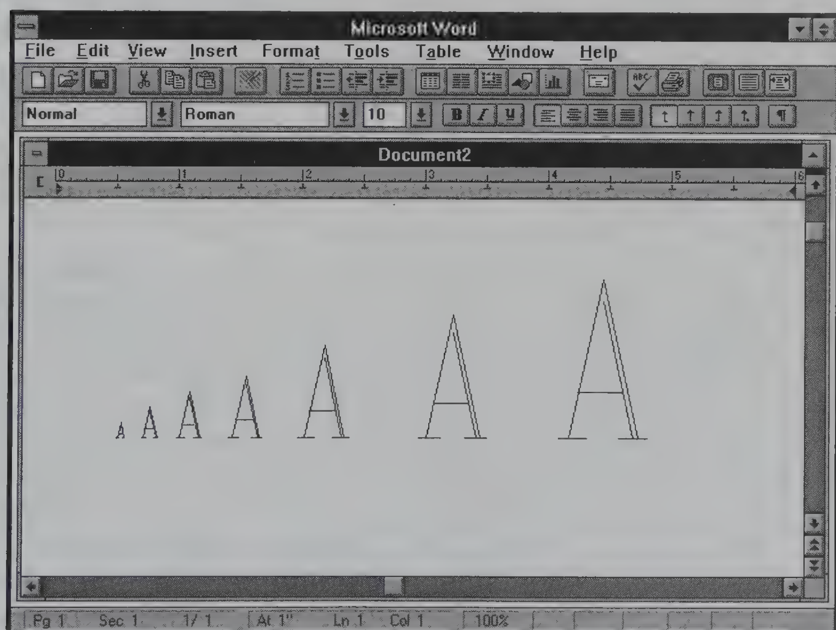


Figure 11.4 Vector fonts scale better at large sizes, since the outline covers a more numerous array of pixels.

raster (bitmap) or TrueType fonts at these sizes. However, vector fonts are required when characters will be output to a plotter or other drawing device that uses a pen to draw the outlines of characters. As you might expect, plotters cannot print bitmapped characters.

Some Windows applications—such as PageMaker—switch automatically to vector fonts (for display purposes) at larger sizes. Since vector fonts look acceptable at large sizes, this approach makes it unnecessary to provide a separate bitmap screen font file for large font sizes. Vector fonts above 12 points also look quite good when they are output on printers that can print at 300 dots per inch (dpi) or higher resolutions. At smaller sizes, vector fonts suffer the same type of resolution problems as we explained for screens: the smaller the print size, the fewer dots are available for creating each character. On very high-resolution printers, such as a 600 dpi laser printer, vector fonts print cleanly even at small sizes.

Windows 3.1 installs three vector font files—ROMAN, SCRIPT, and MODERN. All three files are stored in the WINDOWS\SYSTEM directory and have the extension FON.

TrueType fonts TrueType fonts are new to Windows 3.1; you won't find them in earlier versions of Windows. In a sense, TrueType provides "smart" scalable fonts. That is, TrueType characters are created using the same "connect-the-dots" outlining technology used by vector fonts. However, each TrueType font also contains a set of *hints* that the TrueType engine uses to display characters on the screen and to send characters to the printer. This hinting technology describes how to modify the shape of relatively small-sized characters that are displayed on the screen or on low-resolution printers.

For instance, the letter V in Times Roman uses different widths for the left and right legs of the character, as shown in the top line of Figure 11.5. The left leg (called the *stem* in typography) is typically twice as thick as the right leg (called the hairline). But at small point sizes, a vector font for Times Roman (such as the Roman font supplied with Windows) can make quite a mess of this character. The problem is that the thickness of the outline for the stem activates pixels in a fairly erratic pattern, creating a muddied effect. The hinting feature of TrueType reduces the width of the stem, so that the screen image is sharper and easier to read. The two sentences in Figure 11.5 illustrate this difference. You'll notice that the stems for the letters W, w, V, v, and A all appear sharper in the New Times Roman TrueType font than in the Times Roman vector font. The curves for many lowercase letters also are more cleanly drawn in the TrueType version.

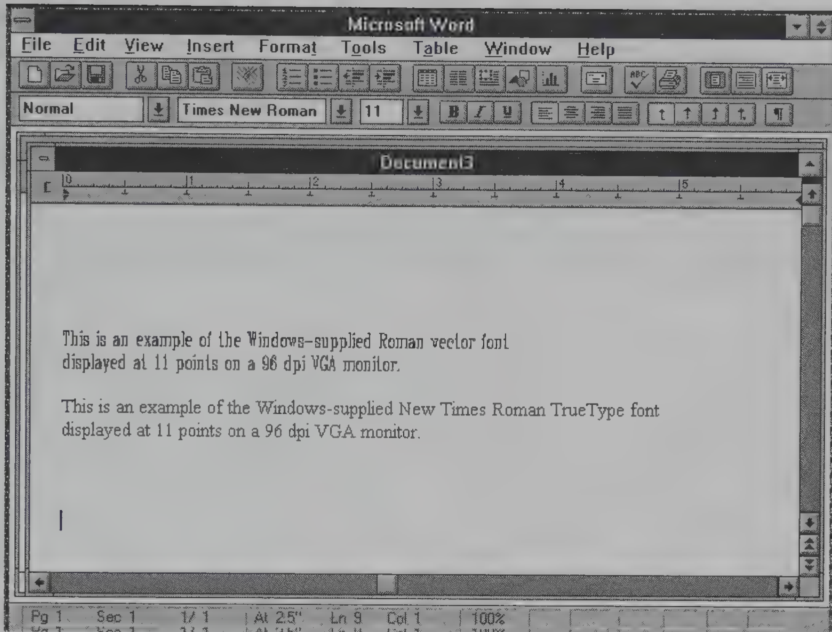


Figure 11.5 TrueType hinting, shown in the second line, adjusts the outline and pixels so that characters display cleanly at all resolutions.

As we suggested in our discussion for vector fonts, hinting is unnecessary for fonts displayed above 18 points on a VGA monitor and above 12 points on a 300 dpi laser printer. Hinting is never required—for any size font—on laser printers that can output characters at 600 dpi or higher.

Another advantage of TrueType is that you can print fonts on virtually any printer, as long as Windows supplies a universal driver for the printer. (Universal drivers are explained in Chapter 13.) You can print TrueType fonts on most dot-matrix printers, PostScript printers, and just about all Hewlett-Packard printers, including the DeskJet and LaserJet series.

You can identify TrueType fonts by the overlapping TT logo that appears next to font names in a Font(s) list box. Actually, each TrueType font has two files—one with the extension FOT and another with the extension TTF. For instance, the two files that are used to display the Times New Roman font are TIMES.FOT and TIMES.TTF The TTF file is the actual font, while the FOT file calls the TTF file. Both files must be present (usually in the WINDOWS\SYSTEM directory) for a TrueType font to work correctly.

We'll provide some in-depth discussions of TrueType and its features in Chapter 12.

Table 11.1 shows the fonts that are supplied by Windows when you install version 3.1. Although we've indicated available point sizes for each font, you'll note that all TrueType (except for Wingdings) and vector fonts are fully scalable (from 2 points to 999 points). However, some applications limit the maximum and minimum sizes for scalable fonts.

Table 11.1 Standard Windows 3.1 Raster, Vector, and TrueType Fonts

Font Name	*Category*	*Width*	*Available Sizes*
Arial	TrueType	Proportional	Scalable
Arial Bold	TrueType	Proportional	Scalable
Arial Italic	TrueType	Proportional	Scalable
Arial Bold Italic	TrueType	Proportional	Scalable
Courier	Raster	Fixed	10,12,15
Courier New	TrueType	Proportional	Scalable
Courier New Bold	TrueType	Proportional	Scalable
Courier New Italic	TrueType	Proportional	Scalable
Courier New Bold Italic	TrueType	Proportional	Scalable
FixedSys	Raster	Proportional	Depends on screen
Modern	Vector	Proportional	Scalable
MS Sans Serif	Raster	Proportional	8,10,12,14,18,24
MS Serif	Raster	Proportional	8,10,12,14,18,24
Roman	Vector	Proportional	Scalable
Script	Vector	Proportional	Scalable
Small	Raster	Proportional	2,4,6
Symbol	Raster	Proportional	8,10,12,14,18,24
System	Raster	Proportional	Depends on screen
Terminal	Raster	Proportional	Depends on screen
Times New Roman	TrueType	Proportional	Scalable
Times New Roman Bold	TrueType	Proportional	Scalable
Times New Roman Italic	TrueType	Proportional	Scalable
Times New Roman Bold Italic	TrueType	Proportional	Scalable
Wingdings	TrueType	Proportional	Scalable (but doesn't use hinting)

Most Printers Cannot Print All Three Categories of Fonts

Certain types and brands of printers are limited in the categories of fonts that they can print. For instance, Hewlett-Packard printers that use the Windows HPPCL.DRV universal printer driver (all LaserJet printers) cannot print Windows raster fonts, nor can PostScript printers. (Most PostScript printers use the Windows PSCRIPT.DRV universal printer driver.)

Table 11.2 shows the font categories that can be printed for four major types of printers—dot-matrix printers, Hewlett-Packard printers that use the HPPCL driver, PostScript printers, and plotters. Note that, even though Hewlett-Packard and PostScript printers can't print Windows raster fonts—such as MS Sans Serif or System—these printers can accept bitmapped fonts if they are sent to the printer in a compatible format. For instance, PostScript printers will accept bitmapped characters that are sent to the printer in the Adobe Type 3 bitmap format.

Using the Installed Printer Driver(s) to Map Built-in Printer Fonts with Screen Fonts

Printer fonts, sometimes called device fonts, are resident in your printer, and therefore aren't supplied by Windows. You can't display printer fonts, although Windows will do its best to select a screen font (usually a TrueType font in Windows 3.1) that closely approximates the appearance of the selected printer font.

All printer fonts appear in an application's Font(s) list box with a small printer icon to the left of the font name. The available printer fonts on your system will vary depending on the currently selected printer. (You might

Table 11.2 Font Categories Supported by Four Major Types of Printers

Type of Printer	*Prints Raster Fonts*	*Prints Vector Fonts*	*Prints TrueType Fonts*
Dot matrix	Yes	No	Yes
HPPCL	No	Yes	Yes
PostScript	No	Yes	Yes
Plotter	No	Yes	No

have installed several printers when you set up Windows; however, you can have only one currently selected printer at a time. Only the fonts for the currently selected printer will appear in an application's Font(s) list box.)

Because a printer font is built into a device, Windows can't display an identical font on your screen unless scaling is defined in the printer driver file. However, Windows maintains an internal font mapping table that it uses to determine an appropriate screen font. The printer driver for the currently selected printer tells Windows the names and attributes of the built-in fonts. Windows uses this information—along with the font mapping table—to display a font on screen that closely approximates the way a printer font will appear when you print your documents.

Since TrueType fonts are fully scalable, Windows will usually use a TrueType font to represent the printer font while you work with documents on screen. Keep in mind, though, that Windows is making a "best guess" substitution effort. You should expect line and page breaks to differ between the font shown on screen and the font that actually prints on your printer.

Cartridge Fonts Are Considered to Be Built-in Fonts

Font cartridges are available for dozens of printers, especially for the popular HP LaserJets. Because these fonts are immediately resident in the printer when you insert a cartridge, Windows considers them to be built-in printer fonts. Cartridge fonts have the same display limitations as we discussed in the previous section.

As is true with any printer font, Windows has to substitute an available screen font for any cartridge fonts that you load into your printer (assuming your printer accepts font cartridges). In many cases, TrueType fonts do a good job approximating the character widths, size, and vertical line spacing of cartridge fonts. However, some cartridges include decorative fonts or special-purpose fonts used to display foreign-language, mathematical, electrical, or other special-use characters. You should keep in mind that these characters might not display correctly if the screen font that Windows selects doesn't include those characters. However, they should still print correctly.

For your Windows applications to use cartridge fonts during printing, you need to use the Printer dialog box to set up the location for cartridge fonts. We'll explain how to do this later in this chapter.

HOT TIP

Laser Printer Languages

Many computer users—even experienced ones—are still confused about the differences between LaserJet printers, PostScript printers, and other laser printers.

All laser printers are driven through the use of a built-in programming language, sometimes called a page description language (PDL), which controls the way the laser mechanism creates and prints text and graphics.

Most laser printers use one or both of these two PDLs: PostScript and/or the Hewlett-Packard Printer Control Language (HPPCL, or HPPCL5 for LaserJet III printers). The HPPCL language is built into all Hewlett-Packard LaserJet printers and HP-compatible laser printers. The PostScript language is built into all printers in the Apple LaserWriter series and in virtually any other laser printer that's labeled as a "PostScript" printer.

When you send a document to a laser printer, Windows uses the universal printer driver (PSCRIPT.DRV for PostScript printers and HPPCL.DRV for HPPCL printers) to generate code and send it to the printer. The code tells the printer which fonts to use and which characters to generate, and also tells the printer how to generate graphic (bitmap) images. In other words, when you send a document created with, say, a PostScript font to a PostScript printer, Windows actually sends PostScript language code, not actual characters, to your printer. You've seen evidence of this if you've ever tried to print a PostScript document to a dot-matrix printer; your printer just spits out page after page of unintelligible code, rather than the document you were expecting to see printed.

The major advantage of PostScript printers over most LaserJet printers (LaserJet III printers are an exception) is that every built-in font is fully scalable. PostScript uses an outlining technology—and some hinting—to create characters in virtually any size, from a single font file. Since most PostScript printers include 35 built-in fonts, all of which are fully scalable, PostScript users often have a distinct advantage over LaserJet users in the number of—and available sizes for—fonts. And because the Windows PostScript driver (PSCRIPT.DRV),uses outlining to create fonts, all fonts can be sent to the screen as well as to the printer. This provides you with scalable WYSIWYG display for all 35 built-in fonts. (We'll explain LaserJet and PostScript technology in more depth in Chapter 13.)

In any event, you need to remember this basic truth about laser printing: you can only print LaserJet (HPPCL or HPPCL5) fonts to a LaserJet (or compatible) printer, and you can only print PostScript fonts to a PostScript printer. And you can't print either LaserJet or PostScript fonts to a dot-matrix printer. (Exceptions to these rules are possible if you are using a third-party font manager, which we'll discuss next.)

Using Third-Party Software to Install and Manage Additional Fonts

Many companies supply additional, Windows-compatible fonts that you can download to your printer. These third-party packages often include a font management program that handles the installation and downloading of the fonts from inside Windows.

You can think of third-party font packages as useful additions to your Windows-supplied and built-in printer font library. However, all font packages are not created equal.

The vast majority of third-party font packages supply *downloadable soft fonts*, which just means that the fonts are loaded into the printer from your PC's hard disk by a software program. (These fonts are "soft" because they aren't "hard coded," or built into, your printer or into a cartridge.) Some packages simply supply additional TrueType, PostScript, or HPPCL fonts. With these packages, you just add the supplied fonts to your hard disk, and then use the Control Panel to install them for use within Windows. (We explain how to do this later in the chapter.)

Other packages come equipped with a separate *font manager*, which is a collection of programs that you can use to install, generate, and manipulate fonts that are supplied with the package. You usually have to use the font manager's installation program, rather than the Control Panel, to make the supplied fonts available from within Windows.

Downloadable soft fonts have long been popular to extend the limited built-in fonts for HP LaserJet printers—prior to the HP LaserJet III series of printers. These font managers use HPPCL to generate fonts.

Note: Third-party fonts that are designed for DOS applications usually won't work with Windows. If you are running Windows and want to use non-Windows-supplied fonts, make sure they are designed specifically for use with Windows applications.

Windows-Compatible Font Managers Typically Use Outlining to Provide Scalable Fonts

Newer font managers, especially those designed to run under Windows, use a scalable outlining technology—in some ways similar to TrueType technology—to generate characters that can be sent to the screen and to a printer. For each supplied font, the characters for both the screen and printer are generated from a single font file. The result, as is true with TrueType fonts, is that text is displayed pretty much the same as it will appear when printed.

The two most popular scalable-type font managers are Adobe Type Manager (ATM) and Bitstream FaceLift. Both packages use outline fonts to create bitmap characters on screen and both packages can print fonts to any printer that Windows supports—including dot-matrix, LaserJet, and PostScript printers.

With dot-matrix and LaserJet printers, the outlines for characters are converted to bitmaps (essentially, graphic images) that can be sent to the printer. With PostScript printers, the outlines are converted and sent to the printer in a format called Adobe Type 1, which is an outline standard that all PostScript printers recognize.

We realize that scalable-type font managers like ATM and FaceLift seem suspiciously similar to TrueType, and that you might not yet understand the benefits or drawbacks in using one approach over another. But we'll provide additional information about the relative merits of ATM, FaceLift, and other font managers when we compare them to TrueType, in Chapter 12.

Using the Add Fonts Dialog Box to Install Most Screen and TrueType Fonts

Although all of the Windows-supplied fonts—including the 14 TrueType fonts—are installed automatically when you set up Windows 3.1, Windows also needs to track any other fonts that you want to use with Windows.

Font installation procedures aren't the same in all situations. For instance, font management utilities typically require you to follow a separate installation procedure; fonts supplied for specific printers, including Hewlett-Packard printers, use a unique dialog box within Windows. But many fonts can be installed using the generic Windows 3.1 Add Fonts dialog box.

You access the fonts dialog box by choosing the Fonts icon in the Control Panel, then choosing the Add button. The Add Fonts dialog box, shown in Figure 11.6, appears. You can use this box to install most TrueType fonts supplied by third-party vendors and many other soft fonts. With many font management packages, you have to use a separate installation routine to make the fonts available to the font manager system. But you then often have to use the Fonts dialog to install the fonts for use by your Windows applications. Here are the steps for installing additional TrueType fonts and many other soft fonts:

1. Choose the Fonts icon from the Control Panel.
2. Click on the Add button to display the Add Fonts dialog box.
3. In the Drives pull-down box, select the drive and directory for the disk that contains the fonts you want to install.
4. In the List of Font(s) box, select the names of all fonts that you want to install. (If you want to install all fonts, click on the Select All button.)

Note: Windows will not allow you to install a font if a font with the same name has already been installed. So, if you want to update a particular font, you must first remove the existing version. If you suspect that a particular font has been installed incorrectly or has been corrupted, you also have to remove the font before you can reinstall it. To remove

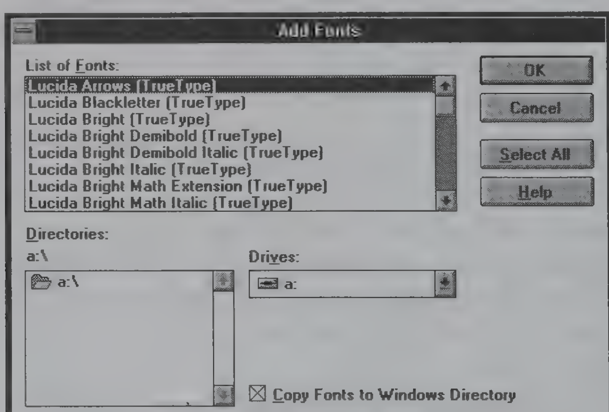


Figure 11.6 Use the Add Fonts dialog box to install screen fonts, TrueType fonts, and some soft fonts.

a font, use the Add Font(s) dialog box, highlight the name of the font you want to remove, then click the Remove button.

Using the Printers Dialog Box to Install Font Cartridges

Because font cartridges are printer-dependent, you must use the Printers dialog box to install cartridge fonts.

If your printer supports font cartridges, you need to install the font cartridge before you can use its fonts. To install a cartridge, follow these steps:

1. Choose the Printers icon from the Control Panel.
2. Select the name of the printer that will use the Font Cartridge.
3. Click on the Setup button.
4. Choose the appropriate cartridge name from the Cartridge list box, then choose OK.

HOT TIP

You Can't Use the Fonts or Printers Dialog Boxes to Install Hewlett-Packard-supplied Fonts

Hewlett-Packard-supplied fonts for all HPPCL printers (most LaserJet printers) are installed using a separate Font Installer utility. Procedures for using the Font Installer are provided in Chapter 13, which also includes a table listing the printers that have their font installation utility built into Windows 3.1.

Using the Character Map to Examine a Font's Character Set or to Copy an Unusual Character to a Document

Every font includes characters for a particular character set, which is a standardized set of codes that can be used to send characters from the keyboard to an application. You can use the Windows 3.1 Character Map to identify and use the character set for a particular font.

Early versions of DOS used the ASCII character set, which used the decimal codes 0 through 127 to specify and send characters through the keyboard. Later versions of DOS included the file ANSI.SYS, which allowed programmers to extend the 128 characters available in ASCII to 255. Most developers for DOS applications chose to keep life simple for users, and stuck with the ASCII character set for their applications.

When Microsoft introduced Windows 3.0, it also legitimized the ANSI (American National Standards Institute) character-set standard by using many of the extended ANSI character codes from 161 to 255 characters.

With the introduction of TrueType, Microsoft elected to use most of the codes that had been left unused in Windows 3.0 to support frequently used symbols in each TrueType font.

In the past, users have encountered a particulary annoying problem when trying to insert characters that aren't shown on the keyboard. That is, you had to press a set of keys that established an "escape sequence," which essentially disengaged the keyboard from the application. Then, you had to type the decimal equivalent for the character you wanted to display, and then press **Enter** or some other key to send the character code to the application.

Although you can still use this basic procedure in most Windows applications, Windows 3.1 also provides an application called the Character Map, which makes it easy to view all of the characters in a font's character set and allows you to copy any character to the Clipboard for easy insertion into a document.

Figure 11.7 shows the Character Map dialog box that appears when you select the Character Map icon from the Accessories group window. The map shown in the figure is for the TrueType Times New Roman font.

To use the Character Map to insert characters, follow these steps:

1. Choose the Character Map icon in the Accessories group window.
2. In the Font pull-down box, find and highlight the font name for the desired font.

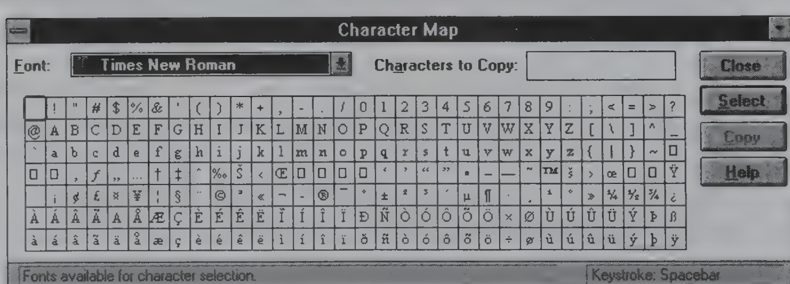


Figure 11.7 Use the Character Map to insert characters that are not shown on the keyboard.

3. In the Character Map grid, locate a character that you want to copy to an application, then double-click on the character. You'll know the character has been selected because it will appear in the Characters to Copy text box. Repeat this step for each character that you want to copy.
4. Click on the Copy button, then click on the Close button to exit the Character Map.
5. Open or switch to the application and document that will contain the character(s).
6. Position the insertion point at the location where you want the characters to be pasted, then click on Edit Paste.

HOT TIP

Use the Symbol and Wingdings TrueType Fonts to Insert Special Characters

The Symbol TrueType font includes many symbols that aren't available in the character sets for other fonts. The Wingdings font contains additional icons and symbols. You should take the time to examine these fonts for symbols and icons that you might want to use eventually. To do so, just display the Character Map and select either the Symbol or Wingdings font name to view the entire character set.

Using Raster Fonts

Windows installs several raster font files in your WINDOWS\SYSTEM directory, all of which have the extension FON. However, the specific FON file names that are installed with Windows 3.1 vary from system to system, and are largely dependent on the type of monitor that is used by the system. Also, some raster fonts are included chiefly for Windows' internal use or for compatibility reasons. You can save yourself some confusion and potential headaches by learning the intended purpose and features of each raster font.

All Raster Fonts Are Resolution-Dependent

Since every raster font is intended for a specific display device, Windows installs only those font files that match the resolution and aspect ratio for your video card. The aspect ratio of a display device is the ratio between the screen's horizontal and vertical resolution.

Five of the available raster fonts—Courier, MS Sans Serif, MS Serif, Small, and Symbol—are available in six possible resolutions. In other words, the Windows installation diskettes include six files for each font. Again, the fonts installed on your system depend on the resolution and aspect ratio of your video card. Windows appends a letter, from A through F, to the name of each font file to indicate the resolution that the file supports. Table 11.3 shows the names of the five font files, where x is the letter appended to the file. Table 11.4 shows which font sets match which device resolution and aspect ratio.

You'll notice in Table 11.4 that two of the font sets are specifically intended to support printer output. Your printer normally will only print a raster font if the resolution and aspect ratio are close to the resolution and aspect ratio of your screen. (They don't have to match exactly.) However, if your printer will not print the font set installed for your video card, you can install the C or D font sets to provide printer support. To do so, you'll first need to use the EXPAND utility to uncompress the font files. (See Appendix A for the procedures on using EXPAND. After you've uncompressed the font files, you can use the Add Fonts dialog box to install the fonts.

The Small font is an alternative for displaying text smaller than 6 points. At sizes below 6 points, TrueType has difficulty—even with hinting—displaying readable text. Unfortunately, Small is only available in three point sizes—2, 4, and 6. On a VGA monitor, 6-point TrueType characters are just as readable or even more readable than Small-font characters. At 4 points, the Small font is difficult to read, but it's marginally better than a TrueType font displayed at 4 points. The 2-point Small-font size is all but useless, since many applications won't allow you to display characters smaller than 4 points, and because 2-point type—even when printed—is not discernable to the human eye.

Raster Fonts Can Only Be Displayed in Documents if the Selected Printer Can Print Windows Raster Fonts

Your applications won't let you use Windows raster fonts if they can't be printed. Because LaserJet and PostScript printers can't print Windows raster fonts, you can't select them in your applications.

Suppose your currently selected printer is the HP LaserJet III. If you're using Excel and you want to select a font for your labels, the list of Fonts that Excel displays will only include HP printer fonts, TrueType fonts,

Table 11.3 File Names for Five Raster Fonts

Font	*File Name*	*Example*
Courier	COURx.FON	COURA.FON (Set 1)
MS Serif	SERIFx.FON	SERIFB.FON (Set 2)
MS Sans Serif	SSERIFx.FON	SSERIFB.FON (Set 2)
Small	SMALLx.FON	SMALLE.FON (Set 5)
Symbol	SYMBOLx.FON	SYMBOLF.FON (Set 6)

Windows vector fonts, and any fonts that you've installed using a font manager. You won't able to select Windows' raster fonts—such as MS Sans Serif, Courier, System, and Terminal—even though these fonts have been installed.

If you want to view and use raster fonts, the only way to do so is to install a printer driver for a dot-matrix printer or some other printer that lets you print Windows raster fonts; you'll then need to select this as the active printer while you're creating a document. However, there isn't much reason for doing this, unless you need to convert and use Windows-based documents with a DOS application and you want to ensure that the formats for the two versions are fairly similar.

If your active printer is a LaserJet or PostScript printer, and you open a document that was created using a Windows raster font, Windows will try to substitute the most closely matching TrueType or vector font for the raster font used originally in the document. Again, if you want to view the document in the format and font in which it was created, you can do so by installing just about any dot-matrix printer driver and making this the active printer. Keep in mind that you won't be able to print the document until you switch back to your "real" printer.

Table 11.4 Main Font Sets that Support Most Monitors and Dot-Matrix Printers

Font Set	*Device*	*Horizontal Resolution*	*Vertical Resolution*	*H:V Aspect Ratio*
A	CGA monitor	96 dpi	48 dpi	2:1
B	EGA monitor	96 dpi	72 dpi	1.33:1
C	Printer	60 dpi	72 dpi	1:83
D	Printer	120 dpi	72 dpi	1.67:1
E	VGA monitor	96 dpi	96 dpi	1:1
F	8514/a monitor	120 dpi	120 dpi	1:1

Windows 3.1 Creates a Font Substitution Table During Installation

Because Windows 3.1 includes improved raster fonts and adds TrueType fonts, Windows replaces a few of the fonts from earlier Windows versions with similar but improved Windows 3.1 fonts. To support any documents you might have that contain these outdated fonts, Windows includes a font substition table so that older fonts in a document can be replaced by their updated versions.

In the font substitution table, MS Serif replaces the Tms Rmn (Times Roman) raster font that was included with previous versions of Windows. Likewise, MS Sans Serif replaces the previously used Helv (a version of Helvetica) raster font. Two vector fonts—Helvetica and Times—are replaced in Windows 3.1 by the TrueType fonts Arial and Times New Roman, respectively. When Windows Setup removes these older fonts during installation, it adds the following entries to the [FontSubstitutes] section of your WIN.INI file:

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

So, if one of your Excel spreadsheets contains, say, labels in the older Helvetica vector font, Windows replaces the text for your labels with the Arial TrueType font. You don't have to worry about changing the font yourself. However, because TrueType fonts use a unique character spacing scheme, you might notice some difference in word spacing and end-of-line wrapping when Windows makes a font substitution.

HOT TIP

You Can Add Entries to the Font Substition Table to Make Substitutions for Virtually Any Font

Suppose your office printer is a 24-pin dot-matrix IBM Proprinter. In the more primitive days of Windows 3.0, you used the fixed-width, Prestige 10-point font (Prestige 12cpi) to print all your correspondence. With Windows 3.1, though, you're not limited to printer fonts and compatible raster fonts. You can use any TrueType font with your dot-matrix printer. So, you might want to tell Windows to replace the Prestige printer font in your documents with a particular TrueType font. (Remember, the Prestige font will appear in a Font(s) list box with the printer icon to the left of the font name.)

In this way, if you reprint any of your older documents, they'll be printed using the more attractive and proportionally spaced TrueType font. Plus, with the TrueType font, you can scale the fonts in your documents to any size. You aren't limited to the single, 10-point size available (in this case) with the Prestige font. To make this substition, all you need to do is add this entry to the [FontSubstitutes] section of WIN.INI:

```
Prestige 12cpi=Times New Roman
```

You can use this technique to substitute any font with any other available font. This technique is especially useful for mapping soft fonts and older scalable fonts to TrueType fonts. For instance, suppose you've been using BitStream's FaceLift to create scalable type in your documents prior to Windows 3.1. If you decide to disable or remove FaceLift from your hard disk, you can still display and print the documents in Windows, because Windows will try to substitute a TrueType font for the FaceLift fonts. However, the printing substitutions that Windows makes might not be to your liking. You're better off controlling these substitutions yourself by adding entries for each FaceLift font that you've used in the past.

Note: Font substitutions affect only your printed documents. Windows will not automatically change the fonts in your documents in response to an entry that you make in the [FontSubstitutes] section of WIN.INI. (Of course, Windows will temporarily substitute fonts if the fonts in the document are not available for the currently selected printer.) If you want to permanently change the fonts used in a document, you must make the changes yourself through an appropriate Windows application.

Windows Includes Several Fonts Intended Specifically for Its Internal Use

If you examine the FON files stored in the WINDOWS\SYSTEM directory, you'll probably notice files for several fonts that you can't identify. Windows installs several raster fonts that are intended to be used solely by Windows. These fonts support specific display or screen-drawing operations, such as the display of text in standard mode, text for non-Windows applications displayed in a window, and for the display of text in Windows components—including menus, title bars, and in the Clipboard.

If you want to have full control over your Windows resources, it's a good idea to learn how Windows uses the FON files stored in the WINDOWS\SYSTEM directory. For most of these internal files, you won't see a corresponding font name in the Font(s) list box for a Windows application. That makes sense, since the fonts aren't intended for your use. However, three of these internal-use fonts—System, FixedSys, and Terminal—will appear in the Font(s) list box for most applications. You can use these fonts to create documents if you want (although these specific fonts won't print on most printers). We'll discuss these three fonts next.

Windows uses the System font to display text in menus, title bars, and dialog boxes. Text displayed in the System font is easy to read because the font automatically displays in bold. You might want to use this font if you find it easy on your eyes. For instance, text displayed in the System font is easy to see on laptops and notebook computers that use LCD screens. Figure 11.8 includes an example of the System font. Keep in mind that you'll only be able to select this font in an application (as is true for any Windows raster font) if your currently selected printer can print Windows raster fonts.

The System font file that Windows installs on your computer depends on your video card. Windows 3.1 provides these three System font files for EGA, VGA, and 8514/a video cards:

System Font File Name	*Supports*
EGASYS.FON	EGA cards (640x350 resolution)
VGASYS.FON	VGA cards (640x480 resolution)
8514SYS.FON	8514/a cards (1024x768 resolution)

FixedSys The Fixed System font (which is called FixedSys in the Font(s) list box of most Windows applications) is included for downward compatibility with Windows applications that were designed for Windows 2.x versions. In fact, FixedSys font is just the older-style System font used in these earlier versions of Windows. FixedSys, like the System font, also appears in bold by default, but text is more expanded in FixedSys than in System.

Figure 11.8 includes an example of text created in FixedSys. All fonts shown in Figure 11.8 are raster fonts, installed for the VGA video driver. These fonts are only available for selection if your current printer can print Windows raster fonts.

Windows 3.1 provides these four FixedSys font files for EGA, VGA, and 8514/a video cards (two resolutions are available for EGA cards):

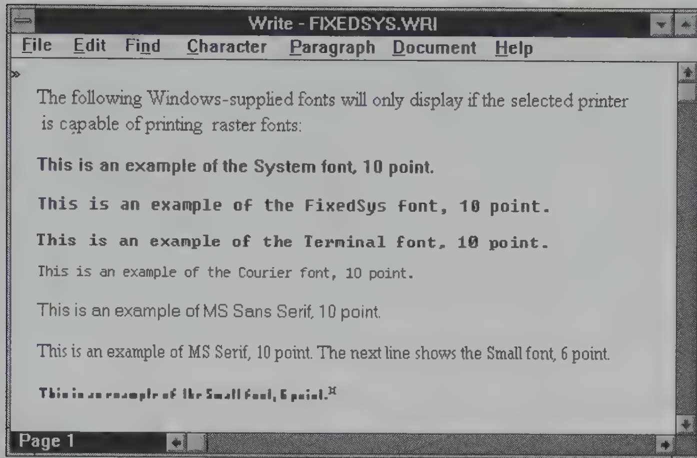


Figure 11.8 Examples of different Windows fonts.

FixedSys Font File Name	*Supports*
EGAFIX.FON	EGA cards (640x350 resolution)
EGAFIX.FON	AT&T EGA cards (640x400 resolution)
VGAFIX.FON	VGA cards (640x480 resolution)
8514FIX.FON	8514/a cards (1024x768 resolution)

If you do not use any older Windows applications that were designed to run under Windows 2.x versions, you can delete the FixedSys font file from your hard disk. Of course, this font will then no longer be available for selection in your applications.

Terminal The Terminal font is designed to support text copied to the Clipboard from Windows and non-Windows applications that use the OEM (also called PC-8) character set, rather than the ANSI character set. Text displayed in the Terminal font looks almost identical to the FixedSys font with boldfacing removed. Figure 11.8 includes an example of text created in Terminal. Windows 3.1 provides these four Terminal font files for EGA, VGA, and 8514/a video cards (two resolutions are available for EGA cards):

Terminal Font File Name	*Supports*
EGAFIX.FON	EGA cards (640x350 resolution)
EGAFIX.FON	AT&T EGA cards (640x400 resolution)
VGAFIX.FON	VGA cards (640x480 resolution)
8514FIX.FON	8514/a cards (1024x768 resolution)

Some Fonts Are Used Solely to Display Non-Windows Applications in a Window

As we've explained in earlier chapters, Windows suspends much of its operations when you run a non-Windows application in standard mode—or if you're displaying the application full-screen, and in the foreground, in enhanced mode. When you turn control of your system over to a DOS application, Windows allows the application to handle the way text is displayed. But this approach won't work when you display a DOS application in a window, in enhanced mode.

For you to create a virtual machine for each DOS application, Windows has to take control of the display environment. That means the routines that your DOS application uses to create text on the screen effectively get put to sleep. Windows is left to its own devices to display text for your windowed application.

But there's a snag. DOS applications need to know your screen resolution in order to operate correctly. When you install a DOS application, it either defaults to a particular screen resolution or asks you to specify the resolution for your video card. Many relatively old DOS applications don't recognize video cards newer than EGA resolutions. Really old DOS applications only recognize CGA resolutions or Hercules monochrome graphics resolutions. Most monitors are downward-compatible—that is, you can display EGA or CGA text on a VGA monitor, but you can't display higher resolution text with an application that only recognizes low-resolution monitors.

To sidestep this problem, Windows includes several raster fonts that are used only when you display a DOS application in a Window. The font that Windows uses depends on the resolution that your DOS application expects and the language that Windows has been set up to support. For instance, some applications allow you to display a character set that supports accents and other marks used in foreign languages. Character sets for different languages are defined by DOS as *codepages.* The U.S. codepage is 437. The multilingual character sets that support diacritical marks for many Indo-European languages is called codepage 850. Other codepages support specific character sets for other languages or special-purpose uses.

Table 11.5 presents the font files that Windows can install on your system to support the display of DOS applications in a window. The Video Support column indicates the number of characters that can display on a line and the number of lines that can display per screen. (For instance, 80 characters x 43 lines means that the application can display 80 characters per line, and 43 lines per screen.)

You should keep these files on your hard disk unless you know that one or more of these resolutions is never used by your DOS applications. You can also delete these files if you never expect to display DOS applications in a window. All of these files are stored in the directory WINDOWS\SYSTEM. Keep in mind that Windows installs only a subset of the files listed in Table 11.5, depending on your video card and the current language setting for Windows.

Windows Includes a Separate Font File for Modifying DOS Text in a Window

The file DOSAPP.FON can modify the resolution of the default font displayed in a DOS window. This file allows you to adjust the vertical and horizontal resolution of the display font to adapt the font to the capabilities of your monitor.

Table 11.5 Windows Installs a Subset of These Files to Support the Display of Text in a DOS Window

Font File Name	*Language Support*	*Video Support*
CGA40WOA.FON	U.S.	CGA 40 characters x 25 lines
CGA40850.FON	Multilingual	CGA 40 characters x 25 lines
CGA80WOA.FON	U.S.	CGA 80 characters x 25 lines
CGA80850.FON	Multilingual	CGA 80 characters x 25 lines
EGA40WOA.FON	U.S.	EGA 40 characters x 43 lines
EGA40850.FON	Multilingual	EGA 40 characters x 43 lines
EGA80WOA.FON	U.S.	EGA 80 characters x 43 lines
EGA80850.FON	Multilingual	EGA 80 characters x 53 lines
HERCWOA.FON	U.S.	Hercules monochrome cards
HERC850.FON	Multiligual	Hercules monochrome cards
VGA850.FON	Multilingual	VGA
VGA860.FON	Portuguese	VGA
VGA861.FON	Icelandic	VGA
VGA863.FON	French Canadian	VGA
VGA865.FON	Norwegian/ Danish	VGA

DOSAPP.FON is a new feature in Windows 3.1 and operates independently of the default that Windows uses to support a particular DOS application. The default font (such as CGA80WOA.FON) supports the screen resolution for the video card that your DOS application expects. DOSAPP.FON then lets you adjust the screen resolution according to the capabilities of your monitor.

To use DOSAPP.FON to change the font (the resolution, really) for a particular DOS application, follow these steps:

1. Start the DOS application and display it in a window (press **Alt+Enter**).
2. Click on the Control-menu box to display the window's Control menu.
3. Click on Fonts to display the dialog box shown in Figure 11.9.
4. Select a different font resolution in the Font(s) list box. An example of the resolution you've selected appears in the Selected Font box.
5. If you want to make the new font resolution permanent for that application, make sure the Save Settings on Exit box is checked.
6. Click OK to exit the Font Selection dialog box.

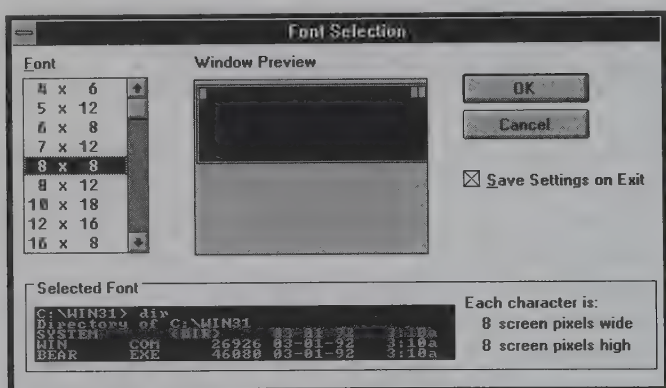


Figure 11.9 Use this dialog box to change the way text displays in a DOS window.

CHAPTER

12

Using TrueType

We've heard a lot of accolades given recently to Microsoft's "new" TrueType fonts. But let's give credit where credit's due. Microsoft licensed the actual fonts from a type foundry company called Monotype Typography, which has been designing typefaces since the turn of the century. And TrueType itself, or a good chunk of it, belongs to Apple Computer, from whom Microsoft licensed the technology.

But Microsoft *is* responsible for fitting TrueType technology with Windows 3.1. And much of the simplicity of TrueType stems from Microsoft's efforts to ensure that TrueType works with *every* Windows-supplied printer driver, except for the Generic/Text Only TTY.DRV driver (which is designed to support daisy wheel and other printers that use preformed characters) and the HPPLOT.DRV plotter driver.

While TrueType has many advantages not found in other font technologies, it does have some limitations as well. In this chapter, we'll describe both its strong points and its shortcomings. Note, however, that we won't deal with TrueType troubleshooting issues that relate to specific printers. You'll find that information in Chapter 13 and in Chapter 16. TrueType problems that are related to specific applications are described in Chapter 17.

TrueType Characteristics

We introduced TrueType fonts briefly in Chapter 11. Every TrueType font uses two files to create characters: an .FOT file and a .TTF file. The .TTF file contains the actual font descriptions, scaling information, and hints that the TrueType engine uses to create TrueType fonts for screen display and for printing. If you use the File Manager to look at the size of your .FOT files (located in the WINDOWS\SYSTEM directory), you'll notice that each .FOT file is only about 1.3K. That's because the .FOT file essentially just tells the TrueType engine where to find the actual TrueType font (the .TTF file)—either in memory if it's already resident or on hard disk.

The major benefit of TrueType fonts is that they are scalable. Windows can scale each font from 1 to 999 points in size, simply by using the outline and hints stored in the font's .TTF file. (Applications and even some printers often impose a limit on the minimum and maximum sizes to which you can scale an outline font.)

The TrueType engine (the program that actually generates fonts) can send a nearly identical image to the screen *and* to the printer—for each size you specify for a given font. In other words, Windows does not have to substitute a comparable raster font in order to display a TrueType font in a given size. The same font outline is used to scale and send characters to the screen and printer. Possibly the single most attractive feature of TrueType is its ability to print fully scalable fonts on any printer for which Windows 3.1 supplies a printer driver, except for the two drivers we've already mentioned. Text set in TrueType fonts can be output to any 9-pin or 24-pin dot-matrix printer, to any HP LaserJet printer, to any ink-jet printer, and to any PostScript printer.

Although virtually all Windows 3.1 drivers support TrueType, you'll need to make sure the TrueType fonts are enabled—otherwise you won't be able to use these fonts on screen or with your printer. To enable TrueType fonts, follow these steps:

1. Start the Control Panel, and choose the Fonts icon.
2. Click on the TrueType button to display the dialog box shown in Figure 12.1.
3. Turn on the Enable TrueType Fonts check box. If you want to prevent other fonts from appearing in your application's Font(s) dialog box, you can also turn on the Show Only TrueType Fonts in Applications check box. This will free up some memory, since Windows won't have to track all screen and printer fonts.
4. Click the OK button to save the new settings.
5. Exit the Fonts dialog box and the Control Panel.
6. Click the Restart Windows button.

The totally printer-independent nature of TrueType gives these fonts an edge over comparable PostScript fonts, which typically require a PostScript printer or typesetting device. And because TrueType is built into Windows 3.1, any Windows 3.1 user can read and print TrueType documents created by any other Windows user.

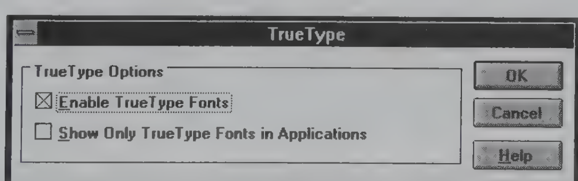


Figure 12.1 Use this dialog box to activate the TrueType engine so that TrueType fonts will be available in your applications.

TrueType Is Device-Independent

Unlike font management packages like FaceLift and Adobe Type Manager (ATM), the engine that generates TrueType fonts is built into Windows 3.1.

Because TrueType is available to every Windows 3.1 user, most documents that contain TrueType fonts can be viewed and printed from any computer system that is running Windows 3.1. Users don't need to purchase a font management package in order to use TrueType.

Also, TrueType fonts do not change their character widths or spacing for different printer resolutions. Line and page breaks in a document that you've created should remain identical regardless of which printer you use to print the document.

HOT TIP

Install the Windows 3.1 HP DeskJet 500 Driver to Support TrueType

If you have an HP DeskJet 500 or 500C printer, the Windows setup program won't update the printer driver when you install Windows 3.1. Microsoft took this precaution to avoid confusion because Hewlett-Packard provides a separate Windows printer driver that allows you to use the DeskJet 500's built-in scalable fonts. By contrast, the Windows 3.1 driver (HPDSKJET.DRV) allows you to use TrueType fonts but doesn't let you use the DeskJet 500's scalable fonts.

If you want to use TrueType fonts with your DeskJet 500, install the Windows 3.1 printer driver *after* you've installed Windows 3.1. When you do this, Windows will add the new HPDSKJET.DRV to your WINDOWS\SYSTEM directory, without deleting the older HP-supplied or Windows 3.0-supplied driver. You can then switch between the two printer drivers depending on whether you want to use TrueType scalable fonts or the LaserJet 500's built-in scalable fonts in your documents. The Windows 3.1 driver will have the description "HP DeskJet 500" or "HP DeskJet 500C" in the Printers dialog box; the HP-supplied driver will have the description "HP DeskJet 500 Scalable Driver."

TrueType Fonts Are Embedded in Your Documents

When you use a font within a document, Windows embeds the full TrueType font within the document in an encrypted (coded)

format. The encryption scheme allows you to display and print the document from any PC running Windows 3.1 and from any Macintosh computer running System 7.

With most vendors of third-party fonts, including PostScript fonts that aren't built into PostScript printers, you cannot move fonts from one system to another without violating the vendor's licensing agreement. Although this kind of licensing violation is commonplace among desktop publishers, it is nonetheless an annoyance.

Suppose you want to send a diskette to a business associate. This diskette contains a report, dozens of pages in length, that has been carefully formatted using, say, PostScript fonts. If you've chosen your fonts from the 35 fonts that are built into most PostScript printers, your associate can view and print the report in the format you've created. But if your report contains fonts that your associate doesn't have, you need to ship the downloadable font files to your associate, along with the report file itself. Without these files, the report won't print correctly—in some cases it won't print at all—even though your associate is using a PostScript printer. Your associate has to install the fonts you've sent before he or she can print your report correctly, and not only is this procedure a hassle—it's illegal. (This same type of font-matching problem has long been a headache for typesetters and service bureaus that have to make sure that all fonts used in a customer's documents have been installed for use by the typesetting device.)

The embedding technology used by TrueType fonts makes this kind of font matching unnecessary. The 14 TrueType fonts shipped with Windows 3.1 are all read/write-enabled. *Read enabling* means that you can display and print documents that use any of the 14 supplied TrueType fonts from any computer system that's running Windows 3.1. *Write enabled* means that you can edit a document that contains write-enabled fonts on any other computer system—using the existing fonts—regardless of whether the fonts have been installed on the system. The 44 additional fonts in the Microsoft TrueType Font Pack for Windows are also fully read/write enabled.

You need to be cautious, though, when you consider buying TrueType fonts from other vendors. Some TrueType fonts are read enabled but write protected. In other words, users who haven't installed these fonts on their system can display and print documents containing the write-protected fonts, but they can't edit the documents.

Currently, many off-the-shelf TrueType packages are read enabled only. This is done to prevent users from creating documents with the purchased fonts, then circulating the documents to other users. In this situation, if the fonts weren't write protected, any user could just delete all the text from the documents and then create new documents using the embedded fonts—which they haven't purchased.

Fortunately, shareware and freeware TrueType packages are usually fully read/write enabled. Hundreds of shareware and freeware TrueType fonts are already available, and you can expect to see an increasing number of them made available in the future. Many of these fonts are of excellent quality, and are available at little or no cost.

TrueType Coexists Peacefully with Most Font Management Packages

Windows treats most font management packages as simply other Windows applications. This means you can run TrueType together with Bitstream's FaceLift, Adobe's ATM, and most other font managers that are designed to run under Windows.

If you want to continue using fonts with a font management package, you don't have to disable TrueType to do so. In fact, you don't have to take *any* special steps. Windows will even allow you to mix TrueType fonts with other packaged fonts—in the same document (although this isn't recommended if you want to port your documents to other computer systems).

However, you do need to make sure your font manager is compatible with Windows 3.1. Adobe recommends that you use ATM version 2.0 or higher with Windows 3.1. Use FaceLift version 1.2 or higher with Windows 3.1. At the very least, if you use an out-of-date version of your font manager with Windows 3.1, the font manager often won't work with the Windows 3.1 printer drivers. At worst, your fonts won't display correctly or your font management software won't run at all.

From a Performance Standpoint, TrueType Is Comparable to Other Font Managers

TrueType can generate fonts for display and printing at speeds that are competitive with ATM, FaceLift, and other font managers. TrueType fonts also print about as fast as PostScript fonts.

When you select a new TrueType font in a document, TrueType creates bitmaps for the screen by using the outline and hints stored in the font's .TTF file. For this reason, you'll notice that font generation is much slower the first time you select a TrueType font than if you had selected a

Windows raster font. But generation of bitmaps with TrueType is usually faster than with ATM, FaceLift, or other font managers, because the TrueType engine is internal to Windows; messages don't need to be passed between Windows and an external font management program.

Each time you select a new font size, TrueType has to generate a new set of bitmapped characters—even if you've previously selected the same font in a different size. However, after TrueType has generated bitmaps for a new font size, the bitmaps are stored in a cache. So, if you apply the font and font size to other parts of the document, you won't notice any performance lag.

ATM, FaceLift, and other font managers also typically store screen bitmaps in a cache, so the performance benefit you get from generating TrueType fonts occurs only the first time you select a font. When you apply a TrueType font again—after the bitmaps have already been generated—you probably won't notice any improvement in speed over ATM, FaceLift, or other font managers.

TrueType is also competitive in its ability to generate fonts quickly for printing. With PostScript and HPPCL (LaserJet) printers, Windows downloads only those characters needed for printing—a technique that helps TrueType perform faster than other font managers. TrueType doesn't download an entire font at one time. Also, some font managers send each page as a graphics image (a bitmap), rather than sending characters to the printer. (For example, on LaserJet printers, ATM downloads each page as one massive bitmap, which takes time to load and consumes quite a bit of printer memory.) Although the TrueType engine *can* send true type fonts to your laser printer as a graphic image, with most printers you have the option to switch between the two downloading approaches.

With PostScript printers, TrueType fonts are first converted to Adobe Type 1 outline fonts in your computer's CPU and then downloaded to the printer. For this reason, downloading and printing with TrueType fonts tends to be slower than using the 35 built-in printer fonts available with most PostScript printers. If your printing needs are complex or if you routinely saddle your PostScript printer with a heavy output load, you might want to stick with printer fonts.

In Chapter 13, we'll discuss some specific tips and techniques for using TrueType fonts with PostScript and LaserJet printers.

TrueType Printing Options

The characteristics and features of TrueType fonts present some interesting options and alternatives for printing. Many of these features vary depending on the type of printer you're using. We'll explore some of these features in the following section.

Printing TrueType Fonts as Graphics on Non-PostScript Laser Printers

The ability to print TrueType fonts as Graphics varies for different laser printers. (This feature does not apply to PostScript printers.) For instance, some printers will only let you print TrueType fonts as graphic images, rather than as downloadable soft fonts. Other printers will only accept TrueType fonts as soft fonts, not as graphic images.

When you send TrueType fonts to a non-PostScript laser printer as downloaded soft fonts, the Windows printer driver sends font commands to your printer's page description language along with the characters to be printed. This is usually faster than printing TrueType fonts as graphic images, since the Windows printer driver has to send a large amount of binary data to the printer to create graphic images.

It's usually best to send TrueType fonts to your laser printer as downloaded soft fonts unless you are printing several pages of graphic images, and only a relatively small amount of TrueType text. Unfortunately, not all printers support downloadable soft fonts. And to compound the problem, the technique for turning the "print TrueType as graphics" feature on or off varies for different printers.

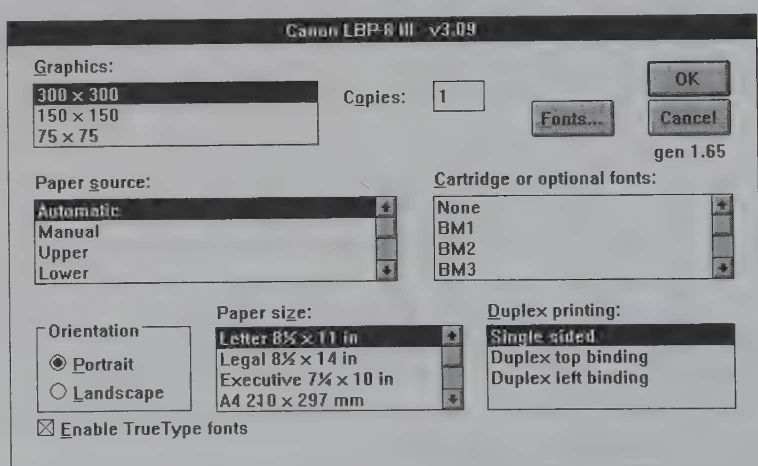


Figure 12.2 For the Canon LBP II and LBP III series of printers, the "Enable TrueType fonts" check box must be turned on to support TrueType fonts.

If you are using a Canon LBP II series or Canon LBP III series printer, you must print TrueType fonts as graphics. To do so, display the Setup dialog box for your printer, and then turn on the "Enable TrueType fonts" check box, as shown in Figure 12.2. This allows the printer to accept TrueType fonts—and as graphic images only.

If you are using an IBM Laser Printer 4019 or an Olivetti PG306 printer, you must turn on the "TrueType fonts as Graphics" check box in the printer's Setup dialog box, as shown in Figure 12.3. If this check box is turned off, your laser printer won't print TrueType fonts because the Windows printer driver will try to download TrueType characters as soft fonts, which the IBM Laser Printer 4019 and Olivetti PG306 printers don't support.

If you use an HP LaserJet printer model that is earlier than the Series II, such as HP LaserJet, HP LaserJet Plus, and the HP LaserJet 500+, you can print TrueType fonts *only* as raster graphics fonts. This is essentially the same technique used to print TrueType fonts on dot-matrix printers. You do not need to take any additional steps to enable this capability. However, you'll note that a "Print TrueType as Graphics" option is not available for these earlier printers.

For LaserJet printer models later than the Series II, you have the option of sending TrueType fonts to the printer either as soft fonts or as graphics. The "Print TrueType as Graphics" check box controls this capability and is available from your printer's Options dialog box.

For additional TrueType information regarding specific printers, see Chapters 13 and 16.

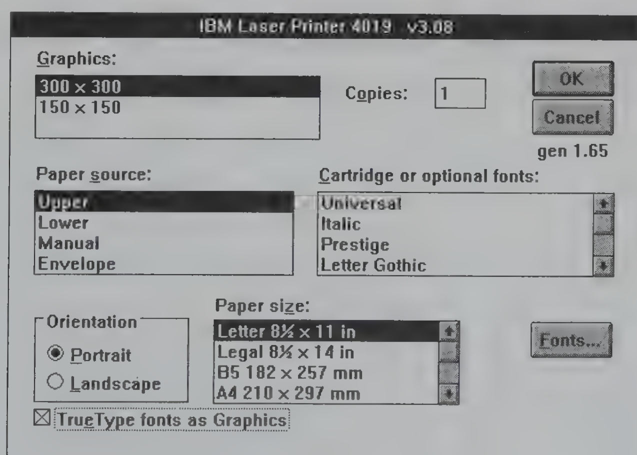


Figure 12.3 For the IBM Laser Printer 4019 and the Olivetti PG306 printer, the "TrueType as Graphics" check box must be turned on to support TrueType fonts.

For the LaserJet III, TrueType Offers Increased Font Selections

Although HP LaserJet III printers provide built-in scalable fonts, only CG Times and Univers are scalable. TrueType adds capabilities to LaserJet III PostScript printers by increasing the number of available scalable fonts.

All LaserJet III printers use the HPPCL5A.DRV printer driver, which supports the PCL-5A page description language used in the HP LaserJet III series. PCL-5A includes the CG Times and Univers scalable fonts, plus the Courier and LinePrinter fonts. With Windows 3.1, you automatically increase the number of scalable fonts available to your LaserJet III , since you can now use the built-in TrueType fonts as well as CG Times and Univers. The Times New Roman TrueType font is very similar to CG Times, while the Arial TrueType font is very similar to Univers—so you won't gain much benefit from these fonts. However, TrueType's Courier New font is scalable, whereas the LaserJet III's Courier printer font is not. And, of course, the Symbol and Wingdings TrueType fonts also are scalable and can be downloaded to the LaserJet III. Also, if you've enabled TrueType fonts for your LaserJet III printer, you can install and use additional, vendor-supplied TrueType fonts.

Even if you own a LaserJet III PostScript printer, TrueType offers you benefits because you can now install any TrueType font as well as any PostScript font for use with your printer.

HOT TIP

Install Only Those Fonts that You Are Sure You Will Use

Each time you add a font to Windows 3.1, information about the font name and its location gets added to your WIN.INI file so that Windows can make the font available to you within your applications. This is true regardless of whether you add TrueType fonts, ATMfonts, FaceLift fonts, or any other Windows-compatible font.

The more fonts you add, the more corresponding entries will get placed in your WIN.INI file. As the size of your WIN.INI file increases, so too does the time required for you to boot Windows, since Windows has to read and interpret every line in the file. If your WIN.INI file includes several hundred font definitions or font substitutions, you might notice a substantial delay (10 seconds or more) in the time Windows requires to load.

For this reason, it's a good idea to install only those fonts that you expect to use. If you buy a font package that automatically installs, say, 100 or more TrueType fonts, but you only expect to use a dozen or so of these, you can still control the size of your WIN.INI file. Simply open WIN.INI, find the section where the fonts you've installed are defined, and comment out the definitions of fonts you don't expect to use in the near future. (To comment out a line in WIN.INI, just add a semicolon at the start of the line.) Later, if you decide you want to use fonts that you've previously commented out, you can delete the semicolons and reboot Windows to make the fonts available.

Bitstream Has Released TrueType Versions of Most of Its Fonts

Bitstream has already seen the handwriting on the wall, realizing that TrueType is likely to become a standard font generator for Windows, and has responded by developing TrueType versions for most of its fonts. The TrueType versions have font names identical to their FaceLift predecessors, which makes it easy for you to install and use the fonts. You can even disable and remove FaceLift from your hard disk if you're using only TrueType fonts.

Since Bitstream's TrueType "upgrades" share the same names as the FaceLift-dependent versions, you can continue to use your favorite Bitstream fonts. If you install the new fonts but don't change the fonts in your existing documents, Windows will print using the TrueType font. However, you won't be able to port these documents to other systems because your documents technically still contain FaceLift fonts. To support portability of your documents, you need to manually change the fonts in your existing documents. By doing so, you embed the TrueType fonts in the documents, which allows you to display and print them on any system that's running Windows 3.1.

Although Bitstream probably doesn't want you to abandon FaceLift, there's really no reason *not* to remove FaceLift from your system if you're using only TrueType fonts. If you want to remove FaceLift, you should first disable it (make it unavailable to Windows) by following these steps:

1. Open the Bitstream FaceLift group window, then double-click on the Bitstream FaceLift program icon.
2. Double-click on the Parameters icon, then turn off the FaceLift active check box.

3. Click on OK to exit the Parameters dialog box, then click on OK to acknowledge the warning dialog box that appears.
4. In the Bitstream FaceLift group window, double-click on the Printers icon.
5. Select a printer name in the Installed Printers list box, then turn on the Unassign SHELL-Driver option button.
6. Click on OK to exit the dialog box, then click on OK to acknowledge the warning dialog box that appears.
7. Click on the Restart Windows button.

Disabling FaceLift does not remove the FACELIFT.EXE program from the WINDOWS directory, nor does it remove the fonts from the BTFONTS directory. You must remove these files manually if you want to remove FaceLift from your hard disk. However, if you think you might need to use FaceLift at a later time, you might want to leave the FaceLift files on your hard disk. You can re-enable FaceLift easily by reversing the procedure described above.

HOT TIP

Disable FaceLift If You Need to Troubleshoot Font-Related Problems

If you need to track down printing problems and are unsure whether a printing problem is related to FaceLift or the Windows printer driver, follow the above steps to disable FaceLift. If your printing problems persist, you can be reasonably sure that the problem does not stem from FaceLift.

Converting ATM Type 1 Fonts to TrueType Fonts

If you purchase a low-cost font converter, you can convert your Adobe Type Manager Type 1 (outline) fonts to TrueType fonts. You might find this useful if you want to continue using your ATM fonts but also want to disable ATM in favor of TrueType. Also, dozens of shareware and freeware ATM fonts are available at little or no cost. If you don't own ATM, you can use these fonts by converting them to TrueType—a low-cost way to expand your library of TrueType fonts.

Several conversion utilities are available to convert Adobe's Type 1 outline fonts (which are really PostScript fonts) to TrueType fonts, including Ares Software's Font Monger for Windows and Atech Software's Alltype.

Keep in mind that ATM version 2.0 and higher will coexist with the TrueType engine, so you might not find it necessary or useful to convert your ATM fonts. But there are a few compelling reasons for doing so. By converting your ATM fonts to TrueType, you can take advantage of TrueType's font embedding technology. In other words, you can display and print documents that contain TrueType fonts using any other computer system that's running Windows 3.1. (Of course, you'll have to manually change the ATM fonts in your existing documents to TrueType fonts.) This capability is not available with ATM. Also, if you convert your ATM fonts to TrueType, you can remove ATM from your hard disk.

If you decide to remove Adobe Type Manager from your hard disk, you should first disable it so that Windows will not try to look for ATM files and fonts when it boots. (You'll conserve memory by reducing the number of fonts that have been installed for use with Windows.) If you want to remove ATM from your hard disk, first follow these steps:

1. Use the System Configuration Editor, Notepad, or the DOS 5 Edit utility to load SYSTEM.INI.
2. Remove the following lines from the [boot] section:

   ```
   SYSTEM.DRV=ATMSYS.DRV
   ATM.SYSTEM.DRV=SYSTEM.DRV
   ```

3. Replace the deleted lines with the following line:

   ```
   SYSTEM.DRV=SYSTEM.DRV
   ```

4. Save the revised SYSTEM.INI file.
5. Open WIN.INI and delete all lines that begin with the entry *Softfont=*.
6. Save WIN.INI and restart Windows.

These steps disable ATM but don't actually remove the Adobe Type Manager system files or the installed ATM fonts. You must delete these files yourself.

HOT TIP

Disable Adobe Type Manager If You Need to Troubleshoot Font-Related Problems

If you need to track down printing problems, and are unsure whether a printing problem is related to Adobe Type Manager or the Windows printer driver, follow the above steps to disable ATM—*with one* *critically important exception.* Wherever one of the preceding steps says to delete a line, instead comment out (disable without deleting) the line by placing a semicolon in front of it. If your printing problems persist, you can be reasonably sure that the problem does not stem from ATM. Go ahead and re-enable ATM by removing the semicolons from the appropriate lines in SYSTEM.INI and WIN.INI.

HOT TIP

Service Bureaus Can Usually Easily Handle TrueType Fonts

We've read a lot of misleading information recently about TrueType and typesetting systems used by service bureaus. Much of the confusion stems from a misunderstanding about the way TrueType converts fonts for PostScript printers and typesetting systems. TrueType does *not* send fonts to a PostScript printer as bitmaps; fonts are sent as Adobe Type 1 fonts by default. (However, you can specify that TrueType fonts be sent to the printer as Adobe Type 3 bitmaps.) This is the same approach used by ATM and of course by all Type 1 PostScript fonts. You can verify this, if you have a PostScript printer installed, by following these steps:

1. Start the Control Panel and choose the Printers icon.
2. Make sure your PostScript printer is the selected printer, then choose Setup.
3. Choose Options to display the Printer Options dialog box.
4. Choose Advanced to display the Advanced Options dialog box, shown in Figure 12.4. If you click on the Send to Printer as: pull-down list box, you'll notice that the *only* two options for sending TrueType fonts to a PostScript device are the Adobe Type 1 (outline) and Adobe Type 3 (bitmap) formats—the same formats used by ATM, the Macintosh, and other PostScript font managers.

So, there is no substantial benefit in using ATM (or other font management systems) over TrueType fonts. Existing ATM fonts can be converted easily to TrueType fonts. And as long as a service bureau's PCs are running Windows 3.1, TrueType will space and kern text just as reliably (and in some cases, even more so) than with other font managers or even with third-party PostScript-compatible fonts.

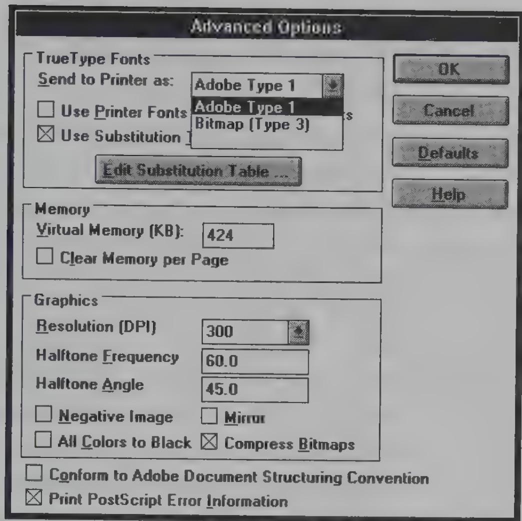


Figure 12.4 This dialog box shows that TrueType fonts are always sent to PostScript printers in either Adobe Type 1 or Type 3 format.

TrueType Fonts Can Be Used on Macintosh Computers That Are Running System 7

TrueType technology is owned by Apple Computer, and licensed to Microsoft. This relationship can prove to be of great benefit if you need to port documents between PCs and Macintosh computers.

The TrueType engine is built into the Macintosh System 7 operating system. Therefore, any TrueType fonts in a document created under Windows can be displayed and printed on a Macintosh that's running System 7. You don't need to convert fonts, since the Macintosh recognizes and retains the TrueType fonts and embedded code in your Windows-created documents. This feature is especially important if you want to send your document to a typesetting company or service bureau that uses Macintosh computers exclusively.

The same is true for documents ported from the Macintosh to a PC running Windows 3.1. You can use Windows applications to display TrueType fonts in documents originally created on a Macintosh. You won't need to perform any font conversions in order to retain the TrueType fonts used in the Macintosh version of the document.

Using the TrueType Font Pack

The Microsoft TrueType Font Pack for Windows Provides a Competitive Alternative to Built-in PostScript Printer Fonts. The Font Pack for Windows must be purchased separately; these fonts aren't bundled with Windows 3.1. But the Font Pack is a useful addition to Windows 3.1, especially if you don't own a PostScript printer. The Font Pack is available from Microsoft for under $100 and includes 22 fonts created by Monotype Typography and 22 Lucida fonts created by Bigelow & Holmes, Inc.

Microsoft introduced the TrueType Font Pack in part to claw its way into the professional publishing world, and in part to convince users that TrueType can meet every possible business and professional printing need. In other words, Microsoft wants Windows to become the standard for desktop publishing and business computing.

The 22 fonts provided by Monotype, added to the 14 TrueType fonts supplied with Windows 3.1, effectively match the 35 scalable fonts built into most PostScript printers. In a sense, Microsoft is trying to convey the message: "If you've got the Font Pack, you don't need a PostScript printer to do professional desktop publishing." Microsoft may have a point, especially since dozens of font vendors have jumped on the TrueType bandwagon. You can expect the number and variety of TrueType fonts to soon rival the number and variety of fonts available for PostScript output—especially since existing ATM fonts can easily be converted to TrueType fonts. In fact, Reasonable Solutions, a shareware distributor, now provides TrueType versions for all of their ATM fonts. This means that more than 300 TrueType fonts are available from this source alone.

Table 12.1 lists the 35 built-in fonts available in PostScript printers that use the Apple LaserWriter Plus standard, along with their corresponding TrueType fonts. The first 14 fonts, of course, are supplied with Windows 3.1. The remaining 22 fonts are part of the Monotype set included in the Font Pack. (Note that the Wingdings TrueType font doesn't have a corresponding PostScript printer font.)

The PostScript printer driver by default sends all of the TrueType fonts listed in Table 12.1 to the printer as printer fonts. In other words, Windows substitutes the corresponding, built-in printer font for each of the TrueType fonts shown in Table 12.1. This approach saves printing time since it isn't necessary to download fonts; the fonts are already resident in the printer. This technique also lets you display scalable fonts in your documents without requiring the use of ATM or some other font manager to create scaled display fonts. You'll essentially be able to view your documents in WYSIWYG, since the TrueType fonts look very similar to

Table 12.1 TrueType Fonts Compared with PostScript Fonts

TrueType Font	*Corresponding PostScript Printer Font*
Arial	Helvetica
Arial Bold	Helvetica-Bold
Arial Italic	Helvetica-Oblique
Arial Bold Italic	Helvetica-BoldOblique
Courier New	Courier
Courier New Bold	Courier-Bold
Courier New Italic	Courier-Oblique
Courier New Bold Italic	Courier-BoldOblique
Times New Roman	Times-Roman
Times New Roman Bold	Times-Bold
Times New Roman Italic	Times-Italic
Times New Roman Bold Italic	Times-BoldItalic
Symbol	Symbol Set
Wingdings	(none)
Arial Narrow	Helvetica-Narrow
Arial Narrow Bold	Helvetica-NarrowBold
Arial Narrow Italic	Helvetica-NarrowOblique
Arial Narrow Bold Italic	Helvetica-NarrowBoldOblique
Book Antiqua	Palatino-Roman
Book Antiqua Bold	Palatino-Bold
Book Antiqua Italic	Palatino-Italic
Book Antiqua Bold Italic	Palatino-BoldItalic
Bookman Old Style	Bookman-Light
Bookman Old Style Bold	Bookman-Demi
Bookman Old Style Italic	Bookman-LightItalic
Bookman Old Style Bold Italic	Bookman-DemiItalic
Century Schoolbook	NewCentury Schoolbook-Roman
Century Schoolbook Bold	NewCentury Schoolbook-Bold
Century Schoolbook Italic	NewCentury Schoolbook-Italic
Century Schoolbook Bold Italic	NewCentury Schoolbook-BoldItalic
Century Gothic	ITC AvantGarde Gothic-Book
Century Gothic Bold	ITC AvantGarde Gothic-Demi
Century Gothic Italic	ITC AvantGarde Gothic-BookOblique
Century Gothic Bold Italic	ITC AvantGarde Gothic-DemiOblique
Monotype Corsiva	ZapfChancery
Monotype Sorts	ZapfDingbats

their corresponding PostScript printer fonts, and share the same spacing characteristics. Your printer will use PostScript fonts when you print your documents, but your applications will display TrueType fonts. However, you can also instruct Windows to send all or selected TrueType fonts to your printer, rather than their substituted PostScript printer fonts. Chapter 13 explains how to do this.

The 22 Lucida fonts appear to have been selected to provide a font set that can support a wide range of business publishing needs. For instance, the Lucida family includes fonts designed specifically to provide clear output on low-resolution fax machines. Other fonts include special-use characters for mathematical and engineering applications. If you already own a PostScript printer, you still might want to consider purchasing the Font Pack in order to use these special-purpose fonts. Figure 12.5 provides examples of all of these Lucida fonts. By default, all Lucida fonts are sent to PostScript printers as downloaded Type 1 or Type 3 fonts.

Lucida Fonts Were Designed for Low-Resolution Laser Printers

Although the 22 TrueType fonts in the Lucida family are not provided with Windows 3.1, they are bundled with several applications, including Ami Pro and Lotus 1-2-3 for Windows. These fonts are available from Microsoft at a relatively low cost, in its TrueType Font Pack for Windows. Since these fonts are all read/

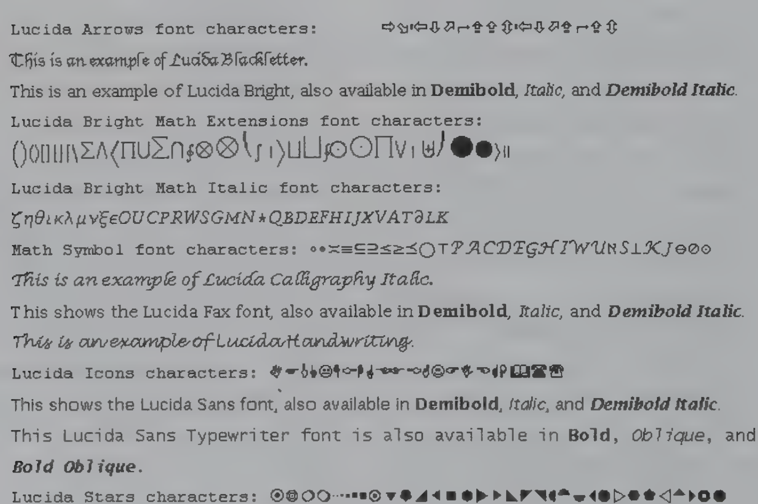


Figure 12.5 This figure shows various Lucida fonts.

write enabled, you can expect to see them distributed widely. For this reason, it is a good idea to recognize that the Lucida fonts, although well drawn, were initially designed for use with low-resolution printers.

The Lucida fonts get their name from the fact that they print clearly and cleanly (they're "lucid") on low-resolution devices. In fact, Lucida character outlines were designed at a time when laser printers were more primitive (usually limited to 300 dpi printing at best) than they are today.

This feature can be both an advantage and a disadvantage to using Lucida fonts. If you are outputting pages to a dot-matrix printer or if you want to quickly output pages using your laser printer's draft (300 dpi) mode, you'll probably find that the Lucida TrueType fonts are a good source for scalable fonts that print exceptionally well at low resolutions. However, if you're using a high-resolution printer or if you're composing material designed for professional publication, you might find that the Lucida typefaces are little on the "simple-looking" side. If so, you might want to opt for more professional-looking fonts.

HOT TIP

You May Need to Manually Install the Font Pack

The setup program that is included with the TrueType Font Pack will generate a general protection error on some systems during the installation of the fonts. If this problem occurs, you can still install the fonts by using the Windows 3.1 EXPAND utility (which is included on Disk 3 of the Windows Installation Disks) to expand and copy all of the .TT_ files to your WINDOWS\SYSTEM directory. Appendix A explains how to use the EXPAND utility. After you've copied the files to your hard disk, follow these steps:

1. If you haven't already done so, rename all of the copied files (*.TT_) so that they have the extension TTF. For example:
   REN C:\WINDOWS\SYSTEM*.TT_ *.TTF
2. Start the Control Panel and choose the Fonts icon.
3. Click on the Add button.
4. Switch to the WINDOWS\SYSTEM directory, click on the Select All button, then click on OK. This will create an FOT file for each TrueType font and will update WIN.INI so that the fonts are installed to your printer port.

Installing the Font Pack to Network Workstations

If you install the Font Pack on your network's file server, you must still set up each workstation to use the fonts.

If you use the setup program provided with the Font Pack to install the TrueType fonts to the server, you'll still need to install the fonts on individual workstations. Here's a quick way to do this:

1. From each workstation, start the Control Panel, and click on the Fonts icon.
2. Click on the Add button, and then switch to the shared Windows directory on the server.
3. Click on the Select All button, then click on OK. This will copy the .FOT files to the workstation's hard disk and will update the workstation's WIN.INI file.

Note: If you don't turn off the Copy To the Windows Directory check box, the .TTF files also will be copied locally. (The .TTF files don't need to be copied to the workstation; however, doing so will allow each user to have access to the TrueType fonts even when the network is down.)

Solving Common TrueType Printing Problems

Although the TrueType engine works with virtually all printers, you might encounter problems using TrueType if your printer driver is old or if your TrueType font files are corrupted or missing. We'll explain how to solve these common printing problems in the following sections. Problems related to specific printers are explained in Chapters 13 and 16.

TrueType Printing and Display Problems Usually Can Be Traced to the Printer Driver

If you're having a problem printing or displaying TrueType fonts, the problem often is that your printer driver is not a Windows 3.1 driver or is not the most current driver available for your printer.

The vast majority of TrueType printing problems result from outdated or faulty printer drivers. And, even though you installed Windows 3.1 "by the numbers," there are still several ways your system could be using a printer driver that doesn't work well (or at all) with Windows 3.1.

First, we should explain why printer drivers tend to be the source of most printing problems in Windows 3.1. Microsoft completely revamped many of its printer driver files for Windows 3.1, chiefly to support TrueType fonts. For dot-matrix printers, Microsoft designed a set of *universal printer drivers*, each of which can support an entire category of printers rather than just one or two specific makes and models of printers. For instance, Windows 3.1 includes the CIT24US.DRV driver file to support all Citizen 24-pin dot-matrix printers, and the PANSON9.DRV to support all Panasonic 9-pin dot-matrix printers.

Most other printer drivers (including the PSCRIPT.DRV file for PostScript printers and the HPPCL.DRV file for LaserJet printers) have been updated to support TrueType fonts and to improve overall printer performance.

If you installed Windows 3.1 using the Express Setup option, Windows automatically updated all of your installed printer drivers. However, if you selected the Custom Setup option, you had to select which printer drivers you wanted to update. If you neglected to update the driver file for one or more of your printers, you could still be using an older Windows 3.0 driver, which won't support TrueType fonts. Since some printer models share nearly identical names, it's also possible to select an incorrect printer driver. For instance, some companies make PostScript and non-PostScript versions for the same printer model. If you selected the wrong printer type during setup, you might also have installed the wrong printer driver.

If you are having problems printing, choose the Printers icon from the Control Panel and make sure the printer name for your active printer is correct. If you are unsure, click the Add button and scroll through the List of Printers box to verify that the name of your Installed Printer does in fact most closely match the name of your printer. Then reinstall the driver for this printer. (Chapter 13 includes step-by-step instructions for re-installing a printer driver.)

Warning: When you reinstall a printer driver, you should first delete the printer driver name from your Printers dialog box, and then delete the actual printer driver file from your hard disk. If you don't delete the existing driver file from your hard disk, and if the existing printer driver was installed during the Windows 3.1 setup process, Windows won't reinstall the driver; it will just use the existing driver file stored on disk. If you are installing a Windows 3.1 driver in order to upgrade over a Windows 3.0 driver, you don't need to delete the existing driver file. However, you will need to make sure you tell Windows *not* to use the existing driver when Windows informs you that a driver for this printer already exists. Chapter 13 includes detailed steps for reinstalling and updating printer drivers.

An additional printer driver problem can result through no fault of your own. Although most of the new printer drivers were tested thoroughly during Windows 3.1's beta testing cycle, some users have reported printing problems—usually with printer models that aren't very widely used. Also, some printers have been placed on the market after the release of Windows 3.1. In most cases, the printer will come equipped with a disk that includes the Windows 3.1 driver. However, it's not unusual for printer-supplied drivers to be out of date, since printers can remain on the manufacturer's or retailer's shelf for months.

In response to these kinds of driver problems, Microsoft has updated several drivers since the initial April 6, 1992 release of Windows 3.1 and has added some new drivers as well. If you experience printing problems, you should always check with Microsoft or your printer manufacturer to see if a more current maintenance release is available for your printer driver. All updated printer drivers are included in the Windows Driver Library (WDL), which is available on many national information services, including CompuServe, GEnie, and Microsoft OnLine. If you don't have access to any of these services but you do have a modem, you can access the Microsoft Download Service (MSDL) by dialing (206) 936-6735 and searching for the current driver file for your printer. Chapter 13 provides additional information on accessing the Windows Driver Library and on using the Microsoft Download Service.

HOT TIP

Check the Version Number for Your Printer Driver to Make Sure It Is Current

You can check the version number for your printer easily from within Windows. Just follow these steps:

1. Choose the Printers icon from the Control Panel.
2. Make sure the name of the printer you want to check is highlighted in the list of Installed Printers.
3. Choose the Setup button, then choose the About button. The version of the printer driver will appear at the top of the About box.

Use this identification number if you are calling Microsoft Product Support, Microsoft's Download Service, or your printer manufacturer to determine whether you have the most current driver for your printer.

TrueType Printing Problems Can Occur If One of a Font's Files Is Missing

As mentioned earlier, every TrueType font uses two files—an FOT file and a TTF file. If one of these files is missing for a particular font, but the font name appears in your application's Font(s) list box, Windows will still try to use the font, but with unpredictable and sometimes disastrous results.

If either a TrueType font's .TTF or .FOT file is missing, Windows might display blank text for the font or might draw the wrong font (as a "best guess" substitution) or even a garbled font. You should suspect this sort of problem if some of your TrueType fonts print fine while others don't.

Although you might have deleted a TrueType font file accidentally, you can also create TrueType havoc by moving your font files from one location to another. It's usually wisest to leave font files in the directory where they're installed—especially with TrueType. Windows expects to find the .FOT file for a TrueType font in the WINDOWS\SYSTEM directory. The .FOT file, in turn, contains the path to the .TTF file. Windows stores all 14 of the supplied TrueType fonts (the .TTF files) in the WINDOWS\SYSTEM directory. If you move any of these .TTF files to a different directory, Windows will not be able to find the files and will behave erratically.

To complicate matters, .TTF files might be installed to a directory other than WINDOWS\SYSTEM if you're installing additional TrueType files from a third-party vendor. In any case, always leave TrueType files in the location where they are installed. If you suspect that a particular TrueType file has been moved or deleted, use the Fonts utility in the Control Panel to reinstall the font.

CHAPTER

13

Using Windows with Your Printer

In this chapter, we'll discuss specific categories of printers and even specific makes and models. We're not going to offer a primer on Windows printing. Instead, we'll focus on questions and problems. In particular, we'll focus on printing issues for HP LaserJet printers and PostScript printers, the two categories of printers most popular with users of Windows and Windows applications.

Much of the material in this chapter is not available in the Windows *User's Guide*, and some of the material deals with printing problems that were uncovered after the initial release of Windows 3.1. Chapter 16 includes additional troubleshooting information for specific printers.

Printer Installation Issues

Getting Windows to work with your printer is usually a simple matter—*if* the Windows installation disks provide a driver for your printer *and if* the printer driver that you install is the most current, bug-free version. In more than a few cases, your particular printer won't be represented in the Windows List of Printers. In other cases, the printer drivers on the Windows 3.1 installation disks might not be the most current ones available. The next few pages of this book deal with these and other general printer-installation issues.

Windows 3.1 Uses Universal Drivers and Several Printer-Support Files

In Windows 3.1, most of the printer drivers that were used in Windows 3.0 have been replaced by universal drivers or have at least been updated to support TrueType fonts. But universal drivers aren't enough to support printing for most printers.

All Windows printer drivers have the extension .DRV. Microsoft uses the term *universal driver* to refer to a driver that can support more than one device model—in this case, more than one printer. For example, the file PSCRIPT.DRV is used to drive all PostScript printers, regardless of the printer's make or model number. Table 13.1 provides a sampling of universal printer driver files along with the specific printers that each driver supports.

It is important to remember that very few printers can be fully supported by their universal printer driver file. In most cases, additional files are required to support model-specific options. These support files

Table 13.1 Examples of Universal Printer Drivers

Printer Driver File	*Supports*
CIT9US.DRV	All Citizen 9-pin printers
CIT24US.DRV	All Citizen 24-pin printers
EPSON9.DRV	All Epson 9-pin printers
EPSON24.DRV	Most Epson 24-pin printers
HPPCL.DRV	Most HP LaserJet printers (except PostScript models)
HPPCL5A.DRV	All HP LaserJet Series III printers (except PostScript models)
NEC24PIN.DRV	Most NEC 24-pin printers (Pinwriter series)
OKI9.DRV	Most Okidata 9-pin printers
PSCRIPT.DRV	All PostScript printers

are used to create dialog boxes or to add printer-specific options to the dialog-box options supplied by the universal driver. Other support files provide help files or install fonts for a specific set of printers.

For example, all PostScript printers use the PSCRIPT.HLP file to display help windows, and the TESTPS.TXT file to test the printer's virtual memory and communications settings. Some PostScript printers require a separate, additional *PostScript description* file. The Epson EPL-7500, for instance, uses the EPL75523.WPD PostScript description file, while the NEC SilentWriter 2 90 printer uses the N2090522.WPD file. (All PostScript description files have the extension .WPD.)

Table 13.2 lists most of the different types of files used to support printers in Windows 3.1.

Windows 3.1 Uses CONTROL.INF and CONTROL.INI to Install Printer Files

Have you ever wondered how Windows 3.1 determines which files need to be installed to support specific printers? To install files for each printer, Windows refers to information stored in CONTROL.INF and CONTROL.INI.

Windows uses essentially the same technique to install printer files during initial setup and at other times when you tell Windows to add one or more printers. In both cases, the Control Panel is responsible for installing printer drivers and support files.

The user interface for installing printer files is the Printers dialog box, which you access by double-clicking on the Printers icon in the Control

Table 13.2 Windows Printer Files and File Types

Printer File(s)	*Description and Use*
*.DRV	Printer driver for one or more compatible printers.
*.WPD	Additional PostScript description file, usually for a particular brand and/or model of PostScript printer that has enhanced capabilities, in addition to those supported in the PSCRIPT.DRV file.
FINSTALL.DLL	Dynamic link library supporting functions for the Font Installer used to Install HP-supplied fonts on LaserJet Series II, Series III, and DeskJet printers.
FINSTALL.HLP	Help file for HP Font Installer.
GENDRV.DLL	Dynamic link library supporting dialog box and other functions with most printer drivers that don't use the UNIDRV.DLL library.
HPPCL.DRV	Printer driver for HP LaserJet Series II printers and many other LaserJet printers.
HPPCL5A.DRV	Printer driver for HP LaserJet Series III printers.
HPPCL5A.HLP HPPCL5OP.HLP	Help files for HP LaserJet Series III printers.
PSCRIPT.DRV	Basic PostScript driver required for all PostScript printers.
PSCRIPT.HLP	Help file for PostScript printers.
TESTPS.TXT	Diagnostic file that tests PostScript virtual memory settings and COM port settings.
UNIDRV.DLL	Dynamic link library used to support options for most universal printer drivers (including most dot-matrix and HP printers).
UNIDRV.HLP	Help file used by most printer drivers that use the UNIDRV.DLL file.

Panel. When you click on the Add button, Windows retrieves a list of printer names from the [io.device] section of CONTROL.INF (stored in the WINDOWS\SYSTEM directory). You can use Write to view CONTROL.INF. Each line in this section of CONTROL.INF includes the name of the driver file (.DRV) for the printer, along with any additional files that the printer requires.

Figure 13.1 shows the partial List of Printers that appears in the Printers dialog box, as well as the corresponding CONTROL.INF lines used to create this list. Take a look at the CONTROL.INF line for the Agfa Compugraphic Genics device:

```
6:HPPCL.DRV,6:unidrv.dll,"Agfa Compugraphic Genics","DEVICESPECIFIC"
```

Each entry in this line is separated by a comma. The first entry, 6:HPPCL.DRV, tells the Control Panel the name of the driver file

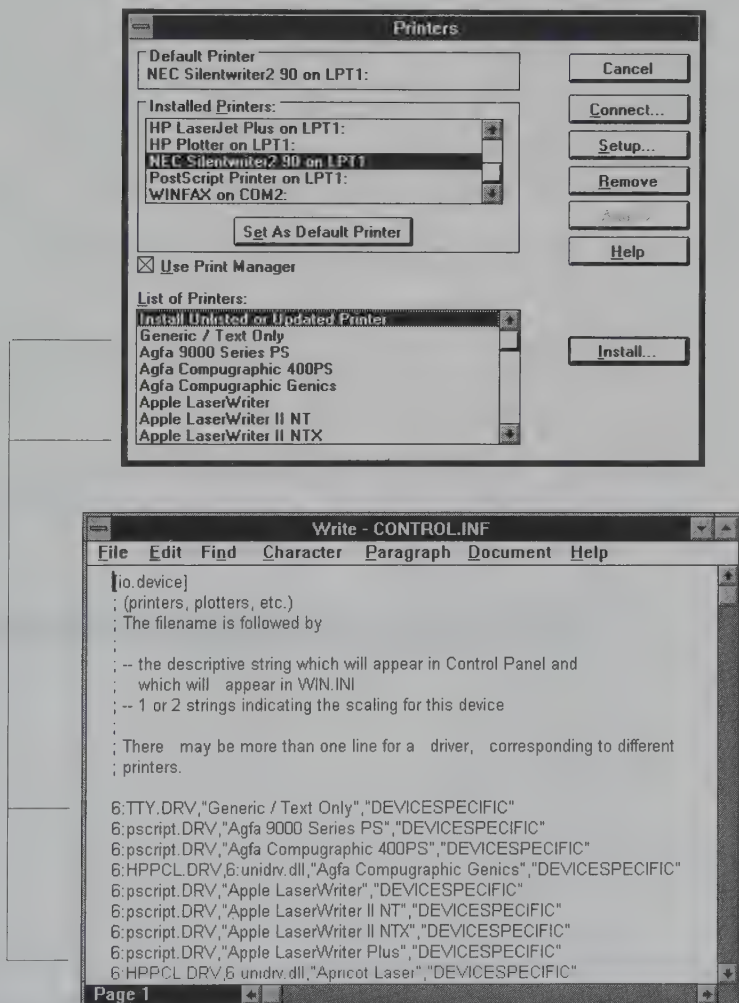


Figure 13.1 The printer names in the Control Panel are extracted from the [io.device] section of CONTROL.INF.

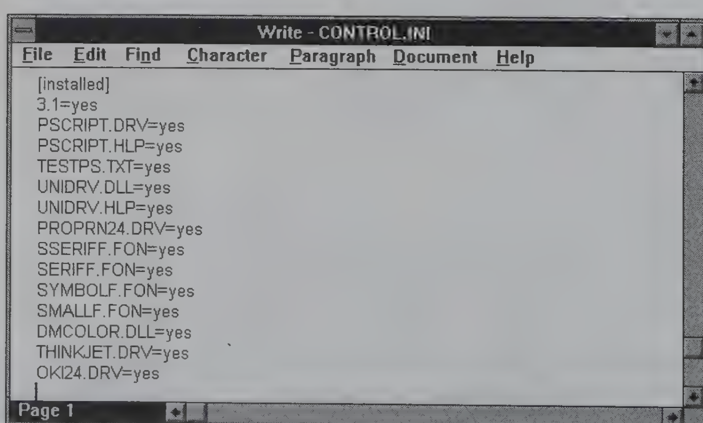
(HPPCL.DRV) plus the installation diskette where this file can be found (6:). The second entry, 6:unidrv.dll, provides the name and diskette location of a second file, UNIDRV.DLL, required to support printing for the Agfa Compugraphic Genics device. The DEVICESPECIFIC string is used to determine the scaling values that the printer allows for scalable fonts.

Before the Control Panel installs these two files, it looks in the [installed] section of the CONTROL.INI file (stored in the WINDOWS directory) to see if these files have already been installed for the current version of Windows. Figure 13.2 is a sample of this section of a CONTROL.INI file.

Whenever Windows installs files for a printer—including universal driver files, any printer-support files, and any font files required by the printer—the file names are entered in the [installed] section of CONTROL.INI. All files listed under the entry 3.1=yes have already been installed for Windows 3.1, provided the file names also include a "yes" entry. For instance, the lines 3.1=yes and PSCRIPT.DRV=yes tell Windows that the Windows 3.1 version of the PSCRIPT.DRV file has already been installed on the hard disk.

The [installed] section is used to save time in installing printers. Since many printers share the same driver and support files, Windows won't try to copy over previously installed files each time you use the Control Panel to add a new printer.

After you have installed the files listed in CONTROL.INF (or they have been verified in CONTROL.INI as having been installed earlier), Windows checks the [io.dependent] section of CONTROL.INF to see if the printer's



Write - CONTROL.INI
File Edit Find Character Paragraph Document Help

```
[installed]
3.1=yes
PSCRIPT.DRV=yes
PSCRIPT.HLP=yes
TESTPS.TXT=yes
UNIDRV.DLL=yes
UNIDRV.HLP=yes
PROPRN24.DRV=yes
SSERIFF.FON=yes
SERIFF.FON=yes
SYMBOLF.FON=yes
SMALLF.FON=yes
DMCOLOR.DLL=yes
THINKJET.DRV=yes
OKI24.DRV=yes
```

Page 1

Figure 13.2 The Control Panel identifies previously installed printer drivers and support files by looking in the [installed] section of CONTROL.INI.

.DRV file requires any additional support files. (Remember: the [io.device] section only identifies required files for a *specific printer.* The section [io.dependent] lists additional files required by specific *printer drivers.*

If the [io.dependent] section identifies additional required files for a printer driver, Windows again checks the [installed] section of CONTROL.INI to see if the files have already been installed. If not, the files are copied from the installation diskette(s) to the WINDOWS\SYSTEM directory of the hard disk. Figure 13.3 illustrates this process for the HP LaserJet Series II printer. Table 13.3 lists all printers supported by the Windows 3.1 April 6, 1992 installation disks, along with the printer driver and related support files required for each printer. You can use this table in conjunction with the [installed] section of CONTROL.INI to verify that all required files have been installed for your printer. If one or more of these files is missing from the WINDOWS\SYSTEM directory, or if CONTROL.INI indicates that your printer files are not current for Windows 3.1, you might need to install or reinstall the appropriate Windows 3.1 files for your printer.

The Windows Driver Library Contains Many Updated and New Printer Files

Microsoft maintains a library of new and updated Windows 3.1 drivers in its Windows Driver Library (WDL). This library includes new or updated printer drivers and support files for more than 100 printers. If Windows 3.1 doesn't list your printer, or if you suspect problems with the supplied driver for your printer, check the WDL for a new or updated driver.

The Windows Driver Library is available on CompuServe, GEnie, and Microsoft OnLine. If you are a member of any of these services, you can download drivers and other printer files directly via modem. If you aren't a member of any of these services, but you do have a modem, you can still access the WDL by dialing the Microsoft Download Service (MSDL), at (206) 936-6735. Downloading WDL files is free on all of these services, although you may have to pay connect-time and other standard charges. CompuServe, GEnie, Microsoft OnLine, and MSDL also maintain a file called WDL.TXT, which lists the devices (including printers) that are represented in the WDL, and provides instructions for downloading driver files. The WDL.TXT file also provides the dates on which the new driver files were last updated in the WDL. Make sure you read this file before you attempt to download files.

If you do not have a modem, you can order the complete WDL on disk, for $20, by calling Microsoft Customer Service at (800) 426-9400 (6:00 A.M. to 5:30 P.M. PST).

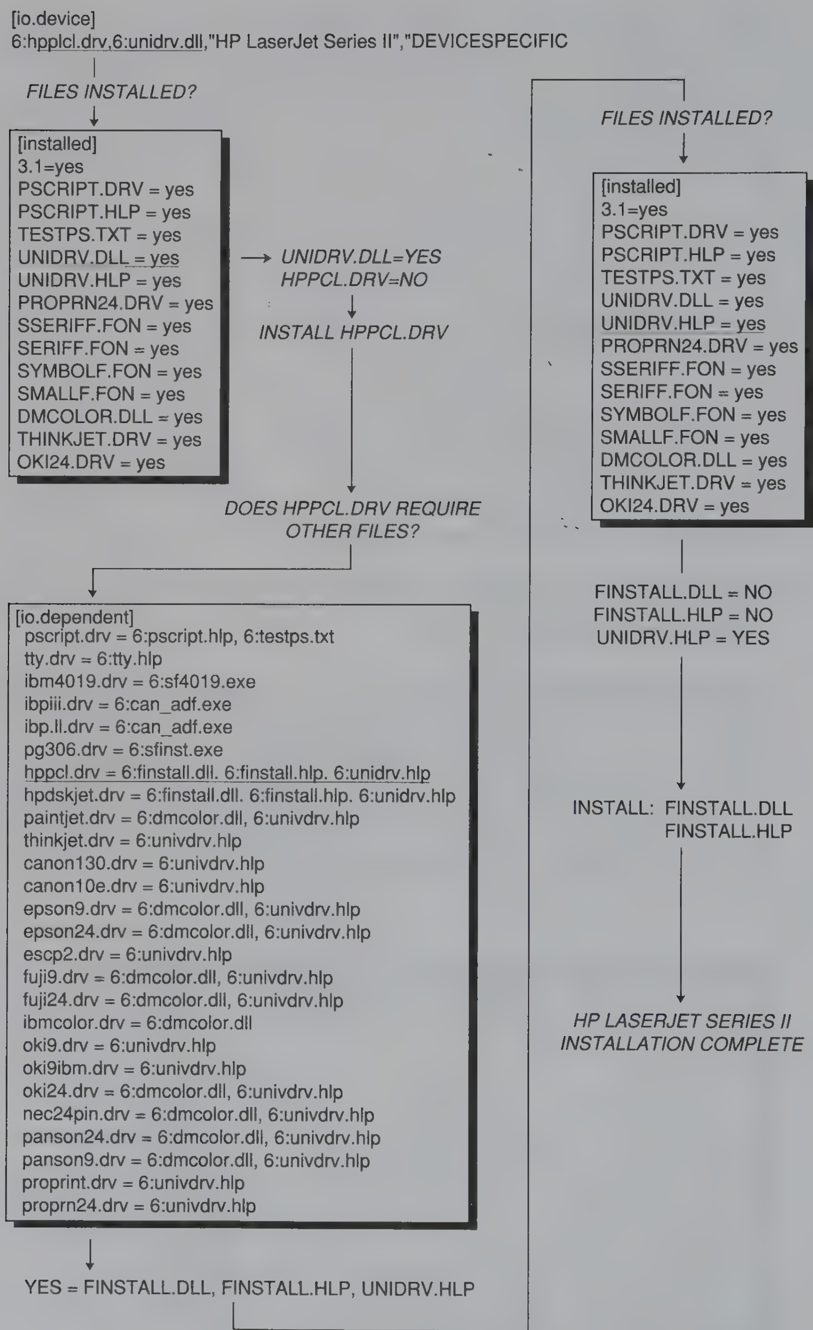


Figure 13.3 This diagram shows the process that the Control Panel uses to install drivers and support files for two popular printers.

Table 13.3 lists all the printers represented in the WDL as of August 7, 1992. Note that for each printer, we've provided the name of an .EXE file. This file includes all driver and support files for the identified printer(s). When you copy this file to disk and then run the .EXE file, all necessary printer files will be extracted and copied to the appropriate directories. You will still need to install the files through the Control Panel.

Note: Do keep in mind that the WDL is updated regularly, as new drivers are made available and as solutions are identified for recognized Windows printing problems. Also, Microsoft adds printer files to the WDL when improvements are made to support certain printers. In other words, the WDL isn't just a resource for solving printer problems; it's also a resource for finding improvements to otherwise problem-free drivers. You should review the WDL from time to time to see if updated files have been included for your printer(s).

Table 13.3 Printers in the Microsoft Windows Driver Library as of August 7, 1992

For This Printer	*Download or Copy This File*
Brother HJ-100	BROHL.EXE
Brother HJ-770	BROHL.EXE
Brother HL-4	BROHL.EXE
Brother HL-8	BROHL.EXE
Brother HL-8D	BROHL.EXE
Brother HL-8e	BROHL.EXE
Brother M-1309	BRO9.EXE
Brother M-1324	BRO24.EXE
Brother M-1809	BRO9.EXE
Brother M-1824L	BRO24.EXE
Brother M-1909	BRO9.EXE
Brother M-1924L	BRO24.EXE
Bull Compuprint 4/12	BULL.EXE
Bull Compuprint 4/14	BULL.EXE
Bull Compuprint 4/22	BULL.EXE
Bull Compuprint 4/23	BULL.EXE
Bull Compuprint 4/40	BULL.EXE
Bull Compuprint 4/41	BULL.EXE
Bull Compuprint 4/51	BULL.EXE
Bull Compuprint 4/52	BULL.EXE
Bull Compuprint 4/54	BULL.EXE
Bull Compuprint 4/64	BULL.EXE
Bull Compuprint 4/66	BULL.EXE
Bull Compuprint 4/68	BULL.EXE
Bull Compuprint 922N	BULL.EXE
Bull Compuprint 923	BULL.EXE

Table 13.3 Printers in the Microsoft Windows Driver Library as of August 7, 1992 (Continued)

For This Printer	*Download or Copy This File*
Bull Compuprint 92	BULL.EXE
Bull Compuprint 924N	BULL.EXE
Bull Compuprint 970	BULL.EXE
Bull Compuprint 1070	BULL.EXE
Bull Compuprint PageMaster 411	BULL.EXE
Bull Compuprint PageMaster 721	BULL.EXE
Bull Compuprint PageMaster 821	BULL.EXE
Bull Compuprint PageMaster 1021	BULL.EXE
Canon Bubble-Jet BJ-10ex (Proprinter)	CANON.EXE
Canon Bubble-Jet BJ-10ex (LQ)	CANON.EXE
Canon Bubble-Jet BJ-130	CANON.EXE
Canon Bubble-Jet BJ-20	CANON.EXE
Canon Bubble-Jet BJ-20 (LQ)	CANON.EXE
Canon Bubble-Jet BJC-800	CANON.EXE
Citizen PROjet	OLIVE.EXE
Digital DEClaser 1100	DEC1.EXE
Digital DEClaser 2100/Plus	DEC1.EXE
Digital DEClaser 2200/Plus	DEC1.EXE
Digital DEClaser 3200	DEC1.EXE
Digital DECmultiJET 1000	DEC1.EXE
Digital DECmultiJET 2000	DEC1.EXE
Digital LA70	DEC1.EXE
Digital LA75	DEC1.EXE
Digital LA75 Plus	DEC1.EXE
Digital LA324	DEC1.EXE
Digital LJ250/252	DEC1.EXE
Digital LN03/PLUS	DEC1.EXE
Epson ActionLaser II	EPLZR.EXE
Epson EPL-4000	EPLZR.EXE
Epson EPL-7000	EPLZR.EXE
Epson LQ-570 ESC/P 2	ESCP2.EXE
Epson LQ-570 Scalable Font	EPLQ.EXE
Epson LQ-870 ESC/P 2	ESCP2.EXE
Epson LQ-870 Scalable Font	EPLQ.EXE
Epson LQ-1070 ESC/P 2	ESCP2.EXE
Epson LQ-1070 Scalable Font	EPLQ.EXE
Epson LQ-1170 ESC/P 2	ESCP2.EXE
Epson LQ-1170 Scalable Font	EPLQ.EXE
Fujitsu DL 900	FUJI.EXE
Fujitsu DL 1100	FUJI.EXE
Fujitsu DL 1100 Color	FUJI.EXE
Fujitsu DL 1200	FUJI.EXE
Fujitsu DL 3600	FUJI.EXE
Fujitsu DL 4400	FUJI.EXE

Table 13.3 Printers in the Microsoft Windows Driver Library as of August 7, 1992 (Continued)

For This Printer	*Download or Copy This File*
Fujitsu DL 4600	FUJI.EXE
HP QuietJet	HPQJET.EXE
HP QuietJet Plus	HPQJET.EXE
IBM Laser Printer 4029	IB4029.EXE
IBM Personal Printer II 2380	IB2390.EXE
IBM Personal Printer II 2381	IB2390.EXE
IBM Personal Printer II 2390	IB2390.EXE
IBM Personal Printer II 2391	IB2390.EXE
Mannesmann Tally MT 130/24	MANNT.EXE
Mannesmann Tally MT 130/9	MANNT.EXE
Mannesmann Tally MT 131/24	MANNT.EXE
Mannesmann Tally MT 131/9	MANNT.EXE
Mannesmann Tally MT 230/18	MANNT.EXE
Mannesmann Tally MT 230/24	MANNT.EXE
Mannesmann Tally MT 230/9	MANNT.EXE
Mannesmann Tally MT 290	MANNT.EXE
Mannesmann Tally MT 330	MANNT.EXE
Mannesmann Tally MT 340	MANNT.EXE
Mannesmann Tally MT 730/735	MANNT.EXE
Mannesmann Tally MT 81	MANNT.EXE
Mannesmann Tally MT 82	MANNT.EXE
Mannesmann Tally MT 90	MANNT.EXE
Mannesmann Tally MT 91	MANNT.EXE
Mannesmann Tally MT 92	MANNT.EXE
Mannesmann Tally MT 92C	MANNT.EXE
Mannesmann Tally MT 93	MANNT.EXE
Mannesmann Tally MT 94	MANNT.EXE
Mannesmann Tally MT 98/99	MANNT.EXE
NEC Pinwriter P3200	NECPIN.EXE
NEC Pinwriter P3300	NECPIN.EXE
NEC Pinwriter P6200	NECPIN.EXE
NEC Pinwriter P6300	NECPIN.EXE
NEC Pinwriter P9300	NECPIN.EXE
Okidata OL-400	OKILED.EXE
Okidata OL-800	OKILED.EXE
Olivetti DM 124 C	OLIVE.EXE
Olivetti JP 150	OLIVE.EXE
Panasonic KX-P1081	PANKX.EXE
Panasonic KX-P1124i	PANKX.EXE
Panasonic KX-P2624	PANKX.EXE
Panasonic KX-P4450	PANKX.EXE
Panasonic KX-P4450i	PANKX.EXE
Royal CJP 450	OLIVE.EXE
Seiko CH 5504	SEIKO.EXE

Table 13.3 Printers in the Microsoft Windows Driver Library as of August 7, 1992 (Continued)

For This Printer	*Download or Copy This File*
Seiko CH 5514	SEIKO.EXE
Seiko SII Personal ColorPoint	SEIKO.EXE
Sharp JX-9300	SHARP.EXE
Sharp JX-9500	SHARP.EXE
Sharp JX-9500E	SHARP.EXE
Sharp JX-9500H	SHARP.EXE
Sharp JX-9700	SHARP.EXE
Star Laserprinter 4	STAR.EXE
Star Laserprinter 8	STAR.EXE
Star Laserprinter 8 DB	STAR.EXE
Star Laserprinter 8 DX	STAR.EXE
Star Laserprinter 8 II	STAR.EXE
Star NB24-10	STAR.EXE
Star NB24-15	STAR.EXE
Star NL-10	STAR.EXE
Star NX-1000	STAR.EXE
Star NX-1000 Rainbow	STAR.EXE
Star NX-1001	STAR.EXE
Star NX-1020 Rainbow	STAR.EXE
Star NX-1500	STAR.EXE
Star NX-2400	STAR.EXE
Star NX-2410	STAR.EXE
Star NX-2415	STAR.EXE
Star NX-2420	STAR.EXE
Star NX-2420 Rainbow	STAR.EXE
Star NX-2430	STAR.EXE
Star SJ-48	STAR.EXE
Star XB-2410	STAR.EXE
Star XB-2415	STAR.EXE
Star XB-2420	STAR.EXE
Star XB-2425	STAR.EXE
Star XR-1000	STAR.EXE
Star XR-1020	STAR.EXE
Star XR-1500	STAR.EXE
Star XR-1520	STAR.EXE
Ti Omnilaser 2108	TIOMNI.EXE
Toshiba ExpressWriter 420	EXPRSS.EXE
Toshiba ExpressWriter 440	EXPRSS.EXE
Toshiba P351SX	P351SX.EXE
Unisys AP-1324	UNI24.EXE
Unisys AP-1337	UNI24.EXE
Unisys AP-1339	UNI24.EXE
Unisys AP-1371	UN1371.EXE
Unisys AP-9205/AP-9210	UNILZ.EXE

HOT TIP

The WDL Contains Two Updated Files That Are Common to Many Printers

Notice in Table 13.3 that the WDL contains an update of the generic PostScript driver (PSCRIPT.DRV) that is stored in the PSCRIP.EXE WDL file, and the universal driver library (UNIDRV.DLL) that is stored in the UNIDRV.EXE WDL file. Because PSCRIPT.DRV is used by *all* PostScript printers, you might want to copy this updated driver to your hard disk to support your PostScript printer. The UNIDRV.DLL file is used by most dot-matrix printers and by many HP printers (including all HP LaserJet Series II printers).

The updated PSCRIPT.DRV file corrects problems in defining paper size on some PostScript printers, and corrects TrueType printing errors that occur on PostScript printers using the Pacific Data Products PostScript cartridge.

The updated UNIDRV.DLL file corrects HPPCL.DRV (HP LaserJet Series II) driver problems, where very small bitmap images sent to the printer sometimes generate general protection (GP) faults. The original UNIDRV.DLL also causes some extended characters to print incorrectly when TrueType and printer fonts are mixed on the same page; the updated UNIDRV.DLL library corrects this problem.

Some Printers Are Shipped with Windows Driver Files

Many printers are boxed and shipped with a drivers disk, which usually includes driver files for Windows. Many of these disks include their own INSTALL or SETUP programs. However, you usually can use the Install Unlisted or Updated Printer option in the Control Panel to install a vendor-supplied printer driver.

The Install Unlisted or Updated Printer option appears first in the Control Panel's List of Printers. If you want to install a vendor-supplied Windows driver, choose this driver option and then follow the instructions that appear on-screen. However, you should keep in mind that printers often remain in warehouses and on store shelves for months. For this reason, many vendor-supplied driver disks include drivers for Windows 3.0 only or, at best, an outdated Windows 3.1 driver. Always use the drivers supplied on the Windows 3.1 installation disks as a first choice, or use the driver files supplied in the WDL as a second choice. *Use vendor-supplied drivers as a last resort only.*

HOT TIP

Using the Generic PostScript Printer Driver

If the Windows 3.1 installation diskettes don't include a driver for your PostScript printer, you can install either the Apple LaserWriter Plus or the PostScript Printer driver. When you choose either of these printer items in the Control Panel's List of Printers, Windows installs the PSCRIPT.DRV, PSCRIPT.HLP, and TESTPS.TXT files, which drive *all* PostScript printers.

If you select one of these "generic" PostScript driver installations, you'll have access to all 35 built-in PostScript fonts and most basic PostScript printer features. However, some specific features of your printer might not be supported. For this reason, you should still look through the Windows Driver Library or call the manufacturer for your printer to find out if an additional Windows 3.1 PostScript Description (.WPD) file is available for your specific printer.

Printers that Emulate Other Printers Often Don't Need a Separate Driver

If your printer isn't listed in the Control Panel or in the Windows Driver Library, you might be able to use the driver files for another printer. This technique works quite well for relatively obscure printers that emulate the capabilities of a more popular printer.

Printer emulation is used in many dot-matrix and laser printers. Many 9-pin dot-matrix printers emulate the IBM Proprinter, the Epson FX-80, or the Epson FX-100 (14-inch carriage). Similarly, many 24-pin dot-matrix printers emulate the IBM Proprinter X24 or the Epson LQ-1500 printers. Many laser printers emulate one or more of the HP LaserJet printers.

You'll need to consult the user's manual for your printer to see which, if any, emulation modes are available. On some dot-matrix printers, you might need to reset some DIP switches in order to switch the printer to emulation mode or to provide the required character set for emulation mode. With most laser printers that support an HP LaserJet emulation mode, you can switch modes by using the printer's on-board menus.

HOT TIP

Using the Generic/Text Only Driver

The Generic/Text Only driver uses the standard ASCII character set, which will let you print from within Windows using even the most archaic dot-matrix and daisy-wheel printers. However, with this driver, you can't print graphics or extended characters, and you can't use TrueType fonts.

Change CONTROL.INI before You Reinstall a Corrupted Driver

If you suspect that one or more files used to support your printer has been corrupted on your hard disk, you can reinstall the printer driver files from the installation diskettes only if you delete the same files from the hard disk or if you modify CONTROL.INI.

We explained earlier that the [installed] section of CONTROL.INI keeps track of all printer files that have been copied to your hard disk. For this reason, you can't simply reinstall printer files by using the Add button in the Control Panel. If you try to do this, Windows will note that the files have already been installed for Windows 3.1, and won't bother to copy files again from the installation diskettes; consequently, any files that you suspect are corrupted will remain on your hard disk. And don't think you can get around this problem by using the Remove button in the Control Panel to remove driver files. The Remove button will delete installed printer names from the Control Panel, but it won't delete any files from your hard disk.

One solution is to first delete all files used by your printer, then use the Control Panel to reinstall your printer. If the files have been physically removed from your hard disk, Windows will reinstall the files from the installation diskettes *even though CONTROL.INI says that the files have already been installed.*

A better solution is to change the 3.1=yes line in the [installed] section of CONTROL.INI to read 3.1=no, reboot your computer, then use the Control Panel to reinstall the printer files. With this technique, you don't have to worry that you neglected to delete one or more printer files from your hard disk. When you change the CONTROL.INI to 3.1=no, you essentially tell Windows that all installed printer files are old. In response, Windows will copy printer files from the installation diskettes whenever you use the Control Panel to reinstall a printer.

Windows 3.0 Drivers Will Work in Windows 3.1

Windows 3.1 is downward-compatible with printer drivers supplied with Windows 3.0. So, if you are having problems with a

Windows 3.1 printer driver and if no other troubleshooting techniques have been successful, you can fix the problem temporarily by installing the Windows 3.0 driver files.

If you reinstall a Windows 3.0 driver, you might be able to use TrueType fonts on screen, but these fonts probably won't print. Of course, before you can reinstall a Windows 3.0 driver, you must have the Windows 3.0 installation disks available.

Warning: Before you attempt this technique, back up your current version of WIN.INI.

To reinstall a Windows 3.0 driver, follow these steps:

1. Insert the Windows 3.0 installation diskette that contains the printer driver file(s) you want to install.
2. Start the Control Panel and choose the Printers icon.
3. In the Installed Printers section, highlight the name of the printer that is not working correctly, then click on the Remove button.
4. Use the File Manager or MS-DOS prompt to delete the .DRV file for the removed printer. (The name of this file is listed in Table 13.5: the file itself will be stored in the WINDOWS\SYSTEM directory.)

   Note: Delete only the .DRV file for the printer; if you delete additional Windows 3.1 support files, you might also remove support for other installed printers.
5. Use the Control Panel to add a printer as you normally would (but make sure the inserted diskette is for Windows 3.0).

If the printer error persists, review WIN.INI to see if entries still exist for the Windows 3.1 driver. Delete these entries, save WIN.INI, and reboot your computer. Then, repeat the above installation procedures.

Using Hewlett-Packard Printers

By far the most popular brand of printer is Hewlett-Packard, and the most widely used line of printers is the LaserJet, although DeskJet and PaintJet printers are also popular. Windows 3.1 provides separate drivers and support files for LaserJet Series II, Series III, DeskJet, and PaintJet printers. The next several sections include tips, techniques, warnings, and other useful information for operating these printers under Windows 3.1. Chapter 16 includes additional troubleshooting information for Hewlett-Packard printers.

Use the Font Installer to Install HP Soft Fonts on LaserJet and DeskJet Printers

If you want to install TrueType fonts for use with your HP printer, you can use the Fonts utility in the Control Panel. If you want to install fonts for a font manager, you need to use the font manager's installation utility. However, if you want to install Hewlett-Packard-supplied fonts, you must use the HP Font Installer.

The HP Font Installer utility was co-developed by Microsoft, Hewlett-Packard, and Aldus Corporation for use in installing Hewlett-Packard-supplied soft fonts. You can use downloadable, HP-supplied, soft fonts with all LaserJet (except PostScript) and DeskJet printers; these soft fonts are not available on ThinkJet printers nor on some printers that emulate HP LaserJet printers, such as the Panasonic KX-P4420.

To use the HP Font Installer, follow these general steps:

1. Insert the diskette that contains the HP-supplied soft fonts.
2. Start the Control Panel, then double-click on the Printers icon.
3. In the Installed Printers box, highlight the name of the printer that will use the HP soft fonts, then click on the Setup button.
4. Click on the Fonts button to display the HP Font Installer dialog box, shown in Figure 13.4.
5. Click on the Add Fonts button, then follow the instructions on screen for installing fonts from the diskette(s).

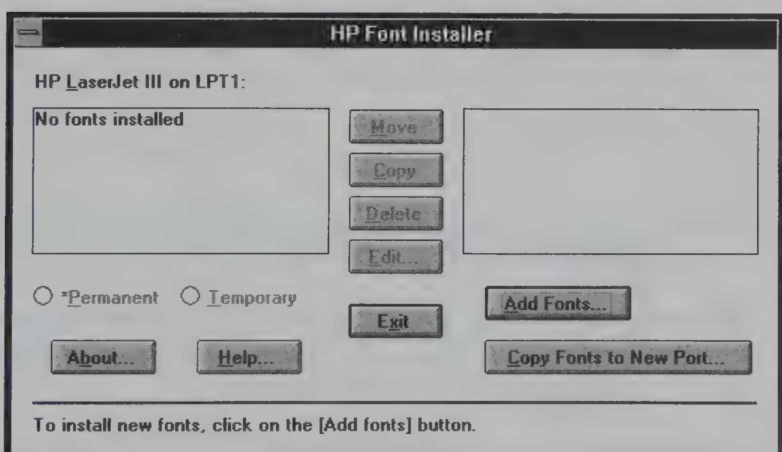


Figure 13.4 Use the HP Font Installer to add HP-supplied fonts.

Use the HP Font Installer Options

The HP Font Installer lets you download fonts on a temporary or permanent basis. You can save printing time and conserve printer memory by using these options intelligently.

Hewlett-Packard printers that support soft fonts allow you to download up to 16 fonts per page. (This includes TrueType fonts.) However, the total number of fonts you can download to the printer depends largely on the amount of memory in your printer. Although different font files have varying sizes, it's generally true that the more memory you have, the more fonts you can download.

If you routinely have several different types of print jobs each day, with different jobs using different fonts, you can conserve memory by selecting these fonts in the HP Font Installer dialog box, then clicking on the Temporary option button. When you mark fonts as Temporary, the fonts are downloaded only when they are needed by a print job, and are deleted from memory at the end of each print job.

If most or all of your print jobs use the same fonts, you can speed up print jobs by selecting the fonts in the HP Font Installer dialog box, then clicking on the Permanent option button. Permanent fonts remain in the printer throughout the day, so fonts used in multiple print jobs won't have to be downloaded at the start of each print job.

When you mark fonts as Permanent, you will have the option of downloading the fonts immediately or whenever you boot your computer. If you choose the Download Now option, the printer driver deletes any soft fonts that are currently resident in the printer's memory, then loads all fonts marked as Permanent as soon as you exit the Font Installer. If you choose the Download at Startup option, the Font Installer will modify your AUTOEXEC.BAT file so that Permanent fonts are downloaded each time you turn on your computer. You won't need to remember to download the fonts each day.

Use the Print TrueType as Graphics Option to Print Large Graphic Files

If your document consists mostly of graphic images, but also contains a small amount of TrueType text, use the Print TrueType as Graphics option to speed up printing.

The Print TrueType as Graphics option is accessed from the Options dialog box, shown in Figure 13.5. From the Printers dialog box, click on

Setup; then in the LaserJet dialog box, click on the Options button to display the Options dialog box. This option is available on non-PostScript HP LaserJet Series II and Series III printers; it is not available for the LaserJet, LaserJet +, DeskJet, PaintJet, and ThinkJet printers.

When you turn on the Print TrueType as Graphics option, each page is composed as a single large bitmap, or graphic image. If this option is turned off, the printer must still generate a bitmap image for graphic images in your document, but also must use the TrueType engine to generate separate bitmaps for TrueType characters—which takes time and uses up printer memory. However, you will actually slow down printing if your documents contain a lot of TrueType text, especially if you use several fonts. Thus, you should use this option sparingly.

Create a Font Summary File Before You Reinstall Windows

Microsoft Product Support professionals recommend reinstalling Windows as a troubleshooting measure in many situations (usually to make sure old drivers, applications, and other files get updated). Whenever you reinstall Windows, the setup program creates a new WIN.INI file. If you've already used the HP Font Installer to install soft fonts, all soft-font entries will be deleted from WIN.INI when you reinstall Windows. However, if you create a FINSTALL.DIR (font summary) file before you reinstall Windows, the Font Installer can regenerate the WIN.INI entries for you.

To create a FINSTALL.DIR file, follow these steps:

1. Start the Control Panel, click on the Printers icon, then click on the Setup button.
2. Click on the Fonts button to display the Font Installer dialog box.

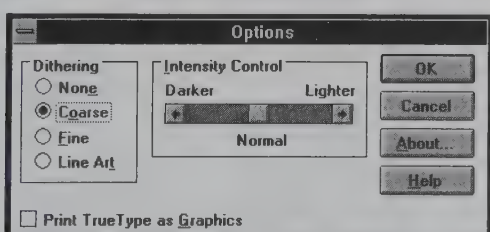


Figure 13.5 Turn on the Print TrueType as Graphics option if your documents are dominated by large graphic images and very little TrueType text.

3. Press **Ctrl+Shift**, then click on the Exit button.
4. When the Create installer directory file dialog box appears, either click on the OK button to exit the path for the FINSTALL.DIR file, or enter a new path and click on OK.

After you've reinstalled Windows, you can copy the font summary information from the FINSTALL.DIR file back into your WIN.INI file by following these steps:

1. Display the HP Font Installer dialog box.
2. Press **Ctrl+Shift**, then click on the Add Fonts button. The Add Fonts dialog box, shown in Figure 13.6, appears. Note that this dialog box is different from the Add Fonts dialog box that you typically use to install fonts.
3. In the top text box, enter the path where the FINSTALL.DIR file is stored, then press **Enter**.
4. When the Font Installer dialog box appears, highlight all of the fonts that appear in the list box on the right side, then click on the Move button.
5. Enter the path where the fonts are currently installed, then press **Enter**. The soft font entries in WIN.INI have now been restored.

Using PostScript Printers

All but a few PostScript printers come with at least 35 built-in PostScript fonts (the Apple LaserWriter Plus standard), and several PostScript printers have a superset of these 35 fonts built in. But with the advent of TrueType fonts in Windows 3.1, PostScript printers have *at least* 49

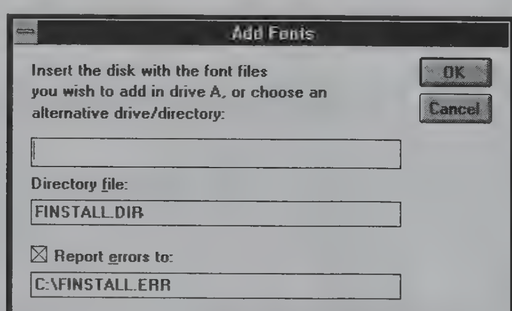


Figure 13.6 Use this Add Fonts dialog box to retrieve font summary information into WIN.INI.

scalable fonts available to them, and can conceivably have hundreds of additional scalable fonts available through the use of Adobe Type Manager or some other Adobe Type 1 and Type 3 font manager, or by simply purchasing and installing additional TrueType fonts.

Font selection with PostScript printers, even at its most basic level, can seem overwhelming. However, Windows 3.1 adds a few more dimensions to the PostScript printing process, namely in the form of PostScript printer dialog box options. Some of these options are difficult for even experienced typesetters and desktop-publishing professionals to decipher, and a few options are downright misleading.

For instance, many users aren't aware that the Arial, Courier New, Symbol, and Times New Roman TrueType fonts *never get used on a PostScript printer* unless the user is willing to wade through several levels of dialog boxes and tinker with a little-known printer component called the *substitution table.*

For the remainder of this chapter, we'll try to demystify many of the PostScript printer options that are available with Windows 3.1.

Enscapsulated PostScript Files Are Valuable, But Have Limited Use

Encapsulated PostScript (.EPS) is a file format used by PageMaker and many other graphics and desktop publishing applications. EPS files are useful if you need to convert a document to a single bitmapped image in PostScript format.

In the past, it was useful to print a file to disk in .EPS format if the file contained a mixture of graphics and non-PostScript text fonts. By converting this type of document to .EPS format (by printing it to disk), the entire file could be read by an application such as PageMaker and then edited as a bitmapped image. It wasn't necessary to convert fonts in order to get PageMaker (or some other application) to recognize an accurate image of the file.

Although .EPS printing is still valuable for this purpose, it is less necessary (and usually unnecessary) when you use TrueType fonts. Why? TrueType fonts are embedded directly within a document, and are printed in either Adobe Type 1 (scalable font) or Adobe Type 3 (bitmap) formats. Since TrueType fonts can be read on PCs running Windows 3.1 and on Macintosh computers running System 7, you often won't need to convert a page to an .EPS file in order to get a desktop publishing application like PageMaker to read in the entire page.

However, you might find it useful to print to an .EPS file if you've created a graphics image in a Windows document, and if the graphic format isn't recognizable to the application that will ultimately be required to format and print the image.

To print to an .EPS file, follow these steps:

1. Use the Control Panel to display the Printer dialog box.
2. Click on Setup to display the printer's setup dialog box.
3. Click on Options to display the basic Options dialog box, shown in Figure 13.7.
4. Click on the Encapsulated PostScript File option, then enter the path and name of the .EPS file that you want to create.
5. Click on OK to exit the Options dialog box, then exit the other dialog boxes.
6. Print the document as you normally would do from within the Windows application.

Use Scaling to Reduce or Enlarge a Page, Independent of Font Size

You've probably done this dozens of times: you print a single-page document on your laser printer, jog over to the photocopy machine, set the reduction or enlargement feature, insert the page you've just printed, and reduce or enlarge your page to the desired size. If you're using Windows 3.1 and a PostScript printer, you can specify reductions and enlargements when you print.

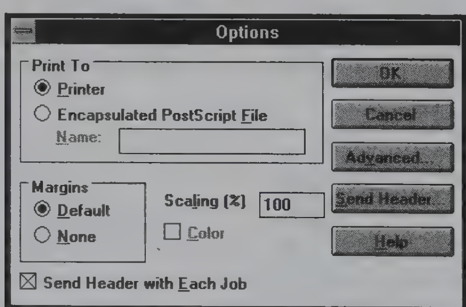


Figure 13.7 Use this Options dialog box to print to an EPS file and to scale (enlarge or reduce) a printed document.

Although most Windows graphics applications and desktop-publishing applications include a scaling feature, most of the day-to-day applications that you use—such as Word for Windows, Excel, Paradox, and so on—don't let you scale an entire page image. The best you can do is scale text by adjusting font sizes.

If you're using a PostScript printer, you can reduce each page in a document down to 10 percent of its original size, or you can enlarge each page up to 400 percent of its original size. Scaling is especially useful if you want to print text in a font size that your application doesn't directly support. Keep in mind, though, that everything, on every page, will be scaled—without exception.

To scale a document, follow these steps:

1. Use the Control Panel to display the Printer dialog box.
2. Click on Setup to display the printer's setup dialog box.
3. Click on Options to display the basic Options dialog box, shown in Figure 13.7.
4. Enter a scaling value from 10 percent to 400 percent in the Scaling text box, then click on OK to exit the dialog box.
5. Exit the other dialog boxes, then print the document.

Using the Advanced PostScript Printer Options

In this section, we'll attempt to explain, in easy-to-understand terms, the cryptic and misleading components of the Advanced Options dialog box that's available for all PostScript printers. the Advanced Options dialog box is shown in Figure 13.8. We won't discuss the Graphics options in this dialog box, though, since the Windows help screens do an excellent job explaining them.

Most of the Windows-Supplied TrueType Fonts Aren't Used by PostScript Printers

On PostScript printers, Windows substitutes printer fonts for the Arial, Courier New, Symbol, and Times New Roman TrueType fonts. (The Wingdings font is not substituted.) This approach helps speed up printing, but can create undesired printed output. However, you can easily prevent Windows from substituting fonts.

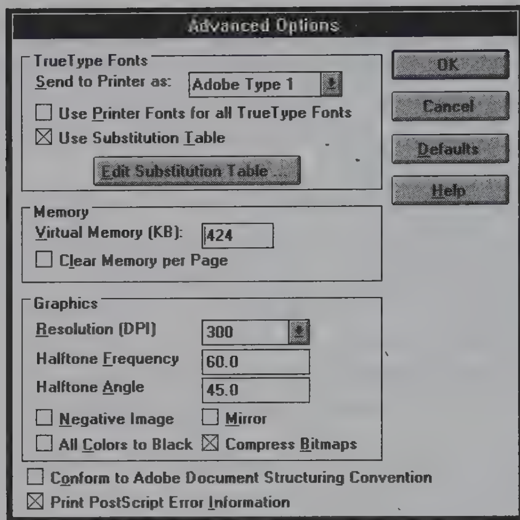


Figure 13.8 Use the Advanced Options dialog box to manipulate font usage, virtual memory, and graphics resolutions for PostScript printing.

Windows uses printer fonts in place of the Arial, Courier New, Symbol, and Times New Roman TrueType fonts because the PostScript fonts are already resident in the printer and don't have to be downloaded. Since TrueType fonts do have to be downloaded to a PostScript printer as soft fonts, printing is faster when you use printer fonts over comparable TrueType fonts.

Windows uses the substitution information in Table 13.4 to swap TrueType fonts with PostScript printer fonts.

Actually, the printer fonts that Windows substitutes are extremely similar to their corresponding TrueType fonts. And the character spacing used by TrueType and PostScript fonts is identical, so you don't have to worry about lines wrapping differently when they're printed.

But even though the substituted PostScript fonts closely match their TrueType counterparts, they aren't *identical.* Characters in TrueType fonts tend to have heavier stroke widths than comparable PostScript

Table 13.4 Font Substitution Defaults Used by PostScript Printers

TrueType Font	*Printer Font Used*
Arial	Helvetica
Courier New	Courier
Times New Roman	Times
Symbol	Symbol

fonts, and the actual design of characters is significantly different. Figure 13.9 shows the difference between selected TrueType and PostScript characters.

If you want all four of these TrueType fonts to be displayed *as well as printed*, disable the substitution table by turning off the Use Substitution Table check box in the Advanced Options dialog box.

HOT TIP

You Can't Use Notepad to Print Documents on PostScript Printers

If you print all of your documents using a PostScript printer, don't use Notepad to print documents. Notepad can only use raster fonts, which can't be printed on PostScript printers. If you need to print text files, use the Write application or a Windows word processor.

Substituting Fonts Selectively by Editing the Substitution Table

Suppose a user has been submitting documents to you using the Courier New TrueType font. By default, Windows substitutes the PostScript Courier printer font for this TrueType font. However, you want Courier New to print as the Bookman printer font. You can make this or any other font substitution by editing the substitution table.

Editing the font substitution table effectively works like a style sheet in a word-processing application. In other words, all instances of a given font in a document are replaced by a printer font during printing. You don't need to change fonts within the document itself.

You can substitute for any TrueType font that appears in a document; however, you can't use the substitution table to replace a screen font.

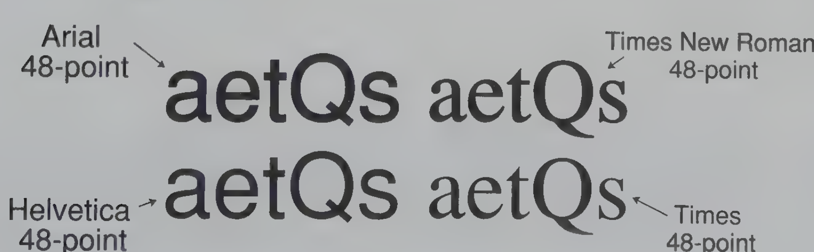


Figure 13.9 TrueType fonts have different stroke widths and slightly different stroke shapes than their corresponding PostScript-printer fonts.

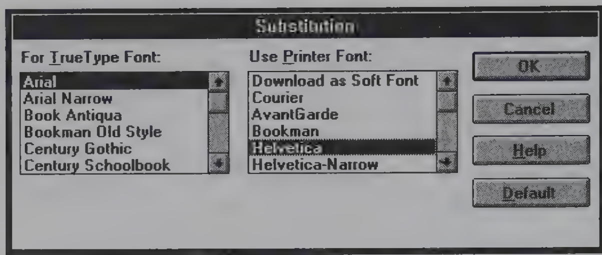


Figure 13.10 Use this table to substitute printer fonts for TrueType fonts.

To edit the substitution table, follow these steps:

1. Display the Advanced Options dialog box.
2. Make sure the Use Substitution Table option is turned on. (Substitution changes will have no effect if this check box is turned off.)
3. Click on the Edit Substitution Table button to display the Substitution dialog box shown in Figure 13.10.
4. In the For TrueType Font list box (left side), select a font to be substituted. In the Use Printer Font list box (right side), select the printer font that will replace the TrueType font during printing.
5. Repeat Steps 3 and 4 for each font that you want to substitute, then click on the OK button.

Note: Font substitutions are permanent and will affect all documents that are printed using the PSCRIPT.DRV (PostScript printer driver) file. If you want font substitutions to affect only the current Windows session, you must remember to return the substitution to its default condition. You can do this easily by clicking on the Default button in the Substitution dialog box (*not* the Defaults button in the Advanced Options dialog box).

If you want your font substitutions to be permanent, but you also want to disable the substitutions temporarily, turn off the Use Substitution Table check box. If you want to disable your substitutions temporarily *and* if you want to use the default substitutions temporarily, turn on the Use Substitution Table check box. (The Use Substitution Table will be turned off automatically in this case.)

HOT TIP

Achieve Optimum Printing Speed the "Use Printer Fonts for All TrueType Fonts" Check Box

When you turn on the Use Printer Fonts for all TrueType Fonts check box, Windows behaves just as you might expect: only printer fonts are used in printing documents. When you use this option, Windows uses its own internal font mapping table (not the TrueType/PostScript substitution table) to match TrueType fonts as closely as possible with device fonts. If you've installed additional TrueType fonts (such as those included with the TrueType Font Pack) and you want to speed up printing as much as possible, turn on the Use Printer Fonts for all TrueType Fonts. Of course, you shouldn't use this option when you want to retain control over the way Windows maps TrueType fonts to printer fonts.

A PostScript Printer's Virtual Memory Has No Relationship to Windows' Virtual Memory

Your PostScript printer's virtual memory has nothing to do with the virtual memory Windows creates to set up swap files. You can use the TESTPS.TXT file to determine the best virtual memory setting for your printer.

The Advanced Options dialog box includes a text box for adjusting the size of part of your PostScript printer's RAM. Virtual memory stores data, including fonts, involved in printing. The term "virtual memory" is used to indicate that you can adjust its size to accommodate additional, installed RAM or to facilitate the speed at which some jobs get printed.

Now that you know—at least in a general sense—what virtual memory means in the context of your PostScript printer, we can explain why you might want to adjust the virtual memory setting.

The virtual memory size that appears in the Advanced Options is the manufacturer's recommended size for the PostScript printer that you're currently using. This value varies widely for different PostScript printers. For instance, the recommended virtual memory (VM) size for the Apple LaserWriter Plus is 168K, while the recommended VM size for the Apple LaserWriter NT is 424K.

If your specific PostScript printer wasn't represented in the List of Printers when you installed Windows, you might have selected the first PostScript printer name you saw, realizing that the PSCRIPT.DRV file drives all PostScript printers. Although you would have been correct in making this assumption, the virtual memory setting for your printer probably is not optimized for your printer's actual VM capability.

When you install any PostScript printer in Windows 3.1, Windows also copies a file called TESTPS.TXT to your hard disk. Most users never do anything with this file, which is unfortunate. The TESTPS.TXT is extremely valuable, because it can assess your PostScript printer's virtual memory capabilities and can provide a recommended setting to use in the Virtual Memory text box in the Advanced Options dialog box.

To use the TESTPS.TXT file, follow these steps:

1. Make sure your printer is turned on and is on line. If you have a multi-mode printer (one that can run in PostScript mode as well as in a Hewlett-Packard LaserJet mode), make sure the PostScript mode is currently selected.
2. Start the File Manager, then display the contents of the WINDOWS\SYSTEM directory.
3. Select the TESTPS.TXT file, then click on File Copy.
4. In the To text box, type **LPT1**: (or the port that your PostScript printer is connected to), then press **Enter**.
5. When the Confirm File Replace dialog box appears, click on the Yes button. Windows prints a few lines of text similar to the ones shown here.

```
Max Printer VM (KB):       389.0
Max Suggested VM (KB):     330.0
Baud Rate:                  9600
Data Bits:                     8
Parity:                     None
Stop Bits:                     1
Flow Control:          Xon/Xoff
```

To set your virtual memory at the maximum recommended size, follow these steps:

1. Display the Advanced Options dialog box.
2. In the Virtual Memory text box, enter the Max Suggested VM value provided by the TESTPS.TXT output.
3. Click on the OK button to save the new setting.

HOT TIP

Reduce the VM Size by 20 to 30 Percent If You Frequently Need to Print Using a Large Number of Fonts

The virtual memory size is the maximum size of memory that can be used to store fonts and other data for print jobs. The more fonts and other data that you currently have stored in memory, the longer it can take to print pages, because the entire contents of memory must be searched each time a new font is used. You won't do any harm to your printer by specifying a virtual memory size larger than the actual maximum size of your printer's virtual memory, because the PostScript driver will flush the printer's virtual memory whenever it becomes full.

But you can improve the efficiency of printing, in some cases, by setting a VM value that's 20 to 30 percent lower than the maximum suggested value. When you adjust this value down, Windows will flush your printer's memory whenever the VM setting you've specified indicates that memory is full.

By flushing virtual memory frequently, you can ensure that only the most recently downloaded fonts are made available. If your documents use half a dozen or more fonts, you might be able to speed up printing by reducing the maximum VM value, because less time will be spent searching through memory for old, unneeded fonts.

The Clear Memory per Page check box was included to provide a service similar to the one achieved by reducing the maximum virtual memory amount, but this option is less efficient and really isn't all that useful. If you flush virtual memory after each page has been printed, Windows must download fonts again at the start of each new page—a time-consuming process that can defeat the benefits of adjusting the virtual memory size.

Part Overview

PART 5

Working with Applications

The two chapters in this part of the book provide some tips and strategies for using applications with Windows. Chapter 14 focuses on information you need to know to get your non-Windows applications to work under Windows. In this chapter, we'll explain some tips for using PIFs, for mutlitasking with non-Windows applications, and for using some specific features available when you run non-Windows applications in enhanced mode.

Chapter 15 zeroes in on a set of features that many Windows users seem to want to take advantage of, but often aren't sure how to implement: object linking and embedding (OLE). We'll explain OLE in down-to-earth terms, and we'll provide several examples of how to put OLE to use with specific Windows applications.

CHAPTER

14

Setting Up and Running Non-Windows Applications

Why would you want to run a non-Windows application under Windows when the application wasn't designed to be used in this way? If it's really a non-Windows application, how come it *does* run under Windows? And is it dangerous to run a non-Windows application under Windows? The answer to the first question is obvious to most experienced PC users. The answers to the second and third questions are a bit more complex, as you'll see as you read through this chapter.

There just aren't enough good Windows applications available to perform all of the tasks that users expect from their PCs. That's not surprising. Programmers have been writing applications for DOS for more than a decade. Because Windows didn't become popular until the advent of version 3.0, most software developers didn't start writing applications for Windows until 1989 or later.

In the next few years, you can expect to see the number of Windows applications increase dramatically—a trend that's already well under way. For now, though, you probably have to rely on at least a few older applications that were designed to run exclusively under DOS. This chapter explains many of the issues involved in using these "non-Windows applications" under Windows. We'll explain how to set up non-Windows applications and we'll provide several tips on using non-Windows applications under Windows 3.1.

A Non-Windows Application Is a DOS Application

In general, a non-Windows application is designed to request all of its I/O services (reading from and writing to disk, printing, requesting memory, and so on) from DOS—therefore, it's also called a DOS application.

A true Windows application, on the other hand, requests all of its I/O services from Windows. (To support a Windows application's requests, Windows might in turn request some I/O services from DOS, but a Windows application almost never directly requests I/O services from DOS.)

In Chapter 4, we explained the benefits of using Windows as a way to pass messages (also called events) among Windows applications and between Windows applications and DOS, so we won't describe message passing here. The fact that DOS applications don't participate in this message-passing approach (called the *Windows API*—for *application program interface*) doesn't mean these applications won't run in the Windows environment. As you probably are well aware, DOS applications *do* run in Windows. However, they might not run very well.

The ability of a DOS application to coexist peacefully with Windows depends on whether the application is *Windows aware.* In other words, the DOS application lets Windows know how well it interacts with I/O devices and with DOS.

The easiest way to make a DOS application windows-aware is to create a Program Information File (PIF), which contains information that Windows can use to set aside memory and other I/O resources solely to support that application. In most cases, you can let Windows set up a PIF for each of your DOS applications. In a few cases, you'll have to create a PIF yourself. Either way, the more you know about how Windows creates and uses PIFs, the easier it will be for you to troubleshoot and optimize your DOS applications.

The next several topics involve setting up and using PIFs. However, we're not going to provide a detailed listing and explanation of every setting and option available in a PIF. That's pretty dull stuff, and it won't really address the specific issues that you probably want to see explained. Instead, we'll focus on techniques, tips, and warnings involved in setting up and using PIFs effectively.

Windows 3.1 and PIFs

With Windows 3.0 and earlier versions, it was up to you to create a PIF for most DOS applications. For instance, in Windows 3.0, you could run Windows Setup to install a few PIFs that were defined in the SETUP.INF file. But for most DOS applications, a custom PIF wasn't available; you either had to use the _DEFAULT.PIF settings, which provided for a worst-case scenario, or you had to create a PIF from scratch—a nightmarish prospect. A few applications included a PIF with their program diskettes, but even in these cases the PIFs were sometimes designed for an older version of Windows or didn't really represent optimal PIF settings.

In Windows 3.1, you can still create a PIF yourself by choosing the PIF Editor icon from the Main group. Figure 14.1 shows the PIF Editor window that appears when you are running Windows in enhanced mode. The basic PIF Editor window for standard mode contains a more limited set of options.

Figure 14.2 shows the Advanced Options dialog box that appears when you click on the Advanced button from within the basic PIF Editor window. Again, this window contains options for running a DOS application in enhanced mode. An Advanced Options window isn't available for standard mode.

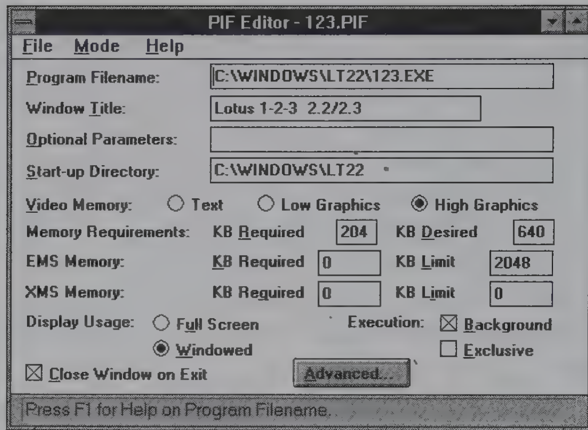


Figure 14.1 Use this PIF Editor window to create or modify PIF settings for an application running in enhanced mode. (An existing Lotus 1-2-3 PIF is shown here.)

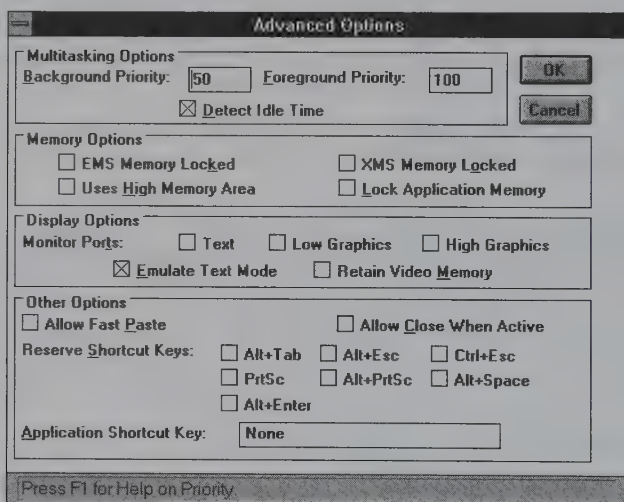


Figure 14.2 Use this PIF window to adjust advanced multitasking, memory, display, and keyboard usage for a DOS application running in enhanced mode.

Fortunately, with Windows 3.1, you'll probably use the PIF Editor mainly to modify and fine-tune existing PIFs; you'll rarely have to create PIFs yourself. The reason: Windows 3.1 can create PIFs automatically for more than 180 DOS applications. All of the information for creating these PIFs is stored in a file called APPS.INF, located in the directory WINDOWS\SYSTEM . The Windows Setup utility uses APPS.INF to create PIFs for your DOS applications when you install Windows. However, you can also use Windows Setup to create PIFs for DOS applications that you install later.

HOT TIP

Create Multiple PIFs for an Application That You Use in Different Ways

If you tend to use a particular DOS application differently in different sessions, you can create different PIFs to support settings for these different sessions. For instance, you might want to create different PIFs if you sometimes run an application solely in text mode, while at other times you run the application in graphics mode (which requires more memory for the video card). Or you might want to specify unique multitasking and priority settings in one version of a PIF but not in another.

If you want to create other PIFs for an application without deleting the original PIF, follow these steps:

1. In the Main group, double-click on the PIF Editor icon, then use File Open to open the existing PIF for the application.
2. Make the desired changes to the PIF.
3. Use File Save As to save the PIF under a new name, then exit the PIF Editor.
4. In the Program Manager, click on the icon for the original version of the application, then choose File Copy.
5. In the To Group list, select the name of the Group where you want the new program icon to appear (usually the same group as the original icon for the application), then click on OK to make a copy of the icon.

   Steps 4 and 5 allow you to keep the original icon for the new versions of the application. If you want to assign different icons to each version of the application, ignore Steps 4 and 5.
6. From the Program Manager menu, choose New. Make sure Program Item is selected in the New Program Object dialog box, then click on OK.
7. In the Description box, create a program description that will help you remember how you want to use this particular version of the application.

8. In the Command Line, enter the name of the revised PIF that you've just created and saved. (You might also want to assign a new icon to differentiate this version of the application from the original and other versions that you use.)
9. Click on OK to exit the Program Item Properties dialog box.

Use this same basic procedure to create new PIFs and icons for other versions of the same application.

Note: If you want to create different icons for an application so that you can use different data directories to store files that you create, make a copy of the icon and modify the Working Directory in the Program Item Properties dialog box. This working directory takes precedence over any directory that you specify in the Start-up Directory box in a PIF. So, it's easier and smarter to create multiple working directories using the Program Item Properties dialog box rather than the PIF Editor.

Writing Over Existing PIFs

When you installed Windows 3.1, you very likely had a few PIF files already residing in your WINDOWS directory. The Windows Setup program discourages you from writing over these PIFs, although there may be many cases where you do want to delete these older PIFs.

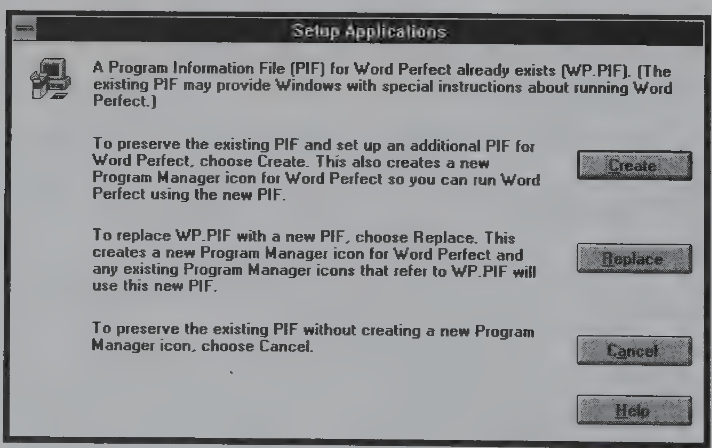


Figure 14.3 Windows Setup will display this dialog box if it detects an existing PIF for a DOS application you are trying to set up.

When the Windows setup program encountered one of these files, it told you that a PIF already exists for the DOS application currently being set up. You were then given the opportunity to use the existing PIF, create a new PIF without deleting the old one, or replace the existing PIF. How you responded makes a big difference in the number of PIF files that are stored in your Windows directory—and in the way PIF names are assigned.

Figure 14.3 shows the dialog box that appears when Windows detects an existing PIF for an application that you're trying to set up. You'll notice that the first option is to create a new PIF for the application without disturbing the existing PIF. There are actually some good reasons for Microsoft wanting you to take this conservative approach.

First, if you overwrite an existing PIF with a new PIF that has the same name, you could be deleting custom PIF settings that you've previously created. Suppose you spent hours experimenting with different settings in order to find the best possible PIF settings for running a particular DOS application under Windows 3.0.

Although it's true that, in most cases, the PIF information supplied in the APPS.INF file for Windows 3.1 is up-to-date and contains optimal settings in most cases, Microsoft doesn't want to be responsible for deleting a PIF that contains custom settings for your memory configuration, graphics card, video card, and so on. Better to install the new 3.1 PIF without deleting the existing version—in the event that you would prefer to use the older PIF or at least will need to refer to the older PIF to note any custom settings that can be added to the new PIF.

Second, a PIF in Windows 3.1 only uses 545 bytes of disk space. So, it make sense to leave older PIFs on your hard disk—at least temporarily—in case you need them later.

Windows Renames Old PIFs

Whenever you elect to preserve an older PIF for a DOS application, Windows appends a two-digit number to the name of the new PIF.

The first time Windows encounters a particular PIF that you want to preserve, the digits 00 are appended to the file name. The second time Windows encounters the original PIF name, the digits 01 are appended for the new PIF. Windows uses this numbering system when you initially install Windows 3.1 and any time thereafter when you run Windows Setup to set up applications.

For example, WordPerfect uses the PIF name WP.PIF. If this file already exists in your WINDOWS directory and you want to set up a new icon for WordPerfect without deleting the existing PIF, Windows will assign the name WP00.PIF to the new PIF. If for some reason you use Windows Setup to create another icon for WordPerfect (perhaps for a different version of WordPerfect), without replacing the existing PIF, the new PIF will be numbered WP01.PIF.

It's helpful to know about this PIF naming/numbering convention when you need to use the PIF editor to open an application's PIF. The most current PIF for the application will always have the highest number. However, Windows uses this same technique to name PIFs for *ambiguous* applications, which are DOS applications that share the same program name. Because of this, it can be difficult to figure out which application uses which PIF.

For instance, suppose you have both WordPerfect and MultiMate installed on your hard disk. Both programs are run by using the file WP.EXE. Windows always assigns PIF names by using the same name of the program file along with the extension PIF. So, Windows will assign the base name WP.PIF to any DOS application that executes using a WP.EXE program file. The sequential-numbering convention is used to distinguish PIFs that share the same base name.

Again, if Windows first encounters WordPerfect on your hard disk when you install Windows 3.1, and you elect to preserve the existing PIF for WordPerfect, the new WordPerfect PIF will be assigned the name WP00.PIF. When Windows next encounters the WP.EXE file for MultiMate, it will assign the name WP01.PIF to MultiMate, since WP.PIF and WP00.PIF have already been used. (Windows 3.0 used a different PIF-naming convention, so in this example MultiMate would not already have a conflicting PIF name, such as WP00.PIF or WP01.PIF.)

Windows will compound this numbering confusion if your hard disk contains different versions of the same application. For instance, all versions of WordPerfect use the same PIF settings. So, if you keep a copy of WordPerfect 4.2 and WordPerfect 5.1 on your hard disk, both will be assigned the base name WP.PIF when you use Windows Setup to create icons for them, but one or both PIFs will have numbers in the PIF name.

HOT TIP

Rename PIFs to Avoid Confusion

If you have two or more PIFs that share the same base PIF name (but are numbered differently), you should assign more descriptive PIF names to the numbered PIFs. This tactic can make it easier to identify which PIFs have been assigned to which applications. It's an especially important technique if you have several applications that share the same base PIF name. When you rename a PIF, make sure you also change the command line in the application's Program Item Properties dialog box to reflect the new PIF name.

The following example shows how to assign a more descriptive PIF name to MultiMate. You can use this same basic procedure to rename any PIF:

1. Click on the existing MultiMate icon to select it.
2. Click on File Properties to display the Program Item Properties dialog box for MultiMate.
3. Note the current PIF name that appears in the Command Line, then exit the dialog box without making any changes.
4. Use the File Manager to search the WINDOWS directory for the PIF name you've just identified.
5. In the File Manager, choose File Rename and then enter a new, more descriptive PIF name (such as **MMATE.PIF**). Make sure you retain the extension .PIF.
6. Exit the File Manager, then return to the Program Manager.
7. Display the Program Item Properties for MultiMate, then change the file name in the Command Line so that it matches the new name you've just created in the File Manager.

   Note: Since icons are associated with PIF names (unless the DOS application comes with its own icon), you will probably lose the current icon when you assign a new PIF name in the command line. The following steps explain how to recover the icon:

8. Choose the Change Icon button. A dialog box will appear to warn you that an icon does not exist for the application.
9. Choose the OK button to bypass the warning and to display the Change Icon dialog box.
10. In the File Name text box, delete PROGMAN.EXE and enter **MORICONS.DLL**. (Most icons for DOS applications are stored in this file.)
11. Click on the Browse button to display the icons stored in MORICONS.DLL. Browse through the list of icons until you find the one you want. (If the desired icon doesn't appear here, review the icons in PROGMAN.EXE.)
12. Click on the desired icon, then click on OK to exit the dialog box.

13. Click on OK to exit the Program Item Properties dialog box.
14. Press **Enter** or click on OK to save the new name and exit the dialog box.

Locating Specific Information about DOS Applications

When you use Windows Setup to set up your DOS applications, Windows creates a PIF for each application by extracting information from the APPS.INF file. If you want to find out which applications Windows can create a PIF for, or if you want to examine specific PIF settings for an application without actually creating a PIF, use Write to open and read through this text file.

There are many reasons why you might want to review information stored in APPS.INF. Perhaps you want to find out if Windows supports a particular DOS application before you decide whether to purchase or install the application itself. If you want to create a PIF for an unsupported DOS application, but you know that the application can use the PIF settings for another application that Windows does support, you'll need to consult APPS.INF to find the appropriate settings. Or, if you want to create a PIF on another computer that's running Windows 3.0, you can use APPS.INF to note the appropriate settings. Table 14.1 lists all of the applications for which PIF data is provided in the Windows 3.1 APPS.INF file.

Table 14.1 DOS Applications for Which APPS.INF Provides Information

Access for MS-DOS	ACCPAC BPI	ACCPAC Plus
APPLAUSE II 1.5	Ashton Tate dBASE IV	AutoCAD
AutoCAD (Batch File)	Autosketch 3.0	
Borland C++	Brief 3.1	
Close-Up 4.0	Crosstalk-XVI 3.71	
DataEase	DataPerfect	DECnet Job Spawner
DisplayWrite 3	DisplayWrite 4	DisplayWrite 5
DisplayWrite Assistant	DM - Directory Maintenance	DrawPerfect
DWDOS286		DWDOS386
DWINF02	DWINF03	
Extra! for MS-DOS		
Flight Simulator 3.0	Flight Simulator 4.0	Formtool
Foxbase+	FoxPro (max. config.)	FoxPro (min. config.)

Table 14.1 DOS Applications for Which APPS.INF Provides Information (Continued)

FTP Utility FTPSRV Utility	Framework III	Freelance Plus 4.0
Generic CADD	Graphwriter	GW BASIC
Harvard GeoGraphics Harvard Project Manager	Harvard Graphics 2.3 Harvard Total Project Manager	Harvard Graphics 3.0 HotWire
IBM Filing Assistant IBM Professional Editor Interleaf 5 for MS-DOS	IBM Graphics Assistant IBM Writing Assistant 2.0	IBM Personal Editor Insight
Kid Pix	KnowledgePro (MS-DOS)	
Lap-Link Pro Learning MS-DOS Quick Reference Lotus 1-2-3 form Lotus Works 1.0	Learning Microsoft Works LetterPerfect Lotus 1-2-3 3.0 Lotus Access System LPQ Utility	Learning MS-DOS 3.0 Lotus 1-2-3 2.2/2.3 Lotus 1-2-3 3.1 Lotus Agenda LPR Utility
Magellan 2.0 Microrim R:Base 3.0 Microsoft Advanced Basic Microsoft C Compiler 5.1/6 Microsoft Macro Assembler Microsoft Online 1.0 Microsoft QBasic Microsoft QuickBasic Microsoft Word 4.0 MultiMate 4.0	Managing Your Money Microrim R:Base 5000 Microsoft Basic Microsoft Chart Microsoft Game Shop Microsoft Mail Microsoft Make Utility Microsoft Pascal Compiler Microsoft Quick C Microsoft QuickBasic Ext. Microsoft Word 5.x	Manifest Microrim R:Base Clout Microsoft Bookshelf Microsoft FORTRAN Compiler 5.1 Microsoft Mail-Admin. Microsift Multiplan Microsoft Project Microsoft Quick Pascal Microsoft Spell Microsoft Works 2.0
Network Control Program	Norton Utilities 4.5	Norton Utilities 5/6.0 Now!
OPTune		
Paradox PC Paintbrush IV Plus PC Tools-FileFind PC Tools PCShell 5.5 PFS:First Choice 3.1 PFS:Plan Procomm Professional File	Paradox 3.0 PC3270 PC Tools-View PFS:Access PFS: First Graphics PFS:Professional Network Procomm Plus 1.1B Professional Write	Paradox 3.5 PCMAIL Utility PC Tools Desktop 5.51 PFS:First Choice 3.0 PFS:First Publisher PING Utility Prodigy Programmer's WorkBench

Table 14.1 DOS Applications for Which APPS.INF Provides Information (Continued)

Q & A Report Writer	Q-DOS 3	QModem
Quick Verse 2.0	Quicken	
Ready!	Reflection 1	Reflection 2
Reflection 4	Reflection 7	Reflection 8
Reflex 2.0	Relay Gold	Remote 2 call
RightWriter	RLOGINVT Utility	RSH Utility
SAS 604	Scheduler	SEDT Editor
Sethost Terminal Emulator	SideKick 1.0	SideKick 2.0
	SideKick Plus	Smartcom II
Soft Kicker	SPSS/PC+	SuperCalc 4.0
SuperCalc 5.0	Symphony 2.2	
TeleMate	TN Utility	Turbo Pascal 6.0
Turbo Tax		
Ventura Publisher	VMAIL Utility	Volkswriter 3.0
WordPerfect	WordPerfect Office	WordStar 2000
WordStar Professional	WPOffice Calculator	WPOffice Editor
WPOffice File Manager	WPOffice NoteBook	Writer Rabbit
XTree Gold	XyWrite	

HOT TIP

Windows 3.0 and 3.1 PIFs Are 100-Percent Compatible with Each Other

Although Microsoft has slightly redesigned the PIF Editor and Advanced Options dialog boxes in Windows 3.1, the settings are essentially the same as those used in Windows 3.0. In fact, a PIF that you created in Windows 3.0 will also work in Windows 3.1, and any PIF created in Windows 3.1 can be used on a system that's running Windows 3.0. Consequently, any PIF settings that appear in the Windows 3.1 APPS.INF file can be used to create PIFs in Windows 3.0.

APPS.INF is stored in the WINDOWS\SYSTEM directory. You'll need to use Write or the MS-DOS Editor to open the file, since it's too large for Notepad.

Warning: Always close the APPS.INF file without saving any changes, unless you are specifically using APPS.INF to add new PIF entries.

The [pif] section contains a list of all executable program file names (but *not* all application names) that Windows Setup can use to create custom PIFs. If two or more applications share the same executable file name (remember, WordPerfect and MultiMate both are loaded using the WP.EXE program file), Windows refers to a list of ambiguous applications to find the correct PIF settings for these alternatives. Each set of ambiguous-name listings begins with the section name [amb_*x*], where *x* is the name of the conflicting executable file name. Figure 14.4 shows the first few entries that appear in the [pif] section of APPS.INF. Note that this section also includes several comment lines explaining the parameters that are used in each line in the [pif] section. We'll explain these parameters a bit later.

```
[pif]
;
; It is VERY important that this list remain and be maintained in
; lexicographical order (by exe name, no extension)
;
; Description strings that contain more than one word (i.e. contain blank
; space) must be enclosed in quotes (").
;
; Parameter order
;
; (0) Exe file = ;  (1)  PIF name
;  (2)  Window Title
;  (3)  Startup Directory
;  (4)  Close Window on Exit flag
;  (5)  File from which to extract icon (default is Progman.exe)
;  (6)  Icon number (default is 0)
;  (7)  Standard PIF settings section (default is [std_dflt])
;  (8)  Enhanced PIF settings section (default is [enha_dflt])
;  (9)  Ambiguous EXEs section (Other applications with same EXE name)
;  (10) Optimized PIFs section
;
123.COM     = 123     ,"Lotus 1-2-3",,cwe,,3,std_gra_256,enha_123c
123.EXE     = 123     ,"Lotus 1-2-3 3.1",,cwe,moricons.dll,50, std_123,
  enha_123,amb_123
ABPI.COM    = ABPI    ,"ACCPAC BPI",,cwe,moricons.dll,30,,enha_BPI
ACAD.EXE    = ACAD    ,"Autocad",,cwe,,16,std_ACAD,enha_ACAD
ACAD386.BAT = ACAD386 ,"Autocad (Batch File)",,cwe,,16,std_ACAD386,
  enha_ACAD386
ACCESS.COM  = ACCESS  ,"PFS:Access",,cwe,,5,std_ACCESS,enha_ACCESS,amb_access
```

Figure 14.4 This listing shows the first few entries in the [pif] section of APPS.INF, along with a brief description of the parameters that can be used in each entry.

```
[amb_123]
123.EXE     = 123     ,"Lotus 1-2-3  2.2/2.3",,cwe,moricons.dll,51,
  std_123R23,enha_123R23
123.EXE     = 123     ,"Lotus 1-2-3  2.3 WYSIWYG",,cwe,moricons.dll,
  51,std_123WYSIW,enha_123WYSIW

[amb_access]
ACCESS.COM  = ACCESS  ,"Symphony (Access)",,cwe,moricons.dll, 95,
  std_SYMPHONY,enha_SYMPHONY

[amb_b]
B.EXE       = B       ,"Brief 2.1/3.0",,cwe,,2,std_B21,enha_B21

[amb_bbc]
BC.EXE      = BC      ,"Microsoft Basic Compiler",,cwe,moricons.dll,2,
  std_BC,enha_BC

[amb_cl]
CL.EXE      = CL      ,"Microsoft C Compiler 5.1",,,moricons.dll,3,
  std_CL51,enha_CL51
CL.EXE      = CL      ,"WPOffice
Calendar",,cwe,moricons.dll,70,std_CLWP,enha_CLWP

[amb_dbase]
DBASE.EXE   = DBASE   ,"Ashton Tate dBase III",,cwe,,4,std_DBASE3,
  enha_DBASE3
```

Figure 14.5 These ambiguous-entry sections of APPS.INF provide PIF information for applications that use the same executable file name as one or more other DOS applications.

Figure 14.5 shows the first few [amb_*x*] sections that appear in APPS.INF. As we'll explain later, entries in the [pif] section direct Windows Setup to the appropriate [amb_*x*] section of APPS.INF. At this point, just keep in mind that the [pif] section plus all [amb_*x*] sections encompass the names of *all* DOS applications that Windows 3.1 can supply a custom PIF for. So, you can use these sections of the APPS.INF file to determine whether Windows 3.1 can supply a custom PIF for a specific DOS application.

Deciphering the Parameters in a [pif] Entry

Each line in the [pif] section of APPS.INF can include up to 10 parameters that Windows Setup uses to create a PIF. A few of these parameters define specific PIF settings, but most of the

parameters identify additional files or sections of APPS.INF that Windows Setup must refer to in order to create a PIF.

Each [pif] entry begins with an executable file name. Windows Setup matches these file names with the executable file names that it finds on your hard disk. When Windows Setup finds a match, it uses the corresponding line in APPS.INF to create a PIF for the application. If Windows Setup can't find a match for an executable file name that's stored on your hard disk, it uses the _DEFAULT PIF for the application.

The 10 parameters that appear to the right of the executable file name (and the = sign) must be listed in the correct order, separated from each other by commas. If a particular parameter uses the default setting, the parameter won't appear, but a comma must be inserted in its place so that Windows Setup can retain the correct order of parameters.

We'll use the 123.EXE entry to illustrate:

```
123.EXE   = 123    ,"Lotus 1-2-3 3.1",,cwe,moricons.dll,50,std_123,
  enha_123,amb_123
```

Parameter 1 (123 ,) The first parameter is the base name of the PIF that Windows will assign—in this case, 123.PIF. The specific PIF name will vary, depending on whether the sequential-numbering scheme must be used (to name PIFs for other applications that share the same executable file name).

Parameter 2 ("Lotus 1-2-3 3.1",) This is the application name that will appear in the title bar when you run the application in a window, and will appear under the icon in the Program Manager.

Parameter 3 (,) The third parameter indicates the startup directory, where the application will store files that you create. The default for this parameter is the directory where the application's executable file is stored. For most applications, this parameter uses the default and allows you to specify a different startup directory if you want by using the PIF Editor. The comma in the third-parameter position, with no other text string, indicates that the default directory will be used.

Parameter 4 (cwe,) The close-window-on-exit flag (cwe) is included for all applications that display a screen or window. This flag simply tells Windows to remove the application's screen display when you exit the application. This flag isn't used for batch files or other executable files that don't display screens of their own, but simply start other programs.

Parameter 5 (moricons.dll,) Windows Setup needs to know which file to use to extract (create) an icon for the application. The default file is PROGMAN.EXE. However, icons for most DOS applications—including Lotus 1-2-3) are stored in the MORICONS.DLL file.

Parameter 6 (50,) This is the icon number for the application; in this case, 50 is the number for the Lotus 1-2-3 icon that's stored in MORICONS.DLL.

Parameter 7 (std_123) This is the section of APPS.INF that contains appropriate standard-mode PIF settings for the application. In this case, Windows Setup refers to the [std_123] section of APPS.INF to retrieve standard-mode PIF settings for Lotus 1-2-3 version 3.1, as shown below:

```
[std_123]
minconvmem  = 204
videomode  = gra
xmsmem    = 320,1024
```

Here, Windows will specify 192K in the KB Required box of the Memory Requirements section, and will turn on the Graphics/Multiple Text button to support graphics mode for Lotus 1-2-3 version 3.1. Windows will also allocate a minimum of 320K of extended memory for use by Lotus 1-2-3 since it can use DOS extender technology, and will allocate a maximum of 1024K of extended memory for Lotus's use. All other standard-mode options will use the settings stored in the [std_dflt] section of APPS.INF.

Parameter 8 (enha_123,) This is the section of APPS.INF that contains appropriate enhanced-mode PIF settings for the application. In this case, Windows Setup refers to the [enha_123] section of APPS.INF to retrieve enhanced-mode PIF settings for Lotus 1-2-3 version 3.1, as shown below:

```
[enha_123]
dispoptvideo = hgr
convmem     = 204,640
emsmem      = 0,-1
execflags   = bgd
procmemflags = dit
```

Here, Windows will monitor all video operations when the application is running in high-resolution graphics mode; Windows will give Lotus 1-2-3 204K of conventional memory when the application starts, and will let Lotus use all available conventional memory if it wants (640K); Windows won't allocate any expanded memory (EMS) to Lotus 1-2-3 by default, but will let Lotus use all available EMS (that's the meaning of the -1 setting) if it wants; Windows will allow Lotus 1-2-3 to run in the background (execflags = bgd); and will not assign Lotus any multitasking time while the application is idle ("procmemflags = dit" turns on the Detect Idle Time check box).

All other enhanced-mode options will use the default settings stored in the [enha_dflt] section of APPS.INF.

Parameter 9 (amb_123) This is the section of APPS.INF that identifies other programs that use the same executable file (123.EXE) as Lotus 1-2-3 version 3.1. Here are the contents of [amb_123]:

```
123.EXE    =123     ,"Lotus 1-2-3 2.2/2.3",,cwe,moricons.dll,51,
  std_123R23,enha_123R23
123.EXE    =123     ,"Lotus 1-2-3 2.3 WYSIWYG",,cwe,moricons.dll,51,
  std_123WYSIW,enha_123WYSIW
```

As you can see, additional entries are included for Lotus 1-2-3 versions 2.2/2.3, including a line for version 2.3 running in WYSIWYG (graphics) mode.

HOT TIP

Use Write's Find Feature to Locate Specific Entries in APPS.INF

The information in APPS.INF is organized in a loose and somewhat haphazard alphabetic order. You can spend an inordinate amount of time looking for specific entries or sections if you simply scroll through APPS.INF screen by screen. (Try looking for the section heading [enha_cs3270] in this way and you'll see what we mean.)

You can save yourself a lot of time by using the Find feature either in Write or in the MS-DOS Editor to locate an entry in APPS.INF. This might seem to be an obvious tip, but we often observe experienced users scrolling through SETUP.INF, APPS.INF, and other large text files—and frankly wasting time. Consider it a gentle reminder: use your text editor's Find or Search feature.

Running DOS Applications

The topics we'll discuss next focus on issues, problems, and tips involved with running DOS applications (non-Windows applications) under Windows. Except in a few cases, we're not going to deal with specific DOS applications. Instead, we'll focus on techniques that apply to all or most of them.

Should You Run DOS Applications in Standard Mode or Enhanced Mode?

If your system has a 386 or higher processor, you have the option of running Windows in standard or enhanced mode. When you run DOS applications under Windows, you should always operate in enhanced mode. This is almost always faster than running DOS applications in standard mode, and provides you with several options that are unavailable in standard mode. Of course, if you have a 286 system, you are limited to standard mode; however, you can still take a few steps to boost the performance of Windows while running DOS applications.

When you run DOS applications under Windows in standard mode, your system can't use the enhanced-mode-only features listed below:

- Copy and paste text between DOS applications and between a DOS application and a Windows application.
- Display a DOS application in a window.
- Adjust the display font for a DOS application.
- Multitask DOS applications.
- Swap DOS applications to virtual memory.

We'll explain all of these features next. The biggest drawback to running DOS applications in standard mode is the memory limitation that Windows enforces. If you're running Windows 3.1 in standard mode, Windows moves most of its own code out of memory and turns over most of your system's resources to the currently active DOS application.

Although you can open multiple DOS applications in standard mode, only one application at a time can reside in memory. All other DOS applications get swapped to the hard disk (called *task switching* in Windows parlance). Also, the currently active DOS application must remain in conventional memory, since Windows can't create virtual DOS machines in standard mode.

For these reasons, you won't boost the performance of DOS applications on a 286 by adding more memory—unless the application can directly access extended memory. However, increased memory used in conjunction with SMARTDrive can speed up task switching, since Windows has more available extended memory for caching applications. (See the following Hot Tip for more details. Also, keep in mind that you can significantly boost the performance of *Windows* applications by adding memory to a 286.)

Using Customized, Up-to-Date PIFs when Running Windows in Standard Mode

The most effective step you can take to ensure the best performance of DOS applications in standard mode is to use only custom PIFs. This is especially important if your DOS applications can make use of a DOS extender. Where available, you should usually use vendor-supplied PIFs rather than letting Windows Setup create the PIFs.

If you install a DOS application after Windows 3.1 has been installed, you should use Windows Setup to create a custom PIF for the application. This is especially important if your DOS application can directly use extended memory, because XMS memory requirements are specified in custom PIFs, but are not included in the _DEFAULT.PIF.

There are two situations where you should *not* let Windows Setup create a PIF from the information in APPS.INF. First, if the DOS application comes with its own PIF, you should use this PIF for the application if the PIF was designed specifically for Windows 3.0 or Windows 3.1. (Many DOS applications include PIFs for older versions of Windows/286 or Windows/386. Use the Windows 3.1-supplied PIF settings rather than these obsolete PIFs.) In most cases, the vendor-supplied PIF will contain either identical or more effective PIF settings than the PIF that Windows Setup will create.

Second, if the DOS application's installation program creates a program group or icon for the application, it's almost a sure bet that the installation program also created or used its own custom PIF (or sometimes multiple PIFs). Always use these vendor-installed PIFs, since the install routine might have modified PIF settings to take advantage of (or safeguard against) a particular hardware or memory configuration that it detected on your system.

HOT TIP

Use the Recommended SMARTDrive 4.0 Configuration for Standard Mode

If you're running Windows 3.1 on a 286 computer, your system might be using an older, slower hard disk controller than those that are sold with 386 and 486 systems. Because standard-mode Windows requires a lot of task switching to support DOS applications, this extensive disk I/O can slow down Windows' performance on many 286 systems.

You can ensure the fastest possible task switching by making sure SMARTDrive 4.0 is running on your system, and by making sure the recommended cache sizes are in use. (Chapter 4 provides information on using and configuring SMARTDrive 4.0.) With SMARTDrive operating at peak performance, Windows can cache more of your DOS applications and Windows code into extended memory, and will need to use the hard disk less often to store and retrieve code for dormant applications.

Use the Fonts Dialog Box to Adjust the Font for a Windowed DOS Application

Many users of Windows 3.0 complained that the display font for a Windowed DOS application was too small or too hard to read. In response, Microsoft shipped a special screen font file called DOSAPP.FON with Windows 3.1. You can use this font to adjust the relative height and width of text that appears in a windowed DOS application.

Microsoft might have gone a little overboard in the range of display font sizes it provides for windowed DOS applications, but you'll probably be able to put at least a few of these font sizes to good use. Keep in mind that the display fonts affect the screen display only, and bear no relationship to printed fonts.

To change the display font for a DOS application, select Fonts from the application's Control menu. Figure 14.6 shows the Font Selection

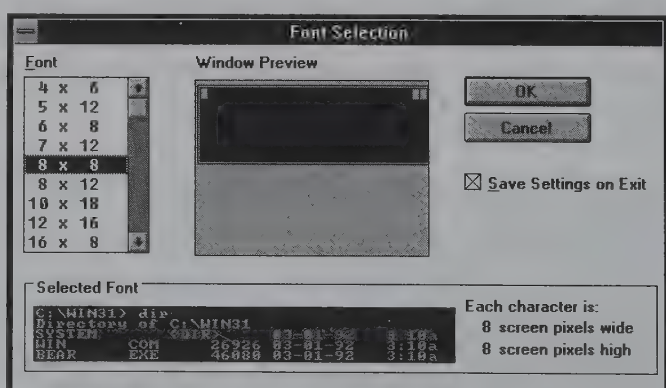


Figure 14.6 Use this dialog box to select a different screen font size for a windowed DOS application.

dialog box that appears when you do this. The "fonts" in the Font list are really just different pixel densities that Windows creates. The ratio that Windows uses for font sizes is the the number of pixels that display horizontally relative to the number of pixels that display vertically—in this case, per character.

By selecting a larger font size (such as 10 x 18 in the Font dialog box), you increase the size of text because Windows uses more screen pixels horizontally and/or vertically to create each character. By selecting a smaller size, you decrease the size of text. The ratios that you'll see in the Font list box will depend on the display capabilities for the screen driver you've installed with Windows.

Figure 14.7 shows the appearance of text in a Lotus 1-2-3 2.2/2.3 window, with the 10 x 18 pixel ratio selected (available on a VGA monitor that uses the standard Windows VGA driver). Figure 14.8 shows the appearance of text for this same window when the 7 x 12 pixel ratio is selected.

Warning: You will not be able to adjust the font size for a windowed DOS application if the application is displayed in graphics mode. You must switch the application to text mode in order to adjust the display font.

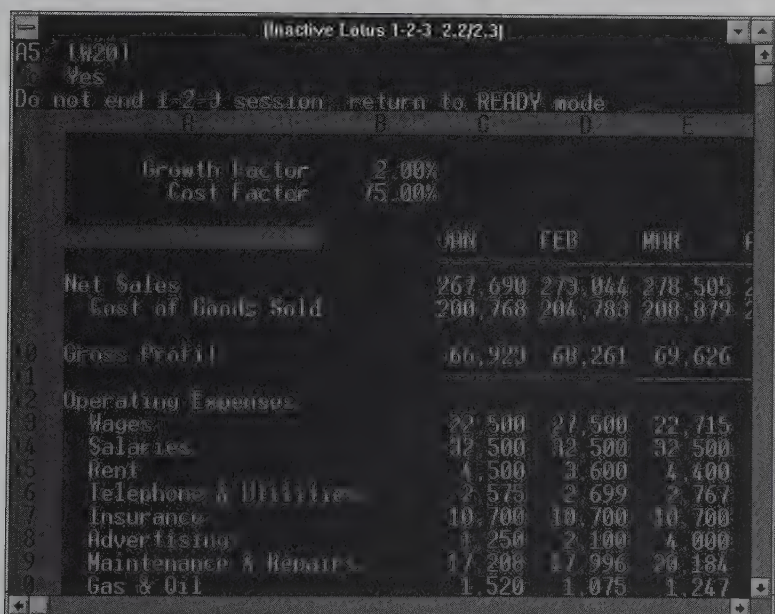


Figure 14.7 This Lotus 1-2-3 window shows a display font that's larger than the default size.

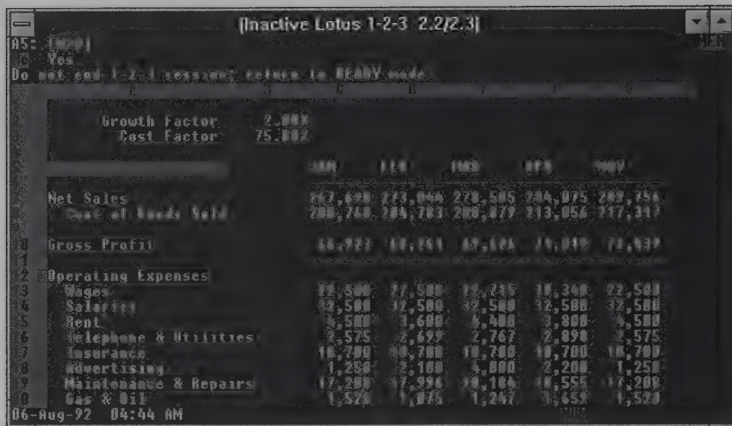


Figure 14.8 This Lotus 1-2-3 window shows a display font that's slightly smaller than the default size.

HOT TIP

The Save Settings on Exit Option Affects Both the Window Size and the Font Size

If you want to retain your font selection permanently for a particular DOS application, make sure the Save Settings on Exit check box is selected in the Font Selection dialog box. Recognize though, that Windows saves both the size of the window as well as the font size.

Windows will adjust the size of a DOS window either larger or smaller to accommodate the new font size that you select. However, you can size the Window to any width and height that you choose. Just remember that the Save Settings on Exit check box will remain turned on in later sessions, unless you turn off this option. If you want to ensure that a particular DOS window always displays at a certain size, follow these simple steps:

1. Change the font size for the window, if you want, then make sure the Save Settings on Exit check box is turned on.
2. Use the mouse to size the window to the dimensions you want.
3. Drag the window to the location you want it to appear each time you run the application in a window. (Windows stores the location of the window, as well as its size.)
4. Exit the application.
5. Start the application again and display it in a window. (You'll notice that the size and location of the window matches the settings you created before you exited the application.)

6. Display the Font Selection dialog box, turn off the Save Settings on Exit check box, then click on OK to exit the dialog box.

Now that you've turned off the Save Settings on Exit check box, you can safely move or resize the DOS window for the current session, without losing the settings you originally created.

If You Want to Display a DOS Application in a Window whenever It Opens, Modify the Application's PIF

The Control menu for a windowed DOS application includes a Settings dialog box that allows you to set multitasking and display options. All of these options affect only the current session for the DOS application.

Figure 14.9 shows the dialog box that appears for Lotus 1-2-3 versions 2.2/2.3. There might be occasions when you'll want to modify the Priority and Tasking Options for the current session only, but there's no reason why you would ever want to use the Display Options buttons; if you want to switch between a windowed and full screen display for the current session of the DOS application, it's easier to press **Alt+Tab** to toggle between the two display options.

If you want the DOS application to display in a window *every time* you start it, you must modify the application's PIF. Just follow these steps (but make sure you're running Windows in enhanced mode):

1. Start the PIF Editor from the Main group window.
2. Choose File Open to display a list of PIF files in the Open dialog box.

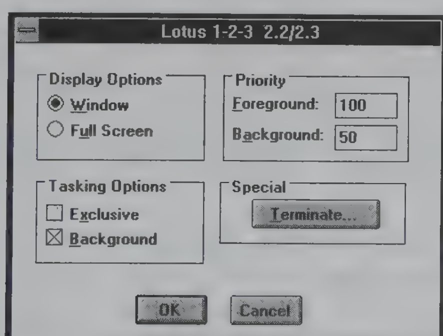


Figure 14.9 Use this dialog box if you want to change a DOS application's multitasking settings for the current session only.

3. In the File Name list box, choose the PIF for the application that you want to modify.
4. Locate the Display Usage buttons near the bottom of the PIF Editor window for the application, then turn on the Windowed option.
5. Choose File Save to save the new settings, then exit the PIF Editor window.

HOT TIP

Adjust Multitasking Options "On the Fly"

In Chapter 4, we explained how multitasking options work—including priority settings. By adjusting the priority values for each DOS application, you tell Windows to give the application relatively more or less processing time when multiple applications are running.

When you allow a DOS application to run exclusively, you can't multitask because the application gets just about all of Windows' processing time. If you allow a DOS application to run in the background, you make it possible to multitask the application with other applications. (If you don't turn on background processing, a DOS application can't do any processing while other applications are open.)

Chapter 4 explains how to use the PIF Editor to modify multitasking options. The problem with this approach is that the settings—especially the foreground and background priority values—are permanent; they are used every time you open the DOS application. But it's far more effective to adjust these settings on the fly—that is, as you need them. In that way, you can adjust priority and tasking options so that they're tailored to your current multitasking situation. Figure 14.9 shows the Tasking Options and Priority boxes that appear in the Settings dialog box for a DOS application.

Note: The tasking and priority options in the Settings dialog box override the tasking and priority options that have been set in an application's PIF. So, using the Settings dialog box is great way to adjust multitasking settings for the current session, without disrupting the permanent settings you've created in PIFs.

Always Exit a DOS Application from Within the Application

You might already have tried to close a windowed DOS application by double-clicking on the application's Control-menu button. If so,

you know that Windows will refuse to close a DOS application in this manner. But there are other Windows-based tricks you can use to exit a DOS application—all of which can be dangerous.

There are three ways to force Windows to close a DOS application:

- In the Advanced Options dialog box (accessed from the main PIF Editor dialog box), turn on the Allow Close When Active check box.
- Press **Ctrl+Alt+Del**.
- In the Settings dialog box (accessed from the Control menu for the application), click on the Terminate button.

When you try to use any of these techniques, Windows will display a warning that encourages you to exit by using the application's own close or exit procedures. (For example, in WordPerfect, you press **F7**, then press **Y** or **N** to answer WordPerfect's two exit prompts; if you're using Lotus 1-2-3, you press **/Q**, then respond to Lotus's exit prompts.) In most—but not all—cases, it is not a good idea, and sometimes a very bad idea, to use any of the above three Windows-based techniques. However, there will be situations when you can use these techniques safely and in fact there will be times when you *must* use either the second or third technique.

The first technique is useful if you want to use the Control-menu button (sometimes called the Close button) or the Close option in the Control menu to exit a DOS application. But when you use this technique, the DOS application cannot follow its standard procedures for exiting—which may include writing information to files immediately before exiting. Therefore, you should use this technique *only* if you are sure that your application doesn't save configuration or other information when it closes. The MS-DOS Prompt window can be safely closed in this way. By contrast, Microsoft Word for DOS (versions 5.5 and earlier) writes configuration information to disk upon exiting, so you should never use this technique with it.

Figure 14.10 shows the dialog box that appears when you try to turn on the Allow Close When Active check box in an application's PIF (Technique 1). If you bypass this warning and save the PIF, Windows will still display a warning whenever you try to use the Control-menu button to close the application—but this message just serves as a reminder that you haven't closed the DOS application from within the application itself. Windows assumes that you are aware of the potential dangers involved in exiting a DOS application in this way, since you've already taken the trouble to turn on the Allow Close When Active check box.

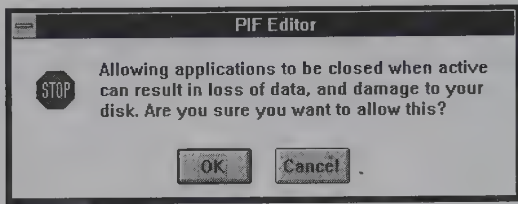


Figure 14.10 This warning dialog box appears when you turn on the Close When Active check box in a DOS application's PIF.

When you use the second technique (pressing **Ctrl+Alt+Del** while the DOS application is open), Windows closes the current DOS application without actually exiting Windows or rebooting your computer. This is the best way—and sometimes the only way—to close a DOS application that has crashed, has locked up Windows, or has begun to behave erratically.

When you press **Ctrl+Alt+Del**, Windows switches to virtual-86 mode and displays a warning screen that explains what will happen if you continue. When you press **Enter**, Windows closes the application so that any memory or other hardware conflicts can be resolved.

Although this approach allows you to remain in Windows after a DOS application has stopped responding, it also prevents the application from saving files and other information prior to exiting. Therefore, you should never use this technique to close a DOS application if you know that the application writes to files before exiting.

And even if you know that the application *doesn't* write to files before it exits, there are some good reasons for not using this technique simply as a convenient way to close the application. First, if you have more than one DOS application open, you might accidentally exit the wrong program. Second, if you get in the habit of using this technique, you might accidentally use it when *no* DOS application is open—which will force your computer to reboot and consequently cause Windows to exit without saving vital information.

The third technique (using the Terminate button) has the same effect as the second, but is a little safer only because you can't accidentally exit the wrong DOS application or accidentally reboot your system. Even so, you should never use this technique to close a DOS application unless you know that the application does not write to files prior to exiting. Because the Terminate button appears to be a convenient way to exit a DOS application, Microsoft has included the rather dire warning shown in Figure 14.11 to discourage you from using this technique.

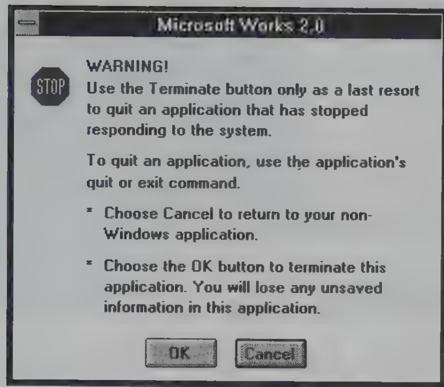


Figure 14.11 Windows displays this warning if you try to use the Terminate button to exit a DOS application.

Using the Windows Clipboard with DOS Applications Running in a Window

One of the most compelling reasons to run a DOS application in a window is that you can use the Windows Clipboard to copy and paste text to and from other applications. You can copy and paste from one DOS application to another—from a Windows application to a DOS application—or from a DOS application to a Windows application.

Microsoft introduced the ability to cut windowed DOS text in Windows 3.0, but many Windows users still aren't aware of its power. Normally, formatted text created in one DOS application can't be copied into a different DOS application, at least not easily, because the two applications don't use compatible formatting codes.

To copy text between applications under DOS, users often first have to convert text to ASCII, save it to a file, exit the application and open the second application, read the ASCII file into the desired document, then reformat the text in this second application.

If you're running a DOS application in a window, you can use the Edit Mark and Edit Copy commands to copy text from one DOS application to the Windows Clipboard and use Edit Paste to paste text from the Clipboard into a different DOS application. The Mark, Copy, and Paste editing commands are all available from the application's Control menu, as shown in Figure 14.12. When you copy and paste in this manner, Windows transfers text characters only, along with spaces and hard

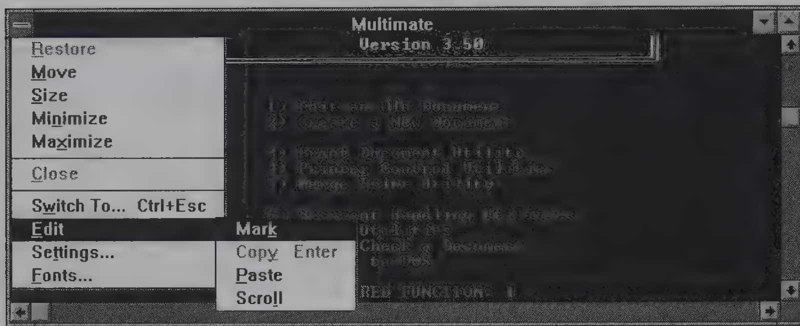


Figure 14.12 Use Mark to highlight a block of text, Copy (currently dimmed) to copy a marked block of text to the Windows Clipboard, and Paste to copy text from the Clipboard into a windowed DOS application.

returns. (You can't copy and paste graphics.) Because Windows uses its own screen fonts when text is displayed, spacing often can be transferred accurately, even though the DOS applications might normally use formatting characters to control spacing (as is the case with different cells in a spreadsheet or tabbed text in a word-processing document).

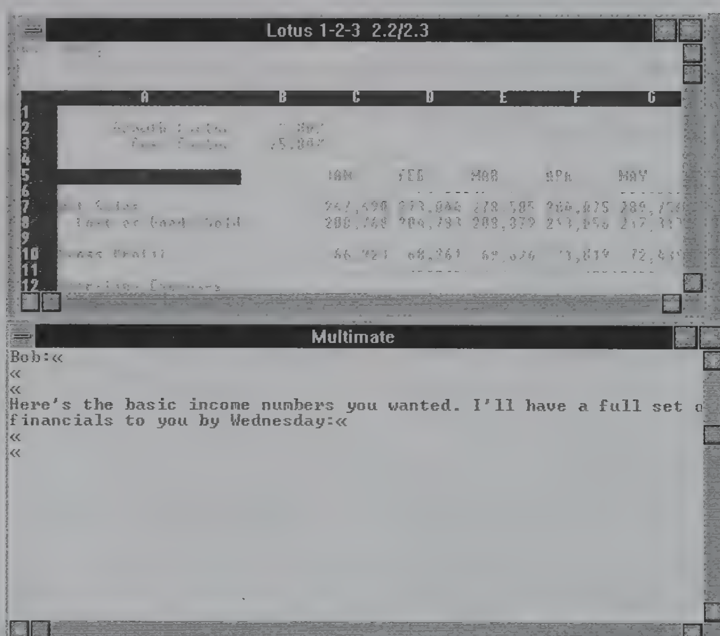


Figure 14.13 This display shows the Lotus 1-2-3 spreadsheet that contains text to be marked and copied, along with the MultiMate document that the text will be pasted into.

To demonstrate this technique, we'll copy and paste text from a Lotus 1-2-3 spreadsheet into a MultiMate document—two applications that normally use incompatible text formats. Figure 14.13 shows the partial Lotus spreadsheet and the MultiMate document that spreadsheet text will be pasted into. We'll copy and paste cells A5 through G10, but you can use these same basic steps to copy and paste text between any two DOS applications:

1. Display Lotus 1-2-3 in a window (by pressing **Alt+Enter**).
2. Display the Control menu and click on Edit Mark.
3. Position the mouse pointer at the starting location for the block of text to be copied (in this case, at the start of cell A5), then click the mouse button to position the highlight cursor.
4. Drag the mouse pointer to the end of the block of text to be copied (in this case, the end of cell G10).
5. Display the Control menu and click on Edit Copy to copy the highlighted text into the Windows Clipboard.
6. Display MultiMate (or the target application) in a window, open the document that will accept the text from the Clipboard, then position the cursor where you want the pasted text to begin.

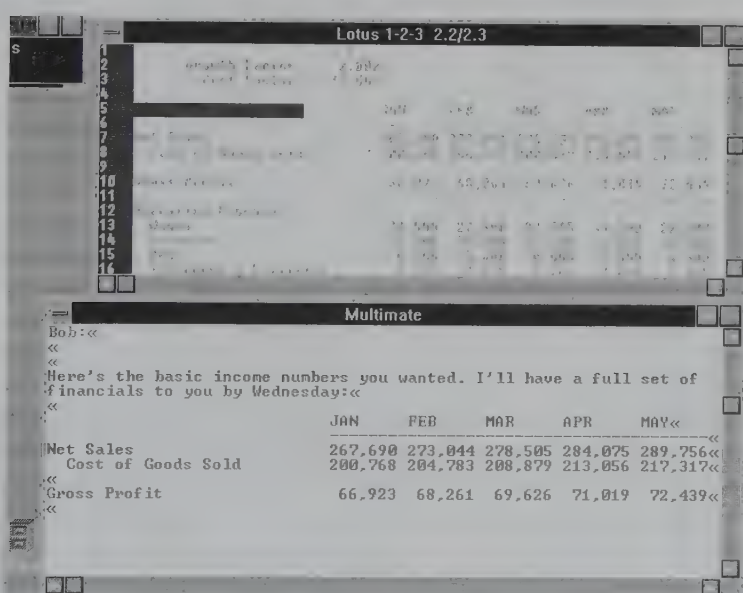


Figure 14.14 This display shows how the Lotus 1-2-3 text appears in the MultiMate document after it has been pasted from the Windows Clipboard.

7. Display the Control menu for MultiMate (or the target application), then click on Edit Paste. Windows pastes the text into MultiMate and retains the appropriate spacing used in the Lotus spreadsheet, as shown in Figure 14.14.

Note: If Windows pastes text incompletely or not at all, the target application probably cannot support the speed at which Windows pastes characters from the Clipboard into the document. To resolve this problem, start the PIF Editor and open the target application's PIF. Click on the Advanced button to display the Advanced Options dialog box, and then turn off the Allow Fast Paste check box. Save the revised PIF.

HOT TIP

Windows 3.1 Can Turn Control of Your Mouse Over to a DOS Application

Even if your mouse does not work inside a windowed DOS application (see the next topic for more information), you can still use your mouse to copy and paste text to and from windowed DOS applications. When you use the Windows Edit Mark or Edit Paste commands, you are using the mouse driver that Windows installed during the setup procedure, and Windows retains control of the mouse driver.

However, when you use the mouse to make routine selections in a DOS application, Windows turns control of your mouse over to the mouse driver that gets installed at the DOS level when your system boots up. This is true regardless of whether you are running the DOS application full screen or in a window. If you are having trouble using your mouse with DOS applications, see the next topic for potential reasons and solutions.

Getting Your Mouse to Work in a DOS Window

If your mouse works fine in Windows applications, but won't work inside a windowed DOS application, you are probably using a video or mouse driver that does not support this Windows 3.1 capability. You can usually resolve the problem by installing a different or updated driver.

Windows 3.0 did not let you use your mouse inside a windowed DOS application, but this feature is available in Windows 3.1—provided that both your video and mouse driver support this feature. Whenever you use a DOS application, Windows disables its own mouse driver so that your DOS mouse driver can take over. So, you need to make sure a mouse driver gets installed when your system boots. Your AUTOEXEC.BAT or CONFIG.SYS file must contain an entry that loads your mouse driver. If a Windows 3.1-compatible mouse driver has been correctly installed at the DOS level, you'll be able to use the mouse in full-screen DOS applications that offer mouse support. However, you will be able to use your mouse in a windowed DOS application only if your mouse driver *and* video driver support this capability.

If your DOS mouse driver files are dated prior to the date of the Windows 3.1 files (3/10/92), the chances are good that your mouse driver doesn't support windowed DOS applications. If you've been using the same mouse driver for years, you *definitely* won't be able to use your mouse in a windowed DOS application. In any case, you'll need to contact your mouse manufacturer for an updated driver. (The Windows Driver Library does not contain any updated mouse drivers.)

HOT TIP

If You Use a Microsoft Mouse, an Updated Driver Is Available with Windows 3.1

If you have a Microsoft mouse, your mouse will work in a windowed DOS application only if you have installed a mouse driver that's version 8.2 or higher. Fortunately, the installation disks for Windows 3.1 include version 8.2 of the MOUSE.COM and MOUSE.SYS driver files, both of which were installed in your WINDOWS directory if your Microsoft mouse was connected when you installed Windows 3.1. However, you won't have mouse capabilities in a DOS window unless you change the path for your DOS mouse driver to the WINDOWS directory. If you currently install MOUSE.SYS from your CONFIG.SYS file, change the DEVICE= line to match this one:

```
DEVICE=C:\WINDOWS\MOUSE.SYS
```

If you currently install MOUSE.COM in your AUTOEXEC.BAT file, make sure your MOUSE.SYS line matches this one:

```
C:\WINDOWS\MOUSE.COM
```

Note: If you don't have a Microsoft mouse, you can still try using the Microsoft mouse drivers. Some mice use the same communications protocol as the Microsoft mouse and therefore are often Microsoft-mouse compatible. However, the version 8.2 MOUSE.COM and MOUSE.SYS probably were not installed in your WINDOWS directory when you set up Windows 3.1. To use either of these files, you must use the Windows EXPAND utility to uncompress and copy MOUSE.CO_ or MOUSE.SY_ from Disk 4 of the Windows 3.1 installation disks. Appendix A explains how to use the EXPAND utility.

All of the video drivers supplied with Windows 3.1 support mouse operations in a windowed DOS application. However, if you're using a vendor-supplied driver, it might not support this feature. If you're sure that you are using a mouse driver that does support windowed DOS applications, but your mouse still isn't working in DOS windows, you can either contact your video manufacturer for an updated video driver or install a Windows-supplied driver that's compatible with your video card (such as the standard VGA or Super VGA display drivers).

Warning: If you already had installed a font manager—including Bitstream Facelift and Hewlett-Packard's Intellifont—the Windows 3.1 setup program might not have updated your video driver. If the setup program fails to update your driver, Windows 3.1 won't provide mouse support in a DOS window. To correct this problem, run Windows Setup *after* Windows has finished installing itself, and use the Change System Settings dialog box to install a Windows 3.1 display driver.

HOT TIP

If You're Using a Vendor-Supplied Video Driver, Try Adding MouseInDosBox=1 to SYSTEM.INI

If Windows detects that your video driver is not 100-percent compatible with Windows 3.1, mouse support for DOS windows will be disabled. However, some newer third-party video drivers can support this feature. You can test this by adding the following line to the [NonWindowsApp] section of SYSTEM.INI, then rebooting your computer:

```
MouseInDosBox=1
```

If the mouse still does not work in a DOS window, you probably need to update your mouse driver. If the mouse works after you make this change to SYSTEM.INI, but behaves erratically, you should delete the MouseInDosBox=1 entry and use either a Windows 3.1 video driver or an updated vendor-supplied video driver.

Correcting Mouse Tracking in WordPerfect 5.1 Running in a Window

When you run WordPerfect 5.1 in a window, your cursor won't be able to track the Windows mouse pointer correctly unless you move the mouse extremely slowly. However, you can correct this problem by changing the mouse pointer's acceleration factor.

Here's the procedure for correcting mouse tracking in a WordPerfect 5.1 window:

1. Open WordPerfect and display it in a window.
2. Press **Shift+F1** to display the Setup menu, then press **M** to select the Mouse option.
3. Press **A** to select the Acceleration Factor, Change the value to 1, then press **Enter**.
4. Press **Enter** twice to return to the WordPerfect screen.
5. If mouse tracking still seems to be slightly off, maximize the WordPerfect window and move the mouse pointer to all four corners of the window.

Older Microsoft and Mouse Systems Mice Cause Problems in Word 5.0 and Word 5.5

If you have a Mouse Systems mouse driver that's version 7.01 or earlier, you'll lose mouse support if you minimize a Word 5.0 or Word 5.5 window in Windows 3.1. If you have a Microsoft mouse driver that's earlier than version 8.2, Word 5.0 or Word 5.5 can crash if you're running Windows 3.1 in enhanced mode.

If you run Word 5.0 or Word 5.5 in Windows 3.1, and if you use an older Mouse Systems or Microsoft mouse driver, you should upgrade to the most current driver. Keep in mind that these problems involve the DOS mouse driver that is installed prior to running Windows, since Windows 3.1 turns control of the mouse over to the DOS mouse driver whenever you are in a non-Windows application. The Mouse Systems and Microsoft mouse drivers supplied by Windows 3.1 will work fine with all Windows applications.

Note: For information about other non-Windows applications that might have problems running in Windows 3.1, use Write to open and read Section 8 of the README.WRI file (stored in the WINDOWS directory).

Associating Your DOS Application's Files with Windows

Many DOS applications allow you to assign any one- to three-letter extension to your file names. In most cases, you won't be able to open these documents directly from the File Manager unless you associate the extensions with your DOS applications.

In a few other cases, your DOS application files might actually get associated with Windows applications by default. For instance, Word for DOS and MultiMate both assign the extension .DOC to documents by default. Since Windows automatically associates .DOC files with Word for Windows, you'll actually start Word for Windows whenever you double-click on a .DOC file that you created in Word for DOS or MultiMate—probably not the effect that you want.

In any case, you can get Windows to recognize your DOS applications and their extensions by creating associations for file extensions that you use frequently and by registering your DOS applications in the Windows registration database. When you use these techniques, you improve your ability to launch application/document combinations from within the File Manager and within the Program Manager.

When you associate extensions for documents that you've created with non-Windows applications, you can open a document and its application directly by double-clicking on file names within the File Manager.

Documents that have associated extensions also can be dragged into the Program Manager to create document icons, which allow you to open a document and its application directly from within the Program Manager.

If you register a non-Windows application, you make it possible to view associations for your non-Windows applications in the File Manager's Associate dialog box.

All of the techniques you need to create associations and register applications are explained in Chapter 8, in the section *Associating Documents with an Application: Some Useful Techniques*, so we won't duplicate that information here. (The techniques for associating documents and registering applications are essentially the same for both Windows and non-Windows applications.) We highly recommend that you put the information in Chapter 9 to work if you use non-Windows applications frequently.

CHAPTER

15

Object Linking and Embedding

To pronounce OLE correctly, just spell out the letters—O-L-E—or use the Spanish cheer *olé!* Either pronunciation is acceptable. But to *use* OLE correctly is another matter entirely, and a complex one. OLE, which stands for *object linking and embedding*, isn't a single Windows feature—it's actually a *collection* of features. And that misunderstanding has led to numerous misconceptions and much confusion about OLE. Perhaps you've heard the old saying: "Ask two economists about the state of the economy and you'll get three opinions." If you ask two Windows consultants how to use OLE, you'll probably get at least three opinions *and* a lot of conflicting information.

The problem is that OLE works differently (and sometimes not at all) depending on which applications you're trying to use OLE features with, and depending on what you're trying to *link* and how you're trying to create your links. And linking is really at the heart of OLE. Stripped of all its finery, OLE essentially just provides a way for you to complete a single task by linking objects in a document with multiple applications. In other words, data, pictures, text, and other objects that you create in different applications all can be combined into a single *compound document* that contains links to all the original, or source, applications.

The compound document (compound because it contains objects created by two or more applications) is controlled by a single application, such as Excel or Word for Windows. The links provide pathways to other applications so that you can edit your objects using the applications in which they were created. But the links also let you keep the objects in a single document; you don't have to continually switch between applications and continually copy and paste text, data, and graphics to complete your task.

Task-Centered Computing

Microsoft calls OLE a *task-centered* approach to using your computer. For example, suppose you're sitting at your desk trying to complete a complicated task. If you continually have to get up to find different files, materials, information, and so on, completing your project becomes more and more time consuming. On the other hand, if you have all of the tools and information you need at your fingertips, you can stay at your desk and complete the task much more quickly.

OLE features, when they're used correctly, allow you to centralize all of your work within one dominant application, with all information stored within one document—called the *client*. If you need any data, graphics, or other information that you've created or have to create within other applications, you can still remain within your client application by linking the objects you need from these other applications, called *server applications.*

If you need to edit text, data, or graphics created in a server application, you can do so from within the client document, usually by double-clicking on the object you want to edit. When you double-click on the object, Windows uses the link you've already created to open the server application and the object that's associated with it, allowing you to edit the object. When you're done making changes to the object, you simply exit the server application and you're automatically returned to the client application and the document you're working on.

Object Linking Is Different than Object Embedding

A lot of the confusion we've seen and heard regarding OLE stems from the way Microsoft uses the verbs *linking* (or its noun counterpart—a *link*), *object linking* (or its noun counterpart—*linked object*) and *object embedding* (or its noun counterpart—*embedded object*). These three sets of terms have vastly different meanings, although they all are incorporated within the same basic OLE technology. In our view, the key to unlocking the mystery of OLE is to understand the differences among these three sets of terms. We'll do that in the next few sections.

Linking a Document to an Application

In its most generic sense, a link refers to a connection that allows one Windows document (called the client) to communicate with another Windows application (called the server).

The term *client* almost always refers to a document, *not* an application. However, the term *server* can refer to either an application or a document, and sometimes to both. The reasons for this confusing use of terms actually stems from the way Windows can form links.

A source document is simply the file that you use to copy data, text, or graphics to the Clipboard so that you can paste or paste-link (by creating either a linked object or embedded object) the Clipboard contents to a different document, called the client. The actual link that gets created, though, is always between the client document and the server application. This link provides a way for the client document to know which application a paste-linked object was created by, and how to start this server application if necessary. (This approach describes an *embedded object*.)

In some cases (specifically, for *linked objects*), two additional links are made—between the client document and the source document and between the client document and the object within the source document that was copied and pasted. With an embedded object, the source document is often called the *server document* because it is always controlled by the server application and because it supplies data to a linked object. When these two additional links are made, changes made to data in the copied source object can be automatically updated in the client's object. Figure 15.1 illustrates the links that can be made between a client document and server applications and server documents.

To gain a quick understanding of the difference between a linked object and an embedded object, take a look at these characteristics:

- A linked object typically stores only descriptors that tell the object where to find the server application, the server document, and the linked item in the server document. The server application then

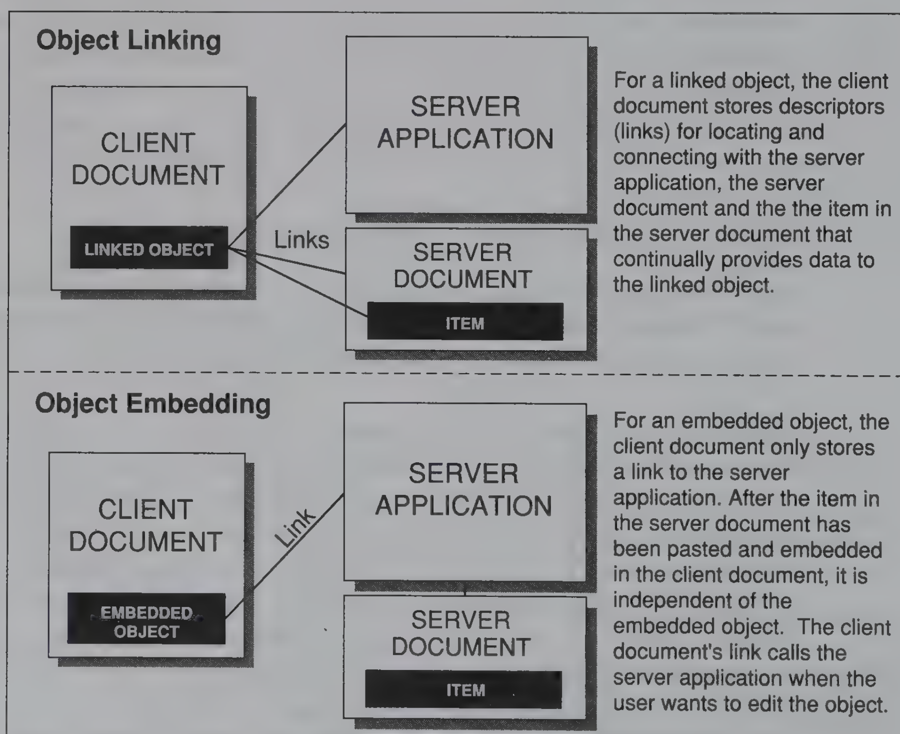


Figure 15.1 This illustration shows the relationships between client and server for linked objects and embedded objects.

updates the client document whenever information changes in the server document. Client documents in some applications also save the most current linked information when you exit the document. You can then elect to update the links when you reopen the document.

- An embedded object is a fully contained version of the pasted item; it contains all of the data, text, and/or graphics that were pasted from the Clipboard to create the object. An embedded object also contains a link to the server application so that when you double-click on the object in the client document, the server application can be started (if it isn't already running) and the object can be edited using the server application's editing tools.

HOT TIP

Determining when to Create an Embedded Object versus a Linked Object

In earlier versions of Windows, you could paste a *static* object simply by copying the object to the Clipboard and then using Edit Paste in another application to paste the Clipboard contents into a document.

In Windows 3.1, you can use Edit Paste to create static objects, *and* you can use the Clipboard to create linked objects and embedded objects. But how do you determine which type of object to paste? In general, create a linked object when you want to share data between the server document and the client document. In this way, the object in the client document will be updated with the most current information whenever you edit that document from within its native application.

Create an embedded object if you still want to be able to use the native application to edit the object, but don't want to maintain a link between the client document and the server document.

Although we recognize these distinctions can be blurry at first, we think they'll become increasingly clear as you continue to read and examine the following sections and examples. For now, just realize that a linked object essentially links a client document to a server document and its application, while an embedded object just links the client document to the server application.

The Difference between DDE and OLE

Dynamic Data Exchange (DDE) is an older linking technology that could be used with Windows 3.0 and even earlier versions of Windows—as long as applications included routines for using the DDE message-passing conventions established for the Windows API. In fact, DDE was made possible by the message-passing features of Windows and the ability to transfer information via the Windows Clipboard. By contrast, OLE is new to Windows 3.1, and makes it easier for Windows applications to create links because the linking routines are part of Windows itself.

So, when applications use DDE technology, they pass messages directly, and are responsible for passing data to and from the Clipboard. Under OLE, though, applications simply call a set of dynamic link *libraries* (DLLs) that are stored in the WINDOWS\SYSTEM directory, which handle all of the message-passing responsibilities.

Although the older DDE approach to linking can still be used by Windows applications running under Windows 3.1, most of the recently released Windows applications use the more efficient OLE technology instead. Even so, the distinction between DDE and OLE can seem fuzzy because some applications, such as Word for Windows 2.0, use DDE features to retain downward compatibility with older applications. Also, a few applications, such as Excel 4.0, always implement OLE features by using older DDE technology. (Excel uses this approach both to maintain backward compatibility and to control the way linked and embedded objects get updated.) To understand the distinction between DDE and OLE technology, just remember that DDE is old, and OLE is new. Recent applications can incorporate the old within the new, but older applications can only support the old technology, not the newer OLE features.

OLE Uses a Set of DDLs Stored in the WINDOWS\SYSTEM Directory

Two main DLL files support OLE: OLECLI.DLL and OLESVR.DLL. Both of these files are stored in the WINDOWS\SYSTEM directory.

OLECLI.DLL is the OLE client library. Applications for client documents call OLECLI.DLL to create, render (draw and format), load, and save client objects. The OLECLI.DLL library can both accept calls from a client application, and send "callback" messages to the client application to provide updated status information regarding changes made in the object.

OLESVR.DLL is the OLE server library. Server applications call OLESVR.DLL to notify it when it saves, renames, or modifies a document that contains a link to a client. A server application can also call

OLESVR.DLL to alert the library that it (the application) has been opened and when it is closed.

Again, OLECLI.DLL and OLESVR.DLL make it possible for applications to use all OLE features without the need to include functions for exchanging messages directly between client and server applications and without having to use the Clipboard directly. The set of OLE DLLs take care of all these functions; all the applications have to do is request DLL services and be able to accept messages from the OLE DLLs.

SHELL.DLL and DDEML.DLL Also Affect OLE Services

Two other DLLs can influence the way OLE features are performed: SHELL.DLL and DDEML.DLL.

SHELL.DLL is used to register a Windows application when its setup program is used to install the application. The application's setup program can call SHELL.DLL to provide information about the OLE features it supports and the extent to which the application can support these features. In turn, SHELL.DLL writes all of this information in the Windows registration database: the file *REG.DAT*, which is stored in the WINDOWS directory. We'll explain more about viewing and using the registration database later in the chapter.

The DDEML.DLL (Dynamic Data Exchange Management Library) file provides support for applications that use the older DDE technology. Although DDEML.DLL was included in Windows 3.1 largely to provide DDE support for older applications that must use this technology, the library does offer a few unique capabilities. Specifically, DDEML.DLL provides extremely efficient routines for applications that have to update many (perhaps hundreds or thousands) of links on a frequent basis. Windows calls this requirement *persistence.*

An example of a highly persistent server application would be a stock market quotation service that must provide updated information to linked documents in a rapid, frequent, and extensive manner. Excel 4.0 uses the DDEML.DLL to facilitate worksheets or other Excel documents that might contain several links that need to be updated frequently.

Some other applications, including Word for Windows 2.0, can use DDEML.DLL to support links that have been made between Word documents and documents in pre-OLE applications, such as Excel 3.0. Therefore, you shouldn't delete the DDEML.DLL file unless you are certain you don't currently have and won't eventually need to use applications that might rely on this library.

Applications That Support OLE Don't Always Support All OLE Services

To link information from one application (the server) into another application (the client), both applications have to support the OLE services you're trying to use—and not all Windows applications do. (DOS applications never do.)

Most Windows applications that have been developed since the release of Windows 3.1 support some, but not all, OLE features. Some applications provide some server functions but not all client functions, while other applications do a better job supporting client functions and are limited in their ability to provide server functions.

For example, the Windows 3.1 Cardfile can accept both linked and embedded objects, as a client, but cannot provide server links for linked or embedded objects. The Works for Windows 2.0 spreadsheet can provide server support for linked and embedded objects, but cannot accept either of these object types in a client document. By contrast, the Works for Windows 2.0 word processor can accept both linked and embedded objects in client documents, but cannot provide either of these objects as a server application. The Works for Windows database can't provide any OLE services.

Table 15.1 provides a sampling of the OLE services supported by Microsoft applications designed for Windows. This table is for illustrative purposes only. We realize that an increasing number of non-Microsoft Windows applications do support OLE services, but we can't possibly provide an up-to-date list of these, since these applications are being continually introduced. At this writing, though, the current versions of Ami Pro and Professional Write both provide full OLE services, and Paradox for Windows also intends to provide full OLE services. Windows 3.1 automatically updates registration information for Ami Pro and Professional Write, but doesn't update REG.DAT files for other non-Microsoft applications.

Using OLE

We've already suggested that object linking and embedding features can seem complex. However, that's no reason to avoid using OLE services. If your work requirements can benefit from creating compound documents, and if your system has enough memory to support OLE, you should take the opportunity to avail yourself of OLE capabilities. In fact,

Table 15.1 OLE Features Supported by Microsoft Applications

Microsoft Application	Client Support Embedded Objects	Client Support Linked Objects	Server Support Embedded Objects	Server Support Linked Objects	Uses DDE
Cardfile	Y	Y	N	N	N
Draw 1.0	N	N	Y	Y	N
Equation Editor	N	N	Y	N	N
Excel 4.0	Y	Y	Y	Y	Y
Graph 3.0	N	N	Y	N	Y
Note-It	N	N	Y	N	N
Paintbrush	N	N	Y	Y	N
PowerPoint 2.0	Y	N	N	N	Y
PowerPoint 3.0	Y	Y	Y	Y	N
Project 3.0	Y	Y	Y	Y	N
Publisher 1.0	Y	Y	N	N	N
Sound Recorder	N	N	Y	Y	N
Word for Windows 2.0	Y	Y	Y	Y	Y
Works for Windows 2.0 Database	N	N	N	N	N
Works for Windows 2.0 Spreadsheet	N	N	Y	Y	N
Works for Windows 2.0 Word Processor	Y	Y	N	N	N
WordArt	N	N	Y	N	N
Write	Y	Y	N	N	N

we think you'll find that OLE is easy to understand after you practice using it a bit.

Toward that end, we'll provide several examples of object linking and object embedding, and walk you through specific techniques for creating objects under several different application scenarios. Along the way, we'll include several tips for using OLE features and for working around some recognized OLE problems.

A Linked Object Allows You to Share Data between Two or More Applications

A linked object contains a link between a client document and a server application, as is true with embedded objects. However, it

also contains descriptors that, in effect, set up two additional links: one between the client document and the server document and one between the client document and the copied-and-pasted item in the client document. These additional links allow Windows to update the contents of the linked object (in the client) whenever you change the contents of the server document.

For clarity, we'll always use the word *item* to refer to an area of a document that gets copied from the source document to the Clipboard, and the word *object* to refer to the area in a client document that contains the paste-linked item.

When you create a linked object, Windows can update the contents of the client object whenever you make a change in the server document. In this sense, the server document *shares* part of its contents with the client document.

For instance, suppose you create a linked object by pasting the range of cells B6:F6 from an Excel worksheet named ANIMART.XLS into a Word document named MEMO.DOC. Now, whenever you make a change in ANIMART.XLS that affects the linked object in MEMO.DOC, Windows automatically updates the contents of MEMO.DOC.

Next, we'll demonstrate the procedures for creating this link. But first, we'll recap by reviewing the components that are involved in creating a linked object for this scenario:

ANIMART.XLS This is the source document in our example, also called the server document because it contains the item to be linked to the client document.

B6:F6 This range of cells is the item in the source document (ANIMART.EXL) that supplies data for the linked object in the client document (MEMO.DOC).

Excel This is the server application because it controls the contents of the server document, ANIMART.XLS.

MEMO.DOC This Word document contains the linked object. Whenever cells B6:F6 change in ANIMART.XLS, the contents of the linked object in MEMO.DOC will be updated to reflect the change.

Example 1: Linking Part of an Excel Worksheet to a Word for Windows Document

To illustrate the creation and benefits of linked objects, we'll link a range of cells in an Excel 4.0 worksheet to a Word for Windows

2.0 document, using the information we've explained in the above section.

Figure 15.2 shows a memo that we've written in Word for Windows 2.0. Note that we intend to include some sales figures in the memo. We'll get these values from the Excel worksheet shown in Figure 15.3, copying and pasting the row of totals that appears near the bottom of the worksheet. However, instead of simply copying and pasting the range of cells from the worksheet into the Word memo, we'll create a linked object. With a link established between the spreadsheet cells and the Word document, the contents of the cells will be updated in the Word document anytime we make a change to the Excel worksheet.

Follow these steps to create the link:

1. Highlight the cells that contain the data to be copied. The highlighted area is the item in the server document that contains all of the data for the link.
2. Click on Edit Copy to copy the data to the Clipboard.
3. Switch to the Word document and position the insertion point where the object is to appear.
4. In the Word document, click on Edit Paste Special. The Paste Special dialog box appears, shown in Figure 15.4. This dialog box allows you

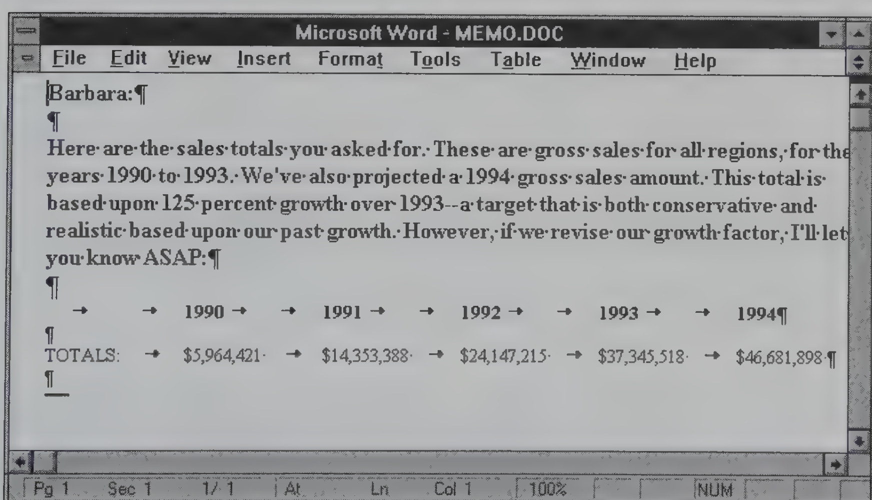


Figure 15.2 This is the client (Word) document, which will receive the pasted and linked data.

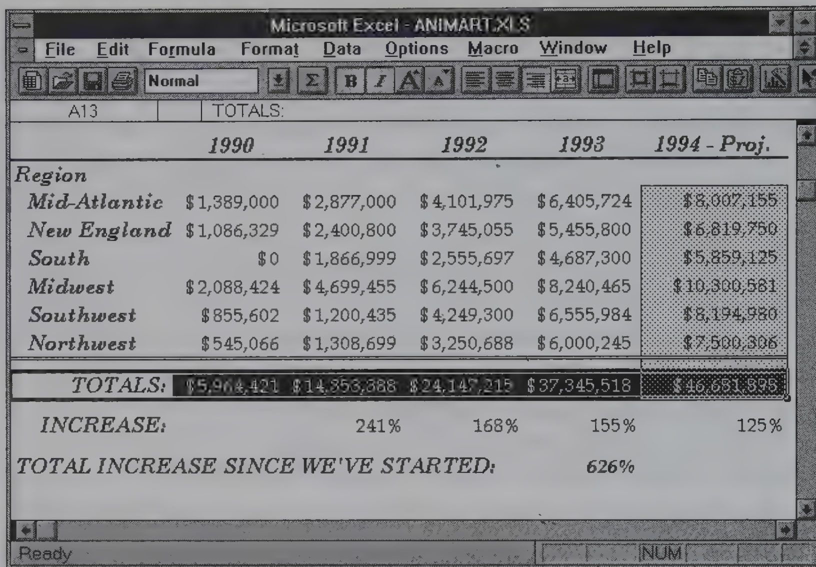


Figure 15.3 This is the server application and document, which will provide the linked data to the client.

to determine the format in which the object will be pasted into the client document. The format you choose determines whether the object is pasted as a static copy, as a linked object, or as an embedded object.

Word provides five formats for the text that's currently in the Clipboard:

Microsoft Excel Worksheet Object Use this format to embed an object in a graphics format called *Windows metafile (WMF).* The benefit of the Windows metafile format is that data, scalable TrueType text, and graphics can be combined in one file and transported from one application to another via the Clipboard. For instance, if you want to paste an object that contains graphics plus text formatted in TrueType fonts, you could paste the object in the picture format. The pasted object would retain all of the font and hinting information that was contained in the original source document. In Word, an object can only be pasted as an embedded object, not as a linked object.

Any linked object or embedded object that contains graphics can only be viewed in Word using the Print Preview screen. However, you can edit the contents of a linked or embedded graphic by double-clicking inside the object's frame. We'll explain more about embedded objects later. For now, just realize that an embedded object does not

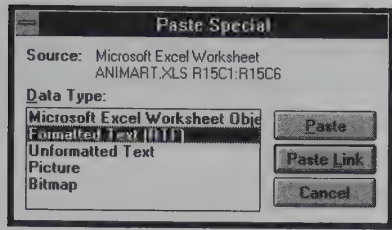


Figure 15.4 Use this dialog box to select the data type that will be inserted into your client document.

contain a link to the server document and therefore doesn't get updated whenever the contents of the server document changes. So, the first option in the Paste Special dialog box can't be used for our current task, which is to create a linked object.

Formatted Text This is the data type that Word suggests for our current task. When you select this format, Word will keep all of the formatting for the text, including font size, font style, and cell width. TrueType fonts and hints also can be brought across the Clipboard when you use this data type. With this option, you can use either the Paste button (to paste a static copy of the cells) or the Paste Link button (to create a linked object that will be updated whenever you make changes in the server document.)

Unformatted Text If you select this format, Word pastes the text with tabs separating each cell value, but with no other formatting. This option is useful if the server document contains elaborate formatting that you don't want to retain within the client document. With this option, you can use either the Paste button (to create a static copy of the text) or the Paste Link button (to create a linked object).

Picture This option pastes the data in the same Windows Metafile format as that used by the Object format. The only difference between the Picture and Object data types is that you have the option of pasting the data as a static object—that is, without creating a link to the server application—or as a linked object. You can't use this option to create an embedded object.

Bitmap This option allows you to create a static object or a linked object in the bitmap (BMP) graphics format used by Paintbrush. You might want to use this option if you will be importing your document to an application that recognizes the BMP graphics format. However, BMP graphics images do not retain TrueType fonts and their hinting information. So, if you scale a bitmap object in Word, you will distort any TrueType text that is stored in the object. Word will not allow you to create an embedded Excel worksheet object in bitmap format.

5. Select the desired paste format, then click on the Paste Link button. (Click on the Paste button only if you want to paste a static copy of the object, with no link to the server application.) Keep in mind that only the Formatted Text and Unformatted Text options will let you create a linked object. The other three choices allow you to create an embedded object only. In our example, we'll select the Unformatted Text option since we don't want our Word document to contain the elaborate font and other formatting used in the Excel worksheet. Figure 15.5 shows the result of the paste link operation.

Now, any time you revise the contents of the Excel (server) worksheet, the changes will be updated automatically in the Word (client) document.

Note: The link between a linked object and its server document only exists in one direction. For instance, suppose we modify the contents of the linked object from within MEMO.DOC (the client document in our example). In other words, suppose we increase the 1990 total (in MEMO.DOC) to $6,964,421. This change won't actually be updated in ANIMART.XLS, because a linked object is only updated in the direction of server document to client document.

If you want to return to the Excel worksheet (perhaps to edit it), you can do so from within Word by following these steps:

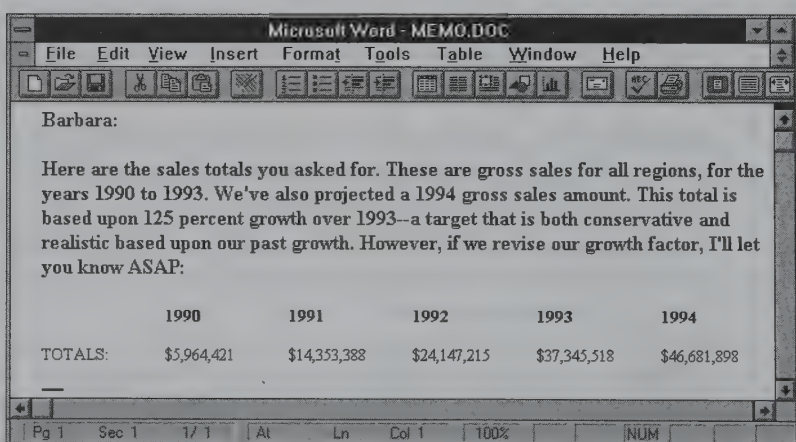


Figure 15.5 This Word window shows the result of pasting and linking data from an Excel worksheet, in Unformatted Text format.

1. From within Word, click on Edit Links to display the Links dialog box shown in Figure 15.6. You can use this dialog box to quickly view all links in a Word document and to change or cancel a link. However, in this case, we're going to use the dialog box to switch to the server application.
2. Select the desired link, and then click on the Open Source button. Word responds by starting Excel (if it isn't already open) and displaying the linked worksheet.
3. Make and save any changes to the document, then either exit or minimize Excel to return to the client document. You should notice that any changes you made from the server are updated automatically in the client document.

Note: You need to use the Links dialog box to return to the server application only for a link created using the Formatted Text or Unformatted Text data types. For other types of linked objects in Word, you can return to the server application simply by double-clicking on the object's frame.

HOT TIP

Paste Link Usually Creates a Linked Object

Application developers have some latitude over the way they use the Paste, Paste Link, and Paste Special menu options boxes. In most applications, the Paste Link option is reserved exclusively for pasting the contents of the Clipboard as a linked object, rather than an embedded object.

If you've copied an item to the Clipboard, and the client application's Edit menu shows the Paste Link option dimmed, you probably have to insert the object as an embedded object or as a static copy of the

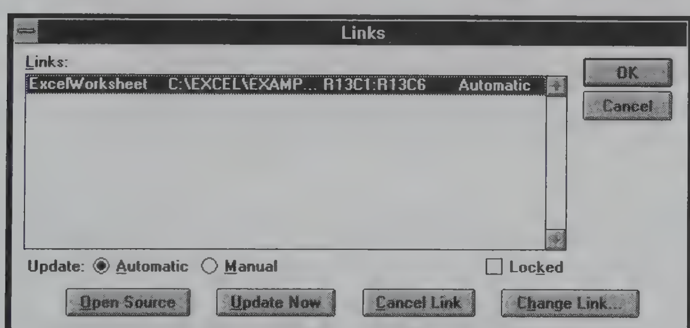


Figure 15.6 The Links dialog box shows linked objects for the current document.

object. A few applications, including Word for Windows 2.0, don't provide a Paste Link option. In these applications, you control the type of link by selecting a particular format from within the Paste Special dialog box.

Example 2: Linking a PowerPoint Slide to a Word Document

In this scenario, you'll see how to link a PowerPoint 3.0 slide into a Word for Windows 2.0 document. We'll also explain why this kind of link can be valuable in creating slide presentations.

Although PowerPoint works great for creating slide presentations, its Notes feature can be cumbersome for creating presentation scripts. For example, you can only view and enter Notes text from within Notes view, and you probably won't be able to read your text at any size less than Actual view. At this size, your screen may redraw very slowly, since it's dealing with a very large graphics image in addition to your text. Another problem is that you can only view Notes text for one slide at a time.

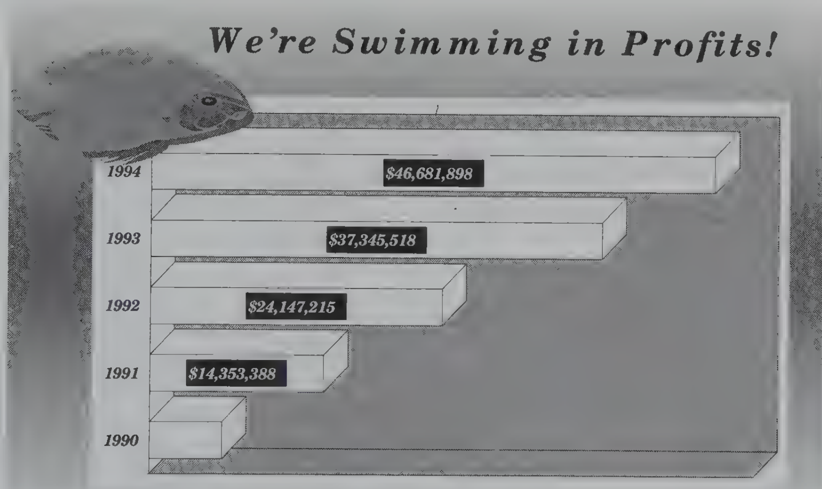
With object linking, you can combine your scripts and slides into a single Word for Windows document. To do so, simply insert PowerPoint slides (as linked objects) into your Word document, then write the narration for each slide within the document. With this approach, you can view and print your slides from within Word, and you can print both your slides and narration as part of a single document. Also, because you're creating a linked object, any changes you make in the original slides are updated in the Word document.

If you want to edit a slide from within Word, simply double-click on the linked object to open PowerPoint and the slide. Figure 15.7 shows a partial slide presentation script with two slides linked to a page, along with the corresponding presentation script.

Here's how to insert a PowerPoint slide into a Word document:

1. Start PowerPoint and open a slide or presentation, then Choose View Slide Sorter. You *must* be in Slide Sorter view in order to link one or more slides.
2. Select the slide (or slides) that you want to link. Keep in mind that if you select multiple slides, they will all be linked as a single object.
3. Choose Edit Copy to copy the selected slide(s) to the Clipboard.
4. Switch to the Word document, then position the insertion point where you want the slide to appear.

Slide 1: "Follow Me" As many of you probably already know, Animart is the hottest new pet supplies company in the nation, both in terms of innovation and in sales. We think the more you see and hear about us, the more you'll like about us.



Slide 2: "Growth" When Animart went public in 1990, our total sales for the year were almost 6 million dollars. In three short years, we've increased that amount by more than 600 percent. In fact, for 1994 we're predicting total sales of more than 46 million.

Figure 15.7 This page shows two PowerPoint slides linked to a Word for Windows document.

5. Choose Edit Paste Special. You'll see the following two Data Type options:

   PowerPoint Slide Object If you select this data type, Word will create an embedded object, not a linked object. Word will create a link to the server application only (PowerPoint), not to the server document (which contains the slides). If you create an embedded object, any changes you make in the original slides will not be updated in the Word document.

   Picture If you select this data type, Word will paste the slide in the Windows metafile format. You can either paste a static copy of the slide (by clicking on the Paste button), or you can paste the slide as a linked object (by clicking on the Paste Link button).

6. Click on Picture, then click on the Paste Link button to paste the slide into Word as a linked object.

You can now size the object's frame within Word. Any changes you make to the slide(s) from within PowerPoint will be updated automatically in the Word document. You can also edit the slide(s) directly from within Word by double-clicking on the object. Double-clicking will start PowerPoint and will open the slide(s).

HOT TIP

PowerPoint 2.0 Does Not Support Linked Objects

If you're using PowerPoint 2.0 rather than PowerPoint 3.0 or higher, you'll have only limited OLE capabilities. PowerPoint 2.0 was one of the first applications to implement OLE, but it was released before Windows 3.1 and therefore doesn't use the full capabilities of the OLE libraries provided in Windows 3.1.

If you're using PowerPoint 2.0, you won't be able to create linked objects—although you can create embedded objects with some limitations. If you want to use full object embedding and linking capabilities with PowerPoint, you should consider upgrading to PowerPoint 3.0 or higher.

Example 3: Linking an Excel Object to a PowerPoint Slide

PowerPoint 3.0 allows you to link all or part of an Excel worksheet, chart, or graph into a slide. In this example, you'll see how to link an Excel worksheet object to a PowerPoint Slide.

Figure 15.8 shows an example of an Excel worksheet that has been linked as an object within a PowerPoint slide.

Here's what we did to create the linked object shown in Figure 15.8:

1. Start Excel and open either a worksheet, chart, or graph. For our example, we'll open an Excel worksheet.
2. Select the range of worksheet cells that you want to paste-link. If you are paste-linking a chart or graph, select the portion of the chart or graph to be paste-linked.
3. Select Edit Copy to copy the selected area to the Clipboard.
4. Switch to PowerPoint and open the slide that will contain the Excel object.
5. Click on Edit Paste Special. You'll see the five data type options that we explained in Example 1. However, in PowerPoint, only the Microsoft Excel Object and Picture data types can be used to create a linked object from a worksheet, chart, or graph. Both of these options produce identical results. The Unformatted Text, Formatted Text (RTF), and Bitmap options can only be used to create static copies of a worksheet. (The two text options will not be available for Excel charts and graphs.)

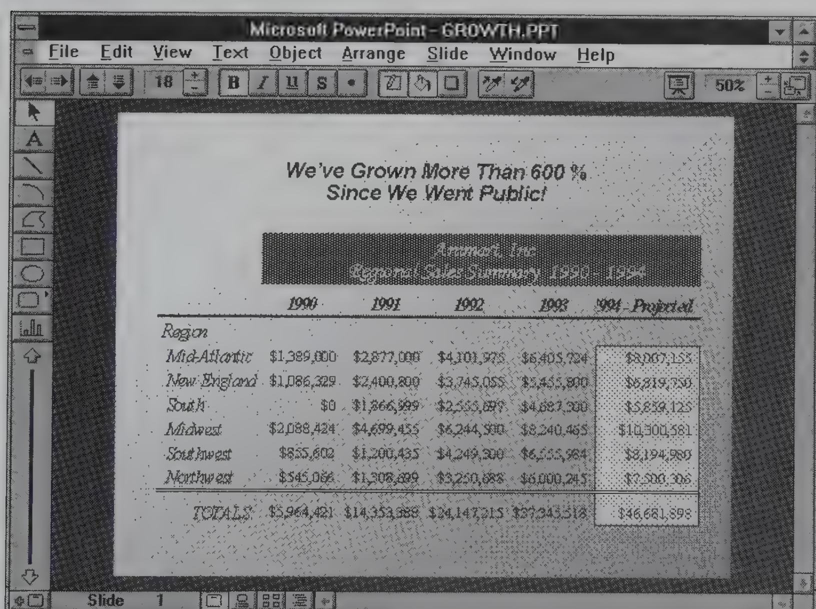


Figure 15.8 This range of worksheet cells has been linked with a PowerPoint slide.

6. Select the Microsoft Excel Object or Picture data type, then click on Paste Link to insert a linked object in your PowerPoint slide. (You would click on the Microsoft Excel Object data type, then the Paste button if you had wanted to create an embedded object rather than a linked object.)

Editing an Embedded Object Using the Server Application

When you create an embedded object, you create a link between the client document and the server application only. The client document does not get linked to the server document or to the copied item in the server document. For this reason, an embedded object is not updated when the contents of the server document change. However, embedded objects are valuable because you can edit the objects from within the client document by using the server application.

Embedded objects are what make the task-centered approach to computing possible. When you create an embedded object in a client document, you can edit the object using its server application, without having to switch away from your current application. This technique is valuable when you need to create compound documents, but don't need to share data between server documents and the client (compound) document. Remember, a compound document contains objects that originate from two or more applications.

Before Windows 3.1, you could create compound documents in some applications, but you couldn't directly edit any objects that weren't *native* to (didn't originate with) the application that controlled the document. For instance, under Windows 3.0, you could copy and paste a graphic image from PowerPoint or Paintbrush into a Word document. However, if you needed to edit this graphic object, you would first have to switch to PowerPoint, Paintbrush, or whatever graphics application you used to create the object. Then you had to edit the graphic from within that graphics application. When you were done, you had to copy the revised graphic to the Clipboard and then return to your Word document. You then had to paste the new version of the graphic into the Word document. The bottom line: there wasn't any way to modify the graphic directly from within Word.

With Windows 3.1 and OLE, the use of compound documents is greatly simplified. For instance, in Word for Windows you can paste a graphic into a Word document as an embedded object. Then if you want to modify the graphic, you simply double-click on the object in the Word document. This action activates the link between the client document and the server application, opens the application in which the graphic was initially created, then opens the embedded object as a separate window within the graphics application.

You can then edit the object using all of the tools available with the graphics program. When you exit the graphics program, Windows 3.1 returns you to the client document (a Word document in this case), which now contains the edited version of the embedded object.

An Embedded Object Is Not Linked to Its Source Document

To understand the difference between a linked object and an embedded object, it's important to keep in mind that the source document for an embedded object is not linked to the embedded object itself.

We've found that the relationship between an embedded object and its source document leads to a lot of confusion among users who try to implement OLE. For this reason, we think it's important to stress how this relationship differs for linked objects and for embedded objects.

We've already stressed that when you create a linked object (as in Example 1), the object gets updated whenever you make a change in the server document. That's because a link exists between the server document and the client document. However, when you create an embedded object, no such link exists. The source document and the client document are totally independent.

So what does this distinction mean from a practical standpoint? Simply this: if you copy an object from, say, a Microsoft Draw file and then paste it as an embedded object within Word, the embedded object no longer belongs to the Microsoft Draw file. So, you could conceivably open Microsoft Draw, open the file from which you copied the embedded object, then make a change to the graphic in this file. When you save the new version of the file, the changes will not be updated in the embedded object because a link does not exist between the source document and the embedded document.

In turn, when you double-click on the embedded object within Word, Windows starts the server application, but *not* the server document. Instead, the embedded object appears in a separate window within the server application—in this case, Microsoft Draw. When you make a change to the embedded object, the change does not get updated to the source document (the document from which you originally copied the graphic). Figure 15.9 illustrates this important difference between linked objects and embedded objects.

HOT TIP

Determine Whether You've Created a Linked Object or an Embedded Object

All linked objects that you've created can be identified by choosing Links from a client application's File or Edit menu. For instance, in Word, Write, Cardfile, and PowerPoint, the Links menu option is located in the Edit menu. In Excel, the Links option is located in the File menu. If the Links option is dimmed, all links you have created are for embedded objects. If the Links option is not dimmed, you can select it to identify all linked objects. Any links that do not appear in the Links dialog box are for embedded objects.

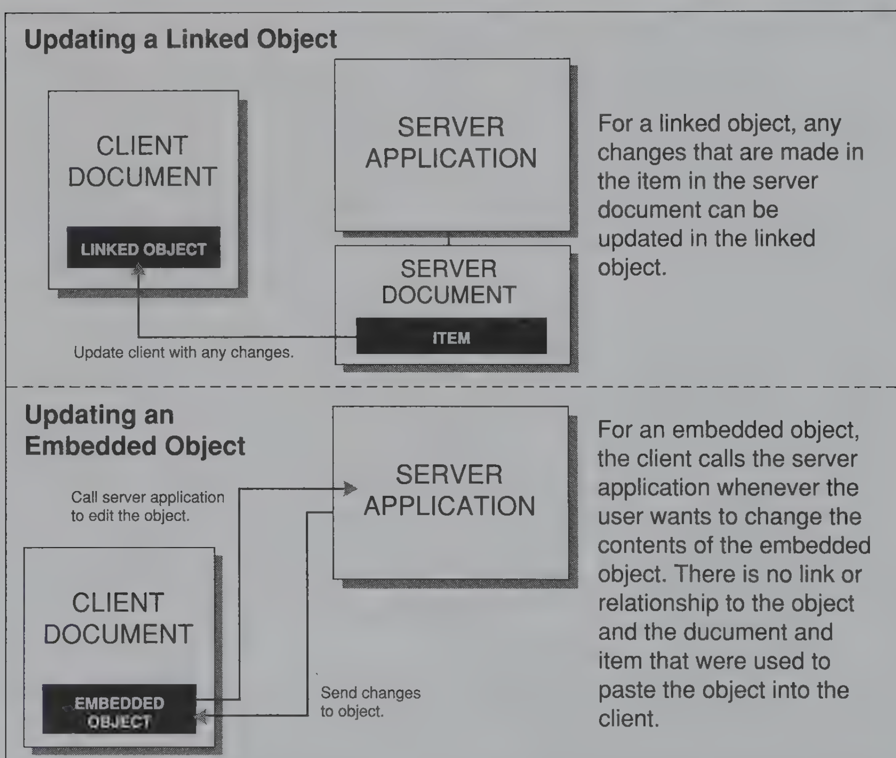


Figure 15.9 A linked object is updated by the server document, while an embedded object is updated by editing the object using the link in the client document.

Creating an Empty Embedded Object

One way to embed an object is to copy it from its source document, switch to the client document, then use the application's Paste or Paste Link option to embed the object in the client document. You can also use the Insert Object option, available in many Windows applications, to insert an empty object into a client document. This second approach lets you use the server application to create the object, without exiting the client document.

The fact that multiple techniques are available to insert embedded objects into a client document causes confusion among many users, and that's not surprising. Depending on the application you're using for your client document, the techniques that are available for inserting embedded objects can vary significantly. In the next few paragraphs, we'll try to explain how these application-dependent differences affect the way you create embedded objects.

If you followed Example 1, you probably noticed that we had to use the Paste Special menu option in order to determine whether to create a linked object or an embedded object, and to specify the format for an imported object. In a few applications—for example, Cardfile—an object will be pasted automatically as an embedded object whenever you choose the Paste command. For most applications, though, you must choose the Paste Special command to create an embedded object. (Even with the Paste Special command, the options and techniques that you can use to embed objects vary from application to application.)

Another way to create an embedded object is to choose the client application's Insert Object command. You can usually find this command in the application's Edit or Insert menu. You can only use this command to create an embedded object—linked objects cannot be inserted in this way.

What's the benefit of using Insert Object? Suppose you're working in a Word document and decide that you want to create a drawing in your document. Since the drawing doesn't already exist, you can't insert it by copying it from another application and paste-linking it into your client document. But with the Insert Object command, you can insert an empty object from a list of available object types. For instance, if you choose to insert a Microsoft Draw object, Word will embed an empty Microsoft Draw object into your Word document. You can then double-click on the object to start Microsoft Draw, create your drawing, then exit Microsoft Draw. The drawing is automatically embedded in your Word document.

You can now edit the drawing at any time by double-clicking on the Microsoft Draw object.

Object types are defined in the Windows registration database (which we'll discuss in greater detail later in the chapter). When you install a Windows application that supports object linking and object embedding, the application's setup program registers its OLE capabilities in the Windows registration database. When you choose an application's Insert Object command, you'll see the Object dialog box shown in Figure 15.10. You can then select one of the object types from the list. Windows will then create a link to the server application for that object type and will allow you to open a window in the server application, as we've just described above.

In the next few sections, we'll demonstrate some different ways you can create embedded objects in applications. Currently, OLE capabilities are available chiefly within Microsoft applications. For this reason, we'll rely on these applications for our examples.

HOT TIP

PowerPoint 2.0 Has Limited Object Embedding Capabilities

Although PowerPoint 3.0 provides full object linking and object embedding capabilities, PowerPoint 2.0's OLE features are much more limited. Linked objects are not supported at all in version 2.0; however, you can create and use embedded objects, with the following limitations:

1. To embed an object, you must use the Insert option available from the File menu. The Paste Link and Paste Special options are not available from the Edit menu.
2. Many server applications (such as Microsoft Draw and Microsoft WordArt) include an Update option in the File menu. Normally, you can use this option to update changes in the client document without exiting the server application. This feature is useful when

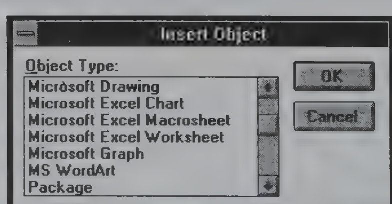


Figure 15.10 This dialog box lists all Windows applications that support OLE.

you want to leave the server application open so that the client document doesn't have to restart the server each time you want to edit an embedded object. However, you can't use Update from the server application in PowerPoint 2.0. Instead, you must exit the server application for the changes to be updated in your client document. Also, PowerPoint 3.0 includes an Update option in its File menu, while PowerPoint 2.0 does not.

3. PowerPoint 2.0 will only recognize the first 20 server applications listed in the [embedded] section of WIN.INI. The items in this section of WIN.INI are listed in alphabetical order by default, so you might need to move some of the entries for them to be recognized as server applications by PowerPoint 2.0. In Windows 3.1, the [embedded] section of WIN.INI is included for downward compatibility only; applications that support the current OLE specification provided with Windows 3.1 (including PowerPoint 3.0) use the registration database, rather than WIN.INI, to identify registered server applications.

Example 4: Embedding a Paintbrush Object into a Cardfile

In Cardfile, you use the Paste command to embed an object from the Clipboard into a card. In this example, you'll see how to embed a Paintbrush object into a card.

1. Start Paintbrush, then open or create a graphic image.
2. Use the Scissors tool to select all or part of the image, as shown in Figure 15.11.
3. Choose Edit Copy.
4. Start Cardfile, then open the file that will contain the embedded graphic.
5. Display the card that will contain the embedded picture.
6. Choose Edit, then make sure the Picture check box in the Edit menu is turned on. You cannot paste a graphic into a Cardfile card unless Picture mode is turned on.
7. Choose Edit Paste to embed the Paintbrush object.
8. Drag the graphic to the desired location in the card, as shown in Figure 15.12.

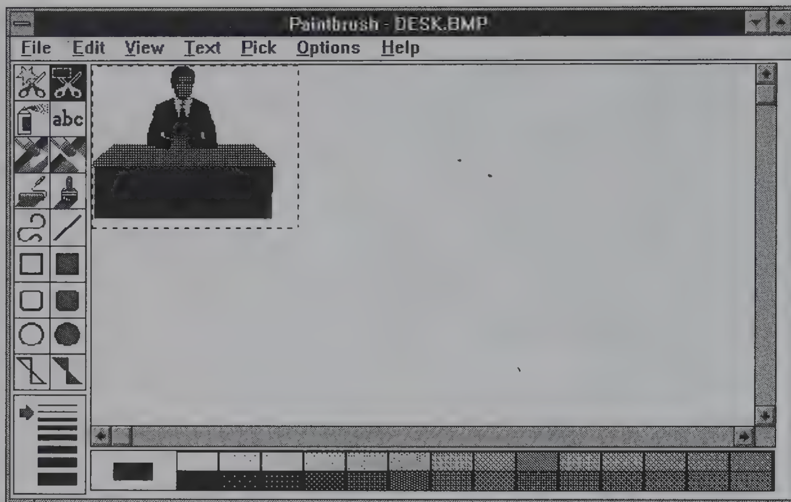


Figure 15.11 The selected area in this Paintbrush object has been copied to the Clipboard.

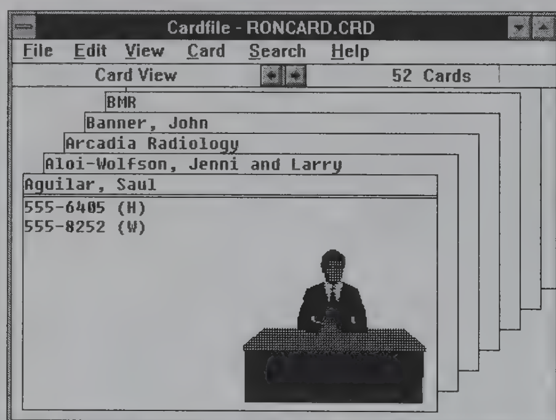


Figure 15.12 This Cardfile card contains an embedded Paintbrush object.

You can now edit the embedded object either by double-clicking on the object or by choosing Edit Paintbrush Picture Object from the Cardfile's Edit menu. Figure 15.13 shows the Paintbrush window that appears when you take either of these actions. Notice that the Paintbrush title line in Figure 15.13 says "Paintbrush Picture in RONCARD.CRD." When you edit this graphic, your changes will affect the embedded object in the Cardfile only. The original Paintbrush file that you used to copy the graphic will not change.

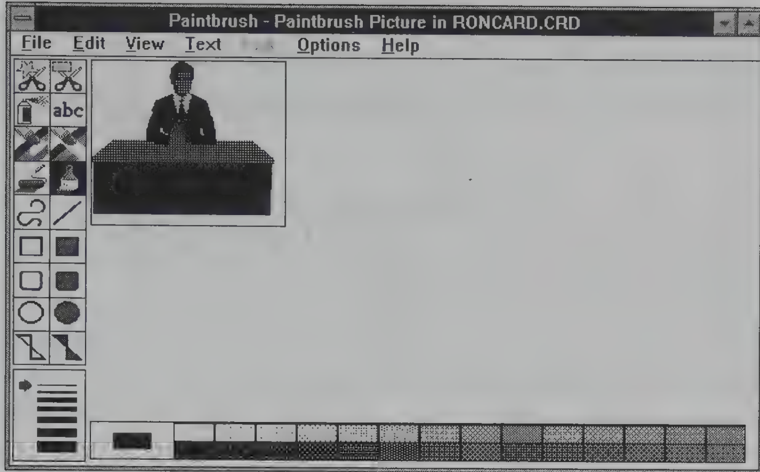


Figure 15.13 When you double-click on an embedded object, the server application opens the object within a window so that you can edit it.

Example 5: Linking or Embedding an Excel Object into a Word Document

In Example 1, we used an Excel worksheet to create a linked object in a Word document. You can also embed an object into Word from Excel. In this example, we'll embed an Excel chart into a Word document. This demonstration can be easily modified to create a linked object instead.

1. Start Excel and open a chart (.XLC file).
2. Select all or part of the chart by clicking on the area that you want to copy.
3. Choose Edit Copy to copy the chart to the Clipboard. Excel will mark the copied area with a flashing dashed border.
4. Start or switch to Word, then open the document that will contain the chart.
5. Position the insertion point where you want the chart to appear, then choose Edit Paste Special. The Paste Special box shown in Figure 15.14 appears. The Paste Special box allows you to choose from the three data type options listed below:

   Microsoft Excel Chart Object If you select this data type, you must click on the Paste button. Word will then paste the chart as an embedded object in picture (metafile) format.

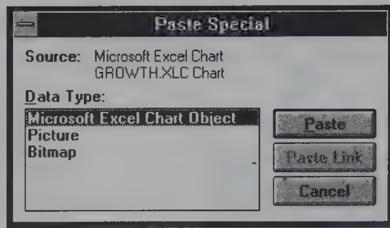


Figure 15.14 These three data type options are available when you paste an Excel chart into a Word for Windows document.

Picture If you select this data type, Word will allow you to use either the Paste or Paste Link button. If you choose the Paste button, Word will insert a static copy of the chart in the Word document and will not create a link to Excel. If you choose the Paste Link button, Word will create a linked object that will be updated whenever you update the chart in Excel. Also, the object will be linked in picture (metafile) format, which allows you to embed TrueType fonts and hinting in a graphic. This format lets you scale a graphic that contains TrueType fonts.

Bitmap If you select this data type, Word will let you use either the Paste (static object) or Paste Link (linked object) button. The only difference between the Bitmap and Picture data types is that a bitmap image does not retain TrueType fonts and hinting. So, if you want to scale the object from within Word, any TrueType fonts that are included in the object will be distorted. You won't have this problem with the Picture data type.

6. To create an embedded object, select Microsoft Excel Object and click on the Paste button. (If you want to create a linked object instead, select either Picture or Bitmap and click on the Paste Link button.)
7. Size the object in Word by clicking the object's frame to select it and then dragging one of the square handles.

You can also edit the object by double-clicking on it. Of course, if you elected to Paste Link the object in Picture or Bitmap format, the object will be linked to the original Excel chart.

View Field Codes in Word to Identify Linked and Embedded Objects

When you link text and data in Word, you'll be able to view it within your Word document. However, if a linked or embedded object

contains graphics, Word will display an empty frame to show the position and size of the object. However, you can quickly identify all objects on a page by turning on Field Codes from the View menu.

Figure 15.15 shows how objects in a Word document appear when the Field Codes option is turned off. Figure 15.16 shows how the same document looks after the Field Codes option is turned on.

A LINK field code denotes a linked object, while an EMBED field denotes an embedded object. The field codes in Figure 15.16 show two linked objects and one embedded object. LINK field codes are displayed in the following format:

```
{LINK ClassName FileName_and_Path [PlaceReference] [\switches] }
```

The ClassName is the registered data type for the object, such as ExcelChart, ExcelWorksheet, MSGraph, and so on. PlaceReference indicates where the item can be found in the server document, such as a range of cells.

Note that the first LINK field code in Figure 15.16 includes a PlaceReference range given in Word's table format (Row1 Column1 to Row17 Column6) rather than in Excel's format (A1:F17). When you link

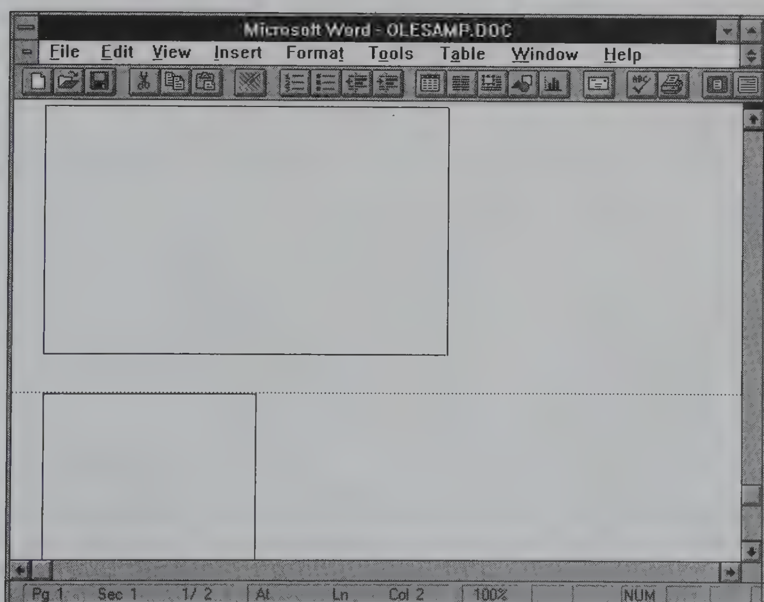


Figure 15.15 Word for Windows displays graphics objects as "empty" frames.

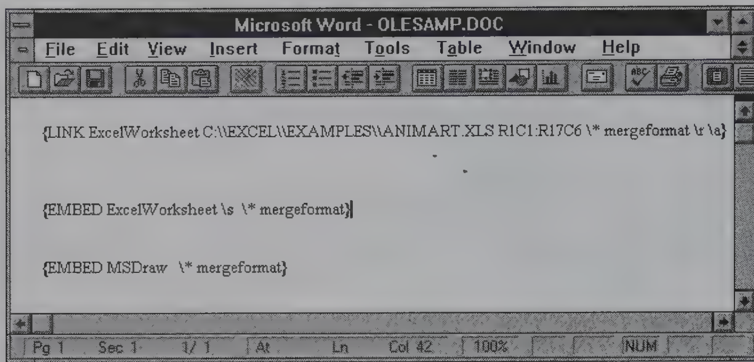


Figure 15.16 When you turn Field Codes on in Word for Windows, you can quickly examine the types of links that exist within a document.

or embed Excel worksheet cells into a Word document, Word automatically converts the worksheet range to its own table format. You can then edit the cells as you would normally edit any other Word table.

The following switches can be used in a LINK field:

\a	Updates the link automatically whenever a change in the server document is made.
\t	Inserts the linked object as unformatted text that can be displayed within a Word document.
\r	Inserts the linked object as formatted text (RTF) that can be displayed within a Word document.
\p	Inserts the linked object in picture (metafile-WMF) format).
\b	Inserts the linked object in bitmap (Paintbrush-BMP) format.
* **mergeformat**	Applies new scaling and formatting whenever changes are made in the server document.

EMBED field codes are displayed in the following format:

```
{EMBED ClassName [\switches]}
```

The following switches can be used in an EMBED field:

\s	Allows you to scale the embedded object from within the client document.
* **mergeformat**	Applies new scaling and formatting whenever changes are made in the server document.

Note: You can edit the contents of a LINK or EMBED field; however, you cannot double-click on the field to open the server application. You must turn off Field Codes to do this.

HOT TIP

Word May Reset OLE Scaling when You Update Word

If you are an experienced Word for Windows user, you know that you can include more than 50 field codes to automatically add information to documents. Word provides field codes for inserting the date and time into a displayed or printed document, for including formulas to make calculations on tables, to insert glossary text, to merge data into a file, and much more. You probably also know that you can update field codes at any time by pressing F9; as an alternative, you can turn on the Update Fields box in the Print Options dialog box so that fields are always updated prior to printing.

When you use either of these techniques to update fields in a Word document, Word will incorrectly reset the size on any embedded objects that use Microsoft WordArt, Microsoft Graph, or Microsoft Equation Editor as the server application. (All three of these "miniapps" are bundled with Word for Windows 2.0.) The size is reset to the original height and width, regardless of any sizing changes you might have made to the objects within Word. (This problem does not occur with the fourth miniapp, Microsoft Draw.)

You can easily prevent this problem from occurring by removing the \s switch from any EMBED field codes for these objects. For example, suppose the following EMBED code exists within your client document:

```
{EMBED MSGraph \s \* mergeformat}
```

To prevent Word from resetting the size for this Microsoft Graph embedded object and any other objects that might be affected, follow these steps:

1. In the Word for Windows View menu, make sure Field Codes is turned on.
2. Find all EMBED codes for Microsoft Graph, Microsoft Equation Editor, and Microsoft WordArt.
3. Delete the \s from each EMBED code. For instance, the Microsoft Graph EMBED code should be edited to look like this example:

{EMBED MSGraph * mergeformat}

Note: Do *not* edit EMBED codes for any other embedded objects, including those for Microsoft Draw.

Example 6: Inserting an Empty Microsoft Draw Object into a Word Document

In this example, we'll demonstrate how to use the Insert Object command to create an embedded object from scratch. We'll create the object in Word, using Microsoft Draw as the server application. Keep in mind that this same basic technique is available in any Windows application that includes the Insert Object command.

Here are the steps to follow:

1. In the Word document, position the insertion point where you want the object to appear.
2. Choose Insert Object. Word displays the dialog box shown earlier in Figure 15.10.
3. Select the Object Type that you want to insert (in this case, Microsoft Drawing), then click on the OK button. Word displays an empty document window, using the server application, as shown in Figure 15.17.

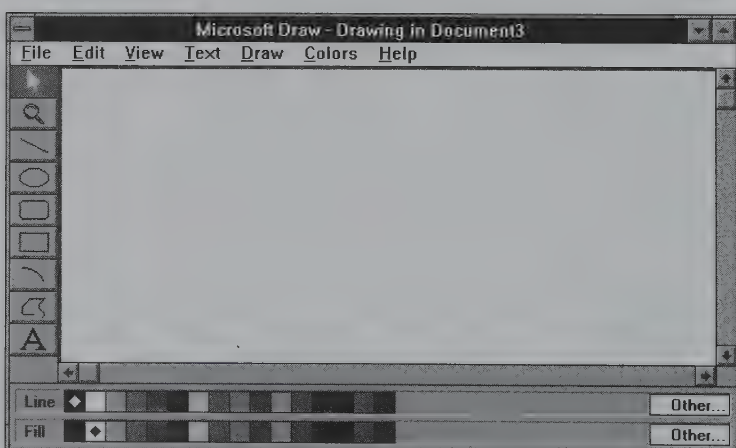


Figure 15.17 When you use Insert Object, the server application displays an empty window that you can use to create the object.

4. Create a drawing, then click on File and select Exit and Return to [filename]. (As an option, you can exit without creating a drawing; you might want to do this to create a place holder frame for a drawing that you plan to create later.
5. When the server application asks you whether you want to update the client document, click on the Yes button.
6. Size and position the object frame.

Note: An object that you embed using the Insert Object command is always embedded in the picture, or Windows metafile (WMF), format. So, TrueType text that exists in your object will be scaled correctly whenever you resize the object.

Combine the Paste and Insert Object Commands to Support "Source Unknown" Objects

You can trick your Windows application into recognizing the server application for an object in the Clipboard, even if the Paste Special dialog box tells you "source unknown." The object must be from a registered OLE server application for the technique outlined below to work.

Occasionally, a Windows application won't be able to recognize the source of an object that you're trying to embed, even though the source is from a registered application. For instance, Word doesn't recognize graphics that have been created in a PowerPoint 3.0 slide. If you want to link or embed a slide, you must switch to Slide Sorter view and then select the entire slide.

But what if you want to create an embedded object in Word using only *part* of a slide's contents? If you switch to Slides view and then copy all or part of the slide to the Clipboard, you'll get a "Source Unknown" message when you try to use Paste Special to embed the object in a Word document, as shown in Figure 15.18.

If you see this message, the client document's application will only let you paste a static copy of the object. You can't create an embedded object because the client doesn't know which application is the server.

You can get around this problem by using the Insert Object command to create an empty object that's linked to the server application. Then, paste the object into the empty object window you've just created. The object is now embedded in your client document.

We'll demonstrate this technique by pasting and inserting part of a PowerPoint slide into a Word document:

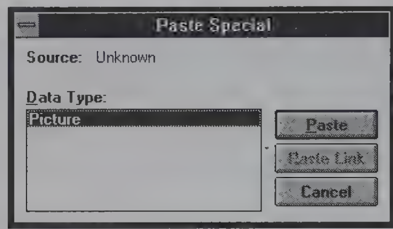


Figure 15.18 A client document will display a "Source Unknown" message if the application can't identify the server application for an object.

1. Start PowerPoint and open a file.
2. Select the portion of the slide you want to paste-link.
3. Choose Edit Copy to copy the selected slide contents to the Clipboard.
4. Switch to the client document (in this case, Word for Windows), then position the insertion point where you want to embed the PowerPoint slide.
5. Choose Insert Object.
6. In the Object Type window, select PowerPoint Slide, then click on the OK button.
7. In the empty PowerPoint document window that appears, choose Edit Paste.
8. Choose File Exit & Return to [filename].
9. Click on the Yes button to update the slide in your client document.

Hot Tip

You Can Create Links on a Network

When Microsoft was developing Windows 3.1, it had announced that the new OLE capabilities would allow network users to create links across the network. Later, Microsoft announced that OLE in Windows 3.1 would not support OLE capabilities between network workstations and the network server. Many people believe this means Windows 3.1 doesn't support OLE at all on a network. However, that assumption is not quite true.

Although you cannot create links between a workstation and the network server, you *can* create links across workstations. However, the applications for both the client document and the server document must be loaded locally, on each workstation that uses the linked data. In other words, if the applications are only located on the network server, you can't create linked and embedded objects on workstations. But if all of the applications required to create a link are loaded on each workstation, links can be formed across workstations.

This distinction confuses many people, in part because Microsoft uses the word "server" in a context that differs from the network use of the word "server." Keep in mind that a network server is the actual file server that provides network services to all workstations. By contrast, "server documents" and "server applications" are Microsoft OLE terms, and are specific to OLE—*not* to the network.

Creating Complex Documents

Because a linked or embedded object always contains a link to its server application, you can easily create links within links. For instance, when you link or embed an Excel object into a Word document, Word starts the server application (Excel) whenever you want to edit the object. Therefore, you are actually working with the server application at this point. You can then link or embed an object into the opened document. At this point, you've inserted an object within an object by chaining one link to another.

There are only two limitations to this capability: First, the client and server applications both must be able to support the type of links you want to create. Second, your system must have enough memory to support the links. Memory becomes an especially important issue here because the original client document might have to start several server applications and several server documents simply to support a single object in the client.

Example 7: Linking an Object to Another Object

In this example, we'll demonstrate how to create complex links by first linking an Excel worksheet to a PowerPoint slide, then embedding the slide into a Word document.

Use the following steps to create a linked object in PowerPoint and then embed the object in Word:

1. Start Excel, then open a worksheet file.
2. Select the range to be copied, then choose Edit Copy.
3. Start or switch to PowerPoint, then open a file.

4. Choose Edit Paste Special.
5. In the Data Type box, select either Microsoft Excel Worksheet Object or Picture, then click on Paste Link.
6. Position and size the worksheet to appropriately fit within the slide. The Excel worksheet is now linked and positioned within the PowerPoint slide.
7. Choose View Slider Sorter, then select the slide.
8. Choose Edit Copy to copy the slide to the Clipboard.
9. Start or switch to Word.
10. Open a document, then position the insertion point where you want the slide to appear.
11. Choose Edit Paste Special.
12. In the Data Type box, select either PowerPoint Slide Object or Picture, then click on the Paste button.
13. Size and position the object.

Figure 15.19 shows what the object looks like when we Print Preview the page in Word. When we embedded the slide in Word, we did not make a link between the original PowerPoint slide and Word (although we could have kept this link by creating a linked object rather than an embedded object). However, a link still exists between the *linked object* in the PowerPoint slide (the worksheet portion) and the original Excel

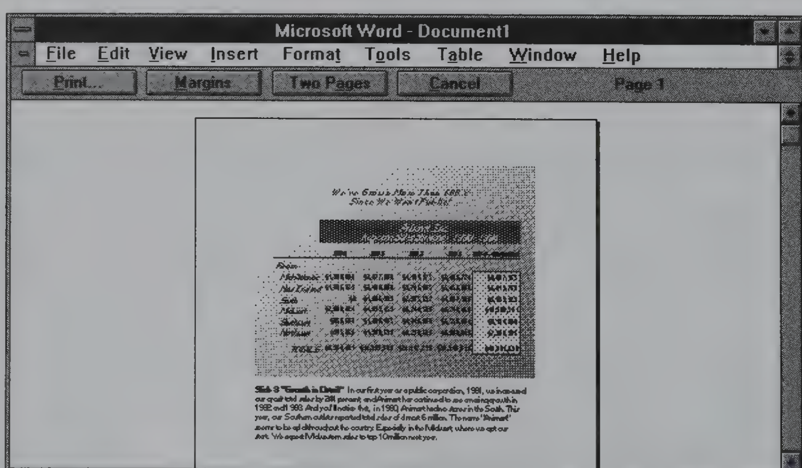


Figure 15.19 This PowerPoint slide embedded in a Word document also contains a linked object that's linked to an Excel chart.

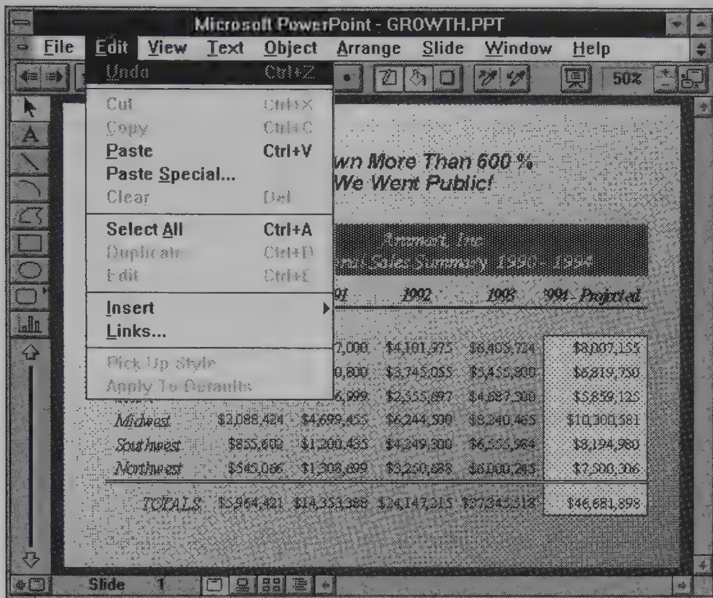


Figure 15.20 Linked objects exist in a document if the Links option is available for selection.

chart. Therefore, the slide in Word will always be updated whenever we modify the worksheet in Excel.

We can verify that this updating will take place by double-clicking on the object to open the slide and start PowerPoint. Then, we can display the Edit menu to see if the Links option is available (not grayed), as we've done in Figure 15.20. Since the Links option is available for selection, we know that the slide still contains a linked object. You can verify this further by clicking on Links and then examining the contents of the Links dialog box.

In our example, we embedded an object that contained a linked object. However, we could have used any linking/embedding combination, and in several different orders. For example, we could have:

- Embedded the slide into the Word document first, then opened the object for editing and then linked the Excel worksheet into the embedded slide.
- Embedded the slide into the Word document first, then opened the object for editing and then *embedded* the Excel worksheet into the embedded slide.

- Linked the slide to the Word document, then embedded the worksheet into the slide (either by switching to PowerPoint or by double-clicking on the object within Word).
- Linked the slide to the Word document, then linked the worksheet into the slide (either by switching to PowerPoint or by double-clicking on the object within Word.)

Of course, we could even link or embed an object within an object that's linked or embedded *within still another object*, and then link or embed *that object* into another object.

We won't tax your patience by demonstrating these additional levels of chained links. Just keep in mind that chaining links is limited only by your system's memory. If you get an "insufficient memory" message when you try to chain linked objects and embedded objects, you'll have to abort your attempt or consider adding more memory to your system.

Using the Registration Database

The registration database (REG.DAT stored in the WINDOWS directory) contains all of the information that the OLECLI.DLL and OLESRV.DLL libraries use to support OLE capabilities for applications. Typically, a Windows application must register itself in the database when you use the application's setup program to install the application itself. However, Windows 3.1 will register some Microsoft-supplied applications automatically.

In any event, an application can use the OLECLI.DLL and OLESRV.DLL libraries only if it is registered in the Windows 3.1 registration database. Therefore, you should be familiar with the registration database in case a vendor provides you with registration information that you must add yourself or if you have to reinstall or repair the registration database.

The SHELL.DLL library also adds information to the registration database so that the File Manager can identify file associations and drag-and-drop facilities provided by applications. In Chapter 8, we explained how to use the registration database to register these features, so we won't dwell on basic registration editor features here. However, we will take the opportunity to explain some techniques for using the registration database to support your Windows applications' OLE capabilities.

Installing an Application's REG File

A Windows application's REG file is typically added when you install the application itself. However, there are two situations where you might need to add a REG file yourself: if the original REG file gets deleted accidentally, or if the vendor supplies a REG file separately from the packaged application.

If you need to install or reinstall a REG file for an application, follow these simple steps.

1. In the Program Manager or File Manager Run dialog box, type **REGEDIT**, then press **Enter** to start the Registration Info Editor, shown in Figure 15.21.
2. In the Registration Info Editor, choose File Merge Registration File.
3. When the Merge Registration File dialog box appears, select the directory and REG file for the desired application, then click on the OK button.

HOT TIP

Install a REG File from File Manager

You can install an application's REG file by starting File Manager, locating an application's REG file, then double-clicking on the file. Windows will automatically install the file's registration information in the registration database.

Repairing the Registration Database

If your registration database (REG.DAT) has been deleted or if you suspect that the database is corrupted, you can reconstruct it easily.

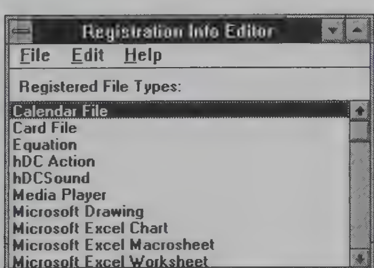


Figure 15.21 Use the Registration Info Editor to add or reinstall a REG file.

Here's how to reconstruct the registration database:

1. Exit Windows.
2. At the DOS prompt, switch to the WINDOWS directory and delete the file REG.DAT.
3. Restart Windows.
4. In the Program Manager or File Manager Run dialog box, type **REGEDIT**, then press **Enter** to start the Registration Info Editor.
5. Choose File Merge Registration File.
6. In the Directories list, choose the WINDOWS\SYSTEM directory.
7. In the File Name list, select SETUP.REG, then click on the OK button. Windows will reconstruct the registration database for all Microsoft applications that provide REG files.

8. If you have non-Microsoft applications that provide REG files, you'll have to add them by merging them separately into the registration database, as we've described in the previous topic.

HOT TIP

Make a Backup Copy of REG.DAT

You can provide yourself some added protection and save time by making a backup copy of your REG.DAT file. Then, if your current REG.DAT file is deleted or corrupted, you can restore the database by copying the backup version into the WINDOWS directory. This technique will prevent you from having to merge REG files for all non-Microsoft applications that support OLE, and will provide a faster way to restore your registration database than reinstalling SETUP.REG.

Part Overview

PART 6

Troubleshooting Strategies

These final two chapters describe many commonplace problems that occur when you run Windows 3.1 with various software and hardware, and explain how to resolve or "workaround" these problems. Chapter 16 explains how to troubleshoot problems related to hardware devices—including memory-related problems and problems with device drivers. Chapter 17 explains how to troubleshoot problems related to specific applications that you are running under Windows.

Obviously, we can't describe every problem that can occur with Windows. ("I've got an XTY SCSI disk controller, an Amazon version 30 BIOS, and an Olympic KT4000 laser printer and everything works fine in Windows unless I try to print half-page greeting cards with 8-point type in color duplex mode; how do I prevent the random green characters that occur under this setup?") Instead, we've tried to identify the major problems that occur with the most widely used hardware and software.

CHAPTER

16

Hardware Problems

In this chapter, you will learn about common Windows-related problems that are directly related to hardware devices and device drivers. Chapter 17 deals with problems you might have when running specific applications under Windows. Although we try to make a clear distinction between these two chapters, the division is somewhat artificial. Solving Windows problems is often a tricky and complex business. At times, a software problem can lead to unpredictable or faulty behavior in one or more hardware devices. In other situations, it's the hardware device or the driver for the device that creates problems for Windows or particular applications.

Also included in this chapter are products that directly control or modify hardware configurations, such as memory managers and the SMARTDrive disk cache. We'll also discuss general TrueType issues here. In turn, if an application causes a problem that leads to unreliable or disastrous behavior in a hardware device, we'll explain the problem and suggested workarounds in Chapter 17. TrueType problems that are related to specific applications are discussed in Chapter 17.

Using the Workarounds

Troubleshooting tips and techniques (workarounds) for many hardware devices, drivers, and other utilities are included featured in this chapter. Keep in mind that the workarounds provided in these two chapters are *suggestions only*. These workarounds come from many sources—including Microsoft, software and hardware vendors, end users, and from our own troubleshooting experience. The usefulness of the workarounds will vary for different system configurations. We cannot guarantee the success or results of any workaround on your system, so please back up files (and your entire hard disk) before you make any modifications.

As you review this chapter and Chapter 17, you'll likely notice that many problems stem from outdated applications, drivers, and hardware designed to support an earlier Windows version (such as Windows 3.0 or the Windows 3.0a maintenance release). Most software and hardware vendors have, by now, created updated applications or drivers to work with Windows 3.1. However, through no fault of your own, you might still be using applications and drivers that are not 100-percent compatible with Windows 3.1.

As a general rule, you should check with the vendors for your hardware devices and your applications to find out whether updated programs or drivers are available for Windows 3.1. Chapter 13 explains how to access updated video and printer drivers from the Windows Download Library (WDL).

HOT TIP

Read README.WRI

When you install Windows 3.1, the Windows Setup program adds a file called README.WRI into your WINDOWS directory. This file contains a wealth of information about drivers, hardware devices, memory managers, and other utilities that can cause problems for Windows 3.1. README.WRI even provides troubleshooting information for several non-Windows applications. Use Write or any text editor to open and read this file. You might also find it helpful to print README.WRI so that you have a hard-copy troubleshooting reference for hardware-related Windows problems.

Many of our suggested workarounds involve using the Windows Setup program in maintenance mode to modify or update the files that support Windows 3.1. Maintenance-mode setup, distinct from the initial Windows Setup program that installed Windows on your system is available at all times from within Windows. In the maintenance mode, the Windows Setup program lets you make additional changes to the Windows environment. To use maintenance-mode setup, choose the Windows Setup icon from the Main group—*after* Windows 3.1 has been installed.

Finally, we've limited the contents of this chapter and Chapter 17 to problems that occur *after* Windows has been installed. If you are having problems installing Windows or want more information before you install Windows, please refer to Appendix A. This appendix also includes information on installing Windows on various networks.

HOT TIP

Troubleshooting General Protection Faults

General Protection (GP) faults typically occur because two applications try to write to the same memory address. The possible reasons for this conflict are numerous, which makes it difficult to diagnose GP faults. Microsoft Product Support Services provides an application note that you can use to help diagnose GP faults. You can obtain this application note by calling Microsoft Product Support Services at (206) 454-2030 or by searching for articles QWW0524 or S13364 in the Microsoft Software Library on CompuServe (GO MSL).

Video Cards, Video Drivers, and Display Problems

This section describes problems that result due to video cards and video drivers that don't work or cooperate well with Windows 3.1. Whenever you have a problem in Windows 3.1 that seems to affect your screen display, you should suspect a driver. In many—but certainly not all—cases, display problems result because your video driver was designed to run under Windows 3.0. And even though you believe the Windows 3.1 Setup program updated your video driver, you might be wrong. There are several situations where the Windows 3.1 Setup program can't or won't update older video drivers. Many updated video drivers are available in the Windows Driver Library (WDL). (Refer to Chapter 13 for information on accessing the WDL.)

HOT TIP

Read SETUP.TXT

When you installed Windows 3.1 on your system, the Windows Setup program copied a file called SETUP.TXT into the WINDOWS directory of your hard disk. Although this file is intended mostly as a troubleshooting document for installing Windows 3.1, it contains some extremely useful information regarding specific video cards and video drivers that are incompatible with Windows 3.1. Use Notepad to open this file, then look for the section titled *Displays That Are Incompatible with Windows 3.1.* The README.TXT file also contains numerous suggestions for configuring video cards to run under Windows 3.1. These suggestions and troubleshooting tips are located in Section 9 of the README.TXT file.

CompuAdd 325TFX and 325NX Laptops: DOS Applications Cause System to Crash

Windows 3.1 upgrades the video driver for these laptops using the standard VGA driver. However, this driver can cause the system to hang when you run a non-Windows application.

This problem manifests itself typically by displaying a DOS application in reverse video before crashing the system.

Suggested workaround: Run Windows Setup in maintenance mode and install the video driver labeled "VGA (Version 3.0)".

Quadram VGA Cards: Windows Behaves Erratically

Video cards that are compatible with the Chips & Technologies 441 chip set (which includes Quadram cards and some ATI and Chips & Technologies display cards) can use the QuadVGA Windows 3.1 video driver. However, with this driver you cannot display the MS-DOS Prompt application in a window, and the Alt+Tab task-switching feature can corrupt the screen.

This problem also existed in Windows 3.0.

Suggested workaround: Run Windows Setup and install the "VGA (Version 3.0)" driver to improve video card performance. However, even with this driver, you might have to reset the screen resolution occasionally by typing the command **MODE CO80** at the DOS prompt.

HOT TIP

The Windows Driver Library Contains Several Updated Video Drivers

The Windows Driver Library (WDL) is available on CompuServe, GEnie, and from Microsoft. As of August 7, 1992, the following new or updated video drivers have been included in this library:

Display Type	*Download This File*
Appian RGDI and compatibles	RGDI.EXE
ATI 8514/Ultra	ATIULT.EXE
CGA Driver	CGA.EXE
DGIS 3.0 and compatibles	DGIS30.EXE
S3 and compatibles	S3.EXE
Tseng ET4000 16 color	TSENG1.EXE
Tseng ET4000 256 color	TSENG2.EXE
Tseng ET4000 32 color	TSENG3.EXE
Tseng ET4000 64K color	TSENG4.EXE

Refer to Chapter 13 for additional information on accessing the WDL. This library is updated regularly by Microsoft, so you might want to check to see whether additional video drivers have been included.

Video Seven 512K Driver: Not Updated by Windows 3.1

If you installed Windows 3.0 using version 3.0a, and if you also installed the Video Seven 512K driver, the Windows 3.1 setup

program will not update the driver if you select the Video 7 512K, 640x480 256 colors driver option during the Windows 3.1 installation.

The Windows 3.1 setup program supplies the V7VGA.DRV driver file, dated 3/10/92 to support Video Seven and compatible cards with 512K of memory. This driver is not installed if you select the 640x480 256 colors driver during setup.

Suggested workaround: Run Windows Setup and install the Video Seven driver again. When you tell Windows Setup to install the Video 7 512K, 640x480 256 driver, Windows will ask whether you want to use the existing driver or a new driver. Make sure you install the new driver file.

HOT TIP

All Windows 3.1 Files Are Dated 3/10/92

All files on the Windows 3.1 installation disks are dated 3/10/92, including all printer and display drivers. You can check to make sure you are using the most current driver by using the File Manager to display a list of all .DRV files in the WINDOWS\SYSTEM directory. If the date for any .DRV file is earlier than 3/10/92, it is either a Windows 3.0 driver or a driver installed by a specific Windows application or hardware setup program and created prior to the release of Windows 3.1. You should check with Microsoft or the original vendor of the driver to see if a more current driver version is available.

TrueType: Displays Distorted Characters for Some Fonts at Certain Sizes

You might notice that some TrueType characters appear distorted on screen. This problem only occurs for certain characters in certain fonts, and at certain sizes.

This problem apparently stems from the TrueType engine's inability to display hinting information correctly in some cases.

Suggested workaround: Do nothing. TrueType characters will print correctly even if they display incorrectly.

Video Modes: DOS Window Locks with Some DOS Applications

If you can run some DOS applications in a window but other DOS applications cause an error message to be displayed, your video

card might not support the video mode that the application is trying to use.

Either of the following two messages indicates an unsupported video mode:

- You cannot run this application while another high-resolution application is running.
- Insufficient Memory.

Suggested workaround: Run these DOS applications in full-screen mode.

HOT TIP

Exclude the UMA Used by VGA Cards

Windows Setup can accurately identify many VGA cards and will prevent Windows from trying to access the portion of upper memory that the card uses to support video operations. However, Windows might not detect some newer, enhanced VGA cards and therefore may try to use a region in the UMA required for your video driver. If you suspect that Windows 3.1 is crashing in enhanced mode due to video memory conflicts, force Windows to exclude the area of the UMA reserved for VGA use. To do so, add the following line to the [386Enh] section of SYSTEM.INI:

```
EMMEXCLUDE=C400-C7FF
```

Memory Management and Memory-Related Problems

Memory management is probably the most difficult area to troubleshoot for Windows 3.1, chiefly because it's difficult to know whether a particular application or driver conflicts with Windows' memory management or whether the application has more general problems using Windows 3.1 features. In this section, we'll focus on Windows-related problems that directly stem from the way memory managers and other utilities use conventional, expanded, and extended memory.

386MAX: Enhanced Mode Windows Does Not Operate Properly

If you use the EXCLUDE= parameter in 386MAX (or BlueMAX) to exclude a range of upper memory, Windows 3.1 might not operate correctly in enhanced mode.

This problem occurs if you exclude ranges above A000.

Suggested workaround: Exclude only ranges below A000, or avoid using the EXCLUDE= parameter.

386MAX: Windows Hangs If You Attempt to Start ASQ

If you are running Windows in standard mode and if your 386MAX Device=386MAX.SYS line (in CONFIG.SYS) contains the EMS=0 parameter, the ASQ memory viewer will hang.

The EMS=0 parameter tells 386MAX not to allocate an EMS page frame in upper memory. 386MAX allows you to run Windows in standard mode even if the EMS=0 parameter is present in your CONFIG.SYS file. However, the ASQ utility will hang if you try to run it from Windows under these conditions.

Suggested workaround: There are two possible workarounds. The most effective solution is to exit Windows before you run ASQ. You will get a more accurate view of memory when you run ASQ outside of Windows. However, if you want to run ASQ while Windows is running in standard mode, replace the EMS=0 parameter with **NOFRAME**, which produces the same results as EMS=0. (You must reboot your computer for any CONFIG.SYS changes to take effect.)

DOS Extenders: DOS Applications Crash or Behave Erratically in Standard Mode

Many DOS applications use their own DOS extenders to make extended memory available to the application. If you are using a memory manager that can emulate EMS memory, these DOS applications might not be able to run in standard mode.

Most memory managers—including EMM386, 386MAX, and QEMM-386—can emulate expanded memory. The memory manager does not have to create an EMS page frame for this problem to occur.

Suggested workaround: Either run these problematic DOS applications in enhanced mode or remove the memory manager to allow the applications to run in standard mode.

The Maximizer: Incompatible with Windows 3.1

At present, The Maximizer memory manager causes the Windows Setup program to fail and will prevent Windows from running if Maximizer's upper memory support is enabled.

The Maximizer memory manager is similar in concept to EMM386, 386MAX, and QEMM-386; it helps remove conventional-memory overhead by configuring the upper memory area and loading drivers and TSRs there. However, Windows will fail to run in either standard or enhanced mode if The Maximizer has configured the UMA.

Suggested workaround: If you want to configure the UMA to load device drivers and TSRs, disable The Maximizer and use EMM386 (supplied with Windows 3.1) or a Windows-compatible memory manager.

QEMM-386: Only 8MB of Memory Available in Standard Mode

If QEMM-386 is running and you try to run Windows in standard mode, Windows will not be able to use more than 8MB of memory, even if your PC has additional RAM available.

The standard-mode limitation exists for QEMM-386 versions 6.0 and earlier.

Suggested workaround: The only way to access RAM above 8MB with QEMM-386 is to run Windows in enhanced mode. Most of the memory-management benefits of QEMM-386 aren't relevant in standard mode. Therefore, you should consider disabling QEMM-386—at least temporarily, if you must run Windows in standard mode and need access to more than 8MB of RAM.

SMARTDrive 4.0: Windows Crashes when Double Buffering Is Disabled

SMARTDrive can cause Windows to crash if double-buffering is required for your system but isn't enabled.

Double buffering is used to keep track of both physical and virtual memory addresses for systems that implement a feature called *bus mastering*. Double buffering prevents DOS from using the wrong address in reading and writing cached data for virtual machines. Double buffering uses 2.5K of conventional memory and causes a slight decrease in performance.

If SMARTDrive detects that your system requires double buffering but you have disabled this feature, your system might behave erratically or crash. You might have disabled this feature accidentally if you removed the device= line for SMARTDrive in your CONFIG.SYS file, believing that SMARTDrive 4.0 only needs to start from AUTOEXEC.BAT.

Suggested workaround: At the DOS prompt, type **SMARTDRV** and read the "buffering" column. If one or more of the entries in this column is a Y or a dash, SMARTDrive has identified a disk controller that requires double buffering. To enable double buffering, add the following line to your CONFIG.SYS file and then reboot your system:

```
DEVICE=C:\WINDOWS\SMARTDRV.EXE /DOUBLE_BUFFER+
```

SMARTDrive 4.0: Two Megabytes of Memory Are Insufficient to Start Windows in Enhanced Mode

When you try to run Windows in enhanced mode, SMARTDrive 4.0 creates a 256K cache if your system has 2MB of memory (1MB conventional and 1MB extended). Under this configuration, Windows will not have the minimum amount of memory required to start in enhanced mode.

To run in enhanced mode, Windows must have access to at least 1.6MB of memory (640K of conventional plus 1MB of extended memory). Windows does not check the upper memory area to determine whether enough memory exists to start in enhanced mode. When SMARTDrive uses its default cache value for Windows (in this case, 256K used in extended memory), the minimum 1.6MB of memory for enhanced mode is not available.

Suggested workaround: There are three possible solutions. Of course, the most effective solution is to add extended memory to your system.

If you have a relatively large amount of hard disk space that can be used for swap files (virtual memory), you can force Windows to run in enhanced mode by typing **WIN /3** at the DOS prompt. Expect system performance to be slow.

If you want to provide more extended memory for use by Windows in enhanced mode, reduce SMARTDrive's "cache size while running Windows" to 128K or less—until Windows will start in enhanced mode without the /3 switch. You must reboot your system for any SMARTDrive changes to take effect.

HOT TIP

Monitoring and Controlling SMARTDrive Performance

You can easily check to make sure SMARTDrive is caching your hard drive. Simply start the Control Panel, then immediately close it. Open the Control Panel again and observe your hard disk drive light. If the light does not go on, you know that the Control Panel program has been cached in RAM; therefore, Windows did not need to read the program from your hard disk on the second access. (The Control Panel should also open very rapidly on the second access.) If the drive light *does* go on, the SMARTDrive cache might not be enabled or might be caching the wrong drive.

You can also check the ongoing status of SMARTDrive. Use the MS-DOS Prompt application to display the DOS prompt, then type **SMARTDRV /S**. SMARTDrive should display a message similar to the following one:

```
There have been 4,018 cache hits and 684 cache misses
```

A cache hit occurs when Windows or another application requests data from disk and SMARTDrive finds it in the cache instead. A cache miss occurs when SMARTDrive cannot find requested data in the cache and the data must be read from the hard disk. You can check SMARTDrive's performance by using SMARTDRV /S periodically during your Windows session. You should notice through the course of the day that the number of cache hits remains substantially higher than the number of misses.

Strangely, on some systems SMARTDrive will display its help screen when you type **SMARTDRV /S** from within a DOS window. If this happens, exit Windows, then type **SMARTDRV** (with no switches) at the DOS prompt. Then restart Windows, start the MS-DOS Prompt application, and type **SMARTDRV /S**. You should now see the appropriate cache status information.

Stack Overflow Messages

If your receive the message "INTERNAL STACK OVERFLOW. SYSTEM HALTED" before your system locks, the number of stacks specified in your CONFIG.SYS file might be too low.

The STACKS= line in CONFIG.SYS sets up additional stack structures in memory to prevent stack overflows that can occur when some applications hook into certain hardware interrupts. Memory managers, SMARTDrive, and TSRs and device drivers (especially network drivers) can trigger stack overflows by trying to add more data to a stack than the stack has been set up to store.

The STACKS= line can take two parameters: *STACKS=x,y*, where *x* is the number of stacks and *y* is the size, in bytes, for each stack. On most systems—true IBM PCs are notable exceptions—Windows sets up nine stacks that are 256 bytes in size (STACKS=9,256). This configuration is adequate for most but not all systems.

Suggested workaround: To avoid this error message and the resulting system crash, increase the number of stacks to 10 or more. (However, if your STACKS= statement is set at 0,0 or is missing from CONFIG.SYS, first set it to 9,256 to see if this corrects the problem.) You might have to experiment with this setting to find the most effective one.

HOT TIP

Increase the Stack Size to 512

If your system behaves erratically—crashing at odd times, especially when you are running DOS programs—try increasing the stack size from 256 bytes to 512 bytes. This is the maximum stack size. With nine stacks set up, this increases conventional memory overhead by only 2K, but might allow your DOS programs to successfully add data to stacks.

VRAM386.SYS and HRAMDEV.SYS: Incompatible with Windows 3.1

Both of these memory managers emulate expanded memory in a manner similar to EMM386. They are incompatible with Windows 3.1 and will cause Windows 3.1 to hang.

If either of these memory managers is installed, the Windows Setup program will hang. If you disable one of these managers to allow Windows Setup to install Windows 3.1, then enable the manager, Windows will hang when you try to start it.

Suggested workaround: Use EMM386 instead of these managers. At this writing, you cannot use these memory managers with Windows 3.1. Contact the vendors for updated versions.

WINA20.386: Windows Reports That the WINA20 Handler Is Missing

Windows 3.0 required the WINA20.386 file in order to run under DOS 5. This file is not required with Windows 3.1; any WINA20 error message that appears when you start Windows 3.1 is usually irrelevant.

The WINA20.386 file was used by Windows 3.0 to avoid clashes with DOS 5 over the use of the high memory area (the first 64K of extended memory). A WINA20 error message is most likely to appear in Windows 3.1 if you are using a version of QEMM-386 earlier than 6.0, and if the NOEMS parameter is used with QEMM-386.

Suggested workaround: Since WINA20.386 is not needed by Windows 3.1, the error message is incorrect. Therefore, you can do nothing. Windows 3.1 will operate properly. However, you can prevent this message from displaying by creating an "empty" WINA20.386 file in the root directory of your hard drive. Also, you should consider upgrading your QEMM-386 memory manager to version 6.0 or higher, since these versions work more reliably and efficiently with Windows 3.1.

Disk Controller Problems

This section explains Windows-related problems that stem directly from procedures used by disk-controller cards and disk drivers. Many of the disk-related problems you will encounter with Windows 3.1 involve the misuse of SMARTDrive 4.0 or the 32-bit disk access (FastDisk) feature of Windows 3.1. In any event, disk-related problems often occur because Windows incorrectly uses or intervenes in the operation of the disk controller.

DISCTEC Removable Storage Drive: Does Not Support 32-Bit Disk Access (FastDisk)

DISCTEC creates a removable hard disk drive by using an interface that makes Windows 3.1 think these drives are compatible with Western Digital 1003. However, they are not.

The DISCTEC hard disk drive interface installs a disk driver file in CONFIG.SYS. This driver reports information about removable disks to DOS. For this reason, the Windows FastDisk utility cannot fully control 32-bit disk access, and therefore may cause system failure.

Suggested workaround: Disable 32-bit disk access. (See Chapter 4 for the appropriate steps to follow.)

HOT TIP

SMARTDrive 4.0 Corrupts Files

Some users have reported that files on their hard disk are corrupted only when SMARTDrive is enabled, and specifically when the write-behind caching feature is enabled.

SMARTDrive 4.0, unlike earlier versions of SMARTDrive, can support both read caching and write-behind caching. Normally, when you install SMARTDrive 4.0, it detects whether your hard disk can safely support write-behind caching. However, there is some evidence that SMARTDrive enables write-behind caching on some disks that cannot safely support this feature.

If you suspect that files on your hard disk are being corrupted, try disabling the write-behind feature of SMARTDrive. To do so, add the drive letter (for each cached drive) to the SMARTDRV line in your AUTOEXEC.BAT file. When you specify a drive letter alone (without the + sign), SMARTDrive will operate with a read-only cache. Here's an example showing how to disable SMARTDrive if your C: drive is the cached drive:

```
SMARTDRV.EXE 1024 512 C
```

There have been other reports that 32-bit disk access causes disk timeout errors only when SMARTDrive with write-behind caching and the DOS SHARE utility are *both* in use. If this problem occurs, try removing SHARE from your AUTOEXEC.BAT file. If you must use SHARE (because one or more of your Windows applications require it), try forcing SMARTDrive to load in conventional memory rather than upper memory by adding the /L switch at the end of the SMARTDrive line in AUTOEXEC.BAT. If this does not solve the problem, disable SMARTDrive's write-behind cache. If this still does not work, turn off 32-bit disk access.

Columbia Data Products SCSI Hard-Disk Controllers: Will Not Support Windows 3.1 in Enhanced Mode

If you are using a driver for these controllers that is older than version 3.35, Windows 3.1 will run correctly in standard mode only.

Early versions of the hard-disk driver for this controller are not compatible with Windows 3.1.

Suggested workaround: To run Windows 3.1 in enhanced mode, upgrade the SCSI driver to at least version 3.35.

Plus Hardcard: Will Not Support Windows 3.1 in Enhanced Mode

Windows Setup detects systems that will not operate correctly in enhanced mode if Windows is allowed to terminate interrupts from the hard disk controller. However, Windows may enable this enhanced-mode feature for Hardcards. Older Hardcards will not operate correctly in enhanced mode with this feature enabled.

This problem occurs with Hardcards manufactured prior to the Hardcard II.

Suggested workaround: Open SYSTEM.INI and look for the following line in the [386Enh] section:

```
VirtualHDIRQ=[On or Off]
```

This entry must be set to Off. If the line does not appear in the [386Enh] section of SYSTEM.INI, add it, as follows:

```
VirtualHDIRQ=Off
```

You do not need to take these steps with the newer Hardcard II or Hardcard II XL devices.

Mouse Problems

This sections describes problems you might encounter using your mouse in Windows. Many mouse-related problems that involve the use of DOS applications are covered in Chapter 14. We won't reiterate these problems here. In general, though, keep in mind that most mouse problems stem from the use of a mouse driver that is not compatible with Windows 3.1 (even though it might work well with Windows 3.0).

Kurta Mouse: Windows Hangs on Startup

If the Kurta mouse driver is installed for use with Windows 3.1, Windows will hang during startup.

The Kurta mouse driver file can be specified in the mouse.drv= line (in the [boot] section of SYSTEM.INI). This effectively tells Windows to install the driver for use with Windows. Although this works in Windows 3.0, it will not work in Windows 3.1.

Suggested workaround: Contact the manufacturer to obtain an updated Windows 3.1 mouse driver.

BallPoint Mouse: Mouse Pointer Will Not Move after Power-Up

Some portable computers include a PS/2 bus mouse port that supports the Microsoft BallPoint Mouse. These computers often use a battery-saving routine that powers down the mouse port when it isn't in use, and then powers up the mouse when it is moved. If so, the mouse pointer might be frozen against the left edge of the screen when power is restored to the mouse.

The driver for the BallPoint Mouse port is a virtual device driver called BPVKD.386. However, Windows 3.1 also hard-codes support for this port into the Windows program.

Suggested workaround: Since you cannot use a different mouse driver for the BallPoint Mouse, you must obtain an updated PS/2 mouse port BIOS in order for the mouse to be supported correctly.

DOS Windows: Mouse Won't Work Even with an Updated Windows 3.1 Mouse Driver

You have installed a mouse driver that you know is designed specifically to support Windows 3.1, including DOS windows in Windows 3.1, but the mouse still won't work in any DOS window.

This problem typically occurs because the video driver in use is not 100-percent Windows compatible. By default, if Windows 3.1 detects a video driver that it perceives to be less than 100-percent compatible with Windows, it will err on the side of safety and prevent mouse operations in DOS windows.

Suggested workaround: Use Notepad or the System Configuration Editor to open SYSTEM.INI. Find the [NonWindowsApp] section, then add the line **MouseInDosBox=1**, save the file, then reboot Windows. If your DOS windows display erratically or cause the DOS application to hang, you probably need to update your video driver. Contact the manufacturer of your monitor to see if an updated driver is available.

MicroSpeed's FastTrap Mouse: Will Not Work in Three-Button Mode

MicroSpeed's FastTrap three-button mouse will only work in Windows 3.1 in two-button mode.

The existing drivers for the FastTrap mouse are not compatible with Windows 3.1. However, the FastTrap mouse is compatible with the Microsoft mouse protocol, which does not support a third button.

Suggested workaround: Run Windows Setup in maintenance mode, then select the "Microsoft, or IBM PS/2" mouse driver. Then, apply the /D switch to either the mouse driver line in either the CONFIG.SYS or AUTOEXEC.BAT file (depending on where the MicroSpeed mouse driver is installed for use with DOS).

In CONFIG.SYS, your device line should be similar to the following:

```
DEVICE=C:\MAP.SYS /D
```

In AUTOEXEC.BAT, your device load line should look similar to the following:

```
DEVICE=C:MAP.COM /D
```

You should also check with the vendor to see if a more current Windows 3.1 driver is available for this mouse.

Mouse Systems' Three-Button Mouse: Does Not Work in Windows 3.1

The Mouse Systems Trackball II, M4, and M5 three-button mice all can behave erratically under Windows 3.1. Specifically, the mouse will act as though buttons are still pressed even after you have released them.

By default, these mice will only work in Windows 3.1 in two-button mode.

Suggested workaround: Find the toggle switch on the bottom of the mouse, then switch it to two-button mode.

Printers and Printer-Driver Problems

In this section, we'll describe problems that can be directly attributed to specific printers and printer drivers. Some printing errors, though, stem from problems in specific Windows applications. Since the culprits of these problems are application related, you should also examine Chapter 17 to locate their potential causes and solutions.

As you review this section, you'll notice that most printer problems stem from the use of an obsolete printer driver. Although the Windows 3.1 setup program updates most printer drivers, there *are* exceptions. We won't repeat printer-driver information that has already been explained in Chapter 13, so you should review the material there as well to determine whether your printer driver is not current for Windows 3.1. Chapter 13 also includes information on obtaining updated drivers from the Windows Driver Library (WDL) as well as techniques for adding or reinstalling printer drivers.

HOT TIP

Read PRINTERS.WRI

When you installed Windows 3.1 on your system, the setup program copied a file called PRINTERS.WRI into your Windows directory. This file contains information that you can use to troubleshoot printing problems for specific printers. Since the information in this file is available to all Windows users, we haven't included most of it in this book. If you need additional help in resolving printer problems, use Write to review the PRINTERS.WRI file.

Canon LBPIII and LBP-4 Printers: Windows 3.1 Drivers Are Not Installed

The Windows Setup program will not install the Windows 3.1 printer driver (LBPIII.DRV) for these two printers if you are upgrading from Windows 3.0.

The Windows 3.0 driver for the Canon LBPIII and LBP-4 printers will remain after you upgrade to Windows 3.1. This problem does not occur for other Canon printers, including the LBP-8 II.

Suggested workaround: Since the Windows 3.0 and Windows 3.1 drivers for these printers share the same name (LBPIII.DRV), you must first delete this file from the WINDOWS directory. Then, use the Printers dialog box (accessed from the Control Panel) to add the Windows 3.1 printer driver for the appropriate printer.

Fuji-Matrix Printers: Windows 3.1 Drivers Are Not Installed

The Windows Setup program will not install the Windows 3.1 printer drivers for Fuji-Matrix printers if you are upgrading from Windows 3.0.

Windows 3.0 used a single driver for both 9-pin and 24-pin Fuji-Matrix printers. Windows 3.1 supplies separate drivers for these two categories. Since Windows has no way of determining whether the currently installed Fuji-Matrix printer is 9-pin or 24-pin, the Windows Setup program won't update the printer driver. This problem only occurs if you are upgrading Windows 3.1 from Windows 3.0.

Suggested workaround: Use the Printers dialog box (accessed from the Control Panel) to remove the Windows 3.0 Fuji-Matrix printer driver. Then, use the Add button to install the appropriate Windows 3.1 printer driver.

Generic/Text Only Printer: Incorrectly Prints 1.5 Line-Spaced Text

If you are using the Generic/Text Only Printer with a word-processing program that allows you to specify 1.5 line spacing—including Write and Word for Windows—the output will be two single-spaced lines, two double-spaced lines, two single-spaced lines, and so on.

Your document will appear correctly on screen with line spacing set at 1.5, but the printed output will be incorrect.

Suggested Workaround: Use a printer-specific driver. If you must use the Generic/Text Only printer driver, line-space your documents at either single spacing or double spacing. These formats will print correctly.

Generic/Text Only Printer: Fonts Are Unavailable in the Write Application

If you are using the Generic/Text Only Printer, you will receive a font error message if the TrueType dialog box has both the Enable TrueType Fonts and the Show Only TrueType Fonts In Applica-

tions check boxes turned on. (This dialog box is available from the Control Panel by clicking on the Fonts icon, then clicking on the TrueType button.)

The following error message will appear when you choose Character Fonts in Write:

```
There are no fonts installed; Run Control Panel to install fonts.
```

This error message apparently occurs because Write recognizes that the selected printer driver cannot print TrueType fonts.

Suggested workaround: In the TrueType dialog box, turn off the Show Only TrueType Fonts In Applications check box.

HP LaserJet Series II Printer Driver: Form Feeds after Printing a Line of Text

If you use the LaserJet Series II driver and try to print documents that contain TrueType fonts, your printer might print a line and then eject the page.

This problem can occur with some LaserJet printers if you try to use more than 16 fonts within a page of your document.

Suggested workaround: Turn on the Print TrueType as Graphics check box. This allows you to send fonts as bitmapped images so that the LaserJet's 16-font-per-page maximum is not violated. To enable this feature, start the Control Panel, then choose the Printers icon. Select the printer you want to use, click the Setup button, then choose the Options button. Turn on the Print TrueType as Graphics check box that appears in the Options dialog box.

HP LaserJet Series II: Large TrueType Fonts Print with Thin Lines

At font sizes larger than 80 points some TrueType fonts print with lines one pixel in width.

This problem does not occur with all LaserJet Series II printers and not with all TrueType fonts. However, it is apparently a printer-related problem and not a problem with the Windows 3.1 TrueType engine.

Suggested workaround: Turn on the Print TrueType as Graphics check box. This allows you to send fonts as bitmapped images rather than

TrueType fonts. To enable this feature, start the Control Panel, then choose the Printers icon. Select the printer you want to use, click the Setup button, then choose the Options button. Turn on the Print TrueType as Graphics check box that appears in the Options dialog box.

Hewlett-Packard LaserJet (Original Model): Cannot Print TrueType Text

The original Hewlett-Packard LaserJet printer cannot print a full page of TrueType text at 300 dpi resolution if only the original 128K of printer memory is installed.

At 300 dpi, the LaserJet printer with only 128K of memory can print only about a quarter page of TrueType before running out of memory.

Suggested workaround: Print TrueType fonts at 150 or 75 dpi. To change the resolution, start the Control Panel and choose the Printers icon. Select HP LaserJet printer, then click on the Setup button. In the Resolution list box, select either 150 dots per inch or 75 dots per inch. Click on the OK button to save the new setting.

Disk-Compression Software Problems

This section describes known problems using the Stacker and SuperStor run-time disk-compression utilities with Windows 3.1. Please note that additional tips and information regarding Stacker can be found in Chapter 10.

Stacker: SMARTDrive Does Not Cache to the Correct Drive

Due to the way Stacker swaps logical hard-disk drives, SMARTDrive by default often will try to cache to the drive that's uncompressed at boot time, not to the drive that appears uncompressed to DOS after Stacker swaps the logical drives.

The Stacker setup program partitions an uncompressed drive on the first unused logical drive of your hard disk. For example, if your hard disk is partitioned for the C: drive only, Stacker will create an uncompressed drive on the D: drive to store the files required to start and configure your computer system. However, these drives get swapped after Stacker loads, which can cause SMARTDrive to cache the wrong drive.

By default, SMARTDrive caches your hard-disk boot drive (usually the C: drive) unless you specify otherwise. If Stacker is running, the C: and D: drives are swapped. However, this swap is transparent to SMARTDrive.

Suggested workaround: To tell SMARTDrive to cache your uncompressed drive, you must specify the drive letter of the actual *compressed* drive. For instance, if your hard drive is drive C, Stacker—while running—will designate the D: drive as the uncompressed drive. Therefore, you need to include the D: drive letter in the SMARTDrive line in your AUTOEXEC.BAT file. Here's an example:

```
SMARTDRV.EXE 1024 512 D
```

Add a + (plus) sign after the drive letter if you want to enable SMARTDrive's write-behind cache.

SuperStor: Create Mountable Drive, Mount, and Dismount Features Cause Unstable Caching

If you use SuperStor's mounting commands in your AUTOEXEC.BAT file, they might prevent SMARTDrive 4.0 from operating correctly.

This problem typically manifests itself when the SMARTDRV line in AUTOEXEC.BAT appears before the mounting commands. Windows Setup places SMARTDRV at the head of your AUTOEXEC.BAT file by default.

Suggested workaround: Use a text editor to move the SMARTDRV line in AUTOEXEC to a position *after* all SuperStor mounting commands.

SuperStor: SMARTDrive Returns Read/Write Errors

SMARTDrive will try to cache to any compressed drives created by SuperStor.

With SuperStor, SMARTDrive will display error messages if it tries to read from or write to any compressed drives since the drives contain only compressed data.

Suggested workaround: Modify the SMARTDRV line in AUTOEXEC.BAT so that caching is disabled for the compressed drive. To disable caching for a drive, add the drive letter and a – (minus) sign at the end of the SMARTDRV line. For instance, if SuperStor creates a compressed drive on drive E, your SMARTDRV line should look similar to this one:

```
SMARTDRV.EXE 1024 512 E-
```

General System and ROM-BIOS Problems

This section includes information related to specific computer systems, CPUs, and ROM-BIOS chips.

Compaq DESKPRO Systems: Diskette Drives Cannot Be Accessed

On Compaq DESKPRO 386/16 and 386/20 systems, SMARTDrive 4.0 can cause problems when you try to read to or write from the diskette drives.

Newer Compaq systems do not exhibit this problem.

Suggested workaround: Disable caching for the diskette drives, which is of marginal use anyway unless you frequently read from or write to the same diskette. To disable caching for a drive, add the drive letter and a – (minus) sign at the end of the SMARTDRV line in AUTOEXEC.BAT. For example, to disable a system that has one diskette drive, your SMARTDRV line should be similar to this:

```
SMARTDRV.EXE 1024 512 A-
```

Dataworld Laptops: Require BIOS Upgrade

DOS applications might hang if you try to run them from Windows 3.1 on a Dataworld laptop.

When you try to run a DOS application from within Windows 3.1 on a Dataworld laptop, the screen might go blank, followed by a system crash.

Suggested workaround: Dataworld recommends that you upgrade the ROM BIOS to support Windows 3.1; however, you can prevent this problem by running only Windows applications from within Windows. (Exit Windows before you run any DOS applications.)

HOT TIP

ROM BIOS Versions that Are Incompatible with Windows 3.1

The following list details problems with various ROM BIOSs confirmed by Microsoft to cause problems with Windows 3.1. In most cases, these are older ROM BIOSs that can be replaced by newer versions compatible with Windows 3.1 operations.

AMI BIOSs manufactured in 1987 can prevent Windows from accessing diskette drives correctly. BIOSs manufactured in 1989 can cause UAEs and other system failures.

AST Some older BIOSs manufactured by AST and installed on Premium 286 machines can cause problems for Windows 3.1 on Premium 286 machines.

AWARD BIOSs earlier than version 3.1 will not work with Windows 3.1.

DTK BIOS versions earlier than version 36 will not work with Windows 3.1. Version 36 requires that the CMOS Setup utilities be disabled.

PEAK/DM (from Chips & Technologies) BIOS versions earlier than 1.3 will not work with Windows 3.1.

PHOENIX BIOSs that have a manufacturing date earlier than 1988 should be upgraded to support Windows 3.1.

QUADTEL BIOS versions lower than 3.05 will not work with Windows 3.1.

TANDON Some older TANDON BIOS cause keyboard failures.

TOSHIBA The T3100/20 will not run Windows 3.1 with a BIOS earlier than version 4.2. The T3100e will not run Windows 3.1 with a BIOS earlier than version 1.7. Windows 3.1 might not detect the T5200 during Setup; make sure this is the selected machine type.

WYSE Windows Setup detects an 84-key keyboard even though a 101-key keyboard is installed. Run Windows Setup in maintenance mode and select the 101-key keyboard.

ZENITH The Zenith 386/16 will not run Windows with a BIOS version lower than 2.6E. The Zenith Turbosport 386 will not run Windows with a BIOS version lower than 2.4D.

NCR 925 Computers: System Causes Memory Conflicts

If you run EMM386.EXE with an NCR 925 system, EMM386 will try to use a range of the UMA required by the system.

Although Windows itself will not use the range of UMA occupied as video memory by this system, EMM386 might. Therefore, this problem should only occur if you are using EMM386 to configure the UMA prior to running Windows.

Suggested workaround: Exclude the range E000-EFFF from the DEVICE=EMM386.EXE line. For example:

```
DEVICE=EMM386.EXE X=E000-EFFF
```

You can exclude and include multiple ranges of memory within this line. Refer to Chapter 2 for more information on EMM386 and memory management.

Note: Microsoft had reported a similar memory conflict when EMM386 was used with Windows 3.0 on an Everex 386/25 system. However, for this system, the problem no longer occurs in Windows 3.1.

Network-Related Problems

This section describes problems you might have running Windows 3.1 on various networks. Specific network issues related to the way Windows 3.1 is installed on networks are provided in Appendix A. Therefore, you should look in Appendix A to review any setup-related issues and problems for networks. In this chapter, we'll focus on problems that can occur during the day-to-day operations of Windows on various networks.

Also, the file NETWORKS.WRI is installed in the WINDOWS directory automatically when you install Windows, unless you or your system administrator specifically instructs the setup program not to copy this file. NETWORKS.WRI should be read before you or your adminstrator installs Windows to the network. Since this file is available on-line, we won't repeat the network information contained there.

DEC Pathworks Network: Network Error Message when Using Print Manager to Print

You may receive a NETWORK_ERROR! message if you try to print to a network printer that isn't directly connected to the file server.

This problem only appears if you are using Print Manager to print to a network printer.

Suggested workaround: Ignore the message. Print Manager is misinterpreting the way the printer has been routed. The printer will function properly and all files will be queued to the Print Manager correctly.

Novell Network: Windows Will Not Run on a Non-Dedicated Server

When a file server for a Novell network is also used as a workstation, Windows 3.1 will not run on the server.

With Novell NetWare, Windows 3.1 will only run on a server that is dedicated (used solely as a server) because a non-dedicated server does not make extended memory available for workstation use. Without a minimum 192K of extended memory available, Windows 3.1 will not run.

Suggested workaround: Configure the server as a dedicated server only, or install Windows locally on all workstations. In this case, of course, the non-dedicated server will still be unable to run Windows; however, other workstation users will be able to run Windows and use network services.

Novell NetWare: In Enhanced Mode, All Windowed DOS Applications Share the Same Network Connections

If your SHELL.CFG file contains the entry Task Mode=0 or Task Mode=1, you might notice that DOS applications you display in a window share the same network connections and working directories.

This problem can occur even if the NWShareHandles option (activated by using the Control Panel) is turned off. A Task Mode=0 settting tells NetWare that you don't want to do any multitasking, while Task Mode=1 indicates that you will only perform multitasking outside of Windows. Task Mode=2 is the correct setting for Windows 3.1, and is also the default Task Mode= value. (So, if a TaskMode= line is not present in your SHELL.CFG file, the correct value (2) for Windows will be used by default and this problem will not occur.)

Your SHELL.CFG file might contain the incorrect setting if you (or someone who used your system) changed the Task Mode value to 0 or 1 before Windows 3.1 was installed. The Windows Setup program does not modify SHELL.CFG.

Suggested workaround: Use Notepad or the MS-DOS Editor to open SHELL.CFG and delete the Task Mode= line. You will need to reboot your system for the default setting to take effect.

CHAPTER

17

Application Troubleshooting

This chapter provides information on problems that stem from specific programs that you run with Windows 3.1. The types of programs we'll cover include TSRs, file and desktop management programs, font managers, and other general-use applications. However, programs and utilities that directly configure memory and other hardware devices—such as memory managers, SMARTDrive, and device drivers—are discussed in Chapter 16.

Please keep in mind that, as in Chapter 16, the workarounds provided in this chapter are *suggestions only*. These workarounds come from many sources—including Microsoft, application and hardware vendors, end users, and from our own direct troubleshooting experience. The usefulness of the workarounds will vary for different system configurations. We cannot guarantee the success or results of any workaround on your system.

Troubleshooting tips and techniques for many applications are included throughout this book. To avoid redundancy, we'll only present additional troubleshooting information in this chapter. If you can't find a solution to your problem in this chapter or in Chapter 16, try using the index at the back of the book to locate the topic in one of the earlier chapters.

As you review this chapter and Chapter 16, you'll likely notice that many problems stem from outdated applications. Most software vendors have, by now, created updated applications to work with Windows 3.1. If you are using an application that was designed for Windows 3.0 or an earlier Windows version, you might need to update to the most current version in order to solve serious problems.

Many of our suggested workarounds involve using the Windows Setup program in maintenance mode to modify or update the files that support Windows 3.1. Maintenance mode setup is available at all times from within Windows, and is distinct from the initial Windows Setup program that installed Windows on your system. In maintenance mode, the Windows Setup program lets you make additional changes to the Windows environment. To use maintenance-mode setup, choose the Windows Setup icon from the Main group—after Windows 3.1 has been installed.

HOT TIP

Read APPS.HLP

Windows 3.1 stores compatibility information about 30 widely used Windows applications in a help file called APPS.HLP. To read this file, follow these steps:

1. From any Windows application, choose Help Index from the Help menu.
2. In the Help window, choose File Open.
3. In the Directories box, select the WINDOWS directory.
4. Double-click on APPS.HLP to open the file.
5. Click on the name of an application to read information about its compatibility with Windows 3.1.

TSR Problems

Many terminate-and-stay-resident (TSR) programs were designed to support DOS features and DOS applications. Other TSRs were designed for Windows 3.0. In any event, a TSR that is not specifically designed to be Windows 3.1-aware might behave unpredictably or might cause your system to crash when you run Windows 3.1. We'll identify many of these problems in this section.

HOT TIP

Read SETUP.TXT

When you installed Windows 3.1, the Windows Setup program copied a file called SETUP.TXT into your WINDOWS directory. This file contains information for many TSRs that cause problems for Windows 3.1. Microsoft provided this file so that you could identify troublesome TSRs and other utilities that might provide the Windows Setup program from installing Windows 3.1 properly. However, much of the TSR information in SETUP.TXT provides troubleshooting tips that you can use to make your TSRs behave well during the routine operations of Windows 3.1.

After Dark: Screen Saver Does Not Blank Out Pull-Down Menus

Windows 3.1 can use a screen drawing feature that was not used in Windows 3.0. This features always displays a pull-down menu as a "topmost window"—in other words, the pull-down menu displays on top of all other displays, including screen savers.

After Dark and other screen savers that were designed for Windows 3.0 are not aware of the topmost window feature and therefore will run incorrectly when the screen saver runs while a pull-down menu is displayed.

Suggested workaround: Contact the screen-saver vendor for an updated set of screen savers (designed for Windows 3.1).

Refer to Appendix A for information regarding TSRs that can interfere with the Windows Setup program. Most TSR problems in Windows are setup-related.

File and Desktop Management Software Problems

File and desktop management software like PC Tools, hDC Power Launcher, and the various Norton utilities must work in close coordination with Windows 3.1, since Windows often performs identical or similar actions as these programs. In fact, some of these programs—such as Norton Desktop for Windows—are designed to replace some Windows features with more customizable and more efficient features. If these file and desktop managers are not designed specifically for Windows 3.1, you might encounter serious problems when you try to use them with Windows 3.1.

Other file and desktop managers might have minor problems that will allow you to use them with Windows 3.1, but with some limitations. We'll describe version-related problems and other situations that might cause Windows to behave poorly, erratically, or to completely fail when various file and desktop managers are in use.

Central Point Backup 7.1: The Include/Exclude List Cannot Be Saved

Under Windows 3.1, if you save a CP Backup Configuration or a Setup, the list of files to Include/Exclude will not be retained when you exit.

Central Point Backup 7.1 ships with the PC Tools 7.1 package. This problem does not occur with Central Point Backup 7.2.

Suggested workaround: For a Configuration, use Notepad to open the WNBACKUP.INI file. Find the INEX= line and change it to read **INEX=1**. Then, change the INEX0= line to read **INEX0=*.***. Save the WNBACKUP.INI file. From this point forward, save Configuration settings using the CP Backup for DOS utility (which is also included with PC Tools 7.1); do not save them under Windows. For Setup files, load the Setup using CP Backup for DOS, then re-save the Setup. From this point forward, save Configuration settings using the CP Backup for DOS utility.

You might also want to consider upgrading to Central Point Backup for Windows 7.2. Central Point currently offers a $20 upgrade, plus shipping, for registered users of the 7.1 version.

Fastback for Windows: Early Versions Will Not Run under Windows 3.1

Fastback for DOS has been available for several years and can be used to back up files at the DOS level. The first release of Fastback for Windows and subsequent maintenance release versions were designed for Windows 3.0 and are not compatible with Windows 3.1.

Fastback for Windows is sold by Fifth Generation Systems. Apparently, the Fastback for Windows program (written for Windows 3.0) checks for Windows version numbers and will hang if you try to run it under Windows 3.1.

Suggested workaround: Fifth Generation Systems has released a version of Fastback for Windows that is compatible with Windows 3.1. Contact Fifth Generation Systems for upgrade information.

The Norton Utilities: Various Problems Occur under Windows 3.1

The Norton Utilities were designed for DOS and often create problems for Windows 3.1. The problems that can occur with The Norton Utilities depend on the version you are currently using with Windows 3.1.

The following problems can occur when you use The Norton Utilities with Windows 3.1.

Disk Monitor When Disk Monitor is turned on, SMARTDrive may cause your system to crash if it tries to write to the protected drive. You can correct this problem by adding the drive letter of the protected drive to the SMARTDrive line in AUTOEXEC.BAT.

NUCONFIG.EXE and NCACHE.EXE NUCONFIG configures Norton's NCACHE utility. The NCACHE utility that comes with The Norton Utilities 5.0 is not compatible with Windows 3.1 and can hang your system if you are running both of these caches. Although newer versions of The Norton Utilities will correct this problem, you should use SMARTDrive instead of NCACHE. SMARTDrive 4.0 supports Windows 3.1 more efficiently than NCACHE and most other third-party disk caches.

TIMEMARK This utility will not track time correctly under Windows 3.1 in standard mode.

NewWave 3.0 (and Earlier Versions): Incorrect System Version Error Message Appears

Hewlett-Packard's NewWave versions 3.0 and earlier will not load under Windows 3.1.

NewWave displays an "Incorrect System Version" error message because the NewWave batch file detects a video driver that it cannot recognize (the Windows 3.1 driver). This problem does not occur with NewWave 4.0.

Suggested workaround: Replace the 386Grabber= line in the [boot] section of SYSTEM.INI with the entry VGA30.3GR. Then replace the Display= line in the [386enh] section of SYSTEM.INI with the entry Display=VDDVGA30.386. Save SYSTEM.INI and restart Windows. Also, consider upgrading to NewWave 4.0, which is compatible with Windows 3.1.

Font Manager Problems

In this section, we'll explain problems and possible workarounds for various third-party font managers. However, keep in mind that the TrueType engine and TrueType fonts provided with Windows 3.1 make many font managers redundant and sometimes obsolete. You can bloat your WIN.INI file and decrease Windows' performance if you are running a font manager that basically just duplicates the scalable-font capabilities of TrueType. If you are using a font manager with Windows 3.1 and it appears to be causing problems for Windows, you should consider disabling the font manager. We've provided general information and some troubleshooting tips for font managers in Chapters 12 and 13. We won't repeat that information here; refer to these earlier chapters for more information on font managers.

Intellifont for Windows 1.1: Intellifont's Screen Driver Causes Windows to Crash in Standard Mode

Intellifont for Windows 1.1 uses a driver called IFW.DRV to create screen fonts. However, this driver is not compatible with Windows 3.1's standard mode and will cause Windows to crash.

Intellifont for Windows is Hewlett-Packard's font manager, providing scalable screen and printer fonts for LaserJet III printers. Windows will try to use this "shell" display driver with Intellifont if the entry "display.drv=IFW.DRV" appears in the [boot] section of SYSTEM.INI. This line is added to SYSTEM.INI by the Intellifont setup program.

Suggested workaround: Run Windows in enhanced mode if you want to use the Intellifont screen fonts with Windows 3.1. If you must run Windows in standard mode, replace the driver name in the display.drv= line to match the name of your video driver. For instance, if you are using the standard VGA driver, change the display.drv=IFW.DRV line to display.drv=VGA.DRV.

HOT TIP

Font Managers Can Prevent Windows 3.1 from Updating Video Drivers Correctly

Hewlett-Packard's Intellifont for Windows and Bitstream's Facelift both change the display.drv= line in the [boot] section of SYSTEM.INI to support their own shell screen drivers (for creating scalable screen fonts). When you upgrade Windows 3.0 to Windows 3.1, Windows Setup will not change this driver entry if it does not recognize the name of a driver installed by Windows 3.0. Consequently, Windows 3.1 may behave erratically since you are not using a Windows 3.1-compatible driver. If you suspect this problem after Windows 3.1 has been installed, try using Windows Setup to change to the correct video driver for your system.

Problems with Other Applications

This section identifies problems that occur when running a wide range of Windows and non-Windows applications, including word processors, spreadsheet programs, database systems, and communications programs. Most (but not all) of the information in this section deals with Windows applications, since we've already covered many non-Windows application problems and issues in Chapter 14. Also, Chapter 15 explains many OLE-related problems that you might have with some Windows applications, so we won't repeat that information here.

Keep in mind that problems you encounter in Windows applications often occur because you are not using the most current version. Windows applications that were designed specifically for Windows 3.0 often do not conform completely to the newer Windows 3.1 API specifications.

Crosstalk for Windows 1.1 (and Earlier Versions): Incompatible with Windows 3.1

System behavior can be unpredictable when you use Crosstalk for Windows 1.1. This version of Crosstalk was designed for Windows 3.0.

Symptoms include slow system performance and strange display behavior.

Suggested workaround: Upgrade to Crosstalk for Windows 1.2 or higher. At this writing, Crosstalk for Windows 1.2.2 is the most current version.

CorelDRAW! 2.0: Can't Use Extrude or Blend Functions

If you try to use the Extrude or Blend functions, CorelDRAW! will display the message, "WALDO has created a fault at (location)."

The program will not crash when this error message occurs; however, you will not be able to use the offending function.

Suggested Workaround: Corel Systems has corrected the problem, and currently offers a maintenance upgrade (version 2.001). Corel Systems provides an upgrade program for registered *CorelDRAW!* users.

CorelDRAW! (All Versions): Scroll Bars and Wire Frames Do Not Work Properly

Scroll bars and wire frames will display incorrectly if you are using the Interruptible Display option with Corel DRAW!.

This problem exists even with the current 2.001 maintenance release.

Suggested workaround: Choose Special Preferences, then turn off the Interruptible Display check box. Contact the vendor to see if a more current upgrade is available.

CorelDRAW! (All Versions): TrueType Fonts Are Not Available

TrueType fonts are not supported in CorelDRAW! because this program contains its own scalable type engine and fonts.

CorelDRAW! will use either its own internal scalable fonts or will import and convert fonts using the WFNBOSS.EXE utility. However, the program WFNBOSS.EXE does not currently provide a filter for TrueType fonts.

Suggested workaround: If you want to import documents that contain TrueType, you can do so by copying the text to the Clipboard and then pasting it into *CorelDRAW!*, which will then convert the TrueType fonts to its own internal fonts.

dBASE IV Hangs on Exit with Novell Networks

If your system loads the DOS SHARE utility when it starts, and if you are running Windows on a network, dBASE IV will not exit properly.

This problem occurs only if you are running dBASE IV from within Windows (as a virtual machine).

Suggested workaround: One solution is to remove SHARE from your AUTOEXEC.BAT file, which will allow dBASE IV to exit properly from within Windows. However, SHARE is required to run some Windows applications, and may also be necessary to share files across the network. If this solution is not feasible, exit Windows before running dBASE IV. The problem will not occur when you start dBASE IV from the DOS prompt.

Equation Editor: Two TrueType Fonts Not Installed

The Equation Editor is bundled with Word for Windows 2.0 and comes with its own set of TrueType fonts. However, if you install Word for Windows 2.0 under Windows 3.1, the MT Extra and Fences fonts will not be installed.

If you open the Fonts dialog box from the Control Panel, you will notice that the MT Extra and Fences fonts are listed. However, they are not correctly installed.

Suggested workaround: In the Fonts dialog box, select MT Extra and click on the Remove button. In the Remove Font box, turn off the Delete Font File From Disk option, then click on Yes. Repeat this procedure for the Fences font. Next, click on the Add button. In the Directories box, choose the WINDOWS\SYSTEM directory. In the List of Fonts box, select MT Extra Plain (TrueType) and Fences Plain (TrueType). Choose OK to add these fonts, then close the Fonts dialog box. You will need to restart Windows so that the new fonts can be read from WIN.INI.

Excel 3.0: The 11" x 14" Paper Size Option Is Not Available

If you are using a Windows 3.1 universal printer driver for a dot-matrix printer, the 11" x 14" paper size will not be available for selection.

The 11" x 14" paper size option normally appears in the Page Setup dialog box (accessed by choosing File Page Setup). However, the description for this option is too lengthy to fit in the dialog box; Excel displays a set of placeholder characters instead.

Suggested workaround: In the Page Setup dialog box, find the characters "|." or "|,". Click on these characters, then click on the OK button. When you print, Windows will use the 11" x 14" paper size. This problem does not exist in Excel 4.0.

Excel 3.0: Generic/Text Only Printer Driver Does Not Work Properly

The Windows 3.1 Generic/Text Only printer driver (TTY.DRV) prints Excel 3.0 worksheets as garbage.

The Windows 3.1 TTY driver is required for daisy-wheel printers and some other older printers that do not have a printer-specific driver.

Suggested workaround: You can reinstall the Windows 3.0 TTY driver to solve this problem. However, since the older TTY driver has the same name as the Windows 3.1 driver (TTY.DRV), you must physically delete the TTY.DRV from the Windows 3.1 directory before you reinstall the Windows 3.0 driver. The Windows 3.1 TTY driver works correctly with Excel 4.0.

Excel 4.0: Crashes with Many Windows 3.0 Printer Drivers

If you are using a Windows 3.0 printer driver, Excel 4.0 may behave unpredictably or may crash if the driver is incompatible with the Windows 3.1 COMMDLG.DLL file.

COMMDLG.DLL is a dynamic link library that applications use to draw common-format dialog boxes. Some third-party printer drivers designed for Windows 3.0 are incompatible with this file and can cause Excel to crash when you try to use the Page Setup dialog box, as well as the Print and Print Preview commands.

Suggested workaround: Upgrade to a Windows 3.1 printer driver. Access the Control Panel and use the Add button in the Printers dialog box to see if Windows 3.1 provides a driver for your printer. If so, install this driver. You can also check with the manufacturer of your printer or check the Windows Driver Library (see Chapter 13) to see if these sources can provide you with an updated driver.

Excel 4.0: Crashes during Setup

If you are running Windows 3.1, and try to install Excel 4.0, the Excel setup program can crash under some configurations.

If the Excel setup terminates prematurely and returns you to the DOS prompt, the problem can stem from several possible hardware situations—including conflicts with TSRs, video memory, and hard disk interrupts.

Suggested workaround: Modify AUTOEXEC.BAT and CONFIG.SYS so that you can perform a clean boot. (This procedure is described in Chapter 2. Disable the load= and run= lines in WIN.INI by placing a semicolon at the start of each line. Then, start Windows by typing **WIN /D:XV** and then pressing **Enter**. Immediately after you do this, press the **Shift** key and hold it down until the Program Manager window appears. (The /D:XV switch and parameters prevent Windows from using the upper memory area and from terminating hard-drive interrupt calls. Holding down **Shift** prevents any programs in your Startup group from loading.)

The net effect of this workaround is to keep your memory configuration and Windows' memory usage as simple as possible. You only need to use this approach to support Excel's setup program; it isn't necessary to support Excel's actual operations. After you have installed Excel 4.0, you can return your AUTOEXEC.BAT, CONFIG.SYS, and WIN.INI files to their original condition, and you can start Windows as you normally do.

Lotus 1-2-3 for Windows 1.0: UAEs Occur when a Standard 1-2-3 Error Should Display

Under Windows 3.1, if you perform an invalid operation in Lotus 1-2-3 for Windows that would normally display a simple error message, you might get an Unrecoverable Application Error, which causes the system to hang.

This problem does not occur with Lotus 1-2-3 for Windows 1.1. Usually, this situation occurs with ATI display cards that are using the ATI S-VGA 1024x768 Crystal font driver.

Suggested workaround: Run Windows Setup in maintenance mode, change the display driver to either the 8514/a or the VGA driver, and then restart Windows. Also consider upgrading to Lotus 1-2-3 for Windows 1.1, which is free to registered users of the 1.0 version.

Lotus 1-2-3 for Windows 1.0: TrueType Fonts Not Available

The Style Font Replace dialog box does not list or allow you to use TrueType fonts.

This problem typically occurs when a dot-matrix or inkjet printer is selected. When LaserJet and PostScript drivers are selected, the problem usually does not occur.

Suggested workaround: Although you can view TrueType fonts in your spreadsheet by installing and selecting a driver for a laser printer, you will not be able to print the spreadsheets when you switch back to your actual printer driver. A better solution is to upgrade to Lotus 1-2-3 for Windows version 1.1, which is available free of charge to registered users of the 1.0 version.

Lotus CC:Mail for Windows: Font Style Dialog Box Shows Garbage

Lotus CC:Mail (dated 10/25/91) has its own COMMDLG.DLL file that is not compatible with Windows 3.1. If CC:Mail is run before any other Windows application that uses the Windows 3.1 COMMDLG.DLL (rather than the CC:Mail version), garbage characters will be displayed in the Font Style dialog box.

The COMMDLG.DLL file contains library routines that Windows applications can use to draw commonly used dialog boxes. The CC:Mail version of this DLL was created prior to Windows 3.1, and therefore should not be used.

Suggested Workaround: Delete the COMMDLG.DLL file that is stored in the directory of the file server that contains the CC:Mail files. Then copy COMMDLG.DLL from the WINDOWS\SYSTEM directory of the server into the CC:Mail directory.

PageMaker 4.0 Does Not Recognize Windows 3.1 Clipboard Formats

Microsoft recently modified and improved the formatting capabilities of the Clipboard file format for RTF (Rich Text Format). Both Word for Windows 2.0 and Excel 4.0 use this new RTF file format. However, when you paste text from Excel 4.0 worksheets or Word for Windows 2.0 documents, PageMaker 4.0 uses an RTF import filter designed for the older RTF format.

The result of the incompatibility between PageMaker's RTF import filter and the newer Microsoft RTF file format is often a blank area or an error message following an attempted copy-and-paste operation. Microsoft also updated the graphics formats used by the Clipboard. In PageMaker, this change can affect your ability to paste Excel charts as well as files from the CLIPART graphics.

Suggested workaround: Try using PageMaker's Place command instead of Paste. At this writing, new import filters are being written to support Word for Windows 2.0 and Excel 4.0 text, as well as Excel 4.0 charts. Check with Aldus to see if these filters are now available. The CLIPART that comes with PageMaker 4.0 is currently owned by Central Point Software. Check with them to see if a new filter is available for the CLIPART graphics.

PageMaker 4.0: Window Will Not Remain Maximized across Sessions

With most applications in Windows 3.1, Windows will save the size of an application's window when you exit. With PageMaker 4.0, the size setting is not retained from session to session.

This problem usually occurs when you maximize the PageMaker window and then exit PageMaker.

Suggested workaround: Use Notepad to open your ALDUS.INI file. Add the following line to the [PageMaker4] section:

```
Maximize=1
```

You need to restart PageMaker for the new setting to take effect. (You don't need to restart Windows.)

HOT TIP

Check to Make Sure Your Application Directories Are in Your PATH

Many Windows applications store essential startup settings and other data (such as internal fonts) in the application's directory (such as ALDUS or PM4 for PageMaker 4.0).

Usually when you install a Windows application, the install program will try to add its directory to the PATH line in your AUTOEXEC.BAT file. However, if you don't allow the install program to make this change, or if your PATH line already contains the maximum 128 characters, the directory for the application will not be included in your PATH. If this happens, the application might not be able to find fonts or other startup information. For instance, if the ALDUS or PM4 directory is not in your PATH, PageMaker 4.0 won't be able to find the .FON files it needs to create screen fonts.

If you experience startup or display problems, or other unruly behavior with a newly installed Windows application, check the PATH in your AUTOEXEC.BAT file and make sure the application's directory is included. You might need to remove older, unused directories from the PATH line to make room for new directories.

PageMaker 4.0: Performance Is Extremely Slow

If your WIN.INI file lists several hundred fonts in the [fonts] section, PageMaker's performance can be severely crippled.

This problem often occurs after PageMaker users install *CorelDRAW!* 3.0 (which installs many of its own fonts) or a set of new third-party fonts. To support its own formatting operations, PageMaker reads through the font information in WIN.INI continually. On reasonably fast systems—such as a 386/33 machine—you can install up to 200 fonts without significantly affecting PageMaker's performance. However, you might experience problems as the number of fonts specified in WIN.INI increases.

Suggested workaround: Remove fonts that you do not use so that you can reduce the size of the [fonts] section in WIN.INI. An even better alternative: at this writing, Ares Software is planning to release a product called FontMinder, which allows you to organize sets of fonts into "font packs" that you can selectively install or remove on the fly. FontMinder used to be available as freeware under the name Font Manager, but this freeware product has been removed from circulation. Contact Ares Software to see if FontMinder is now available.

PowerPoint 3.0: LaserJet Series II Will Not Print TrueType as Graphics

If you use the LaserJet Series II driver, the TrueType as Graphics option will not work with PowerPoint 3.0 if you select the option from within PowerPoint.

You normally turn on the Print TrueType as Graphics check box from within PowerPoint by choosing Printer Setup, then clicking on the Setup button, clicking on the Options button in the Setup dialog box, then turning on the check box that appears in the Options dialog box. However, this approach does not work.

Suggested workaround: To ensure that the Print TrueType as Graphics option gets turned on (if you do want it on), select it from the Control Panel, *not* from within PowerPoint. To do so, choose the Printers icon from the Control Panel, select the printer you want to use with PowerPoint, click the Setup button, then choose the Options button. Turn on the Print TrueType as Graphics check box that appears in the Options dialog box. Although this option can slow down printing, you might need to use it to prevent your LaserJet printer from form-feeding a blank page after it prints a line of text.

PowerPoint 3.0 Will Not Paste Tables Created in Word for Windows 2.0

If you try to paste a table created in Word for Windows 2.0 into a PowerPoint 3.0 slide, you will be able to view and print only the Word for Windows icon.

This same problem will occur if you try to use Paste Special to paste or paste-link the table as a Word Object.

Suggested workaround: There are several possible solutions, although none of these is ideal. You can use Paste Special to paste the table into PowerPoint as formatted or unformatted text. However, you will not be able to edit the table easily in PowerPoint under this approach. You can also paste the table if you first convert it to text in Word for Windows. Again, this procedure prohibits easy editing from within PowerPoint.

One fairly effective solution is to first paste the table into an Excel worksheet, then paste or paste-link the worksheet into a PowerPoint slide. Although PowerPoint can paste Excel worksheets and worksheet objects successfully, Excel often has a difficult time pasting Word for Windows tables that have complex formatting.

Perhaps the best way to place a Word for Windows table into a PowerPoint slide is to first insert a Microsoft Graph object into the slide. Then double-click on the object to start Microsoft Graph. Use Select All and Clear Data to remove the default data and graph that appear. Next, switch to Word and copy the table to the Clipboard. Switch back to Graph and paste the table. Graph can retain all the formatting of a Word table. In the PowerPoint slide, each table cell will appear as a separate text object. You can group and ungroup the cells as necessary to support editing and sizing. Although this approach is extremely cumbersome, it can be faster than re-creating large tables in PowerPoint.

Publisher 1.0: "Error in loading printer driver" Message Appears

In Microsoft Publisher running under Windows 3.1, if you try to print a file that you initially created in Windows 3.0, Publisher will display an error message.

This problem occurs because Publisher identifies that the currently selected printer driver differs from the driver used to create the file. You will not be allowed to print the file.

Suggested workaround: In Microsoft Publisher, choose File Printer Setup. In the dialog box, click on the OK button. The error message will no longer appear and you can now print the file. You must repeat this procedure for each file that produces the error message.

Publisher 1.0: Word for Windows 2.0 Can't Be Imported

Publisher will either not allow you to import documents that were created in Word for Windows 2.0 or will import the documents incorrectly.

The import filter used in Publisher 1.0 was designed prior to Word for Windows 2.0 and therefore is not compatible with Word for Windows 2.0 documents.

Suggested workaround: You can save Word for Windows 2.0 documents as Word for Windows 1.x documents, then import these documents successfully into Publisher. However, a maintenance release (version 1.0a) of Microsoft Publisher is available, and contains an import filter for Word for Windows 2.0. Consider upgrading to this version.

Publisher (All Versions): Will Not Import .DRW Files from Micrografx Draw 'N Publish

Microsoft Publisher does support the .DRW graphics format, but not the one used by Micrografx Draw.

This problem results because Micrografx Draw updates its versions more frequently than does Publisher, resulting in incompatibilities between the two applications.

Suggested workaround: Save Micrografx Draw files in the Windows Metafile (WMF) format and then import these files into Publisher. You can also use the supported Clipboard formats to simply copy and paste images from Draw into Publisher.

Word for Windows 2.0: Font Names and Sizes Do Not Appear in Ribbon

Normally, the font and font size boxes in the ribbon display the name and size of the font for the current location of the insertion point. However, you might notice that these boxes are occasionally or always empty.

If you select a new printer using the Print Setup dialog box in Word for Windows, the fonts will appear on the ribbon *unless* the printer you select is not specified as the default printer in the Control Panel. If you select a printer that uses a Windows 3.0 driver, the fonts cannot be viewed.

Suggested workaround: If you are using a Windows 3.0 printer driver, you can't view font names and sizes in the ribbon. However, if you are using a Windows 3.1 printer driver, you can access the Control Panel from within Word, then set the current printer as the default printer. To do so, click on Word's control-menu button, then choose Run. Click on the Control Panel option button, then click on OK to start the Control Panel. Choose the Printers icon to start the Printers dialog box. Make sure the highlighted printer name is the one you are using with Word, then click on the Set as Default button. Close the Printers dialog box to return to Word.

Word for Windows 2.0: Works for Windows Documents Cannot Be Converted to Word Format

Word for Windows did not ship with a conversion utility for Works for Windows. Consequently, documents that you have created in Works for Windows cannot be opened in Word for Windows without some loss in formatting.

To conserve disk space, Microsoft did not include several conversion utilities, including the one for Microsoft Works for Windows, with the diskettes that ship with Word for Windows 2.0.

Suggested workaround: Microsoft offers a diskette that contains conversion utilities for Microsoft Works for Windows and several other text formats. This diskette is free of charge to registered Word for Windows users.

WordPerfect for Windows: TrueType Fonts Are Not Available

When you are using WordPerfect for Windows printer drivers, TrueType fonts are not available.

The WordPerfect printer drivers were designed prior to the release of TrueType fonts. However, the Windows printer drivers do allow you to use TrueType fonts with Word for Windows.

Suggested workaround: To use TrueType fonts, choose File Select Printer. In the Printer Drivers box, turn on the Windows option button. You must repeat this procedure for each printer driver that you select.

Appendix A: Windows Setup Information and Tips

If you've purchased this book, you've probably already successfully installed Windows 3.1 on at least one computer. However, if you've had difficulty installing Windows 3.1 or if you need to perform multiple installations of Windows 3.1 on different systems, you may need additional information and help. This appendix is designed to support these types of needs.

General Setup Information

There are two ways to use the Windows Setup program (SETUP.EXE). You can type **SETUP** at the DOS prompt to perform a first-time installation of Windows 3.1, or you can use the Windows Setup icon (in the Main group) to modify an existing Windows 3.1 installation. This second technique is called *maintenance-mode* setup. All of the information in this appendix applies to first-time installations of Windows 3.1, *not* to maintenance-mode setup. Much of the setup information described elsewhere in this book deals with maintenance-mode setup issues.

Also, Windows allows you to specify whether you want to perform an *Express setup* or a *Custom setup*. An Express setup works in most cases, and allows Windows to determine most hardware settings. With a Custom setup, you must specify all hardware settings, and you will have the opportunity to determine which categories of files are installed onto your hard disk.

If you need to perform customizable setups on a network, you can usually do this most effectively by creating a system settings template and then performing an automated setup (by using the SETUP /H switch). Chapter 5 explains the techniques you can use to set up Windows for use on a network. Additional network setup information is provided later in this appendix.

Minimum Space Requirements

Windows Setup requires at least 6,144,000 bytes of space to be available on a hard disk before it will install Windows 3.1. Normally, the setup program will determine free space correctly. However, on hard disks that use nonstandard cluster sizes, Windows will not be able to determine the size used by each directory. The result is that Windows will often begin the setup process, but will abort at some point because the remaining free space following creation of directories is insufficient to support Windows. If this problem occurs, you should remove any unwanted files from your hard disk, then begin the Windows Setup program again. If you have a disk defragmentation utility, you should also run this utility to free up additional space so that Windows can support its swap files.

Table A.1 lists the *minimum requirements* for installing and running Windows 3.1 in standard mode, as well as the *recommended minimum* configuration for running Windows 3.1 in standard mode. Table A.2 provides the same information for enhanced-mode Windows.

TSRs and Drivers Incompatible with Windows Setup

Many TSRs (terminate-and-stay-resident programs) and device drivers interfere with the ability of the Windows Setup to assess and configure memory correctly. Some TSRs and drivers can cause the Windows Setup

Table A.1 Minimum and Recommended Configurations for Standard-Mode Windows

Standard Mode Minimum Requirements	*Recommended Minimum Configuration*
80286 CPU or higher	80286 CPU or higher
640K conventional memory	640K conventional memory
256K extended memory	1408K extended memory
6.5MB free hard disk space	9MB free hard disk space
EGA video or higher	VGA video

Table A.2 Minimum and Recommended Configurations for Enhanced-Mode Windows

Enhanced Mode Minimum Requirements	*Recommended Minimum Configuration*
80386 CPU or higher	80386 CPU or higher
640K conventional memory	640K conventional memory
1024K (1MB) extended memory	3456K extended memory
8MB free hard disk space	10.5MB free hard disk space
EGA video or higher	VGA video

to hang or crash, while other TSRs can create less serious problems that might go undetected until after you've set up Windows.

As a general rule, you should disable all TSRs and unnecessary drivers from your AUTOEXEC.BAT and CONFIG.SYS files before you install Windows 3.1. (Necessary drivers include your mouse driver files or other hardware that Windows won't be able to detect unless it has already been configured at the DOS level.)

Windows Setup can, itself, detect many drivers that are incompatible with the setup program. In most cases, Windows will remove these programs from memory temporarily during setup—although you can prevent the setup program from doing so. Table A.3 lists the TSRs and drivers that the Windows Setup program can detect and remove during installation. All of these programs are described in the [incompTSR1] and [incompTSR2] sections of SETUP.INF.

Table A.3 TSRs in [incompTSR1] and [incompTSR2] Sections of SETUP.INF

File Name	*Description*
ALLEMM4.SYS	All Charge 386
ANARKEY.COM	Anarkey
APPEND.COM	DOS APPEND Utility
ASPLOGIN.EXE	ASP Integrity Toolkit
ASSIGN.COM	DOS ASSIGN Utility
CACHE.EXE	Disk Cache Utility
CED.COM	CED Comand Line Editor
CED.EXE	PCED Command Line Editor
CMDEDIT.COM	Command Line Editor
CUBIT.EXE	Cubit
DATAMON.EXE	PC Tools Datamon
DESKTOP.EXE	PC Tools Desktop TSR
DISKMON.EXE	Norton Disk Monitoring TSR
DOSCUE.COM	DOSCUE Command Line Editor
DUBLDISK.SYS	Double Disk Data Compression Utility
EP.EXE	Norton Desktop/Windows Erase Protect TSR
FLASH.EXE	Flash Disk Cache Utility
GRAPHICS.COM	DOS GRAPHICS Utility
HPEMM386.SYS	HP Expanded Memory Manager
HPEMM486.SYS	HP Expanded Memory Manager
HPMM.SYS	HP Memory Manager
HYPER286.EXE	Hyper Disk Cache Utility
HYPER386.EXE	Hyper Disk Cache Utility

Table A.3 TSRs in [incompTSR1] and [incompTSR2] Sections of SETUP.INF (Continued)

File Name	*Description*
HYPERDKC.EXE	Hyper Disk Cache Utility
HYPERDKE.EXE	Hyper Disk Cache Utility
HYPERDKX.EXE	Hyper Disk Cache Utility
IEMM.SYS	Memory Manager
ILIM386.SYS	Intel Expanded Memory Emulator
JOIN.EXE	DOS JOIN Utility
KBFLOW.EXE	KBFlow TSR by Artisoft
LANSEL.EXE	Lansight Network Utilities TSR
LE.COM	Le Menu Menuing Package
LSALLOW.EXE	Lansight Network Utilities TSR
NCACHE.EXE	Norton Disk Cache Utility
NDOSEDIT.COM	Command Line Editor
NEWRES.EXE	Newspace Disk Compression Utility
NEWSPACE.EXE	Newspace Disk Compression Utility
PA.EXE	Printer Assist
PC-CACHE.COM	PC Tools Disk Cache Utility
PC-KWIK.EXE	PC-Kwik Disk Cache Utility
PCPANEL.EXE	Lasertools Printer Control Panel
PCSXMAEM.SYS	pcsxmaem Utility
PRINT.EXE	DOS PRINT Utility
PYRO.EXE	Pyro! Screen Saver
QCACHE.EXE	386MAX Disk Cache Utility
QMAPS.SYS	QMAPS Memory Manager
RAMTYPE.SYS	Ramtype Utility
RM386.SYS	NetRoom Memory Manager
S-ICE.EXE	SoftIce
SK.COM	Sidekick Version 1.0
SK2.EXE	Sidekick Version 2.0
SKPLUS.EXE	Sidekick Plus
SPEEDFXR.COM	Speedfxr
SUBST.EXE	DOS SUBST Utility
SUPERPCK.EXE	Super PC-Kwik Disk Cache Utility
TSCSI.SYS	Trantor T100 SCSI Driver
UMBPRO.SYS	UMB Pro Memory Manager
VACCINE.EXE	Vaccine Antivirus Program
VDEFEND.COM	PC Tools VDefend
VDEFEND.SYS	PC Tools VDefend
VDISK.SYS	IBM RAM Disk Utility

Table A.3 TSRs in [incompTSR1] and [incompTSR2] Sections of SETUP.INF (Continued)

File Name	*Description*
VIRALERT.SYS	Data Physician Plus TSR
VSAFE.COM	Central Point Anti-Virus TSR
VSAFE.SYS	Central Point Anti-Virus
XGAAIDOS.SYS	8514/a Emulation Driver
XMAEM.SYS	xmaem Utility

For further information regarding the problems these TSRs can cause, refer to the *Program Listing* section of SETUP.TXT. The SETUP.TXT file is stored in an uncompressed format on Disk 1 of the Windows installation disks and also is copied into the WINDOWS directory during the installation of Windows. You can use Notepad, the MS-DOS Editor, or any text editor to read this ASCII file. SETUP.TXT also contains additional information on changes that you might need to make to AUTOEXEC.BAT and/or CONFIG.SYS in order to support certain TSRs after Windows has been installed.

The following sections identify and describe additional TSRs and drivers that are known to cause problems for Windows Setup or the Windows program itself.

ATDOSXL.SYS

If the HardCard Plus device driver ATDOSXL.SYS is loaded before you run Windows Setup, the setup program will not modify CONFIG.SYS. Instead, proposed changes to CONFIG.SYS will be written to a file called CONFIG.WIN. If this file already exists, the setup program writes the proposed changes to a file called CONFIG.001 or uses the next-higher number extension if .001 or other numbered CONFIG files already exist. You can either comment out the ATDOSXL.SYS line in CONFIG.SYS before you install Windows, or you can manually write or copy the changes in the proposed CONFIG file into your CONFIG.SYS file.

INSTALL=MOUSE.COM Is Undetected by Windows Setup

If your CONFIG.SYS file contains the INSTALL=MOUSE.COM line to install the Microsoft mouse driver, Windows Setup will fail to detect the mouse driver and thus will not update it. To avoid this problem, you can change the INSTALL=MOUSE.COM line to DEVICE=MOUSE.SYS.

Windows Setup Incompatible with Procom Technology's SCSI Driver

If your system uses Procom Technology's SCSI disk driver file SCSIDISK.SYS, version 1.3 (dated September, 1991), Windows Setup will crash. To prevent this problem, open CONFIG.SYS and type **REM** at the start of the DEVICE=SCSIDISK.SYS line, then reboot the system. You can remove the REM comment after Windows Setup has been completed.

QEMM-386's LOADHI.COM and LOADHI.SYS Utilities

The LOADHI.COM and LOADHI.SYS utilities used by QEMM-386 interfere with the Windows Setup TSR detection routine. TSRs and device drivers that are loaded into upper memory using these QEMM-386 utilities will not be detected by Windows Setup. Therefore, if these TSRs are incompatible with Windows Setup, you might not be able to install Windows successfully. It's usually best to disable all unnecessary drivers and TSRs temporarily, before you install Windows 3.1.

SHELLSTB.COM (PS/1 Systems)

SHELLSTB.COM was designed to create user friendliness on PS/1 systems by displaying a system menu each time the system is started. However, when this TSR is running, Windows Setup will fail. Setup detects SHELLSTB.COM and asks if it can disable this TSR; however, if the user chooses the Ignore option, Windows Setup will fail. Always allow Windows Setup to disable this TSR. Also, it is advisable to disable this TSR permanently if you will be running Windows. In its place, you should enter the command **WIN** to AUTOEXEC.BAT so that Windows starts automatically when the system boots.

Incompatibility Information in APPS.HLP

Windows 3.1 includes a little-known help file that lists and explains problems for many programs that are known to be incompatible with Windows 3.1. To read this file, called APPS.HLP, choose Help Contents (or Help Help Index) from any menu bar. Next, choose File Open to display a list of .HLP files for the current directory. Display the .HLP contents of the WINDOWS directory, if necessary, then open the file APPS.HLP. Click on the name of any program to read additional incompatibility information. Table A.4 lists the programs mentioned in APPS.HLP.

Table A.4 Programs Mentioned in APPS.HLP

Ace Software AceFile
Adobe Illustrator
Adobe TypeAlign
Aldus Freehand 3.0
Aldus Persuasion
Bitstream FaceLift 1.2
Borland C 3.0 Winsight
Campbell Services On Time 1.0
Central Point Software PC Tools
Channel Computing Forest and Trees 2.0a
Claris Hollywood
Code Finale
Computer Support Arts&Letters
Computer Support Picture Wizard
First Byte Monologue for Windows
hDC First Apps Memory Viewer 1.0
Hewlett-Packard NewWave
Lotus Ami Pro
Microsoft Bookshelf for Windows
Microsoft PowerPoint 2.0e
Microsoft Productivity Pack 1.0
Microsoft Word for Windows 1.1
NBI Legacy
Norton Desktop 1.0
PFS:WindowsWorks
Powersoft Powerbuilder 1.0
SofNet FAXit for Windows
Software Publishing Harvard Graphics for Windows
WordPerfect for Windows
WordStar for Windows

Using BOOTLOG.TXT to Troubleshoot Windows 3.1 Installation Problems

Windows 3.1 Setup writes setup information to an ASCII file called BOOTLOG.TXT. If Windows Setup aborts prematurely, you can often view this file to locate the source of the problem.

If Windows Setup crashes or locks, restart your computer and then use the MS-DOS Editor or another text editor to view the contents of

BOOTLOG.TXT, which is stored in the WINDOWS directory. Look at the end of the BOOTLOG.TXT file. If a particular driver has failed to initialize, you'll be able to identify the faulty driver at the end of the BOOTLOG.TXT file. If you have installed more than one version of Windows 3.1 on a particular system, or if you have tried more than once to install Windows 3.1 on a system, BOOTLOG.TXT will contain multiple [boot] sections. If so, the last [boot] section will represent the most recent boot attempt.

If all entries in BOOTLOG.TXT appear normal, your SETUP.EXE or SETUP.INF files might be corrupted. If BOOTLOG.TXT does not exist, Windows Setup has either had a problem accessing extended memory or has identified a loaded driver or program that conflicts with Windows in protected mode.

Device Options Available During Setup

Table A.5 lists the device options that you can select when you perform a Custom setup. All of these options, except for the language options, are also available when you use maintenance-mode setup.

Table A.5 Device Options Available with a Custom Setup

Display Options
8514/a
8514/a (Small fonts)
Compaq Portable Plasma
EGA
EGA black and white (286 only)
EGA Monochrome (286 only)
Hercules Monochrome
IBM MCGA (286 only)
Olivetti/AT&T Monochrome or PVC Display
QuadVGA, ATI VIP VGA, 82C441 VGAs
TIGA (Small fonts)
TIGA (Large fonts)
VGA
VGA (Version 3.0)
VGA with Monochrome display
Super VGA (800x600, 16 colors)
Video 7 512K, 640x480 256 colors
Video 7 512K, 720x512 256 colors
Video 7 1Mb, 800x600 256 colors

Table A.5 Device Options Available with a Custom Setup (Continued)

Video 7 1Mb, 1024x768 256 colors (Small fonts)
XGA (640x480, 16 colors)
XGA (Large fonts)
XGA (640x480, 256 colors)
Other (Requires disk provided by a hardware manufacturer)

Mouse Options

HP Mouse (HP-HIL)
Logitech
Microsoft, or IBM PS/2
Genius serial mouse on COM1
Genius serial mouse on COM2
Mouse Systems serial or bus mouse
Mouse Systems serial mouse on COM2
No mouse or other pointing device
Olivetti/AT&T Keyboard Mouse
Other (Requires disk provided by a hardware manufacturer)

Keyboard Options

All AT type keyboards (84 - 86 keys)
AT&T '301' keyboard
AT&T '302' keyboard
Enhanced 101 or 102 key US and Non-US keyboards
Hewlett-Packard Vectra keyboard (DIN)
Olivetti 101/102 A Keyboard
Olivetti 83-key keyboard
Olivetti 86-key keyboard
Olivetti M24 102-key keyboard
PC-XT 83-key keyboard
PC/XT - Type keyboard (84 keys)
Other (Requires disk provided by a hardware manufacturer)

Language Options

English (American)
English (International)
Finnish
French
French Canadian
German
Icelandic

Table A.5 Device Options Available with a Custom Setup (Continued)

Italian
Norwegian
Portuguese
Spanish
Spanish (Modern)
Swedish
Other (Requires disk provided by a hardware manufacturer)

Network Options

No Network Installed
3Com 3+Open (versions 1.x)
3Com 3+Open (version 2.0 Basic)
3Com 3+Open (version 2.0 Enhanced)
3Com 3+Share
Artisoft LANtastic (versions below 3.0 unsupported)
Artisoft LANtastic (versions 3.X)
Artisoft LANtastic (versions 4.X)
Banyan Vines (versions below 4.0 unsupported)
Banyan Vines (versions 4.0X)
Banyan Vines (version 4.1)
IBM OS/2 LAN Server (versions below 1.2 unsupported)
IBM OS/2 LAN Server (without /API option)
IBM OS/2 LAN Server (version 1.2 or 1.3)
IBM OS/2 LAN Server (version 1.3 CSD 5015/5050)
IBM OS/2 LAN Server (version 2.0)
IBM PC LAN Program (all versions)
Microsoft LAN Manager (version1.X)
Microsoft LAN Manager (version 2.0 Basic)
Microsoft LAN Manager (version 2.0 Enhanced)
Microsoft Network (or 100-percent compatible)
Novell NetWare (shell versions below 3.01 unsupported)
Novell NetWare (shell versions below 3.21)
Novell NetWare (shell versions 3.21 and above)
Novell NetWare (shell versions 3.26 and above)
DEC Pathworks (versions below 4.0 unsupported)
DEC Pathworks (version 4.0)
DEC Pathworks (version 4.1 or higher)
TCS 10Net (versions below 4.1 unsupported)
TCS 10Net (versions 4.1X)

Table A.5 Device Options Available with a Custom Setup (Continued)

TCS 10Net (versions 4.1X with DCA 1M card)
TCS 10Net (versions 4.2 and above)
TCS 10Net (version 5.0)
Other (Requires disk provided by a network manufacturer)

Files on Windows Installation Disks

Table A.6 lists and identifies the usage for all files on the Windows installation disks. Most of the files are stored on the installation disks in a compressed format and, therefore, must be uncompressed before they can be used. The Windows Setup program uses a built-in file expansion utility to uncompress files before they are copied to your hard disk; the Control Panel uses the LZEXPAND.DLL library to expand files that you install later to support additional printers, fonts, and so on.

Each compressed file appears on the installation disk with an underscore in the last character position of the file extension (such as GDI.EX_ or VGA.DR_). When the files are expanded, the underscore is replaced by the correct character for the extension. (For example, GDI.EX_ becomes GDI.EXE and VGA.DR_ becomes VGA.DRV after they are uncompressed.)

For your convenience, we've included the appended extension character in parentheses for each file name. Keep in mind that if you view a directory listing of any of the installation disks, you will see an underscore in place of this letter. Also, you might encounter situations where you will need to use the EXPAND utility yourself to copy and uncompress additional driver or font files after the setup program has been completed. In many cases, EXPAND automatically will be able to identify the correct final character for a particular file's extension. However, for some files, you might need to specify the complete file name.

Table A.6 Contents of Windows Installation Disks

DISK 1 (46 Files)

File Name	*Usage*
8514.DR(V)	Supports 8514/a displays
8514SYS.FO(N)	System font for 8514/a displays
APP850.FO(N)	Code-page 850 font support for windowed DOS applications

Table A.6 Contents of Windows Installation Disks (Continued)

COMM.DR(V)	Supports serial and parallel port communication
CPWIN386 CP(L)	Additional enhanced-mode code for Control Panel
DISK1	Disk ID for setup
EGA.3G(R)	Enhanced-mode grabber for EGA displays
EGA.DR(V)	Supports EGA displays
EGACOLOR.2G(R)	Standard-mode grabber for EGA color displays
EGAFIX.FO(N)	Fixed-system font for EGA display
EGALOGO.LG(O)	Code used to display startup logo for EGA displays
EGALOGO.RL(E)	Startup logo screen for EGA displays
EGAOEM.FO(N)	OEM system font for EGA displays
EGASYS.FO(N)	System font for EGA displays
GDI.EX(E)	Graphics device interface
HERCLOGO.RL(E)	Startup logo screen for Hercules displays
HERCULES.2G(R)	Standard-mode grabber for Hercules displays
HPEBIOS.38(6)	EBIOS virtual device driver for Hewlett-Packard systems
KBDHP.DR(V)	Keyboard driver for Hewlett-Packard systems
MMSOUND.DR(V)	Multimedia sound driver
MSCVMD.38(6)	Mouse systems virtual mouse device driver
MSNET.DR(V)	Generic network driver (supports 3Com 3+Share, 3Com 3+Open LAN Manager, Banyan VINES 4.0, Microsoft LAN Manager 1.X, Microsoft LAN Manager 2.0 Basic, Microsoft Network, and IBM PC LAN)
OLIGRAB.2G(R)	Standard-mode grabber for Olivetti/AT&T PVC displays
PMSPL20.DL(L)	Library to support Microsoft LAN Manager printer API
SETUP.EX(E)	Windows setup program
SETUP.HL(P)	Windows setup help file
SETUP.INF	Windows setup information file
SETUP.SHH	Template file for automated setup
SETUP.TXT	User information regarding setup and setup-related problems
SYSTEM.DR(V)	Generic system driver used for most computer systems
SYSTEM.SR(C)	Source template used to create SYSTEM.INI
TIGA.DR(V)	Supports TIGA displays
V7VDD.38(6)	Video Seven virtual display device driver
VDD8514.38(6)	8514/a virtual display device driver
VDDCT441.38(6)	82C441 VGA virtual display device driver
VDDEGA.38(6)	EGA virtual display device driver
VDDTIGA.38(6)	TIGA virtual display device driver
VDDVGA30.38(6)	VGA (Version 3.0) virtual display device driver
VDDXGA.38(6)	XGA virtual display device driver
VGA.3G(R)	VGA virtual display device driver
VGASYS.FO(N)	System font for VGA displays

Table A.6 Contents of Windows Installation Disks (Continued)

WIN.CN(F)	Windows startup code (used by SETUP.EXE to create WIN.COM)
WIN.SR(C)	Source file used to create WIN.INI
WINHELP.EX(E)	Windows help engine
XGA.DR(V)	Supports XGA displays
XMSMMGR.EXE	Extended-memory manager used by Windows Setup if HIMEM.SYS is not detected

Disk 2 (127 files)

File Name	*Usage*
386MAX.VX(D)	Standard mode virtual device driver for 386MAX
8514FIX.FO(N)	Fixed system font for 8514/a displays
8514OEM.FO(N)	OEM system font for 8514/a displays
BANINST.38(6)	Virtual device driver for Banyan VINES 4.0 instancing
BLUEMAX.VX(D)	Virtual device driver for BlueMAX (used with PS/2 systems)
CGA.2G(R)	Standard-mode grabber for CGA displays
CGA40850.FO(N)	Multilingual font for windowed DOS applications with CGA 40-column displays
CGA40WOA.FO(N)	U.S. font for windowed DOS applications with CGA 40-column displays
CGA80850.FO(N)	Multilingual font for windowed DOS applications with CGA 80-column displays
CGA80WOA.FO(N)	U.S. font for windowed DOS applications with CGA 80-column displays
CGALOGO.LG(O)	Code used to display startup logo for CGA displays
CGALOGO.RL(E)	Startup logo screen for CGA displays
CONTROL.HL(P)	Control Panel help file
DECNB.38(6)	DEC Pathworks virtual device driver
DECNET.38(6)	DEC Pathworks virtual device driver
DISK2	Disk ID for Windows Startup
DOSAPP.FO(N)	Code page 437 scalable screen font for windowed DOS applications
DOSX.EX(E)	Extended memory manager for standard mode
EGA.SY(S)	EGA DOS device driver
EGA40850.FO(N)	Multilingual font for windowed DOS applications with EGA low-resolution displays
EGA40WOA.FO(N)	U.S. font for windowed DOS applications with EGA low-resolution displays
EGA80850.FO(N)	Multilingual font for windowed DOS applications with EGA higher-resolution displays

Table A.6 Contents of Windows Installation Disks (Continued)

EGA80WOA.FO(N)	U.S. font for windowed DOS applications with EGA higher-resolution displays
EGAHIBW.DR(V)	Supports EGA displays with 128K of RAM
EGAMONO.2G(R)	Standard-mode grabber for EGA monochrome displays
EGAMONO.DR(V)	Supports EGA monochrome displays
EGAMONO.LG(O)	Code used to display startup logo for EGA monochrome displays
EGAMONO.RL(E)	Startup logo screen for EGA monochrome displays
HERC.3G(R)	Enhanced-mode grabber for Hercules displays
HERC850.FO(N)	Multilingual font for windowed DOS applications with Hercules displays
HERCLOGO.LG(O)	Code used to display startup logo for Hercules displays
HERCWOA.FO(N)	U.S. font for windowed DOS applications with Hercules displays
HPMOUSE.DR(V)	Hewlett-Packard mouse driver
HPSYSTEM.DR(V)	Standard-mode driver for the HP Vectra system
IPX.OB(J)	Workstation communications driver for Novell NetWare
IPXODI.CO(M)	Workstation communications driver for Novell NetWare (ODI model)
KBDBE.DL(L)	Belgian keyboard library
KBDCA.DL(L)	French-Canadian keyboard library
KBDDA.DL(L)	Danish keyboard library
KBDDV.DL(L)	U.S. Dvorak keyboard library
KBDFC.DL(L)	Canadian multilingual keyboard library
KBDFI.DL(L)	Finnish keyboard library
KBDFR.DL(L)	French keyboard library
KBDGR.DL(L)	German keyboard library
KBDIC.DL(L)	Icelandic keyboard library
KBDIT.DL(L)	Italian keyboard library
KBDLA.DL(L)	Latin-American keyboard library
KBDMOUSE.DR(V)	Olivetti/AT&T keyboard mouse driver
KBDNE.DL(L)	Dutch keyboard library
KBDNO.DL(L)	Norwegian keyboard library
KBDPO.DL(L)	Portuguese keyboard library
KBDSF.DL(L)	Swiss-French keyboard library
KBDSG.DL(L)	Swiss-German keyboard library
KBDSP.DL(L)	Spanish keyboard library
KBDSW.DL(L)	Swedish keyboard library
KBDUK.DL(L)	British keyboard library
KBDUS.DL(L)	U.S. keyboard library
KBDUSX.DL(L)	U.S./International keyboard library
KEYBOARD.DR(V)	Supports standard keyboards (installed by default)

Table A.6 Contents of Windows Installation Disks (Continued)

KRNL286.EX(E)	Windows kernel for standard mode
KRNL386.EX(E)	Windows kernel for enhanced mode
LANGDUT.DL(L)	Dutch language driver
LANGENG.DL(L)	General international language driver
LANGFRN.DL(L)	French language driver
LANGGER.DL(L)	German language driver
LANGSCA.DL(L)	Finnish/Icelandic/Norwegian/Swedish language driver
LANGSPA.DL(L)	Spanish language driver
LANMAN.DR(V)	Supports LAN Manager 2.0
LANMAN.HL(P)	Help file for LAN Manager 2.0
LANMAN10.38(6)	Virtual device driver for LAN Manager 1.0
LMOUSE.CO(M)	DOS-level Logitech mouse driver
LMOUSE.DR(V)	Supports Logitech serial mouse
LSL.CO(M)	Workstation link support layer (ODI model) for Novell Netware
LVMD.38(6)	Virtual device driver for Logitech mouse
LZEXPAND.DL(L)	File expansion library used by Control Panel to uncompress files on Windows installation disks
MOUSE.DR(V)	Supports Logitech or IBM PS/2 bus mouse, and Microsoft or IBM PS/2 serial mouse
MSC3BC2.DR(V)	Supports Mouse Systems three-button mouse on COM2
MSCMOUSE.DR(V)	Supports Mouse Systems serial and bus mouse
NETAPI20.DL(L)	Microsoft LAN Manager network API library
NETWARE.DR(V)	Supports Novell Netware 2.10 or above, and Novell NetWare386
NETWARE.HL(P)	Help file for Novell Netware
NETX.CO(M)	Workstation shell for Novell Netware
NOMOUSE.DR(V)	Supports Windows on systems with no mouse attached
NWPOPUP.EX(E)	TSR to support Novell network pop-up messages on workstations
OLIBW.DR(V)	Supports Olivetti/AT&T monochrome and PVC displays
PCSA.DR(V)	Supports DEC Pathworks networks
PLASMA.3G(R)	Enhanced-mode grabber for Compaq portables with plasma display
PLASMA.DR(V)	Supports plasma displays on Compaq portables
POWER.DR(V)	Supports portable systems using advanced power management features
POWER.HL(P)	Advanced power management help file
SL.DL(L)	Library for advanced power management
SL.HL(P)	Additional help file for advanced power management
SUPERVGA.DR(V)	Supports Super VGA (800x600 resolution, 16-color) displays

Table A.6 Contents of Windows Installation Disks (Continued)

TBMI2.CO(M)	Supports task switching on Novell networks
TIGAWIN.RL(M)	Code for TIGA firmware
USER.EX(E)	User portion of Windows core
V7VGA.3G(R)	Enhanced-mode support for Video Seven VGA displays
V7VGA.DR(V)	Supports Video Seven VGA displays
VDDCGA.38(6)	Virtual device driver for CGA displays
VDDHERC.38(6)	Virtual device driver for Hercules displays
VER.DL(L)	Version resource and file installation library
VGA.DR(V)	Supports standard VGA displays
VGA30.3G(R)	Enhanced-mode grabber for VGA (Version 3.0) display driver
VGA850.FO(N)	Multilingual font for windowed DOS applications with VGA displays
VGA860.FO(N)	Portuguese font for windowed DOS applications with VGA displays
VGA861.FO(N)	Icelandic font for windowed DOS applications with VGA displays
VGA863.FO(N)	French-Canadian font for windowed DOS applications with VGA displays
VGA865.FO(N)	Norwegian/Danish font for windowed DOS applications with VGA displays
VGACOLOR.2G(R)	Standard-mode grabber for VGA color displays
VGADIB.3G(R)	Enhanced-mode grabber for 8514/a monochrome displays
VGAFIX.FO(N)	Fixed font for VGA displays
VGALOGO.LG(O)	Code used to display startup logo for VGA displays
VGALOGO.RL(E)	Startup logo screen for VGA displays
VGAMONO.2G(R)	Standard-mode grabber for VGA monochrome displays
VGAMONO.DR(V)	Supports VGA monochrome displays
VGAOEM.FO(N)	OEM screen font for VGA displays
VIPX.38(6)	Virtual IPX device support for Novell networks
VNETWARE.38(6)	Virtual device support for Novell networks
VPOWERD.38(6)	Virtual advanced power management device support for portable computers
WINDOWS.LO(D)	Load module for 386MAX and BlueMAX
WINPOPUP.EX(E)	Supports pop-up messages for LAN Manager 2.0 and compatible networks
WINPOPUP.HL(P)	Help file for pop-up messages (used with WINPOPUP.EXE)
XLAT850.BI(N)	Translation table for multilingual fonts
XLAT860.BI(N)	Translation table for Portuguese font
XLAT861.BI(N)	Translation table for Icelandic font

Table A.6 Contents of Windows Installation Disks (Continued)

XLAT863.BI(N)	Translation table for French-Canadian font
XLAT865.BI(N)	Translation table for Norwegian/Danish font

Disk 3 (63 files)

File Name	*Usage*
256COLOR.BM(P)	256-color wallpaper file
APPS.HL(P)	Help file for applications known to be partially or fully incompatible with Windows 3.1
ARGYLE.BM(P)	Argyle wallpaper file
CALC.EX(E)	Calculator program
CALC.HL(P)	Help file for Calculator
CALENDAR.EX(E)	Calendar program
CALENDAR.HL(P)	Help file for Calendar
CANYON.MI(D)	Canyon MIDI sound file
CARDFILE.EX(E)	Cardfile program
CHARMAP.EX(E)	Character Map utility
CHIMES.WA(V)	Chimes sound file
CHORD.WA(V)	Chord sound file
CLOCK.EX(E)	Clock program
DISK3	Disk ID for Windows Setup program
EGYPT.BM(P)	Egypt wallpaper file
EXPAND.EXE	DOS-level file expansion utility used to uncompress files on Windows installation disks
FLOCK.BM(P)	Flock wallpaper file
GLOSSARY.HL(P)	Help file of Windows terms and definitions
HERCULES.DR(V)	Supports Hercules displays
MORICONS.DL(L)	Icons for non-Windows applications
MPLAYER.EX(E)	Media Player program
MPLAYER.HL(P)	Help file for Media Player
MSADLIB.DR(V)	Supports MIDI Adlib compatibles
NETWORKS.WR(I)	Readme file containing setup and troubleshooting information for networks
NOTEPAD.EX(E)	Notepad program
PACKAGER.EX(E)	Object Packager program
PACKAGER.HL(P)	Help file for Object Packager
PBRUSH.DL(L)	Paintbrush program support library
PBRUSH.EX(E)	Paintbrush program
PBRUSH.HL(P)	Help file for Paintbrush
PIFEDIT.HL(P)	Help file for PIF Editor
PRINTERS.WR(I)	Readme file containing setup and troubleshooting information for printers
PRINTMAN.HL(P)	Help file for Print Manager

Table A.6 Contents of Windows Installation Disks (Continued)

README.WR(I)	Readme file containing general troubleshooting information for hardware and some applications
RECORDER.DL(L)	Recorder support library
RECORDER.EX(E)	Recorder program
SCRNSAVE.SC(R)	Blank screen saver
SETUP.INI	Initialization file for Windows Setup
SNDBLST.DR(V)	Supports SoundBlaster 1.5
SNDBLST2.DR(V)	Supports SoundBlaster 2.0
SOL.EX(E)	Solitaire program
SOL.HL(P)	Help file for Solitaire
SOUNDREC.EX(E)	Sound Recorder program
SOUNDREC.HL(P)	Help file for Sound Recorder
SSFLYWIN.SC(R)	Flying windows screen-saver file
SSMARQUE.SC(R)	Marquee screen-saver file
SSMYST.SC(R)	Mystify screen-saver file
SYSINI.WR(I)	Readme file explaining SYSTEM.INI settings
TADA.WA(V)	Tada sound file
TERMINAL.HL(P)	Help file for Terminal
THATCH.BM(P)	Thatch wallpaper file
TIMER.DR(V)	Supports multimedia timers
VADLIBD.38(6)	Virtual DMA device driver for MIDI Adlibs
VSBD.38(6)	Virtual device driver for SoundBlaster
VTDAPI.38(6)	Virtual device driver for multimedia timers
WINFILE.EX(E)	File Manager program
WINHELP.HL(P)	Help file for basic Windows operations
WININI.WR(I)	Readme file explaining WIN.INI settings
WINMINE.EX(E)	MineSweeper program
WINTUTOR.DA(T)	Data file for Windows on-line tutorial
WINTUTOR.EX(E)	Windows on-line tutorial program
WRITE.EX(E)	Write program
WRITE.HL(P)	Help file for Write

Disk 4 (57 files)

File Name	*Usage*
APPS.IN(F)	PIF settings and other information for non-Windows applications
ARCADE.BM(P)	Arcade wallpaper file
ARCHES.BM(P)	Arches wallpaper file
CARDFILE.HL(P)	Help file for Cardfile
CARS.BM(P)	Cars wallpaper file
CASTLE.BM(P)	Castle wallpaper file
CHARMAP.HL(P)	Help file for Character Map

Table A.6 Contents of Windows Installation Disks (Continued)

CHITZ.BM(P)	Chitz wallpaper file
CLIPBRD.HL(P)	Help file for Clipboard Viewer
COMMDLG.DL(L)	Library of common Windows dialogs
DDEML.DL(L)	Dynamic Data Exchange Management Library
DISK4	Disk ID for Windows Setup
DRIVERS.CP(L)	Control Panel support for installing printer drivers
DRWATSON.EX(E)	Windows error-detection program
DSWAP.EX(E)	Standard-mode support for non-Windows applications
EMM386.EX(E)	DOS-level expanded memory manager
HONEY.BM(P)	Honey wallpaper file
LEAVES.BM(P)	Leaves wallpaper file
MAIN.CP(L)	Main Control Panel extension
MARBLE.BM(P)	Marble wallpaper file
MCICDA.DR(V)	Media control interface (MCI) CD audio
MCISEQ.DR(V)	MCI driver for MIDI devices
MCIWAVE.DR(V)	MCI driver for waveform audio
MIDIMAP.CF(G)	MIDI Mapper Control Panel extension
MIDIMAP.DR(V)	Supports MIDI Mapper Control Panel extension
MMSYSTEM.DL(L)	Multimedia system library
MONOUMB2.38(6)	Virtual display device driver for generic monochrome displays
MOUSE.CO(M)	DOS-level Microsoft Mouse (8.2) driver
MOUSE.SY(S)	DOS-level Microsoft Mouse (8.2) driver
MOUSEHP.CO(M)	DOS-level mouse driver for Hewlett-Packard systems
MOUSEHP.SY(S)	DOS-level mouse driver for Hewlett-Packard systems
MPU401.DR(V)	Supports MIDI MPU401 devices
MSD.EXE	Microsoft Diagnostics program
NOTEPAD.HL(P)	Help file for Notepad
OLESVR.DL(P)	OLE server library
PRINTMAN.EX(E)	Print Manager program
PROGMAN.EX(E)	Program Manager shell
RAMDRIVE.SY(S)	RAMDrive utility
RECORDER.HL(P)	Help file for Recorder
REDBRICK.BM(P)	Red brick wallpaper file
REGEDIT.EX(E)	Registration Info Editor program
REGEDIT.HL(P)	Help file for Registration Info Editor
RIVETS.BM(P)	Rivets wallpaper file
SHELL.DL(L)	Shell library
SMARTDRV.EX(E)	SMARTDrive 4.0 disk caching utility
SND.CP(L)	Sound extension for Control Panel
SQUARES.BM(P)	Squares wallpaper file
SSSTARS.SC(R)	Stars screen-saver file

Table A.6 Contents of Windows Installation Disks (Continued)

TARTAN.BM(P)	Tartan wallpaper file
TERMINAL.EX(E)	Terminal program
WIN386.EX(E)	Windows memory manager and loader for enhanced mode operations
WIN87EM.DL(L)	Library for 80x87 coprocessor emulation
WINFILE.HL(P)	Help file for File Manager
WINLOGO.BM(P)	Windows logo wallpaper file
WINMINE.HL(P)	Help file for Mine Sweeper
WINOA386.MO(D)	Supports non-Windows applications in enhanced mode
ZIGZAG.BM(P)	Zigzag wallpaper file

Disk 5 (64 files)

File Name	*Usage*
ARIAL.FO(T)	Arial TrueType font initialization file
ARIAL.TT(F)	Arial TrueType font file
ARIAL.FO(N)	Arial screen font for EGA resolution displays
ARIALBD.FO(T)	Arial bold TrueType font initialization file
ARIALBD.TT(F)	Arial bold TrueType font file
ARIALBI.FO(T)	Arial bold italic font initialization file
ARIALBI.TT(F)	Arial bold italic TrueType font file
ARIALI.FO(T)	Arial italic TrueType font initialization file
ARIALI.TT(F)	Arial italic TrueType font file
CLIPBRD.EX(E)	Clipboard Viewer utility
CONTROL.EX(E)	Control Panel utility
CONTROL.INF	Information used by Control Panel to create many settings
CONTROL.SR(C)	Template used to create CONTROL.INI
COUR.FO(T)	Courier TrueType font initialization file
COUR.TT(F)	Courier TrueType font file
COURB.FO(N)	Courier bold TrueType font initialization file
COURBD.FO(T)	Courier bold TrueType font initialization file
COURBD.TT(F)	Courier bold TrueType font file
COURBI.TT(F)	Courier bold italic TrueType font initialization file
COURE.FO(N)	Courier screen font for VGA displays
COURF.FO(N)	Courier screen font for 8514/a displays
COURI.FO(T)	Courier italic TrueType font initialization file
COURI.TT(F)	Courier italic TrueType font file
DING.WA(V)	Ding sound file
DISK5	Disk ID for Windows Setup
HIMEM.SY(S)	DOS-level extended memory manager
MMTASK.TS(K)	Multimedia background task manager
MODERN.FO(N)	Modern vector screen font

Table A.6 Contents of Windows Installation Disks (Continued)

MSD.IN(I)	Initialization file for Microsoft Diagnostics program
OLECLI.DL(L)	OLE client library
PIFEDIT.EX(E)	PIF Editor utility
PROGMAN.HL(P)	Help file for Program Manager
REGEDITV.HL(P)	Help file for advanced-level Registration Info Editor
ROMAN.FO(N)	Roman vector screen font
SCRIPT.FO(N)	Script vector screen font
SERIFB.FO(N)	MS Serif screen font for EGA displays
SERIFE.FO(N)	MS Serif screen font for VGA displays
SERIFF.FO(N)	MS Serif screen font for 8514/a displays
SETUP.RE(G)	Registration Database template
SMALLB.FO(N)	Small screen font for EGA displays
SMALLE.FO(N)	Small screen font for VGA displays
SMALLF.FO(N)	Small screen font for 8514/a displays
SSERIFB.FO(N)	MS Sans Serif screen font for EGA displays
SSERIFE.FO(N)	MS Sans Serif screen font for VGA displays
SSERIFF.FO(N)	MS Sans Serif screen font for 8514/a displays
SYMBOLB.FO(N)	Symbol screen font for EGA displays
SYMBOLE.FO(N)	Symbol screen font for VGA displays
SYMBOLF.FO(N)	Symbol screen font for 8514/a displays
SYSEDIT.EX(E)	System Configuration Editor utility
TASKMAN.EX(E)	Task List utility
TIMES.FO(T)	Times New Roman TrueType font installation file
TIMES.TT(F)	Times New Roman TrueType font file
TIMESB.FO(N)	Times screen font for EGA displays
TIMESBD.FO(T)	Times New Roman TrueType font initialization file
TIMESBD.TT(F)	Times New Roman bold TrueType font file
TIMESBI.FO(T)	Times New Roman bold italic TrueType font initialization file
TIMESBI.TT(F)	Times New Roman bold italic TrueType font file
TIMESI.FO(T)	Times New Roman italic TrueType font initialization file
TIMESI.TT(F)	Times New Roman italic TrueType font file
TOOLHELP.DL(L)	Tool Helper library
WIN386.PS(2)	Virtual device support for PS/2 systems
WINOLDAP.MO(D)	Standard-mode support for non-Windows applications
WINVER.(EXE)	Windows version utility (the entire EXE extension is appended during installation)
WSWAP.EX(E)	Standard-mode support for Windows applications

Disk 6 (110 files)

File Name	*Usage*
40291730.WP(D)	PostScript description file for IBM LaserPrinter 4029 (17 fonts)

Table A.6 Contents of Windows Installation Disks (Continued)

40293930.WP(D)	PostScript description file for IBM LaserPrinter 4029 (39 fonts)
CANON10E.DR(V)	Supports Canon Bubble-Jet BJ-10e printers
CANON130.DR(V)	Supports Canon Bubble-Jet BJ-130e printers
CANON330.DR(V)	Supports Canon Bubble-Jet BJ-300/330 printers
CAN_ADF.EX(E)	Font installer for Canon LBP-8 II and LBPIII printers
CIT24US.DR(V)	Citizen 24-pin printers
CIT9US.DR(V)	Citizen 9-pin printers
CITOH.DR(V)	C-Itoh 8510 and AT&T 470/475 printers
COURBI.FO(T)	Courier bold italic TrueType font initialization file
DEC1150.WP(D)	PostScript description file for Digital DEClaser 1150 printers
DEC2150.WP(D)	PostScript description file for Digital DEClaser 2150 printers
DEC2250.WP(D)	PostScript description file for Digital DEClaser 2250 printers
DEC3250.WP(D)	PostScript description file for Digital DEClaser 3250 printers
DECCOLOR.WP(D)	PostScript description file for Digital ColorMate PS printers
DECLPS20.WP(D)	PostScript description file for Digital LPS Printer Server
DICONIX.DR(V)	Supports Kodak Diconix printers
DISK6	Disk ID for Windows Setup
DM309.DR(V)	Supports Olivetti DM 309 printers
DMCOLOR.DL(L)	Universal color printing library
EPL75523.WP(D)	PostScript description file for Epson EPL-7500 printers
EPSON24.DR(V)	Supports Epson 24-pin printers
EPSON9.DR(V)	Supports Epson 9-pin printers
ESCP2.DR(V)	Supports Epson ESCP2 printers
EXECJET.DR(V)	Supports IBM ExecJet printers
FINSTALL.DL(L)	Font installer for HPPCL5/A printers
FINSTALL.HL(P)	Help file for HPPCL5/A font installer
FUJI24.DR(V)	Supports Fujitsu 24-pin printers
FUJI9.DR(V)	Supports Fujitsu 9-pin printers
GENDRV.DL(L)	Generic printer library
HERMES_1.WP(D)	PostScript description file for Hermes H 606 PS (13 fonts) printers
HERMES_2.WP(D)	PostScript description file for Hermes H 606 PS (35 fonts) printers
HPDSKJET.DR(V)	Supports Hewlett-Packard DeskJet printers
HPELI523.WP(D)	PostScript description file for HP LaserJet IIISi PostScript printers

Table A.6 Contents of Windows Installation Disks (Continued)

HPIID522.WP(D)	PostScript description file for HP LaserJet IID PostScript printers
HPIII522.WP(D)	PostScript description file for HP LaserJet III PostScript printers
HPIIP522.WP(D)	PostScript description file for HP LaserJet IIP PostScript printers
HPPCL.DR(V)	Supports HP LaserJet II printers
HPPCL5A.DR(V)	Supports HP LaserJet III printers
HPPCL5A.HL(P)	Help file for HP LaserJet III printer driver
HPPCL5OP.HL(P)	Additional help file for HP LaserJet III printer driver
HPPLOT.DR(V)	Supports HP Plotters
HP_3D522.WP(D)	PostScript description file for HP LaserJet IIID PostScript printers
HP_3P522.WP(D)	PostScript description file for HP LaserJet IIIP PostScript printers
IBM17521.WP(D)	PostScript description file for IBM 4019 (17 fonts) printers
IBM39521.WP(D)	PostScript description file for IBM 4019 (39 fonts)
IBM4019.DR(V)	Supports IBM Laser Printer 4019 printers
IBM5204.DR(V)	Supports IBM Quickwriter 5204 printers
IBMCOLOR.DR(V)	Supports IBM Color printers
L200230&.WP(D)	PostScript description file for Linotronic 200/230 devices
L330_52&.WP(D)859	PostScript description file for Linotronic 330 devices
L530_52&.WP(D)	PostScript description file for Linotronic 530 devices
L630_52&.WP(D)	PostScript description file for Linotronic 630 devices
LBPII.DR(V)	Supports Canon LBP-8 II printers
LBPIII.DR(V)	Supports Canon LPPIII printers
MT_TI101.WP(D)	PostScript description file for Microtek TrueLaser printers
N2090522.WP(D)	PostScript description file for NEC Silentwriter 2 90 printers
N2290520.WP(D)	PostScript description file for NEC Silentwriter 2 290 printers
N2990523.WP(D)	PostScript description file for NEC Silentwriter 2 990 printers
N890X505.WP(D)	PostScript description file for NEC Silentwriter LC890XL printers
N890_470.WP(D)	PostScript description file for NEC Silentwriter LC890 printers
NCM40519.WP(D)	PostScript description file for NEC Colormate PS/40 printers
NCM80519.WP(D)	NEC Colormate PS/80 printers
NEC24PIN.DR(V)	Supports NEC 24-pin printers

Table A.6 Contents of Windows Installation Disks (Continued)

O5241503.WP(D)	PostScript description file for OceColor G5241 PS printers
O5242503.WP(D)	PostScript description file for OceColor G5242 PS printers
OKI24.DR(V)	Supports Okidata 24-pin printers
OKI9.DR(V)	Supports Okidata 9-pin printers
OKI9IBM.DR(V)	Supports Okidata 9-pin (IBM model) printers
OL840518.WP(D)	PostScript description file for Oki OL840/PS printers
OLIVETI1.WP(D)	PostScript description file for Olivetti PG 306 PS 913 fonts) printers
OLIVETI2.WP(D)	PostScript description file for Olivetti PG 306 PS (35 fonts) printers
P4455514.WP(D)	PostScript description file for Panasonic KX-P4455 printers
PAINTJET.DR(V)	Supports HP PaintJet printers
PANSON24.DR(V)	Supports Panasonic 24-pin printers
PANSON9.DR(V)	Supports Panasonic 9-pin printers
PG306.DR(V)	Supports PG 306 printers
PHIIPX.WP(D)	PostScript description file for Phaser II PX printers
PROPRINT.DR(V)	Supports IBM Pro series printers
PROPRN24.DR(V)	Supports IBM Pro 24-pin series printers
PRTUPD.INF	Information used by Windows Setup to upgrade printer drivers
PS1.DR(V)	Supports IBM PS/1 printers
PSCRIPT.DR(V)	Generic PostScript printer driver
PSCRIPT.HL(P)	Help file for PostScript printers
Q2200510.WP(D)	PostScript description file for QMS-PS 2200 printers
Q820_517.WP(D)	PostScript description file for QMS-PS 820 printers
QWIII.DR(V)	Supports IBM QuietWriter III printers
SEIKO_04.WP(D)	PostScript description file for Seiko ColorPoint PS Model 04 printers
SEIKO_14.WP(D)	PostScript description file for Seiko ColorPoint PS Model 14 printers
SF4019.EX(E)	Font installer for IBM Laser Printer 4019
SFINST.EX(E)	Font installer for PG 306 printers
SYMBOL.FO(T)	Symbol TrueType font initialization file
SYMBOL.TT(F)	Symbol TrueType font file
TESTPS.TX(T)	Settings and memory test file for PostScript printers
THINKJET.DR(V)	Supports HP ThinkJet printers
TI850.DR(V)	Supports TI 850/855 printers
TIM17521.WP(D)	PostScript description file for TI microLaser PS17 printers
TIM35521.WP(D)	PostScript description file for TI microLaser PS17 printers

Table A.6 Contents of Windows Installation Disks (Continued)

TKPHZR21.WP(D)	PostScript description file for Phaser II PX I printers
TKPHZR31.WP(D)	PostScript description file for Phaser III PX I printers
TOSHIBA.DR(V)	Supports Toshiba P351/P1351 printers
TRIUMPH1.WP(D)	PostScript description file for Triumph Adler SDR 7706 PS (13 fonts) printers
TRIUMPH2.WP(D)	PostScript description file for Triumph Adler SDR 7706 PS printers
TTY.DR(V)	Generic/Text Only printer driver
TTY.HL(P)	Help for Generic/Text Only printer driver
U9415470.WP(D)	PostScript description file for Unisys AP9415 printers
UNIDRV.DL(L)	Universal printer driver library
UNIDRV.HL(P)	Help file for universal printer drivers
WINGDING.FO(T)	Wingdings TrueType font initialization file
WINGDING.TT(F)	Wingdings TrueType font file

Using EXPAND to Uncompress Files on the Windows Installation Disks

Although SETUP.EXE contains a built-in file expansion (uncompression) utility and the Control Panel uses the LZEPAND.DLL library to uncompress files, you might encounter situations where you will need to expand Windows files yourself.

For example, if you accidentally delete a file from the WINDOWS or WINDOWS\SYSTEM directory, you might have to recover the file by copying and expanding it from the appropriate Windows installation disk. Another example might be the need to reinstall a Windows 3.0 driver as a temporary workaround to a hardware-related problem. A third example might involve the need to reinstall a driver without using Control Panel. (Control Panel often takes additional steps when it installs drivers, fonts, and other files, such as updating WIN.INI or other .INI files.) If you simply want to reinstall a file—and you specifically don't want Control Panel to update .INI files—you should use EXPAND.

The EXPAND.EXE file appears in uncompressed format on Disk 6 of the Windows installation disks, and is copied into the WINDOWS directory during setup. Here is the format to use with EXPAND:

```
EXPAND -R SOURCE DESTINATION
```

The -R switch can be placed before or after the source path and file name and tells EXPAND to automatically rename the file(s) being expanded. For example, the following command uncompresses the file XGA.DR_ stored on the disk in drive B and copies it to the WINDOWS directory of drive C, then automatically renames the file to XGA.DRV:

```
EXPAND -R B:XGA.DR_ C:\WINDOWS
```

You can include wildcards in the source if you want to expand multiple files. For instance, the following EXPAND command is valid:

```
EXPAND -R B:*.DR* C:\WINDOWS
```

This command tells EXPAND to copy, uncompress, and rename all .DR_ files located on the disk in drive B, then write the files to the WINDOWS directory on the hard drive.

Uncompressed Files on Windows Installation Disks

Table A.7 lists all of the files on the Windows installation disks that are provided in uncompressed format. (You don't need to use the EXPAND utility when you copy these files to your system's hard drive.)

For more information on the purpose of each of these files, locate the file name in Table A.6.

General Windows Setup Troubleshooting

The following sections describe several problems that can occur during the setup of Windows 3.1.

Windows Setup Prematurely Requests Disk 5

If Windows 3.1 Setup finds a SETUP.EXE file in the WINDOWS\SETUP directory, it will incorrectly prompt for disk 5 of the Windows installation disks; Setup may ask you to insert disk 5 when none or only some files on disk 3 have been installed, and will skip disk 4.

The Windows Setup program will look for a SETUP.EXE file in the WINDOWS directory, and delete this file. However, it will not delete a SETUP.EXE file found in the WINDOWS\SYSTEM directory. As a result, when the Setup program is called again during its operation, it will try to load from the WINDOWS\SYSTEM directory if this directory contains a SETUP.EXE file for Windows 3.0 or for another application.

To correct this problem, delete any SETUP.EXE file that exists in the WINDOWS\SYSTEM directory.

Table A.7 Uncompressed Files on Windows Installation Disks

File Name	*Disk #*
DISK(x)	All disks (This is just a disk ID used by the Windows Setup program to verify that you've inserted the correct disk, and serves no other purpose for users)
SETUP.EXE	Disk 1
SETUP.INF	Disk 1
SETUP.SHH	Disk 1
SETUP.TXT	Disk 1
XMSMMGR.EXE	Disk 1
EXPAND.EXE	Disk 3
SETUP.INI	Disk 3
MSD.EXE	Disk 4
CONTROL.INF	Disk 5
PRTDUPD.INF	Disk 6

Windows 3.1 Setup Will Not Upgrade Windows 2.X Versions

Although documentation for Windows 3.1 indicates that it can upgrade Windows over Windows versions 2.X, this is not really true. Windows can install version 3.1 on a system that contains a 2.X version, but only in a different directory; Windows 3.1 cannot upgrade a Windows 2.X version. To upgrade Windows 3.1, either delete all files for the Windows 2.X version and then perform a "clean installation," or install Windows 3.1 in a separate directory if you want to retain the Windows 2.X version.

You might also see an error message if Windows 3.1 tries to upgrade a Windows 3.0 configuration that in turn was used to upgrade a Windows 2.X version. In this case also, you should either delete the files for Windows 2.X or install Windows 3.1 in a different directory.

Windows Setup (for version 3.1) Cannot Be Run within Windows 3.0

If you try to run the Windows 3.1 Setup program (to upgrade Windows 3.1 from Windows 3.0) by using the Run command from within Windows 3.0, you will receive the following error message:

```
This program requires a new version of Windows
```

Whenever you run Windows Setup from within Windows, Setup tries to run in maintenance mode, which will fail if you try to run Windows 3.1 Setup from within Windows 3.0. Instead, exit to the DOS prompt and start Windows Setup from there.

Windows Setup Can't Install on Some PS/1 Systems

Some PS/1 systems boot automatically by using DOS code stored in ROM. In these cases, Windows Setup will display the following message during installation:

```
Setup cannot find or update your system files on drive D:.

   If you started your system from a floppy disk, be sure to remove
write protection from your system startup disk. Then, insert the disk
into drive A and choose Retry. To copy system files to your Windows
directory choose Cancel.

Retry    Cancel
```

The above message occurs because CONFIG.SYS and AUTOEXEC.BAT are hard-coded into the system and the Windows Setup program is unable to modify them. When this problem is encountered, Windows writes the proposed changes to the files CONFIG.WIN and AUTOEXEC.WIN, and stores the files in the WINDOWS directory. For Windows to run on these systems, you must copy and rename the files to the root directory of the hard disk. The following command demonstrates how to copy and rename CONFIG.WIN to the hard disk:

```
COPY C:\WINDOWS\CONFIG.WIN C:\CONFIG.SYS
```

Windows Setup Does Not Preserve Printer Codes for Generic Printer Driver

If you have installed the Generic Text printer driver under Windows 3.0 and then upgrade to Windows 3.1, you may receive the following error message:

```
Unable to retain printer codes.
Use default printer codes.
```

If you see these error messages, you may need to reinstall the Generic Text printer driver for Windows 3.0. Procedures for doing this are explained in Chapter 13.

Network Setup Information

Chapter 5 focuses on various setup options and techniques that you can use to install Windows 3.1 for network use. The following sections provide additional information that can be of use to you if you are setting up one or more copies of Windows 3.1 to run on a network.

NETWORKS.WRI

NETWORKS.WRI contains important setup information for many networks. In fact, the majority of problems that can occur during or as a result of a network setup of Windows are explained in this file. NETWORKS.WRI is copied to the WINDOWS directory during the installation of Windows 3.1 and appears as a compressed file (NETWORKS.WR_) on Disk 3 of the Windows installation disks. It is extremely important that you read this file before you begin a network installation of Windows 3.1. Table A.8 lists the networks for which setup-related information is provided in NETWORKS.WRI.

Table A.8 Networks Discussed in NETWORKS.WRI

3Com 3+Share
3Com 3+Open LAN Manager
Artisoft LANtastic (general)
Artisoft LANtastic 3.x
Artisoft LANtastic 4.x
Banyan VINES (general)
Banyan VINES 4.0x
Banyan VINES 4.1x
DEC Pathworks
IBM OS/2 LAN Server (general)
IBM OS/2 LAN Server 1.2 and 1.3
IBM OS/2 LAN Server 2.x
IBM PC LAN Program
NET/30 for Windows (Invisible Software)
Microsoft LAN Manager (general)
Microsoft LAN Manager Basic
Microsoft LAN Manager 1.x
Microsoft LAN Manager 2.0 Enhanced
Microsoft LAN Manager 2.1 Basic
Microsoft LAN Manager 2.1 Enhanced

Table A.8 Networks Discussed in NETWORKS.WRI (Continued)

Microsoft Network and Compatibles
Novell NetWare
TCS 10Net (from Tiara Computer Systems and formerly called DCA 10Net)
Ungermann-Bass Net/One

Setup Options (Switches)

Windows lets you use seven different switches with the SETUP command in order to modify the way the Windows Setup program will operate. Most of these switches are designed to help customize the setup process for networks. Table A.9 lists and briefly describes the switches that you can use with SETUP. See Chapter 5 for more detailed information regarding the /N, /A, and /H switches.

Table A.9 SETUP Switches

Switch	*Purpose*
/I	Tells Windows to ignore automatic hardware detection. ARCNET cards and other network devices sometimes cannot be detected accurately by the Windows Setup program. The /I switch often prevents these hardware-related errors from occurring during setup. If you are using multiple switches in a SETUP command line, the /I switch *must* appear before the /A, /N, or /H switches.
/N	Installs a shared copy of Windows from a network server. See Chapter 5 for more information regarding a network setup.
/A	Initiates an administrative setup, where all Windows files are copied and expanded to the network server, and are marked as read-only. See Chapter 5 for more information regarding an administrative setup.
/B	Sets up Windows using monochrome display.
/H:<filename>	Performs an automated setup using custom system settings specified in <filename>. See Chapter 5 for more information regarding the creation of a system settings file and for performing an automated setup (also called batch mode setup).
/O:<filename>	Specifies the SETUP.INF file to use during setup.
/S:<filename>	Specifies the SETUP.INF file to use during setup, including a path where the installation disks (files) are located.

Network-Related Setup Problems

The following sections provide troubleshooting information and tips regarding network installations of Windows. Some sections provide general information, while others relate to specific networks. In all cases, the information is either not included in NETWORKS.WRI or is provided here to clarify and expand on information in the NETWORKS.WRI file.

Use Spaces between Windows Startup Switches

The Windows Setup program requires that you insert a space between each switch in the startup command line. For example, the following startup command is erroneous—Windows will execute in monochrome mode (the /B switch) but will disregard the second "ignore hardware" (/I) switch:

```
SETUP /B/I
```

With a space between each space between each switch, however, the command line will be executed successfully:

```
SETUP /B /I
```

SETUP /A Exits when Write-protected Files Are Encountered

If Windows Setup encounters a write-protected file during installation, the setup operation will abort and you will be returned to the DOS prompt. This will most likely occur when updating Windows 3.1 over Windows 3.0 on workstations where Windows Setup needs to update one or more files that are write-protected. To work around this problem, you should use the DOS ATTRIB command to turn off the read-only attribute for *all* files in a workstation's WINDOWS and WINDOWS\SYSTEM directories, as in:

```
ATTRIB -R C:\WINDOWS\*.*
```

SETUP /N Prevents EGA Driver from Loading

When you use the SETUP /N command to install a shared copy of Windows on a workstation that has an EGA monitor, the setup program will add the following line to the workstation's CONFIG.SYS file:

```
DEVICE=<NET_DRIVE>:\WINPATH\EGA.SYS
```

The result is that EGA.SYS will not be loaded when the user starts his or her system because the network has not yet been loaded and hence the network drive is not available.

To solve this problem, copy EGA.SYS from the network drive to the workstation's local hard drive, then modify the DEVICE= line in the file CONFIG.SYS to reflect the local hard drive rather than the network drive.

Network Setup Creates a Nonworking Copy of Windows

If you are updating a copy of Windows 3.0 to Windows 3.1, Windows Setup might create a nonworking copy of Windows if either of the following two conditions exists:

- Windows 3.0 was installed using the SETUP /N switch, but Windows 3.1 is installed with no switches.
- Windows 3.0 was installed as a stand-alone copy, but Windows 3.1 is installed from the network with the /N switch.

If either of these problems occurs, you should perform a "clean installation" by deleting all files in the WINDOWS and WINDOWS\SYSTEM directories, and then rerunning the Setup program.

"Error Building WIN.COM" Message

You will receive this message if you try to perform a network installation (SETUP /N) from the Windows installation disks. A network installation can only be performed from a shared Windows directory—typically after an administrative setup has already been performed.

You might also see this message if the WIN.CNF or a required .LGO file is damaged or missing. These files are used by Windows Setup to build the WIN.COM program.

Windows Setup might also display this message if it is having a difficult time detecting network hardware correctly. Try using the /I switch to resolve this problem. Make sure you add the /I switch before you add the /N switch—for example, SETUP /I /N and *not* SETUP /N /I.

"File Copy Error" Messages

You might receive a file copy error message if Windows Setup is having a difficult time detecting network hardware correctly. ARCNET cards often cause this problem. Try using the /I switch to resolve this problem. Make sure you add the /I switch before you add the /N switch—for example, SETUP /I /N and *not* SETUP /N /I.

Installing DEC Pathworks on Diskless Workstations

Diskless workstations on a DEC Pathworks network must know how to boot the required Windows files. For instance, any drivers installed by Windows 3.1 for use at the DOS level (such as HIME.SYS, EMM386.EXE, and SMARTDRV.EXE), must be copied to the boot disk for floppy-based systems, or to the bootable share for diskless workstations.

Network Directory Not in PATH

With the following networks, Windows Setup might not identify the presence of the network correctly if the network directory is not in the workstation's PATH line: 3Com 3+Open, 3Com 3+Share LAN Manager, and Microsoft Network (and compatibles). If you have not yet installed Windows 3.1 and you are using one of these networks, add the network directory to the PATH statement before you install Windows. If you have already installed Windows and the Windows Setup dialog box (accessed by choosing the Windows Setup icon in the Main group) reports no network installed (or the wrong network installed), choose Options Change System Settings and specify the correct network. You will need to reboot your system for the change to take effect.

Network Setup Error Occurs on Workstations Booted from a Disk

During network setup, the Windows Setup program will report the following error for any workstation that is booted from a floppy disk, rather than from a hard disk:

```
Setup cannot find or cannot update your system files
on drive A. If you started your system from a floppy
disk, be sure to remove write-protection from your system startup
(boot) disk. Then insert the disk into drive A and choose Retry. To
copy system files to your Windows directory, choose Cancel.
```

If this message occurs, remove the Windows installation disk from drive A if necessary, then place the boot disk in drive A and select Retry.

"System Error: Cannot Read from Device NETWORK" Message

This error can occur with ARCNET and Western Digital/SMC ETHERNET cards that are configured to use D000 as the starting memory address. To resolve this problem, try changing the address of the card to D800. Make sure you change any related network settings to reflect this new address.

VPICDA.386 No Longer Required

Windows 3.0 was incompatible with network cards that used the hardware interrupt IRQ2 (often used by Novell network cards). To solve this problem, the VPICDA.386 virtual device driver file could be installed. Windows 3.1 no longer needs this file, because all VPICDA requirements have been incorporated into the built-in Windows VPICDA driver. Since this driver is installed by default on network workstations, you do not need to take any additional steps to prevent IRQ2 conflicts.

Windows 3.1 Requires IPX 3.1 for Novell Network Workstations in Enhanced Mode

On a Novell Network, workstations can run Windows in enhanced mode only if IPX version 3.1 is used. The required IPX.OBJ file, version 3.1, is provided on the Windows installation disks, so this version requirement usually is fulfilled by default when you install Windows 3.1 on a Novell Network.

To run Windows 3.1 on a Novell network, your network drivers must have versions equal to or higher than the ones shown in Table A.10.

Table A.10 Driver Versions Required for Novell Networks

File Name	*Version*
IPX.COM	3.1
NETX.COM	3.26 (where *X* is the major DOS version number—such as NET5.COM for DOS 5)
TBMI2.COM	2.1
IPXODI.COM	1.2
LSL.COM	1.21

Windows 3.0 Files Remain in Novell Search Paths Following Network Installation of Windows 3.1

If you try to start Windows 3.1 in enhanced mode after upgrading over Windows 3.0 using the SETUP /N switch on a Novell network, you may receive the following messages:

Invalid VxD Dynamic Link Call to Device # 0021, Service 8006
Invalid VxD Dynamic Link Call to Device # 0010, Service 8001

These messages usually indicate that one or more Windows 3.0 files are in the Novell Search or Map paths. You can easily identify these files by examining their dates. Delete any older Windows files in order to run Windows 3.1 in enhanced mode.

Appendix B: The Windows Insider Disks

If you've ever wanted "the best of the best" Windows shareware utilities on one set of disks, then the *Windows Insider Disks* were designed with your interests in mind. Shareware utilities number in the thousands and vary widely in quality. If you've spent any time examining shareware utilities, you've probably already discovered that many of these programs (possibly even the majority of them) are downright disappointing. Some programs contain bugs, while others are either poorly documented or provide no documentation. Other programs were designed specifically to fill some of the gaps and flaws in Windows 3.0, and are obsolete with Windows 3.1. However, several excellent shareware programs—updated or designed specifically for Windows 3.1—*are* available, *if you know where to find them.*

Now you don't have to spend hours of your valuable time reviewing bulletin board services and shareware catalogs looking for up-to-date and genuinely useful Windows 3.1 shareware utilitities, because we've already done the work for you. We've reviewed and tested hundreds of Windows-based shareware programs and selected only those that meet our high standards for performance and usefulness. You won't find any games or clever icons on the *Windows Insider Disks*—just the best productivity tools available. The programs that you'll receive with the diskettes include all of those mentioned in this book, including all of the network utilities described in Chapter 7. We've also included a few programs that weren't available at the time the book was being written.

The On-Line Help Files

The *Windows Insider Disks* provide much more than shareware. When you purchase this set of disks, you'll also receive a set of files designed specifically to support your troubleshooting efforts. These files include complete listings of PIF, WIN.INI, and SYSTEM.INI settings, plus tips for using these settings to solve your thorniest Windows-related problems. We've also included more than 100 troubleshooting bulletins related to Windows 3.1 problems that have only recently been identified by users and technical support professionals.

All of this information has been created as on-line help files that you can open directly from within Windows, and include all of the underlined jump topics and definitions you would expect to find in any Windows 3.1 help file. You'll find these features to be of great help in cross-referencing information related to a single troubleshooting category or problem.

Index

P

Q

R

V

W

Windows 3.1 Insider Disks Offer!

The Windows Insider Disks are a complete set of the best shareware utilities and information to help you troubleshoot problems with Windows. (For more information, see Appendix B.)

To order, just fill out the coupon below, telling us how many diskettes you want. Check the appropriate box for diskette type: 5 1/4" 360K, or 3 1/2" 720K. Add 6% sales tax if the order is to be shipped within Arizona.

Include payment in the form of check or money order in U.S. funds, or else VISA/MasterCard authorization. CODs and open account purchase orders are not accepted.

Mail the coupon to:

The Coriolis Group
Insider Disks Offers
7721 E. Gray Road, #204
Scottsdale, AZ 85260

Shipping and handling is $2 for orders shipped to the U.S. and Canada, and $6 for airmail shipping to the rest of the world. Please make all checks and money orders payable to The Coriolis Group.

- -

Qty	Set of disks for:	Each	Total
___	*Windows 3.1 Insider Disks*	$15	______
	Arizona orders please add 6% sales tax:		______
	Shipping and handling:		______
	Total due, in U.S. funds:		______

Disks size: 5 1/4" ☐ 3 1/2" ☐

Phone (602) 483-0192
Fax: (602) 483-0193

Name ______________________________
Company ______________________________
Address ______________________________
City/State/ZIP ______________________________
Phone ____________________

VISA/MC # ______________________________ Expires: __________

Signature for charge orders: ______________________________

The book you are holding was designed and produced
by the Coriolis Group, with the aid of the latest desktop publishing
software, including Aldus PageMaker, Adobe Illustrator, Aldus Freehand,
Adobe PhotoShop, Microsoft Word, and of course, Microsoft Windows 3.1.
The cover concepts were designed by Marek Antoniak
The interior design is by Keith Weiskamp and Frank Grazioli.
The body text is 10.5/12.5 Garamond Light.
Technical illustrations were provided by Rob Mauhar and Brad Grannis.
Editorial direction was provided by Paul Farrell.
The text was edited by Pat Vincent.

About the Authors

Keith Weiskamp is the President and Publisher of the Coriolis Group, a leading developer of technical magazines and books including the highly acclaimed *PC TECHNIQUES* Magazine. Keith is an accomplished author, editor, designer, and book developer. During the past five years, he has authored and coauthored over twenty books on a variety of computer-related topics, including programming languages and operating systems. His best-selling books have been translated into numerous languages, including Italian, French, German, Spanish, Portuguese, and Japanese.

Ron Pronk has been writing books on PC software for almost a decade and is considered to be one of the foremost instructional design experts in the field of computer education. In recent years, Ron has focused his technical and educational expertise on the Windows platform. In fact, his Windows instructional materials have been used to train tens of thousands of adults in hundreds of corporations worldwide. Ron has a knack for explaining complex topics in down-to-earth, easy-to-understand, and often entertaining language. He believes computer users don't buy books, they buy knowledge—and have a right to expect results.

Contacting us:
You can reach both Keith Weiskamp and Ron Pronk by writing to the Coriolis Group, 7721 E. Gray Rd., Suite 204, Scottsdale, AZ, 85260. Keith Weiskamp can be reached on MCI as KWEISKAMP and Ron Pronk can be reached on CompuServe at 71033,1502.

WINDOWS

$26.95

Expert Advice at Your Fingertips...

- Do you want to run internal DOS commands from within Windows? According to Windows, you can't do this. But, see page 230 for a way around this limitation.
- Need to print a group of documents quickly? See page 212 for a useful technique for printing unopened documents from the File Manager.
- Perplexed by Windows memory terminology? See page 18 for an explanation of conventional memory, expanded memory, and virtual memory.
- Unsure about how large to make your permanent swap file? See page 115 for an easy way to estimate the size you need.

This INSIDER guide shows you how to get more out of Windows 3.1 by customizing the program to match your style of working and your individual goals. You'll find many tips and techniques that aren't documented anywhere else to make the Windows tasks you perform every day much easier and less time-consuming. You'll also get valuable insights into just what makes the program tick.

ABOUT THE AUTHORS

KEITH WEISKAMP, author of more than a dozen books on computers, is Publisher of PC TECHNIQUES magazine and a principal of the Coriolis Group, a publishing and programming concern in Scottsdale, Arizona. RON PRONK has written more than a dozen books on computers and designed numerous computer training programs. His instructional materials for Windows have been used to teach in hundreds of corporations worldwide.

THE WILEY INSIDER SERIES

"Imagine having an expert by your side—someone always ready to show you how to find undocumented shortcuts and enhancements to get the most out of your software. The INSIDER guides are designed to do just that. They provide clear, practical, accessible advice on the what, why, and how of running your software. Focused on *your* specific needs—on turning the features of the program to your advantage—the guides help you set up, customize, and run your software programs the way *you* want them to run."

— Keith Weiskamp, INSIDER Series Editor

£23.95

John Wiley & Sons, Inc.
Professional, Reference and Trade Group
605 Third Avenue, New York, N.Y. 10158-0012
New York • Chichester • Brisbane • Toronto • Singapore

ISBN 0-471-57984-X

52695

9 780471 579847