Tim Stanley

DOS Stuff Microsoft® Forgot!

The Affordable, Easy Way to Add Power to DOS

que®

DOS STUFF

MICROSOFT FORGOT!

Timothy S. Stanley

que

DOS STUFF MICROSOFT FORGOT!

©1993 by Que® Corporation

All rights reserved. Printed in the United States of America. No part of this book may be used or reproduced in any form or by any means, or stored in a database or retrieval system, without prior written permission of the publisher except in the case of brief quotations embodied in critical articles and reviews. Making copies of any part of this book for any purpose other than your own personal use is a violation of United States copyright laws. For information, address Que Corporation, 11711 N. College Ave., Carmel, IN, 46032.

Library of Congress Catalog No.: 93-85562

ISBN: 1-56529-475-0

This book is sold *as is,* without warranty of any kind, either expressed or implied, respecting the contents of this book, including but not limited to implied warranties for the book's quality, performance, merchantability, or fitness for any particular purpose. Neither Que Corporation nor its dealers or distributors shall be liable to the purchaser or any other person or entity with respect to any liability, loss, or damage caused or alleged to be caused directly or indirectly by this book.

96 95 94 93 8 7 6 5 4 3 2 1

Interpretation of the printing code: The rightmost double-digit number is the year of the book's printing; the rightmost single-digit number, the number of the book's printing. For example, a printing code of 93-1 shows that the first printing of the book occurred in 1993.

Screen reproductions in this book were created by means of the program Collage Plus from Inner Media Inc., Hollis, NH.

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Que cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Publisher: David P. Ewing

Associate Publisher: Rick Ranucci

Managing Editor: Corinne Walls

Marketing Manager: Ray Robinson

DEDICATION

To my wife, Brenda, and son, Joshua. My appreciation and affection go to you two, who put up with me while I wrote this book.

ACKNOWLEDGMENTS

I would personally like to thank each and every author of the shareware products included in this book. Each program is quality work. Thanks to these hard-working, talented individuals, people like you and me have access to affordable software which provides the stuff that Microsoft Forgot.

CREDITS

Publishing Manager
Joseph B. Wikert

Product Development Specialist
Bryan Gambrel

Production Editor
Phil Kitchel

Technical Editor
Randall G. Bryant

Production Manager
Scott Cook

Proofreading/Indexing Coordinator
Joelynn Gifford

Production Analyst
Mary Beth Wakefield

Book Designer
Amy Peppler-Adams

Cover Designer
Katherine Hanley

Graphic Image Specialists
Dennis Sheehan
Susan VandeWalle
Tim Montgomery
Teresa Forrester

Production Team
Jeff Baker
Angela Bannan
Danielle Bird
Laurie Casey
Charlotte Clapp
Heather Kaufman
Bob LaRoche
Mike Mucha
Sandra Shay
Amy Steed
Michelle Worthington
Lillian Yates

Indexer
Craig A. Small

Composed in *Cheltenham* and *MCPdigital* by
Prentice Hall Computer Publishing

ABOUT THE AUTHOR

Timothy S. Stanley works as a Product Development Specialist for Que. He has contributed to quite a number of Que's books, including *Using Windows 3.1*, Special Edition; *Tuning Windows 3.1*; *Killer Windows 3.1 Utilities*; *Using MS-DOS 6*, Special Edition; and *Using PC Tools for Windows*. When not working on computer books, Tim enjoys spending time with his wife, Brenda, and son, Joshua. Tim also enjoys model railroading, and frequents the TRAINNET forum on CompuServe.

OVERVIEW

Introduction 1

1 Managing the Screen with DOS 7

2 Working Better with Disks and Files 29

3 Using a Smart Print Spooler 55

4 Enhancing Communications with DOS 69

5 Don't Just Follow the Path—Make It! 91

6 Memory: Don't Lose It, Use It 103

7 Automating Your Work 139

8 Making DOS Multitask 169

Index 195

TABLE OF CONTENTS

Introduction **1**

What Does This Book Give Me? 2
How Do I Interpret Syntax Lines in This Book? 5
What Is Shareware? 6
Remembering What You Forgot 6

1 Managing the Screen with DOS **7**

Saving Your Screen with ShowMe Saver 8
Installing ShowMe Saver 10
Using ShowMe Saver 12
Customizing ShowMe Saver 13
ShowMe Saver Command Parameters 15
Recalling Your Scrolled Screen 18
Using SCROLLit 19
Using SCROLLit Command Line Parameters 23
Summary 27

2 Working Better with Disks and Files **29**

Formatting a Disk While You Work 30
Using Con>Format 30
Configuring Con>Format 33
Duplicating a Disk Quickly 35
Using Disk Copy Fast Interactively 36
Using Disk Copy Fast Command Line Options 41
Completely Eliminating a File 44
Understanding the SHREDFILE Screen 46
Selecting Files to Shred 46
Customizing the SHREDFILE Screen 50
Quickly Finding Directories 51
Grouping Files by Dates 52
Summary 54

3 Using a Smart Print Spooler **55**

What is Spooling? 56
Using the Spooler/Despooler 57
Starting Spooler/Despooler 58
Printing To Spooler/Despooler 60

Understanding the Separate Spooling, Monitoring, and Despooling Programs 61
Using Spooler 61
Using Despooler 63
Using Monitor 64
Using Despooler/Monitor 65
Using Spooler/Monitor/Despooler 66
Queueing DOS Files 67
Summary 68

4 Enhancing Communications with DOS 69

Getting Messages with PortaCall 70
Transferring Files Using ZIP 72
Getting Ready to Use ZIP 73
Transferring Files 74
Using Server Mode 78
Using the Command Line 81
Transferring ZIP to Another Computer 84
Configuring ZIP 85
Defining DOS Error Messages 89
Summary 90

5 Don't Just Follow the Path—Make It! 91

Using PATH 92
How DOS Looks for Commands 92
How PATH Helps DOS Look for Commands 93
Paths Never Eternal 94
Expanding Your Path Using Patha 95
Changing Your Search Path 96
Powering with Patha 99
Where To Use Patha 100
Summary 101

6 Memory: Don't Lose It, Use It 103

Managing Memory 104
Understanding DOSMAX 105
Managing Memory with DOSMAX 106
Sample DOSMAX Configurations 117

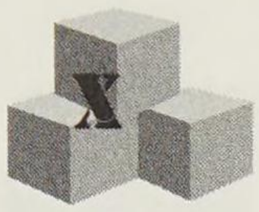
Loading Programs in Memory for Speed 122
Using Fast Load 123
Monitoring Programs You Use 128
Protecting Your System 129
Capturing RAM 131
Recording Vital Hard Disk Information 132
Storing Your CMOS Setup in a File 133
Preserving System Information 134
Examining Memory and Files 135
Displaying Memory Allocation 137
Summary 138

7 Automating Your Work 139

Understanding EXC 141
What Are Scripts? 142
Starting EXC 143
Memory Usage 143
Creating a Script 144
Planning a Script 145
Writing a Script 147
Working with a Script 153
EXC Commands 154
Controlling the Script 154
Displaying and Finding Text 159
Controlling the Screen 163
Summary 167

8 Making DOS Multitask 169

Multitasking DOS with VMiX 170
Configuring DOS for VMiX 171
Starting VMiX 172
Operating VMiX 177
Starting Processes 177
Quitting a Process 178
Controlling Remote Access 179
Controlling the Screen 181
Loading Programs Into UMBs 186
Issuing Commands in the VMIX.INI File 187



Using VMiX with VW Menus 188
VMiX Utility Programs 191
Summary 192

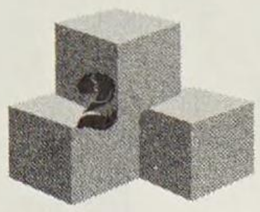
Index 195

INTRODUCTION

It's April 15—what do I do? (Pssst...Mail in your tax forms.) Oh, I forgot! It's my anniversary—what do I do? (Hey, hey...get your spouse a present and a card.) Oh, I forgot! It's been three months since I talked to Mom—what do I do? (Yoo-hoo...better call Mom if you know what's good for you.) Oh, no, I forgot! I've got a big test tomorrow—what do I do? (Cram, baby, cram.) I don't believe it—I forgot!

Forgetting. A common thing we imperfect humans do. "I'd forget my head if I didn't see the ugly thing every morning," is an old saying (slightly modified, of course). The point is, we forget things on a daily basis, right? But come on; some things you just can't forget. Like my wife says, "If it's really important to you, you don't forget it." (I hate it when she has a point.)

How true—you don't forget what's important. But just how important do you feel DOS (*d*isk *o*perating *s*ystem) is to your computer? Without DOS, what would life since 1981 have been like? ('81 is when DOS was introduced with the IBM PC, you know.) Why, without DOS, the Berlin Wall might not have crumbled. Without DOS, we might still be stuck with disco. Without DOS—well, you get the idea. Suffice it to say, DOS has been, and still is, quite important to computers.



With something as important as DOS, you assume that nothing is missing. DOS is the be-all, end-all, got-everything-you-need operating system. Everything that DOS should do, it *does* do. Right?

I think you can tell what I'm leading up to: DOS does *not* do all it *could*. There's stuff missing! (And thank goodness—or else I couldn't write this book.)

WHAT DOES THIS BOOK GIVE ME?

Man, am I glad you asked that. You won't believe the things DOS is missing.

This book, and the attached disk, gives you 17 shareware programs which add things DOS is missing. With this book, you get utilities for your screen, for managing your files, for automating your work, and even for telling you (when you're away from home) that you have messages on your answering machine.

Enough with the generalizations. Here's a list, chapter by chapter, of utilities that DOS forgot, but this book provides:

Chapter 1, "Managing the Screen with DOS," includes two programs for the screen. They are ShowMe Saver and SCROLLit.

- ShowMe Saver is a screen saver program that displays great-looking graphics, while at the same time preventing your screen from image burn-in.
- SCROLLit enables you to scroll back through screens of information that have left the screen—like those long directory listings that seem to fly by while you grumble.

In Chapter 2, "Working Better with Disks," you'll wonder how you ever got by before you had these five utilities. In this chapter, you find: Con>Format, Disk Copy Fast, ShredFile, FD, and Command-Date.

- Con>Format is a concurrent diskette formatter. This little memory-resident program sits in memory, enabling you to format a diskette no matter what program is running. The program formats a disk in the background while you continue working.

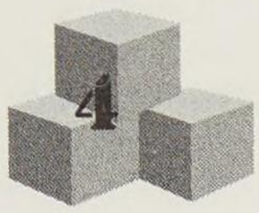
- Disk Copy Fast is a quick disk copy program. This program lessens the time it takes to duplicate a diskette.
- ShredFile enables you to completely eliminate a file, keeping valuable information from accidentally getting into the wrong hands. ShredFile completely overwrites the file or files that you select.
- FD enables you to quickly find a directory on your hard disk. Can't remember the path to that directory? No problem; FD will find it for you.
- Command-Date provides you with a way to group files by date. You can copy and delete files within specific dates or groups of dates using Command-Date. You can continue using any other command-line utility that you currently use for managing files, along with Command-Date.

Chapter 3, "Using a Smart Print Spooler," gives you the Spooling/Despooling Software program.

- Spooling/Despooling Software increases your productivity by enabling you to print in the background. Using Spooling/Despooling Software, you can continue working with your software programs while printing long documents.

In Chapter 4, "Enhancing Communications with DOS," you are introduced to utilities that keep you in touch with your answering machine, enable you to transfer files between computers, and help you with those cryptic DOS error codes.

- PortaCall is a utility that notifies you (when you're not home) when you receive a message on your answering machine. Simply indicate to PortaCall the number where you can be reached, and, using the modem attached to your computer, PortaCall rings that phone when you have a message.
- ZIP is a speedy file-transfer program. With ZIP and a *null-modem cable* (described in detail in Chapter 4), you can quickly and easily transfer files between computers. This capability is especially helpful when the two computers have incompatible disk drives.



- ERROR is a handy program that gives you an English-language meaning of cryptic DOS error codes.

Chapter 5, "Don't Just Follow The Path—Make It!" provides you with a utility that you can use to manage your search path.

- Patha is similar to Path Editor in that you can use it to edit your search path. However, Patha goes a bit further. Patha enables you to increase the size of your search path beyond DOS' 127-character limit. With Patha, you can create a search path as long as 1,024 characters.

Chapter 6, "Memory: Don't Lose It, Use It," provides you with utilities that enable you to make better use of your computer's memory.

- DOSMAX is a set of utilities you can use to squeeze as much conventional memory out of your computer as possible.
- Fast Load enables you to load often-used programs into extended or expanded memory. Loading programs from memory is much faster than loading them from disk.
- SYS_CAP provides several utilities helping you to trouble-shoot your computer. These utilities enable you to capture and examine memory, the CMOS Setup RAM, and critical segments of your hard disk.

In Chapter 7, "Automating Your Work," you learn how to completely automate your computer system—further than batch file automation.

- EXC is a scripting program that creates and replays scripts (programs) to automate repetitive tasks. You have access to a complete scripting language that even enables you to alter the way the script works based on information that appears on-screen.

Chapter 8, "Making DOS Multitask," introduces VMiX—a program which will dramatically boost your productivity.

- VMiX is a multitasking program. With VMiX, you can run more than one program at a time on your computer.

While your spreadsheet program is calculating in the background, you can write a letter using your word processor in the foreground. You can even attach a computer terminal to your computer so that while you are using the word-processor from your keyboard someone else can use that spreadsheet program from a remote keyboard and monitor.

HOW DO I INTERPRET SYNTAX LINES IN THIS BOOK?

You start utilities by typing a command from the command line. In any syntax line, there are parts of the syntax you *must* use, and parts you can *optionally* use. There are actual commands that you *must* use, and there are items in the syntax that vary based on how you use your computer. This section enables you to interpret a syntax line, so you can determine what is necessary versus what is optional, what is actual versus what is a variable.

In a syntax line, what appears in UPPERCASE is what you actually type from the command line. A lowercase item, then, is a variable. What appears in **bold**, you must include in the syntax. What appears in ***bold italic***, you optionally can include. For example, notice this syntax line:

CHKDSK ***d: /F***

The **CHKDSK** command is both uppercase and bold. So, to use this command, you must type CHKDSK. The first parameter, *d:*, is lowercase and italic. Since it's italic, you do not have to include the parameter—it's optional. The ***d*** in ***d:*** is lowercase. That means you replace ***d*** with the actual drive letter you want to check, for example C:. The final parameter is ***/F***. It's italic, designating that you don't have to include the parameter. However, if you do, you must use the parameter as **/F**. The F is uppercase, so you cannot replace the F with another letter.

WHAT IS SHAREWARE?

Shareware is software you can try before you buy. It is *not* free software, but it *is* a neat idea.

Shareware utilities are written by talented, creative individuals. Their software, quite often, provides the same power as programs you can purchase at the local computer or software store. However, you have the advantage of knowing what you are getting before you buy it.

Included in the book, at the beginning of the discussion of each shareware product, is information about the author of the specific program. Additionally, each program on the disk has with it a text file showing you how to register the software. Any software that you like and plan to use on a regular basis, you are obligated to register.

What benefits do you gain by registering the shareware? First, you have a clean conscience, knowing that you have paid the author for his many hours spent in creating such a useful program.

Second, registering the software gives you technical support from the author. The author will provide you with a way to get answers to questions you have with the software.

Finally, registering the software enables you to keep informed of updates that the author makes to the software. This is important, as you benefit from increased features with new versions, and the software remains compatible with new versions of DOS.

REMEMBERING WHAT YOU FORGOT

Now you know what you need to know. Or, you learned to remember what DOS forgot. (Does that make sense?) Anyway, get into the book. Find out about the utilities and programs that you may have no idea are missing. Increase your computer productivity using *DOS Stuff Microsoft Forgot!*

MANAGING THE SCREEN WITH DOS

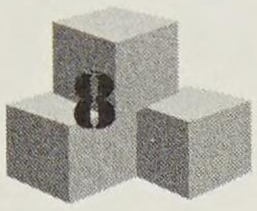
1

You know, people are funny. I've noticed that, given the choice between something cerebral and something flashy, most people will choose the flashy thing every time. When thinking of how to start this book, I decided that I, too, would choose the flashy.

When you use your computer, most of your interaction is with the video display. Sure, you type on the keyboard, and you move that little mouse-thing (if you have one). But think about it—what do you spend hours staring at, the keyboard or the screen? I hope you stare at the screen rather than the keyboard. And why do you stare at the screen? Because you're looking for something flashy, even if it's flashy in a cerebral sort of way.

MICROSOFT FORGOT:

- A screen saver
- A scrolling utility



This chapter shows you how to use two flashy utilities. The first, ShowMe Saver, is a screen saver program that displays fun-to-watch pictures on-screen while preserving the life of the screen. The other utility, Scrolllt, enables you to review information that scrolled off the screen.

What is shareware?
Briefly stated, shareware is *try-before-you-buy* software. You can use a shareware program for a period of time to see if the program meets your expectations and your needs—without laying out a dime. If you decide to keep the shareware, you send the author the price he or she asks for the software. If, however, you find that the software is not quite what you were looking for, simply erase the program from your disk and pay nothing.

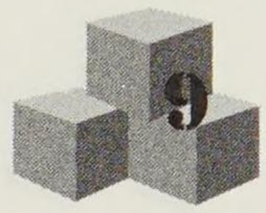
You will find that most shareware is high-quality software at a very reasonable price. When you pay for the software, you have registered the program with the author. This generally entitles you to continue using the program, get technical support, and notice of upgrades for the program.

SAVING YOUR SCREEN WITH SHOWME SAVER

Many people leave their computer on, including the monitor, without changing the display on the screen. Now, your computer doesn't get *mad* about continually displaying the same information. But displaying the same picture for an extended period of time really does burn your monitor. (This is a pun—if you didn't get it yet, just stick with me.)

When your computer displays the same image on the screen for an extended period of time, the image can get *burned* into the screen. That is, when you shut off the monitor, you still see a ghost of the image on the screen.

A screen saver keeps your monitor from getting image burn-in by changing the display on the screen after a set period of



minutes. Generally, a screen saver either completely removes the image from the screen, leaving a blank screen, or it displays an interesting graphic image on the screen. Press a key (or move the mouse), and the original screen reappears.

ShowMe Saver is a screen saver that does more than just save your screen. Like other screen savers, ShowMe Saver displays fun graphic images on the screen after a set amount of time. Additionally, ShowMe Saver enables you to display a message on the screen.

To register ShowMe Saver 3.82, contact:

Harold Schwartz
P.O. Box 104482
Jefferson City, MO 65110

Complete registration and price information is available in the documents included with this utility. If you're going to keep it and use it, please register it!

ShowMe Saver is a TSR program. That is, it's a program that stays in memory when you start the program, but only uses about 8K of memory. When the computer is idle for a set amount of time, ShowMe Saver comes to life and displays graphic images on the screen. Additionally, there are key combinations that ShowMe Saver recognizes, enabling you to redefine the way it works while it's in memory.

"Why display a message?" you ask. (I knew you were going to ask that, so I've got an answer ready.) Quite often, the reason the image on the screen doesn't change is that you are away from your computer. If you're away from your computer, the little message tells other people just where the tutti-frutti you are. Follow along, and you, too, will be enamored with ShowMe Saver (and we'll leave that cerebral stuff for later in the book).

INSTALLING SHOWME SAVER

Before you can use ShowMe Saver, you must install the program. But, don't worry—installation is a breeze. (Ooh, I forgot...just because you like something flashy doesn't mean you can't handle something cerebral like installation.)

When you transferred the utilities with the book from the diskette to your hard disk, the transfer program copied ShowMe Saver to \DSMF\SHOWME on your hard disk. Before running Install, switch to that directory. Then, type **INSTALL** and press Enter.

When Install starts, it displays the following message on the screen:

```
Installation utility for ShowMe saver, the 'Message'
screen saver.
Do you wish to install ShowMe Saver at this time (Y/N)?
```

Type **Y** to install ShowMe Saver.

The next prompt displays the following:

```
The default path for installation of ShowMe Saver is
C:\SHOW
Is this acceptable? (Y/N)
```

If the directory C:\SHOW (the ShowMe Saver installation program copies from the \DSMF\SHOWME directory to \SHOW) is OK for use, type **Y**. Otherwise, type **N**, type a new directory

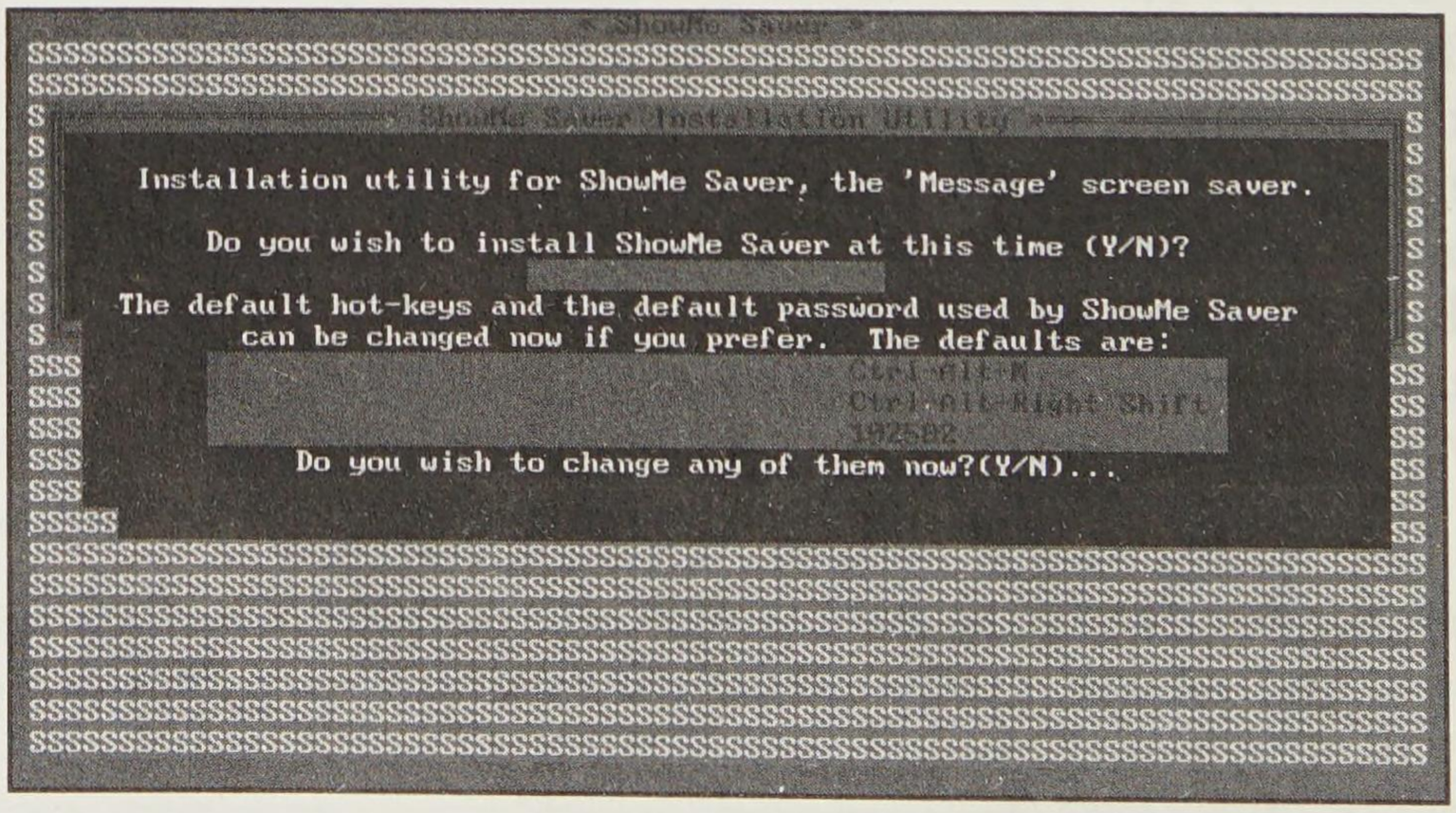


Fig. 1.1 *Customizing ShowMe Saver during installation.*

name, and press Enter. Install begins copying the ShowMe Saver files into the directory that you specified.

Once Install copies the ShowMe Saver files to the directory you selected, it enables you to configure ShowMe Saver. Install displays the box shown in figure 1.1.

If you don't need to change the settings for ShowMe Saver, type **N**. If you do want to make changes, type **Y**, then follow the prompts for changes to make. The configuration changes that you can make enable you to change hot-key choices and select a password (hot-keys and passwords are discussed later).

NOTE

A hot-key is a combination of keys that you press to make an event occur. For example, the default hot-key to display the ShowMe Saver configuration window is Ctrl-Alt-M. That means you press and hold the Ctrl key, press and hold the Alt key, press the letter M, then release all three keys. In this case, the event that occurs is the configuration menu appearing on the screen.

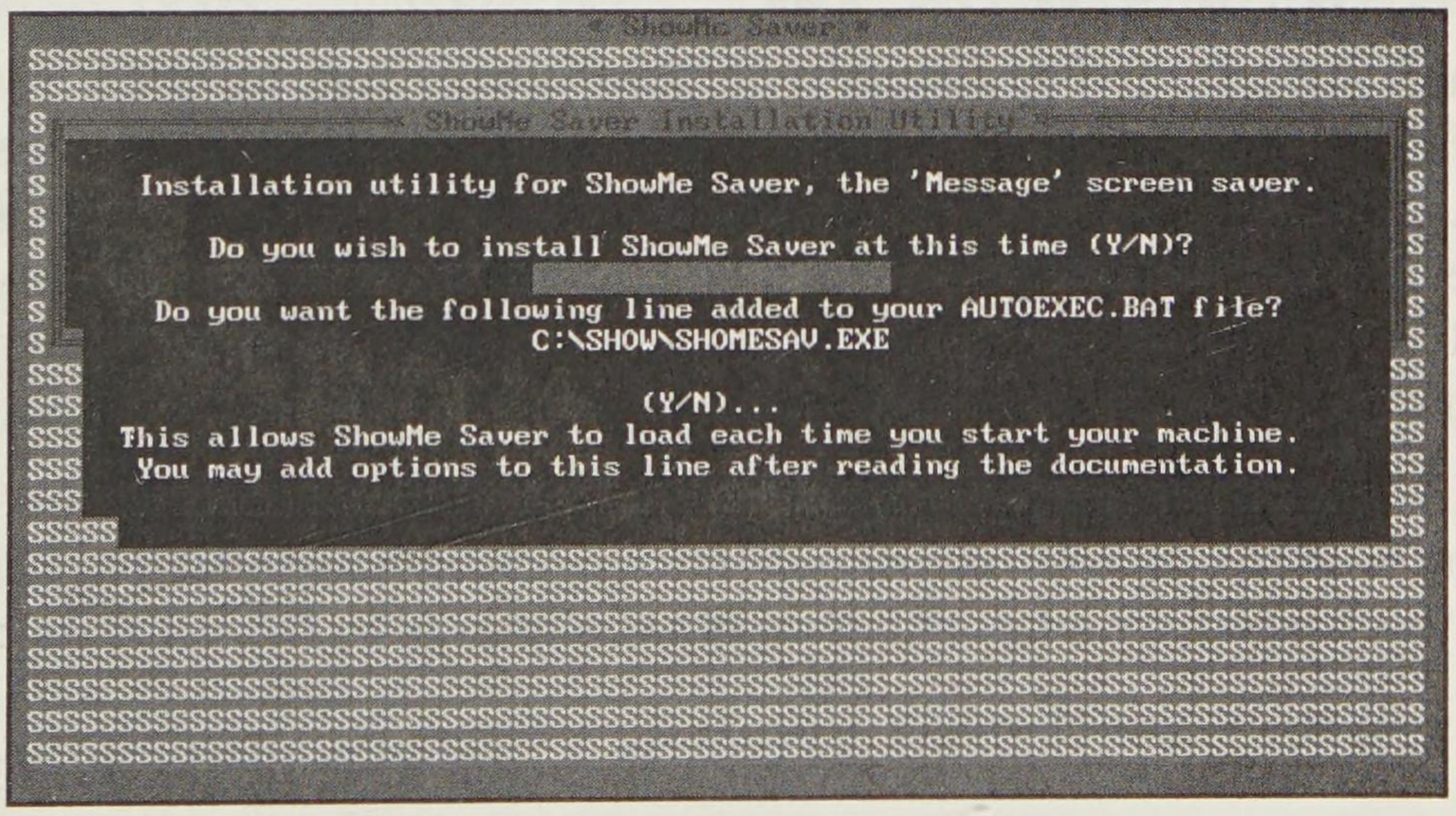
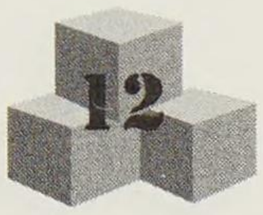


Fig. 1.2
Install enables you to add ShowMe Saver to the AUTOEXEC.BAT file.

When configuration changes are made, Install displays a box, shown in figure 1.2, enabling you to start ShowMe Saver every time you turn on your computer. Type **Y** to add ShowMe Saver to AUTOEXEC.BAT, or **N** to not add ShowMe Saver to AUTOEXEC.BAT.



Another prompt displayed by Install asks if you want to see the documentation from the software author. If you do, type **Y**, read the documentation, then press Esc when you are finished. If you don't want to review the documentation, type **N**.

One final prompt asks

```
Do you wish an automatic reboot of your machine now
to load ShowMe Saver? (Y/N)...
```

Type **Y** to reboot your computer or **N** to return to DOS. (Unless you have some real good reason not to, let Install reboot your computer.) Now, sit back and get ready to get flashed. (Oops! I mean, er, to see something flashy. Heh, heh....)

Suppose you didn't let Install add ShowMe Saver to your AUTOEXEC.BAT file. To start ShowMe Saver, switch to the directory where you installed ShowMe Saver (probably C:\SHOW), type **SHOMESAV**, and press Enter.

USING SHOWME SAVER

When ShowMe Saver starts, it displays a message on the screen telling you about the program and how to register it. It also tells you to press any key to continue. (So what are you waiting for? Press a key!) When you press a key on the keyboard, the computer returns the DOS prompt. That's it! To see ShowMe Saver work, simply wait 5 minutes, doing nothing with your computer.

OK, OK, you don't want to wait 5 minutes. Well, take another gander at the screen. Information about ShowMe Saver lingers on the screen. The information is as follows:

```
SHOMESAV loaded. PopUp time is 5 minutes.

Configuration window Hot-Key is CTRL-ALT-M

Immediate pop-up Hot-Key is CTRL-ALT-RIGHT SHIFT
```

To see what ShowMe Saver does, press the Ctrl-Alt-Right Shift hot-key (hold them all down for just a moment before releasing them) and watch your screen. Wow!

> The hot-keys shown in this chapter are based on the default hot-keys. If you made a change during installation, the hot-keys you selected will display rather than the default keys.

NOTE

ShowMe Saver begins displaying interesting designs on your screen. See the default message? You can replace that message with your own message.

Now you've got to be impressed. And do you want to see what else you can do? Press a key on your keyboard to return to the DOS prompt.

Press Ctrl-Alt-Right Shift again. ShowMe Saver starts again, this time with a different design. Version 3.82 of ShowMe Saver comes with 56 different screen savers. And the software author has more available for you to purchase.

ShowMe Saver has another neat feature that's worth mentioning. Even though ShowMe Saver is a TSR, that is, a program that remains in memory even when not active, it doesn't take up much memory if you have expanded memory (EMS) or extended memory (XMS) available. (EMS and XMS is memory that gives your computer more than the conventional 640K of memory.) If you have EMS or XMS, ShowMe Saver loads most of itself in EMS or XMS memory. Only 8K resides in conventional memory.

Why is that important to know? Your programs use conventional memory. Any conventional memory that TSR programs use takes memory away from other programs that you use. (You know, the *real* reason you bought a computer in the first place.) ShowMe Saver uses only a little conventional memory, not enough for most programs to miss.

CUSTOMIZING SHOWME SAVER

ShowMe Saver enables you to customize the way it works. To customize ShowMe Saver, press Ctrl-Alt-M. The menu displays on the screen. The menu has two main areas. First is the status area. In the status area, ShowMe Saver displays the following:

Status Information	*Description*
Current message	Displays the current message that ShowMe Saver displays on the screen.
Saver Status	Displays whether ShowMe Saver is active or not.
Minutes to Pop-up	Displays the number of minutes your computer must be inactive before ShowMe Saver saves your screen.
System Lock	Displays whether or not you must use the password to return from the screen saver to your screen.
Pop-up order	Displays the order that the different screen savers appear on the screen.
Module cycling	Displays whether or not ShowMe Saver cycles through the different screen saver modules.

The second area on the configuration menu enables you to make changes to the configuration. There are eight choices, which are as follows:

Message displayed—When you select this option, enter a new message of up to 12 characters to display on the screen. Then press Enter.

Status of saver—Choosing this option enables you to toggle the status of ShowMe saver between ON and OFF. ShowMe Saver still resides in memory when OFF. Refer to the status area when choosing this option.

Time to pop-up—Enter the number of minutes from 1 to 60 to delay before ShowMe Saver saves the screen. Then press Enter.

Password—This option enables you to use or not use a password to return from a screen saver to your original screen. Additionally, use this option to change the password. If you use a password, ShowMe Saver requires you to enter the password before it clears the screen saver from the screen, then redisplays your original screen.

Enable saver modules—Enables you to select the saver modules to use. When you select this option, a window appears displaying all saver modules. Select the modules to use using the arrow keys and the space bar, or press Ctrl-Enter to select or deselect all savers. Press Enter to continue.

Order to pop-up modules—Use this option to toggle between Random display of modules, or to display the modules in the order that they appear.

Cycling of saver modules—Enables you to select how many times to cycle through all the saver modules. Enter a number from 1 to 15.

ESC to exit the configuration menu.

Make all changes as you see fit to configure ShowMe Saver. When you are finished making changes, press Esc. ShowMe Saver remembers all your changes, and displays the opening welcome screen again. Press Enter to reinitialize ShowMe Saver and return to the command line. The next time ShowMe Saver starts, it will use all the changes that you made.

ShowMe SAVER COMMAND PARAMETERS

You can set up ShowMe Saver in a number of ways when you start the screen saver. You use command line parameters to customize ShowMe Saver when starting it from the command line or from a batch file. The syntax you use is as follows:

```
C:\SHOW\SHOMESAV parameter
```

Table 1.1 lists the command line parameters that you can use with ShowMe Saver.

Table 1.1. ShowMe Saver Command Line Parameters

Parameter	*Description*
/Bn	Corrects the screen-flashing that occurs on some monitors. Enter a number between 1 and 250 for *n*.
/D	Loads ShowMe Saver in memory as disabled.
/E	Indicates that ShowMe Saver should *not* use EMS memory.
/F	Indicates that ShowMe Saver should *not* use XMS memory.
/I*n*	Causes ShowMe Saver to pop up in the number of seconds indicated by *n*, and activate the default password.
/M*msg*	Specifies that ShowMe Saver should display the message *msg* when active.
/N	Indicates that ShowMe Saver should not swap itself from memory.
/O	Indicates that ShowMe Saver should display saver modules in order.
/P*path*	Specifies that ShowMe Saver swap itself to the directory indicated by *path* when not active.
/R*n*	Indicates the number of times that ShowMe Saver cycles through each module, where *n* is a number from 1 to 15.
/S*mod*	Indicates the ShowMe Saver modules to use, where *mod* is module codes.
/T*time*	Specifies the time that ShowMe Saver waits before popping up, where *time* is a number from 1 to 60.

Parameter	*Description*
/U	Unloads ShowMe Saver from memory.
/X	Indicates that if both EMS and XMS memory are available, ShowMe Saver should use XMS memory first.

From the entries in Table 1.1, you can see that ShowMe Saver is very customizable. Most of the parameters in the table correspond to a configuration option on the configuration menu, but there are a few parameters that deserve extra explanation.

CONTROLLING VIDEO FLASHING

First is the /B*n* parameter. You may notice that ShowMe Saver modules seem to flash on some very high-speed video systems. If this happens to you, then you should use this parameter to clear up the flashing. This parameter simply slows ShowMe Saver down to get rid of the flashing. You use a number from 1 to 250 in place of *n*, where 1 has the least effect and 250 has the greatest effect. Start with a lower number, then work your way up until you are satisfied the flashing doesn't occur any longer. You can only use this parameter when you load ShowMe Saver into memory—for example, from the AUTOEXEC.BAT file.

CONSERVING MEMORY

ShowMe Saver detects both EMS and XMS memory when you start the program. If your computer has either EMS or XMS memory available, ShowMe Saver loads most of itself into one of these types of memory to conserve conventional memory. By default, ShowMe Saver will use EMS memory first, then XMS memory. You can control which type of memory ShowMe Saver uses.

If you want ShowMe Saver to only use EMS memory, start ShowMe Saver using the /F parameter. But if you want ShowMe Saver to use only XMS memory, then start the program using the /E parameter. If you use the /X switch, the ShowMe Saver will use XMS memory first, if both EMS and XMS memory are available in your computer.

Suppose you don't have EMS or XMS memory available in your computer, or you don't want ShowMe Saver to use EMS or XMS memory. In that case, ShowMe Saver will swap some of itself to disk in order to conserve conventional memory. You can control where ShowMe Saver swaps itself using the /P*path* parameter. If you want ShowMe Saver to use the C:\TEMP directory for swapping, then use the parameter /PC:\TEMP.

You can disable swapping altogether, if you like. Use the /N parameter to keep ShowMe Saver from swapping out of memory.

Some programs that you use might require as much memory as your computer has available. In this case, you may have to remove ShowMe Saver from memory to enable the memory-hungry program to operate. You can remove ShowMe Saver from memory using the /U parameter. However, ShowMe Saver must have been the last TSR program started.

RECALLING YOUR SCROLLED SCREEN

When was the last time you issued a DIR command only to have the name of the file you wanted quickly scroll by, off the screen—even when using DIR /W or DIR /P. Or, you press the Enter key, then realize that a piece of information that you needed at the top of the screen just scrolled off the screen. Aggravating isn't it? Don't you wish DOS enabled you to scroll the screen so you could again see that information you need?

Well, now you can get back information that scrolled off the screen. SCROLLit, one of the utilities on the disk, enables you to recall information that scrolled off the screen.

SCROLLit is a TSR program that maintains information that has scrolled off your screen. SCROLLit uses memory in your computer to maintain screen information. By pressing a key, you activate SCROLLit, and are able to recall the information that scrolled off your screen. Not only can you scroll back to look for information that scrolled off the screen, SCROLLit enables you

to search for information automatically. Additionally, SCROLLit enables you to print information to your printer.

> *To register SCROLLit 1.7, contact:*
>
> Bromfield Software Products
> P.O. Box 70081
> Ottawa, Ontario K2P 2M3
>
> Complete registration and price information is available in the documents included with this utility. If you're going to keep it and use it, please register it!

USING SCROLLIT

You will find SCROLLit in the \DSMF\SCROLLIT directory on your disk. Though you can start SCROLLit from the command line, the best place to start SCROLLit is from your AUTOEXEC.BAT file. At the minimum, to start using SCROLLit, you must specify the amount of memory that SCROLLit should use to hold information that scrolls off the screen. For example, to start SCROLLit using 20K of memory to store information that scrolls off the screen, enter the following command:

```
SCROLLIT 20
```

When SCROLLit starts, it displays information on the screen telling you the size of the screen buffer (the area of memory where SCROLLit stores scrolled information) and what type of memory the buffer is located in (such as EMS or XMS memory). Additionally, the information on the screen tells you to press Scroll Lock to activate SCROLLit.

> SCROLLit compresses information stored in the screen buffer. Generally, even though you specify to use a 20K screen buffer, you can store 30K worth of information in the buffer.

NOTE

Now that you have started SCROLLit, enter the DIR command a couple of times, then enter CHKDSK. The purpose of this, of course, is to get information to scroll off the screen. Now, press Scroll Lock. (The first time you press Scroll Lock, SCROLLit displays registration information. Read it, then press the key specified on the screen to continue.) Next, SCROLLit activates, displaying a status bar at the bottom of the screen (see fig. 1.3.) Except for the status bar, the screen looks no different. However, you can begin scrolling the screen, searching for information or printing information from the scroll buffer.

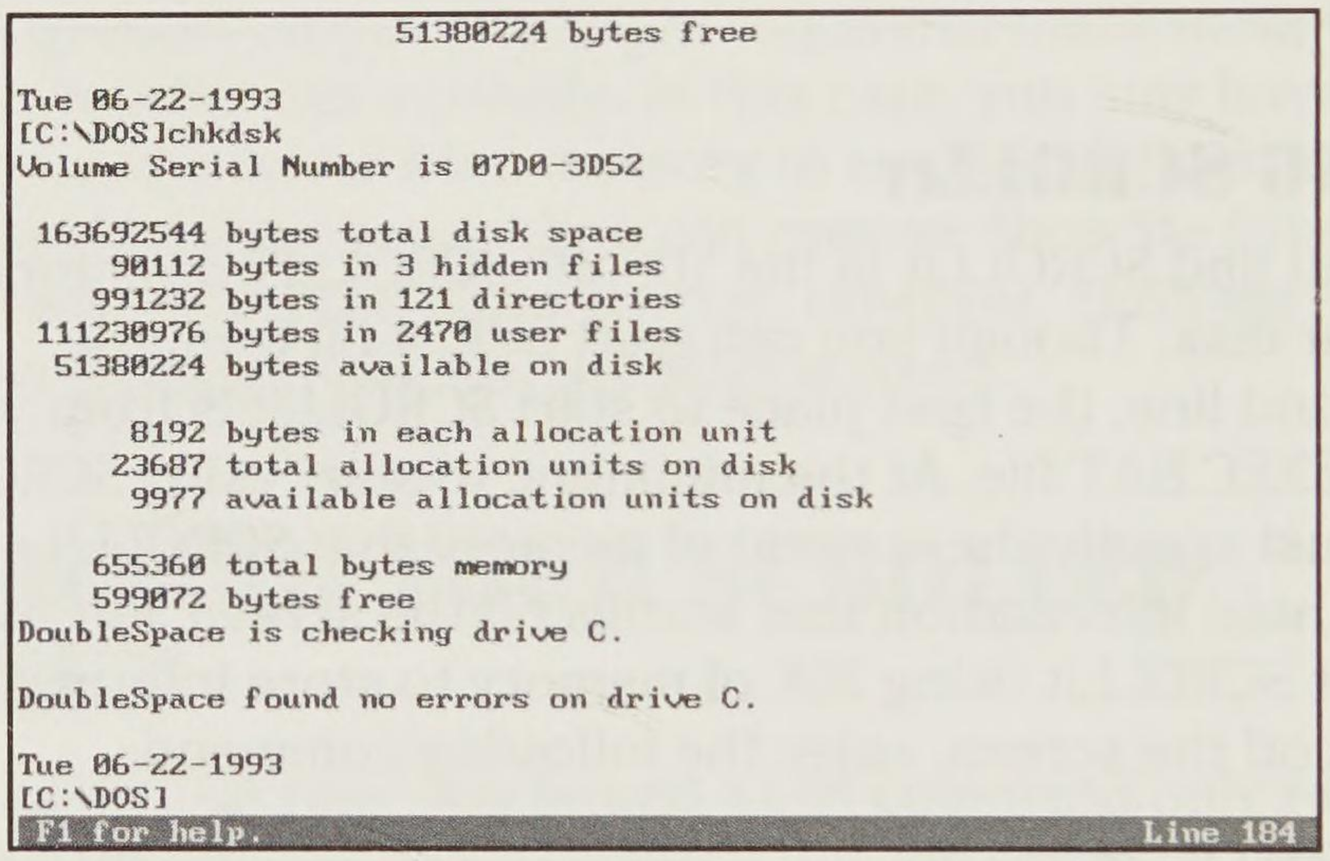


Fig. 1.3
After pressing Scroll Lock, SCROLLit activates, enabling you to review information that scrolled off the screen.

You can use the arrow keys and the other cursor movement keys to scroll through the screen buffer. Table 1.2 lists all the keys that you can use with SCROLLit.

Table 1.2. SCROLLit Command Keys

Command Key	*Description*
↑	Moves the cursor up one line in the screen buffer.
↓	Moves the cursor down one line in the screen buffer.
PgUp	Moves the cursor up one full screen in the screen buffer.

Command Key	*Description*
PgDn	Moves the cursor down one full screen in the screen buffer.
Home	Moves the cursor to the beginning of the screen buffer.
End	Moves the cursor to the end of the screen buffer.
Ctrl-Home	Returns the cursor to the last screen buffer location.
F1	Displays help, including Number of lines saved, Average bytes per line, and Percent of buffer compression.
Esc	Exit the screen buffer.
F	Search for text from the current cursor position forward.
B	Search for text from the current cursor position backward.
A	Perform the previous search again.
W	Write a block of text from the screen buffer to a disk file or to a printer.

SEARCHING FOR INFORMATION

Generally, when you press Scroll Lock to view the contents of the screen buffer, you want to see information that just scrolled off the screen. However, SCROLLit keeps more information than what just scrolled off the screen. Depending on the size of the SCROLLit buffer, the buffer may contain information that scrolled off the screen that morning, when—well, look at the time!— it's afternoon. You can use commands to search for information still held in the screen buffer.

You can search Backward and Forward through the screen buffer to find information. To search for information, first press Scroll Lock to enter the screen buffer. Then, press **B** to search backward or **F** to search forward in the screen buffer. (Searching backward means that you search from the current cursor location to the beginning of the buffer. Searching forward means you search from the current cursor location to the end of the buffer.)

When you press either **B** or **F**, a box appears on the screen enabling you to type in text to search for. For example, figure 1.4 displays the box that appears on-screen when you press **B**.

In the box, type the text to search for. For example, type **DATA** and press Enter. SCROLLit searches for the text, and selects the text, highlighting it in reverse-video.

To repeat the same search again, press **A**. Then, the next occurrence (backwards, of course) displays on the screen in reverse video.

Fig. 1.4
A box appears enabling you to enter text to search for.

SCROLLit is not case-sensitive. That means searching for **DATA** finds DATA, Data, data, or any other combination of upper- or lowercase letters.

SAVING AND PRINTING INFORMATION

You can print or write to a file any information that SCROLLit saves in the screen buffer. Follow these steps to print information or write information to a file from the screen buffer:

1. Press Scroll Lock to activate SCROLLit.
2. Press **W**. A highlight bar appears on the screen.
3. Scroll through the screen buffer until the beginning of the information to be printed or saved appears highlighted on the screen.

4. Press Enter to mark the beginning of the block.
5. Scroll through the screen buffer to the end of the information to be printed or saved. As you scroll, all information appears selected.
6. Press Enter to mark the end of the block. A box appears on the screen asking for a file name.
7. Type the name of the file to save the block to. Type the name PRN to print the block.
8. Press Enter. The blocked information is either saved to the file you specified or is printed to the printer.

USING SCROLLIT COMMAND LINE PARAMETERS

You can alter the way SCROLLit works by using parameters when you start SCROLLit. The complete syntax to start SCROLLit is as follows:

```
SCROLLIT kbufsize options
```

kbufsize is the amount of memory you want SCROLLit to use to maintain screen information, expressed in kilobytes. For example, using the number 50 in place of ***kbufsize*** installs SCROLLit using 50K of memory to maintain screen information. You must specify at least 20.

There are a number of ***options*** that you can specify with SCROLLit. The ***options***, shown in Table 1.3, enable you to choose the type of memory to use with SCROLLit or change the hot-key which activates SCROLLit, among other things.

Table 1.3. SCROLLit Command Line Parameters

Parameter	*Description*
/-	Disables SCROLLit, but does not remove the program or buffer from memory.
/+	Reenables SCROLLit once disabled.

continues

Table 1.3. Continued

Parameter	*Description*
/E	Specifies that SCROLLit should use EMS memory for buffer space. Only use this parameter when starting SCROLLit.
/C	Specifies that SCROLLit should use conventional memory for buffer space. Only use this parameter when starting SCROLLit.
/N=*name*	Enables you to enter your registration name. (Use only when you register the program.)
/K=*key*	Enables you to enter your registration code. (Use only when you register the program.)
/H=*hotkey*	Enables you to specify the hot-key to use, where *key* is the hot-key code. (See the section "Specifying a SCROLLit Hot-key," later in this chapter)
/AC=*color*	Enables you to set the color to use for video mode 3. (See the section "Specifying SCROLLit Colors," later in this chapter.)
/AM=*color*	Enables you to set the color to use for video mode 7. (See the section "Specifying SCROLLit Colors," later in this chapter.)
/NT	Indicates that SCROLLit should not use the DESQView timer. (DESQView is a multitasking product produced by QuarterDeck Office Systems.)
/R	Displays SCROLLit registration information.
/U	Removes SCROLLit from memory.

You can use the command parameters from the command line even though SCROLLit resides in memory. For example, you can stop SCROLLit from recording information in the screen buffer by using the command **SCROLLIT /-**. Use the command **SCROLLIT /+** to start recording information in the screen buffer again. The + and - commands stop SCROLLit from recording information. They do not activate SCROLLit, so you can scroll through the recorded screen information.

NOTE

You only use the /K and /N parameters once—when you register the program with the author and receive the registration information in the mail.

SPECIFYING A SCROLLIT HOT-KEY

Although the hot-key that you use to activate SCROLLit is Scroll Lock, a key not normally used, you can change the SCROLLit hot-key. You change the hot-key with the /H=*hotkey* command parameter.

When you use the /H parameter, you specify the key code for the hot-key you want to use. You can change the hot-key to any key or key combination on the keyboard. For example, you can change the hot-key to Ctrl-Left Shift-Scroll Lock (that is, press and hold the Ctrl key, the left-hand Shift key, and the Scroll Lock key, then release all three keys). Use the following command to change to this hot-key:

```
SCROLLIT /H=646
```

NOTE

The key codes that you can use to change the SCROLLit hot-key are quite extensive. While this chapter describes how to change the hot-key for a single key code, refer to the SCROLLIT.DOC document in the SCROLLit directory for a complete listing of key codes.

SPECIFYING SCROLLIT COLORS

The default colors that SCROLLit uses are blue and white on cyan. These colors are generally fine, but you might have a reason to change them sometime. For example, if you use SCROLLit on a laptop computer with an LCD screen, the default colors may not be as readable as other colors. You can use the /AC or /AM command parameters to change the screen colors. The /AC parameter changes the colors in color mode, and the /AM parameter changes the colors in monochrome mode.

With each parameter, there are four color settings. Specify each color with the parameters as follows:

```
/AC=bc,nt,ht,st

/AM=bc,nt,ht,st
```

where *bc* is the border color, *nt* is the normal text color, *ht* is the highlighted text color, and *st* is the color of the text being selected.

You set each color setting using two digits—the first digit for the background color, and the second digit for the foreground color. The following table lists the color codes and the colors that each represents. (Note that you can only use codes 0 through 7 for the background.)

Code	*Color*
0	Black
1	Blue
2	Green
3	Cyan
4	Red
5	Magenta
6	Brown
7	White

Code	*Color*
8	Grey
9	Bright Blue
A	Bright Green
B	Bright Cyan
C	Bright Red
D	Bright Magenta
E	Yellow
F	Bright White

Suppose you want to change the four color settings to white on cyan, white on blue, bright white on blue, and blue on white. Use the following command to set these colors:

```
SCROLLIT /AC=37,17,1F,71
```

SUMMARY

In this chapter, you saw the beginning of some impressive utilities. You saw how ShowMe Saver can save your screen from image burn-in. Additionally, you saw that SCROLLit can help you review information that has scrolled off the screen. However, these two utilities are only the beginning. Chapter 2 will introduce you to utilities that make managing disks a breeze. (And I'm not blowing hot air!)

WORKING BETTER WITH DISKS AND FILES

2

As you know, DOS stands for Disk Operating System. Did you catch the clue? That's right: *disk*. So, DOS should help you work with disks—and it does. But have you ever gotten help from a "friend" who only does what he or she *has* to do—no more, just what gets him or her by? Well, I don't want to be the one to complain, but our old buddies Format, Diskcopy, Del, and CD (alias CHDIR) are like those "do what it takes to get by" friends. But that's why you bought this book, right? You're looking for some new friends.

Enough anticipation building. Would you like a utility that enables you to continue working while you format a disk? Would you like a utility that quickly copies an entire diskette without making you do the diskette shuffle? Would you like a utility that completely overwrites a file when you delete it? Would you like a utility that finds the directory you want to change to, even if you don't know the entire path of the directory? You'll find those

MICROSOFT FORGOT:

- A way to format disks without interrupting your work.
- A fast way to copy disks.
- A means of (*completely*) erasing disks.
- An easy way to track down directories.
- A way to group and copy files by date.

things described here, and you will find the utilities on the disk. Additionally, you'll find a bonus utility—one that enables you to group files by date to perform actions on.

FORMATTING A DISK WHILE YOU WORK

"Just whis-tle while you work. Hmm, hmm, hmm, hmm, hmm, hmm, hmm..." So I'm not a great singer; but this is a good introduction to Con>Format, a concurrent diskette-formatting program.

Con>Format is a TSR (using about 12K of memory) that enables you to format diskettes while you continue working in an application. With Con>Format in memory, you can format a diskette while you are calculating a spreadsheet, or printing a letter. You don't even have to exit the program you are using.

> *To register Con>Format 1.8A:*
>
> SYDEX
> P.O. Box 5700
> Eugene, OR 97405
>
> Complete registration and price information is available in the documents included with this utility. If you're going to keep it and use it, please register it!

USING CON>FORMAT

Before you can use Con>Format, you must load it into memory. You can find Con>Format in the \DSMF\CONFMT directory on your hard disk (the directory to which you initially install the program).

To load Con>Format into memory, first change to the directory where CONFMT.EXE is located. Then type **CONFMT** at the DOS command line and press Enter. Con>Format displays an opening screen, then prompts you to press the space bar to continue.

When you press the space bar, the DOS command prompt returns to the screen. You have successfully loaded Con>Format into memory; it is ready for use.

> Con>Format works just fine loaded into upper memory. If you are using DOS 5, DOS 6, or a memory manager that supports upper memory blocks, you can load Con>Format into upper memory, conserving conventional memory.

TIP

To activate Con>Format once it is loaded into memory, press the hot-key combination. By default, the hot-key combination is Left Shift-Alt-F. When you activate Con>Format, the window shown in figure 2.1 displays on the screen.

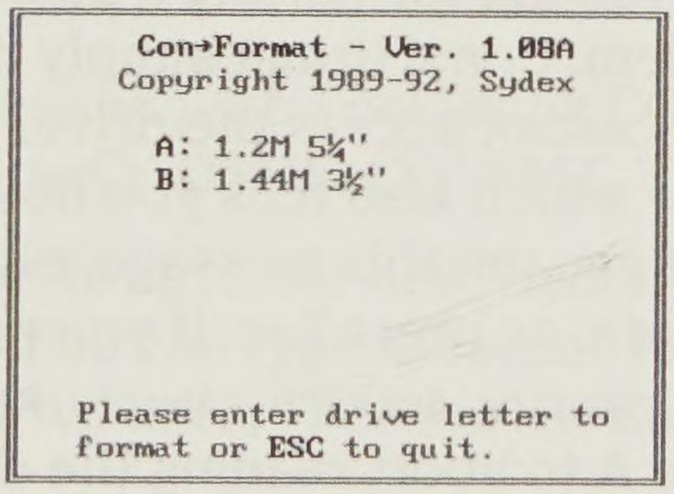


Fig. 2.1
The Con>Format Main Window.

The Con>Format window displays the drives you have available in your computer. To format a disk, press the letter of the drive; to format a disk in drive B, press **B**. If the drive is capable of formatting diskettes with different capacities, a window like the one in figure 2.2 displays on the screen. From this window, choose the correct capacity to format. For example, from the window shown in figure 2.2, type **2** to format a 1.44M diskette.

When you have chosen the capacity, another window displays on the screen asking you to insert the diskette in the drive. Place the diskette to be formatted in the drive and press Enter. Con>Format begins formatting the diskette, and you can continue working. Pretty cool, huh?

When Con>Format finishes formatting the disk, it displays another message on the screen telling you whether or not the format was successful. You are also prompted to insert another

diskette to format into the drive and press Enter, or press Esc to quit formatting.

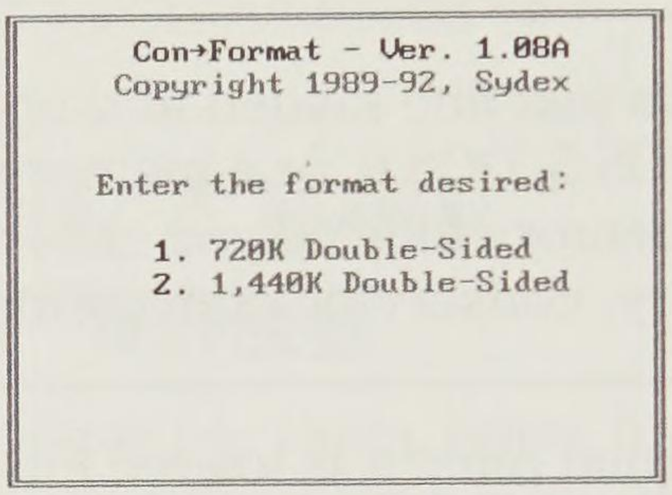


Fig. 2.2
Choose the diskette capacity to format.

One thing to keep in mind—you cannot access the diskette drives while you are formatting a disk. No doubt, at some time you will forget that you are formatting a disk and try to use the disk drive. No problem. Con>Format simply displays a message telling you that you cannot access the drive. The message appears in a window which also tells you how far along the format process is. If you get this message, either wait until formatting is complete, or press Esc. If you press Esc, DOS displays a `Not ready reading drive B--Abort, Retry, Fail` message on the screen. Press **A** to abort reading the drive.

NOTE

If you are using an application program, and try to access a diskette while using Con>Format to format a diskette, the application program displays some type of `Drive not ready` message. Simply respond to the message in whatever way the application program specifies. When Con>Format completes formatting the diskette, you can then access the drive.

Because Con>Format is a memory-resident program, it does take up memory. However, you can remove Con>Format from memory, provided it is the last memory resident program you loaded into memory. To remove Con>Format from memory, type **CONFMT R** and press Enter.

CONFIGURING CON>FORMAT

You can change the configuration Con>Format uses. You can change the hot-key combination that activates Con>Format, the colors that display on the screen, and other attributes of Con>Format. The following steps show you how to configure Con>Format:

1. Change to the directory containing Con>Format. For example, type **CD \DSMF\CONFMT** and press Enter.
2. Type **CFCONFIG** and press Enter to configure Con>Format. CFCONFIG displays information on-screen about configuration.
3. Press Esc to quit configuring Con>Format or any other key to continue.
4. Specify the drive and path containing CONFMT.COM. For example, type **C:\DSMF\CONFMT**. Then press Enter.
5. Specify the drive to save the newly configured CONFMT.COM. You can type the same drive as in step 4, or type a new drive letter. Then press Enter.
6. Specify whether your computer is one of the following:

   PC AT, IBM PS/2 or compatible

   PC XT, IBM PS/2 MODEL 25 or 30 or compatible

   CRCONFIG displays the type of computer it *thinks* you have. If the type displayed is correct, press **Y**. Otherwise, press **N**, then **Y** to choose the correct type of computer.
7. Choose the foreground and background colors to display for normal text. Type the number displayed for the correct color combination. Then press Enter.
8. Choose the foreground and background colors to display for highlighted text. Type the number displayed for the correct color combination. Then press Enter.
9. Choose the foreground and background colors to display for the product name text. Type the number displayed for

the correct color combination. Then press Enter. CFCONFIG displays the color combinations that you selected on the screen.

10. Press **Y** to accept your color choices and continue. If you press **N**, however, repeat steps 7 through 9.

11. Press **Y** if your computer's display adapter can support fast access or **N** if it cannot. Generally, only older Color Graphics Adapters cannot. If you have a newer CGA, or an EGA or VGA display, press **Y**.

12. Press **Y** to enable the configuration program to detect the types of diskette drives that you have in your system. However, if you have an unusual diskette configuration, press **N** and select your diskette drives manually.

13. CFCONFIG displays a screen explaining the diskette configuration report that it is about to show you. Read the information, then press any key to continue.

14. Review the diskette configuration that CFCONFIG determined. If there are no configuration changes to make, press **N**. Otherwise, press **Y** to manually select the diskette configuration.

15. Press **N** to format diskettes using the standard DOS boot sector, or press **Y** to use a special boot sector.

    If you press **Y**, CFCONFIG asks you to place a diskette containing the special boot sector in a diskette drive, then press the letter of the drive. CFCONFIG reads the special boot sector, then asks you to press any key to continue.

CFCONFIG can only read the boot sector from a physical disk drive, not from a logical drive. You cannot, for example, read the boot sector of a compressed drive, such as one created with the DOS 6 DoubleSpace program or the Stac Electronics Stacker program. You can, however, read the boot sector of the DoubleSpace or Stacker host drive.

16. Enter the hot-key combination you want to use to activate Con>Format when it is loaded in memory. For example, simultaneously press Ctrl-Left Shift-F, then release the keys. Press the keys again to make sure you selected the correct keys, then press any key to continue.
17. Specify the way that Con>Format should alert you when it completes formatting a diskette. Press **W** for a window to display, **T** for a tone to sound, or **B** to have both happen when formatting is complete.
18. Specify the directory to write the configured copy of CONFMT in. For example, type **\UTILS** then press Enter. Then, press any key to continue.
19. Press **Y** to include a line in AUTOEXEC.BAT to start Con>Format each time you start your computer. Press **N** to not include the line.
20. Press any key to quit CFCONFIG and return to DOS.

You have successfully reconfigured Con>Format. You can run CFCONFIG anytime that you want to reconfigure Con>Format.

DUPLICATING A DISK QUICKLY

You probably use the COPY command quite often to copy files. COPY is a pretty nice command to use—you issue the one command, sit back, and let COPY do its thing. From time to time, however, you have to copy entire disks, rather than just files. DISKCOPY, like COPY, requires just a single command. Unlike COPY, DISKCOPY requires a lot of baby-sitting, especially if you are copying to and from diskettes in the same diskette drive. In that case, you probably have to perform the diskette shuffle. You know how that goes—stick in the disk and press Enter...read a little of the disk then switch disks...press Enter and write a little to the disk...swap disks and press Enter to read a little more. Down-right confusing, and a pain, to boot.

Disk Copy Fast is a fast disk copying program that saves you time in three ways. First, Disk Copy Fast reads an entire diskette

into memory (or even to a disk file), so you don't have to exchange the source and target disks several times to complete the copy. Second, Disk Copy Fast writes only the portion of the diskette that actually contains data. If a disk is only half full, then Disk Copy Fast only reads and writes half the diskette. Third, once Disk Copy Fast reads the source diskette, you can copy to more than one diskette, so you don't have to issue a command over and over again.

To register Disk Copy Fast 4.2:

DCF Software
P.O. Box 60064
Palo Alto, CA 94306

Complete registration information is available in the documents included with this utility. If you're going to keep it and use it, please register it!

You can use Disk Copy Fast in two ways. The first way shown is using Disk Copy Fast interactively, choosing commands from the menus. The second way, and probably the quickest, is from the command line.

USING DISK COPY FAST INTERACTIVELY

You can operate Disk Copy Fast interactively, choosing commands from menus. Disk Copy Fast also supports a mouse. To start Disk Copy Fast, change to the directory containing Disk Copy Fast, type **DCF** and press Enter. (By default, Disk Copy Fast is located in the C:\DSMF\DCF directory.)

When you start Disk Copy Fast, information about the program displays on the screen. Press a key to continue with the program. The Disk Copy Fast main screen, shown in figure 2.3, displays on the screen.

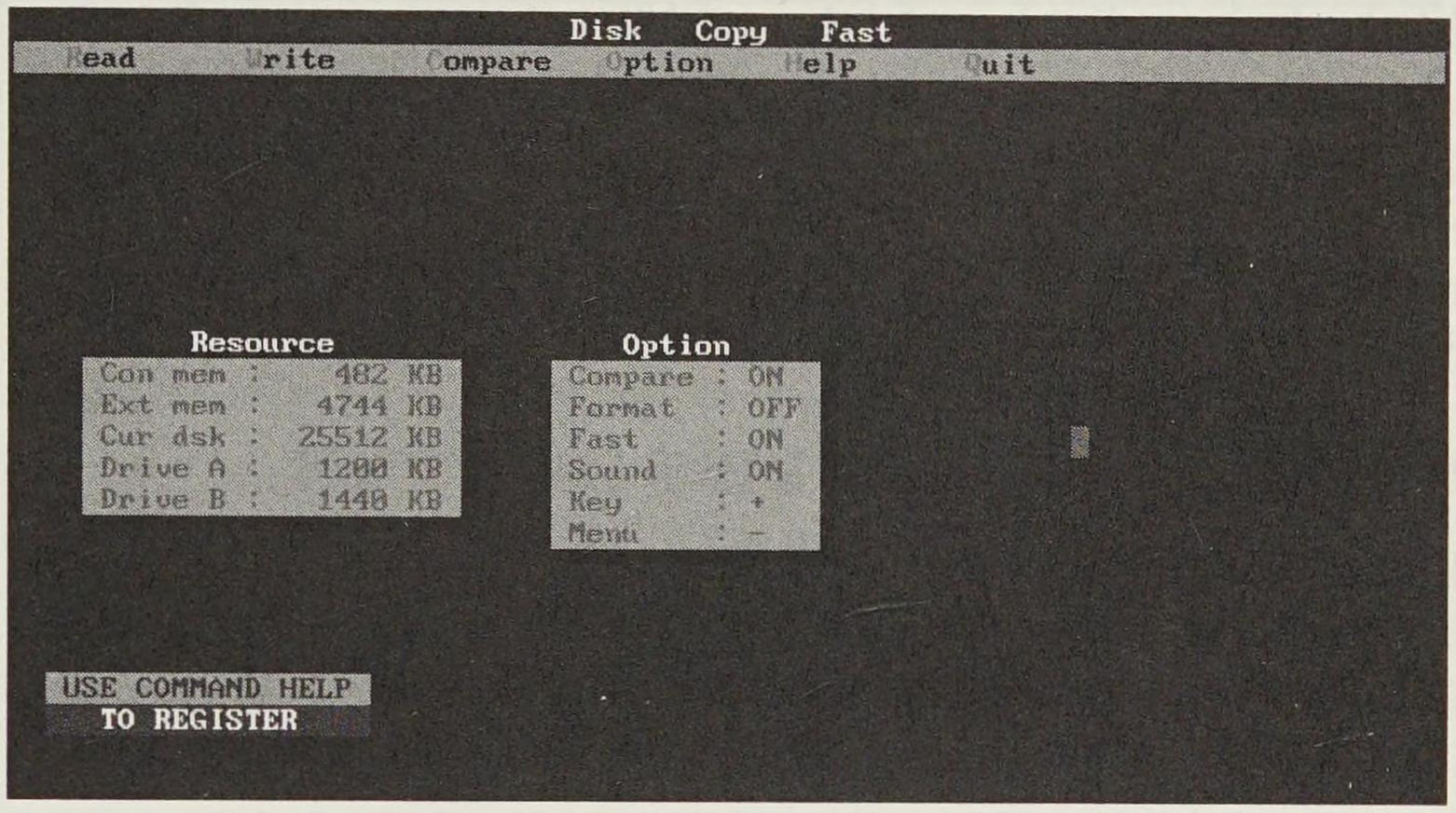


Fig. 2.3
The Disk Copy Fast main screen.

UNDERSTANDING THE DISK COPY FAST SCREEN

Take a closer look at the screen shown in figure 2.3. At the top of the screen is the main menu. From this menu, you can select to read a disk, write a disk, compare a disk, set options, get help, or quit Disk Copy Fast. You can open the menus and choose menu options with the keyboard by pressing the menu's and option's highlighted letter. If you have a mouse, point to the menu or menu option and click the mouse button.

In the middle of the screen are two windows—the Resource window and the Information window. The Resource window displays how much memory your computer has—conventional memory, expanded memory, extended memory, and hard disk space. Additionally, the Resource window displays the maximum capacities of the diskette drive attached to your computer.

The Option window displays the setting of several of the Disk Copy Fast options. Any changes that you make in the Options menu are immediately reflected in the Option window.

> If you are using a mouse, you can set the options in the Option window by clicking on the option with the mouse to toggle the option setting.

TIP

A third window displays in the middle of the screen after you read from or write to a diskette—the Information window. The Information window displays the capacity of the disk copied, the number of tracks used on the disk, including where the information is stored, the amount of time the last read and write took, and the number of copies made.

COPYING A DISK WITH DISK COPY FAST

Copying a disk is a two-step process. First, you must read the information from the source disk. Second, you must write the information to the destination disk. The following steps enable you to read from the source disk:

1. Start Disk Copy Fast as shown above.
2. Choose **R**ead from the main menu. Either press **R** or click on **R**ead with the mouse. The Read menu displays on the screen.
3. From the Read menu, choose the drive or file to read from. For example, to read from Drive B, select Drive B from the menu and press Enter. A message appears on the screen telling you to `Insert SOURCE diskette into drive B`.

CAUTION

> If a disk is in the drive you choose to read from or write to, Disk Copy Fast displays the message, detects that the disk is already in the drive, and then starts reading from or writing to the disk. Make sure you have the correct disk in the drive before you use Disk Copy Fast to read from or write to a disk.

4. Place the source diskette in drive B. A screen, shown in figure 2.4, appears on the screen showing you the progress of the read.
5. Once Disk Copy Fast is finished reading from the source disk, remove the disk from the drive.

After you read the information into memory (or to a disk file), you can write the information to the destination disk. Follow these steps to write information to the destination disk:

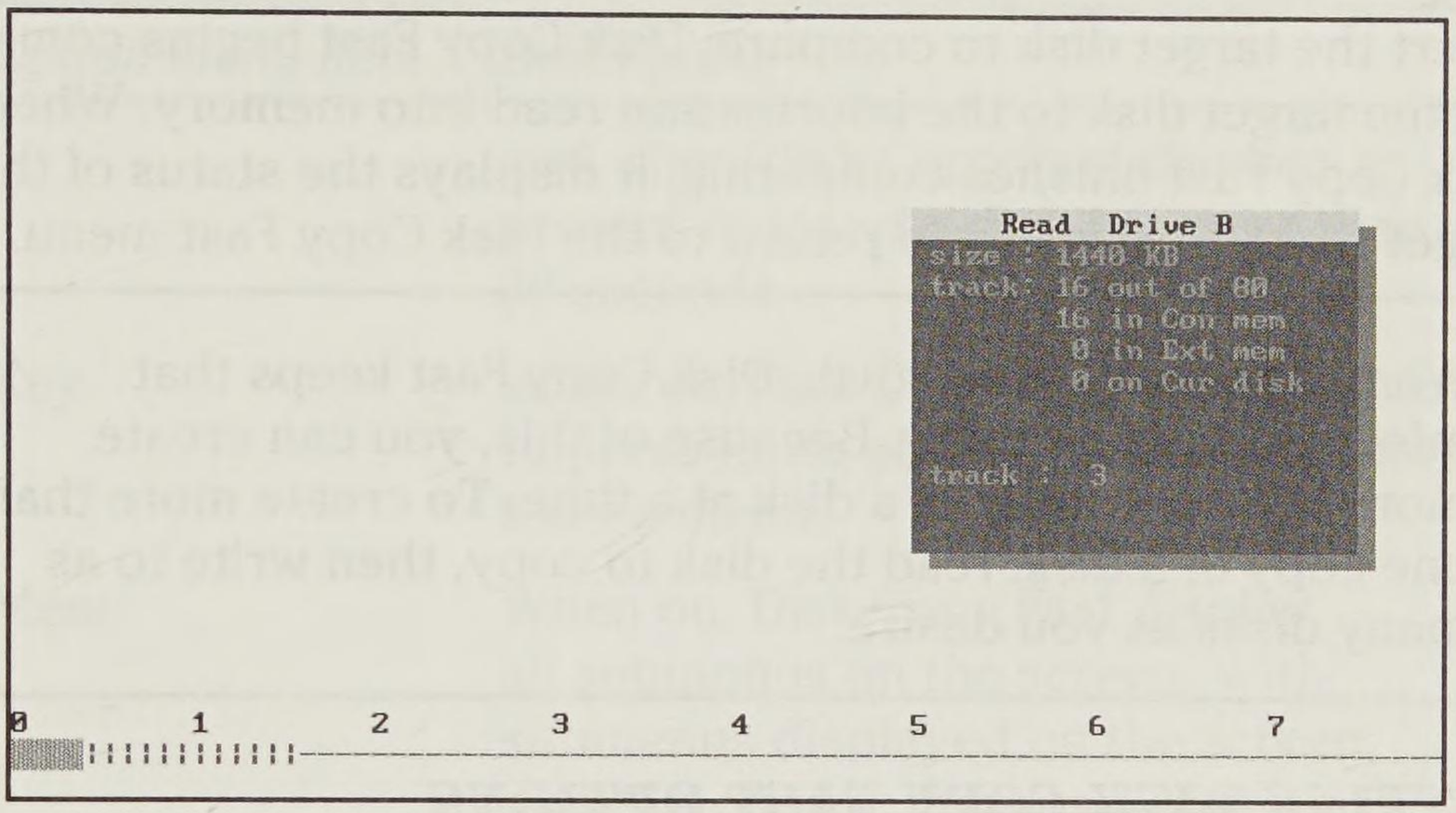


Fig. 2.4
A screen displaying the progress during a disk read.

1. Choose **W**rite from the main menu. Either press **R** or click on **W**rite with the mouse. The Write menu displays on the screen.
2. From the Write menu, choose the drive or file to write to. For example, if you read from the disk in drive B, then select Drive B to write to. Then press Enter. A message appears on the screen telling you to `Insert TARGET diskette into drive B`.
3. Place the diskette to write to in drive B. Disk Copy Fast begins writing the information to the diskette. A progress screen displays as Disk Copy Fast writes the information to the disk.

> A target disk must be the same size and capacity as the source disk. You cannot duplicate a 1.2M disk to a 1.44M disk, for example.

NOTE

When Disk Copy Fast finishes writing to the target disk, the main screen returns, and a new Information box displays on-screen.

When you are finished writing the information that you read to the target disk, you can compare the target disk to the information still in memory. Choose **C**ompare from the main menu, and

insert the target disk to compare. Disk Copy Fast begins comparing the target disk to the information read into memory. When Disk Copy Fast finishes comparing, it displays the status of the target disk. Press a key to return to the Disk Copy Fast menu.

TIP

When you read from a disk, Disk Copy Fast keeps that information in memory. Because of this, you can create more than one copy of a disk at a time. To create more than one copy of a disk, read the disk to copy, then write to as many disks as you desire.

SETTING DISK COPY FAST OPTIONS

There are several options you can set while you are using Disk Copy Fast. For example, you can set up Disk Copy Fast to always compare the information it is writing to a disk. Or, you can specify that Disk Copy Fast always format a destination disk. You can set the options from the **O**ption menu, or if you use a mouse, from the Option window. Each option is a toggle—that is, you set the option on or off.

The following tables list the options that you can set from the **O**ption menu:

Option Menu Item	*Description*
Compare	When on, Disk Copy Fast compares information written to the destination disk.
For**m**at	When on, Disk Copy Fast formats a destination disk as it writes information to the disk.
Fast	When on, Disk Copy Fast reads and writes only the part of the disk containing active data.
Sound	When on, Disk Copy Fast emits a sound when you must insert a disk, when Disk Copy Fast completes a read or write,

Option Menu Item	*Description*
	and when Disk Copy Fast displays a prompt to insert a disk for more than 20 seconds.
Key	When on, Disk Copy Fast requires you to press Enter after selecting an option from a menu.
Me**n**u	When on, Disk Copy Fast display all submenus on the screen. With submenus displayed on the screen, you can choose options from the submenus without first selecting the main menu options.

USING DISK COPY FAST COMMAND LINE OPTIONS

You can choose from several command-line options when starting Disk Copy Fast. From the command line, you can set options that change the way Disk Copy Fast works. Additionally, you can copy a disk from the command line without any menu interaction.

The syntax for Disk Copy Fast is as follows:

DCF *sd: dd: /options*

Where *sd:* is the source disk drive letter, *dd:* is the destination drive letter, and */options* are the command options listed in Table 2.1.

The command options listed in Table 2.1 are toggle options. That is, you use the options to turn the option on or off. Use a plus symbol to turn on the option, and the minus symbol to shut it off. For example, to start Disk Copy Fast and turn compare on, use the following command:

DCF /C+

You can copy a disk from the command line. For example, to use Disk Copy Fast to read a source disk in drive B and write to a destination disk in drive B, without menu interaction, use the following command:

DCF B: B:

Disk Copy Fast starts, and reads the disk in drive B. Then, Disk Copy Fast prompts you to place the target disk in drive B and press Enter.

You can use a combination of options with Disk Copy Fast. For example, suppose you want to read the disk in drive B, make multiple copies, and format each target disk while copying the information to the disk. Use the following command:

DCF B: B: /X+/M+

Table 2.1 lists all of the Disk Copy Fast command line options.

Table 2.1. Disk Copy Fast Command Line Options

Command Option	*Description*
/A	Indicates whether Disk Copy Fast should check to make sure the source and destination disks are compatible capacities. Disk Copy Fast checks for compatible capacities by default.
/B	Indicates whether you desire Disk Copy Fast to alternate reading the source disk and writing the destination disk, or whether you want Disk Copy Fast to read the complete source disk then write the complete destination disk. The default is to read and write complete disks.
/C	Indicates whether Disk Copy Fast should compare while it is copying. Disk Copy Fast compares by default.

Command Option	*Description*
/D	Indicates whether Disk Copy Fast should operate at its fastest speed, or whether your computer requires Disk Copy Fast to delay. The default is to operate at fastest speed.
/F	Indicates whether Disk Copy Fast should copy only the parts of the disk that contains active information. By default, Disk Copy Fast only copies sectors containing information.
/K	Specifies whether you can press highlighted keys to perform a command, or whether you must select a command and press Enter. By default, you press highlighted keys to select commands.
/M	Indicates whether Disk Copy Fast should format the target diskette as it copies. The default is off.
/N	Indicates whether the menus display on-screen already opened, or whether you must open a menu before you can select commands from the menu. By default, you must open a menu before selecting the menu options.
/O	Indicates whether Disk Copy Fast should display in monochrome or color. The default is color.
/S	Indicates whether Disk Copy Fast should sound a signal when finished reading or writing a diskette, or if you do not respond to a prompt for more than 20 seconds. The default is on.

continues

Table 2.1. Continued

Command Option	*Description*
/T	Indicates whether Disk Copy Fast should begin reading a high-density disk and switch if a low-density disk is inserted, or should begin reading a low-density disk, then switch if a high-density disk is inserted. The default is to begin reading a high-density disk.
/W	Indicates whether a source disk must be write-protected before Disk Copy Fast reads from the disk. By default, Disk Copy Fast does not check to ensure that the source has its write-protect notch covered.
/X	Indicates whether Disk Copy Fast will copy the source disk to multiple destination disks or to a single destination disk. By default, Disk Copy Fast copies to a single destination disk.

COMPLETELY ELIMINATING A FILE

You no doubt use DEL or ERASE quite often to remove files from your disk. But did you know that you do not *really* delete the file from your disk? When you use DEL to delete a file, for example, DEL simply alters the file name in the directory list, and releases the disk space used by the file. However, your file remains on the disk.

OK, so the fact that the file is not completely wiped from the disk does give you the opportunity to *undelete* a deleted file. I've used that feature when I accidentally deleted a file, and I'm sure you have, too. But suppose you want to delete a file which contains sensitive information—that is, information you don't want other people to get their hands on. Well, forget it if you use

DEL or ERASE. Any Tom, Dick, or Harry can undelete the information from the disk and read that sensitive information.

What do you do with paper documents that you want to destroy? Probably run the documents through a shredder. What do you do to destroy *electronic documents*? Run them through an electronic shredder, of course. And that is just what the SHREDFILE utility enables you to do.

> *To register SHREDFILE 1.0:*
>
> Seth W. Comstock
> 4835 W. Montecito
> Phoenix, AZ 85031-2027
>
> Complete registration and price information is available in the documents included with this utility. If you're going to keep it and use it, please register it!

As with the other utilities described in this book, SHREDFILE is a pretty easy utility to use. You can find SHREDFILE in the \DSMF\SHRED directory on the drive to which you installed the utilities.

To start SHREDFILE, change to the directory containing the utility, type **SHRED** and press Enter. The main SHREDFILE screen, shown in figure 2.5 appears on-screen.

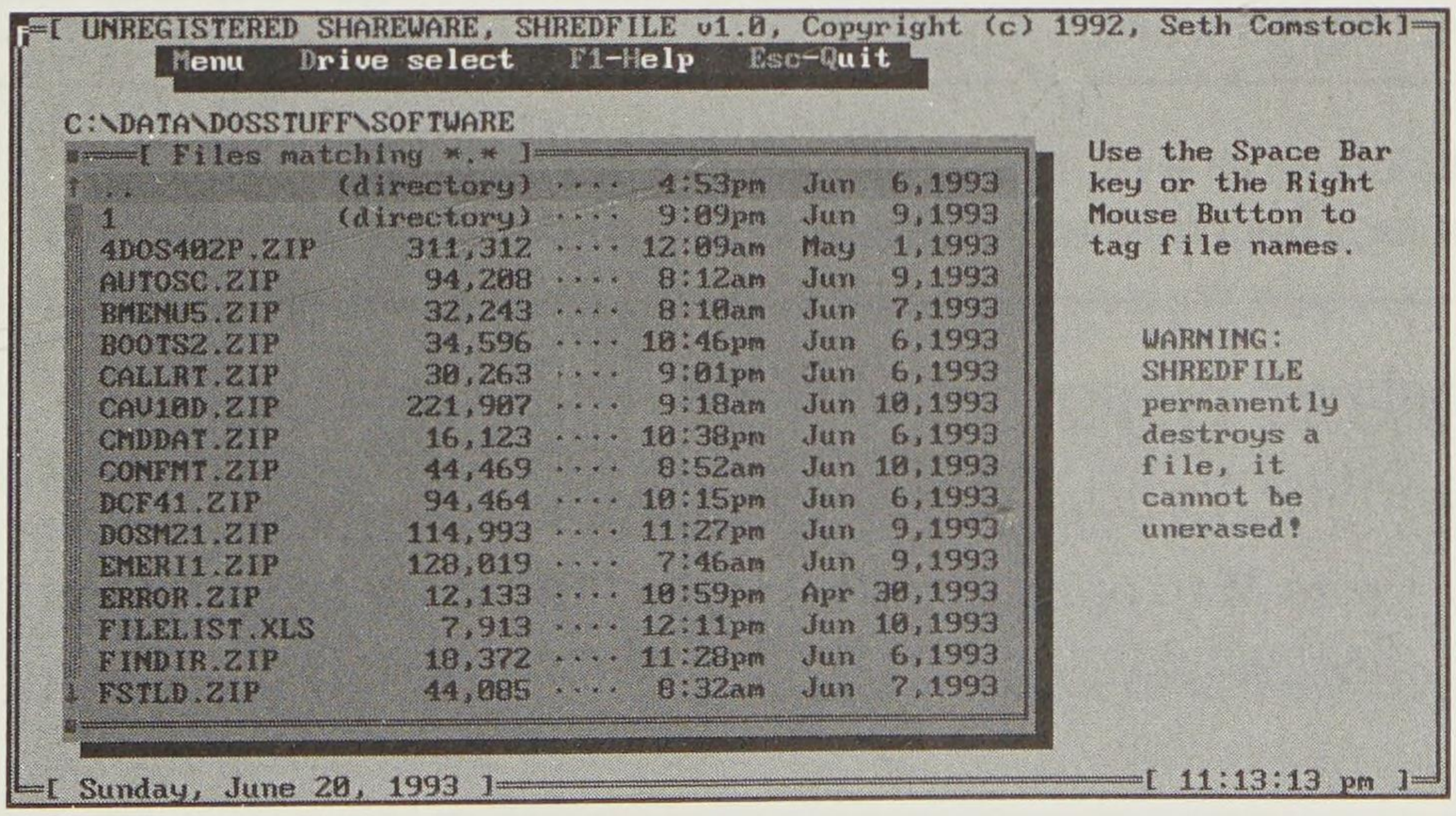


Fig. 2.5
The SHREDFILE Main Screen.

UNDERSTANDING THE SHREDFILE SCREEN

The SHREDFILE screen is divided up into three areas. At the top of the screen is a menu. From this menu, you can open the main SHREDFILE menu, select a new drive to use, display help or quit the program. To select an item from this menu, press one of the highlighted keys. For example, to display the SHREDFILE menu, press M, since the option **M**enu displays the highlighted letter M.

The center of the screen is where you do most of your work. This area displays the current directory. The window displays files in the current directory. Use the highlight bar in the window to select files that you want to shred. To move the highlight bar, use the arrow keys. When you select a file with the highlight bar, you can *tag* or mark a file to delete by pressing the space bar. Alternatively, you can highlight and tag files using the mouse. Point to a file and click the left mouse button to highlight a file. Point to a file and click the right mouse button to tag a file.

The bottom of the screen is the status area. This area displays the current date and time.

SELECTING FILES TO SHRED

When you start SHREDFILE, the current directory displays on the screen. You can sort the list of files, display files meeting a file specification, select files to shred, or change to a different directory to work.

NOTE

> If you have a mouse attached to your computer, you can select files and menu items using the mouse.

SORTING FILES

To sort the file list, follow these steps:

1. Press **M** to open the main **M**enu. The main menu opens displaying options on the **S**hred menu.

2. Press **O** to open the s**O**rt menu.
3. From the s**O**rt menu, choose **N**ame, **E**xt, **D**ate, or **S**ize to sort by the name of the file, the type of file (extension), the file date, or the file size, respectively.

REFINING THE FILE LIST

To display files matching a file specification, follow these steps:

1. Press **M** to open the main **M**enu. The main menu opens displaying options on the **S**hred menu.
2. Press **F** to open the **F**spec menu.
3. Choose **1** to display all files (*.*), **2** to display all backup files (*.BAK), or **E** to **E**nter file spec.

   If you choose **E**nter file spec, you are prompted to `Enter new file spec`. Type the file spec in the text box and press Enter. For example, to list all files with the TXT extension, type ***.TXT** and press Enter.

SHREDDING AN INDIVIDUAL FILE

To shred a single file, follow these steps:

1. Move the highlight bar using the arrow keys to select the file to shred.
2. Press **M** to open the main **M**enu. The main menu opens displaying options on the **S**hred menu.
3. Press **H** to choose **H**ilighted-shred.
4. Choose **F**ull or **Q**uick to shred the file. Shred completely overwrites the file. Or, press Esc to abandon shredding.

   When you choose **Q**uick, SHREDFILE overwrites the file one time. When you choose **F**ull, however, SHREDFILE overwrites the file three times, then a fourth time using random characters.
5. Press any key to continue, returning to the Shred screen.

CAUTION

If you are using a diskcache that uses delayed writes (other than SmartDrive, which comes with DOS), you must disable delayed writes before using SHREDFILE. For example, if you use the disk cache that comes with the Norton Utilities, use the command **NCACHE /-W** to turn off delayed writes. After you shred files, you can turn delayed writes back on. Again, using the Norton Utilities NCACHE program as an example, type **NCACHE /+W**.

SHREDDING MULTIPLE FILES

To shred multiple files, follow these steps:

1. Move the highlight bar using the arrow keys to select the first file to shred.
2. Press the space bar to tag the file. A triangle appears to the left of the file, and the highlight bar moves to the next file. Select the file again, and press the space bar to untag the file.
3. Continue performing steps 1 and 2 until you have tagged all the files to shred. Then continue with step 4.
4. Press **M** to open the main **M**enu. The main menu opens displaying options on the **S**hred menu.
5. Press **T** to choose **T**agged-shred.
6. Choose **F**ull or **Q**uick to shred the tagged files. Shred completely overwrites the files. Or, press Esc to abandon shredding.
7. Press any key to continue, returning to the Shred screen.

CHANGING THE DRIVE AND DIRECTORY

To change to another drive, follow these steps:

1. Press **D** to choose **D**rive select. A list of drives attached to your computer displays on the screen.

2. Press the letter of the drive to select. For example, to change to drive C, type **C**. A list of the files in the current directory of the drive you selected displays on-screen.

To change to another directory, press the arrow keys to move the highlight bar to the directory to change to. Either highlight the directory name to change to, or highlight the two periods (..) to change to the parent directory. Then, press Enter.

RUNNING SHREDFILE FROM THE COMMAND LINE

You can specify options when starting SHREDFILE. You can select the drive to use and select files to delete from the command line. To use a directory other than the default drive, use the following syntax:

SHRED d:

For example, if drive C is the current directory, and you want to shred files on drive A, type **SHRED A:** then press Enter.

To specify files to shred, use the following syntax:

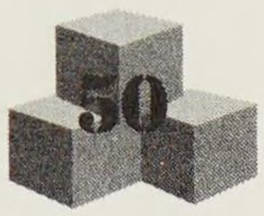
SHRED *filespec*

For example, to shred all files with the TXT extension in the current directory, type **SHRED *.TXT** and press Enter. When the menu appears on the screen, choose **F**ull or **Q**uick to shred the files that you specified. Of course, you can press Esc to abandon shredding files.

SHREDDING UNUSED DISK SPACE

You can use SHREDFILE to shred space on the disk not used by files. "Why," you ask, "would I want to shred space that no file uses?" Remember, if you use DOS' DEL or ERASE commands, only the file name is altered. All the information still resides on the disk. So you can use SHREDFILE to eliminate any information lurking on your disk that you don't want anyone else to get their hands on. To shred information in unused disk space, follow these steps:

1. Press **M** to choose the main **M**enu. The main menu opens displaying options on the **S**hred menu.



2. Press **P** to choose **P**urge freespace.
3. Choose **F**ull or **Q**uick to shred the free space on the disk. Shred completely overwrites the free space on the disk. Or, press Esc to abandon shredding.
4. Press any key to continue, returning to the Shred screen.

CUSTOMIZING THE SHREDFILE SCREEN

You can change the way that SHREDFILE operates by changing the setup. SHREDFILE enables you to display more information on the screen if you have an EGA or VGA video adapter and display. Additionally, SHREDFILE enables you to set a password that must be used before SHREDFILE operates.

To change the SHREDFILE setup, press **M** to open the main **M**enu. Press **U** to display the set**U**p menu. From the set**U**p menu, choose one of the following options:

Menu Option	*Description*
25 rows (reset to normal)	Displays 25 lines on the screen (the default setting).
28 rows VGA / 25 rows EGA	Displays 28 rows on a VGA screen or 25 rows on an EGA screen.
50 rows VGA / 43 rows EGA	Displays 50 rows on a VGA screen or 43 rows on an EGA screen.
Enter + save password	Enter a password you'll use to start SHREDFILE.

The first three options on the set**U**p menu changes the way that SHREDFILE displays on the screen. The fourth option enables you to set a password of up to 20 characters. If you set a password, SHREDFILE requires you to enter that password and press Enter each time you start SHREDFILE.

QUICKLY FINDING DIRECTORIES

There's nothing worse than looking for something and not being able to find it. Don't you hate that? It seems like you look and look and look—you even go back to places where you've already looked. Well, directories are no different.

Hard disks are a lot bigger than they used to be. About 10 years ago, the standard hard disk was 10M. Today, the standard hard disk is more than 200M. What does that mean? Simply that you can create more directories than before, making directories harder to find than before. But, using Find Directory, you not only can find a directory on your hard disk, but you can easily change to the directory once you find it. Heck, why be limited to a single hard disk? Find Directory enables you to change to a directory on another disk. You can find Find Directory in \DSMF\FINDIR on your hard disk.

To register Find Directory 4.0:

Guy Campbell
5397 Maricopa Drive
Simi Valley, CA 93063

Complete registration and price information is available in the documents included with this utility. If you're going to keep it and use it, please register it!

How do you use Find Directory? Use the following syntax:

FD *d:directory.*

where ***d:*** is the drive to search and ***directory*** is the directory that you are looking for.

NOTE

One thing to keep in mind—FD finds the first occurrence of a directory. Suppose that you have the directory C:\SPRDSHT and C:\DATA\SPRDSHT. If you type **FD SPRDSHT** and press Enter, FD changes to the first occurrence of SPRDSHT that it finds—every time.

Notice the period after ***directory*** in the syntax line above. When you specify ***directory***, FD assumes that the text you typed in place of ***directory*** is just a part of the directory name. However, when you include a period, FD represents ***directory*** as the entire directory name.

Suppose you want to find the directory SPRDSHT on your hard disk. You can type **FD SPRD** and press Enter, or type **FD SPRDSHT** and press Enter. Both commands will find the SPRDSHT directory.

Suppose, however, that you have two directories with similar names—WPLTR1 and WPLTR. If you type **FD WPLTR** and press Enter, Find Directory may find either of the directories, depending on which comes first. However, if you want to ensure that you find the WPLTR directory (and not WPLTR1) include the period—type **FD WPLTR.** and press Enter.

You should place FD.EXE in a directory included on your DOS path statement so that, no matter what your current directory is, DOS will find FD.

GROUPING FILES BY DATES

Anytime you work with files, such as copying files or deleting files, you can group them by name, using global characters. But, grouping files by name may not always be the best way to group them. From time-to-time, you might like to group files by date.

Suppose you worked on special project files between the dates of 12-15-93 and 12-22-93—a period of 8 days. Wouldn't it be great to specify that you want to copy the files within that eight-day period. That is what Command-Date can do for you. Command-Date enables you to use standard DOS commands, affecting files fitting within a certain date period. (You can find Command-Date in the \DSMF\CMDDATE directory.)

Command-Date doesn't just limit you to grouping files by date. You still can group files by name, using wild-card characters, and additionally group those files by date.

So, how do you use Command-Date? Just like this:

> **CMDDATE *command filespec commandoptions offset date /option***

where ***command*** is the command to perform (such as DEL or COPY), ***filespec*** is the files to affect, ***commandoptions*** are any options that you can use with ***command***, ***offset*** is the number of days to include, ***date*** is the beginning date, and ***/options*** are the command options that you can use with **CMDDATE**. The following table lists the options that you can use with CMDDATE.

To register Command-Date 1.0

RebelWare
Rt. 2 Box 210-A
Decatur, MS 39327

Complete registration and price information is available in the documents included with this utility. If you're going to keep it and use it, please register it!

CMDDATE options	*Description*
/C	Confirm a command before the command executes.
/N	By default, answer No to a prompt.
/S	Perform the command on subdirectories within the specified directory.
/T	Use test mode—emulates the command so that you can see the results before performing the actual command.

When you use CMDDATE, you must specify the command that you want to use, and either the date to begin the group, or the offset from the date; that is, how many days before or after the date the command should include.

Specify the date as mm/dd/yy or mm-dd-yy. For example, to perform a command on files created or changed on February 12, 1993, indicate the date as 2/12/93, or 2-12-93. Specify the offset using the @ symbol, a plus or minus, and the number of days in the offset. To copy files created from February 12, 1993, to February 15, 1993, use the following:

CMDDATE COPY *.* B: @+4 2/12/93

You do not have to specify the date. If you do not specify the date, CMDDATE assumes the current date. So, to copy to drive B all files created today and 4 days earlier (5 days worth of files in all), type the following command:

CMDDATE COPY *.* B: @-5

On the other hand, suppose you only want to copy all files to drive B created on July 10, 1993. Use the following command:

CMDDATE COPY *.* B: 7-10-93

Think you can find use for Command-Date? I *know* you will.

SUMMARY

So, what do you think? Great utilities, huh? Using these utilities will save you more time than you thought possible. You can use Con>Format to format diskettes without having to stop using your computer. Using Disk Copy Fast, you can duplicate disk-ettes without having to swap diskettes everytime memory fills up. SHREDFILE enables you to *completely* delete a file, providing ultimate security. With Find Directory, you can find directories and change to them quicker than ever before. And, how about that bonus? Before, you could always perform actions such as copying files by grouping them by name. But, with the bonus Command-Date, you can perform actions on files grouped by date.

USING A SMART PRINT SPOOLER

3

Computing is great—you can save so much time using a computer. And you can get a lot more done with a computer in a given amount of time than you can without one. It's amazing to think about the number of papers you used to wad up and throw away when writing a report using pen and paper. Get to the end of the page and make a mistake—oooh! Doesn't that burn you up? So now you use a word processor and fix those errors *before* you print.

Printing? It's the worst part of computing. You can do so many things quickly with your computer. Then you have to print. While you're printing, you wait. And wait. And wait. Hmmmph. "If only I could continue working with my computer while I'm printing. But *DOS* won't *let* me," you grumble sarcastically.

MICROSOFT FORGOT:

- A print-spooling utility to maximize computing time.

Grumble no more! With Spooling/Despooling Software, you can print even the longest document and continue working while your printer is humming away.

WHAT IS SPOOLING?

The actual meaning of *spooling*, according to the American Heritage Dictionary of the English Language, 3rd Edition, on CompuServe, is "the temporary storage of information that occurs while that information awaits further processing." In other words, spooling is a holding tank for information traveling from one place to another.

Generally, when you print information from an application, the computer sends the information directly to the printer. The printer interprets the information and prints the information to the paper. The printer can only process a small amount of information at a time. So, when the computer sends as much information as the printer can handle, the printer sends a signal back to the computer saying, "Hold up, will you?" The computer responds, "Just let me know when I can send more." While the computer and printer are having this friendly little chat, you sit there glaring at the screen and mumbling, "When will this fool thing finish printing so I can get on with other work?"

A *print spooler* is a program that stays in memory and captures information that you print from an application. In turn, the print spooler sends information to the printer. The kick is that although the information is being sent to the print spooler, your application thinks it is printing to a printer. The application just keeps spewing information to the print spooler like nobody's business. When finished sending information to the print spooler, that nasty little "Busy Printing" message leaves the screen and you can again use your application. (For that matter, you can exit the application and start another.) All of this took just a few seconds—far less time than if you had waited for the entire document to print on the printer.

"Fine," you say, "but what happens to the information in the spooler?" This is the great part: the print spooler *despools* the information to the printer. That is, the print spooler begins

sending information to the printer. While you continue working with your computer, the print spooler prints to the printer—in essence, your computer is performing two tasks at once.

You see, your computer doesn't spend all its time processing. For example, when you type a letter, you can't type fast enough to occupy all the processing power your computer has. The print spooler uses some of this otherwise wasted time to send information to the printer. Even though the computer seems to be doing two things at once, it really is only sharing its processing power between two tasks. But, hey, who cares? With print spooling, you can get more done than ever before.

USING THE SPOOLER/DESPOOLER

From the previous section, you can see that spooling saves a considerable amount of time. Included with this book is a great and versatile print-spooling program—Spooling/Despooling Software. You can find this program in the directory \DSMF\SP3.

This print-spooling software is actually broken into several programs. Each program enables you to perform a different function of print spooling. (For more information, see the section "Understanding the Separate Spooling, Monitoring, and Despooling Programs," later in this chapter.) This section describes the program that most computer users will implement—SP3SPDSP.EXE. With this program, you can spool and despool information. When you print, Spooler/Despooler captures into a file the information that is going to the printer, and sends that file to the printer.

> *To register Spooling/Despooling Software 2.0:*
>
> Budget Software Company
> P.O. Box 621534
> Littleton, CO 80162-1534
>
> Complete registration and price information is available in the documents included with this utility. If you're going to keep it and use it, please register it!

STARTING SPOOLER/DESPOOLER

To start Spooler/Despooler, use the following syntax:

```
SP3SPDSP /pn: DIR=splpath\ KEEPFILES=Y/N BIOS=Y/N
BUFF=buffsize /T
```

The following table lists the parameters you can use with SP3SPDSP and a description of each one.

Parameter	*Description*
/pn:	Enables you to specify the port and port number to use. Replace ***p*** with L for parallel or C for serial, and ***n*** with the numbers from 1 through 4. /L3: means to use parallel port 3. The default is /L1:.
DIR=*splpath*	Indicates the directory to store the temporary spool files. You must include the final backslash character. **DIR=C:\TEMP** means to use the directory C:\TEMP to store the spool files. The default is C:\DS4\SPOOL\L1\ for L1, C:\DS4\SPOOL\L2 for L2, and so on.
KEEPFILES=*Y*/*N*	Indicates whether to keep the temporary spool files, or delete them when the document prints. **KEEPFILES=N** means to delete the spool files after they print. The default is **KEEPFILES=Y.**
BIOS=*Y*/*N*	Indicates whether or not to use the BIOS as the printing mechanism. The default is **BIOS=N.**
/BUFF=*buffsize*	Enables you to indicate how much memory to use as buffer space to

Parameter	*Description*
	speed up information sent to the printer. ***buffsize*** is a number from 01 to 64, representing 1K buffer space. **/BUFF=10** means to use 10K memory as buffer space. The default is **/BUFF=2.**
/T	Removes SP3SPDSP from memory. Use this parameter by itself.

How about some examples of starting Spooler/Despooler? The first example shows starting Spooler/Despooler capturing information going to LPT1. This example stores spool files in C:\SPOOL, deletes the spool files once they print, uses the BIOS for printing, and sets aside 5K for buffer space. Use the following syntax:

```
SP3SPDSP DIR=C:\SPOOL\ KEEPFILES=N BIOS=Y /BUFF=5
```

Notice that the example did not specify the ***/pn:*** parameter. That is because /L1: is the default. Also, the directory you specify for the temporary spool files must exist. The only exception is the default directory, which Spooler/Despooler creates automatically.

The second example starts Spooler/Despooler using COM2, capturing the files to C:\SPOOL, deleting spool files, using the BIOS for printing, and using 10K for buffer space. Use the following syntax:

```
SP3SPDSP /C2: DIR=C:\SPOOL\ KEEPFILES=N BIOS=Y /BUFF=10
```

Since Spooler/Despooler is a TSR program, Spooler/Despooler takes memory. The third example shows how to remove Spooler/Despooler from memory:

```
SP3SPDSP /T
```

PRINTING TO SPOOLER/DESPOOLER

Actually, you don't need to do anything special when printing with Spooler/Despooler loaded in memory. But this section gives you some information to help you use Spooler/Despooler.

When you print files, Spooler/Despooler creates files in the directory you specify to contain the temporary spool files. It names the first file it creates L1000001.SPL. The second, L1001001.SPL. The third, L1002001.SPL. Spooler/Despooler continues naming files in this fashion.

The first two places of the file name refer to the port being captured. For example, the first spool file created when capturing LPT2 is L2000001.SPL. The second spool file created when capturing COM2 is C2001001.SPL.

The next three places in the filename is the spool file number. For example, L1012001.SPL is the twelfth spool file printed. You can maintain up to 1,000 spool files. The last three characters in the file name are always the same: 001.

Why is it important to know about these file names, and why is it important to keep files? This is simply a time-saving mechanism. Suppose you start Spooler/Despooler using the following parameters:

```
SP3SPDSP /L1: KEEPFILES=Y
```

Then you start a CAD program and load an intricate drawing. Processing takes quite some time to display the drawing, calculating the many angles and lines that make up the drawing. Then you print the drawing. Of course, Spooler/Despooler captures the information going to the printer and frees up your computer. So, you exit the CAD program and wait for the printing to finish.

Oh, no. You were supposed to print *two* copies of the drawing. What a pain to have to start the CAD program again, load the drawing, wait for the processing, and print the drawing. "Oh," you think to yourself, "when I started Spooler/Despooler, I specified to *keep* spool files." Yes, since you kept the spool file, you can simply spool that file to the printer again.

You can use the SP3PRT.COM program from the command prompt to reprint the spooler file with the following command:

```
SP3PRT C:\DS4\SPOOL\L1\L1000001.SPL LPT1
```

This command reprints the spool file to the LPT1 port. If you think you might have to reprint the file again, copy this file to another filename, preserving the spool file for later use. For example, use the following command to preserve the first spool file:

```
COPY C:\DS4\SPOOL\L1\L1000001.SPL C:\DRAWINGS\DRAWING1.PRN
```

OK, in this example, the drawing is called DRAWING1.PRN. The name DRAWING1.PRN is not as explanatory as it could be. You can use a more explanatory file name—for example, BIGHOUSE.PRN. This *is* just an example. The point is, you can save the spool file for later use. Don't save too many of these files, however. If you print a large amount of information, especially graphics, the spool files are large. You can waste a lot of hard-disk space by saving old spool files.

UNDERSTANDING THE SEPARATE SPOOLING, MONITORING, AND DESPOOLING PROGRAMS

In the previous section, you saw how to use Spooling/Despooling software to spool information going from your application to your printer. You used the SP3SPDSP.EXE program. However, several programs are included in Spooling/Despooling Software, enabling you to spool, despool, and monitor information going to your printer. The following sections show you how to use each of the different programs that come with Spooling/Despooling Software.

USING SPOOLER

You can use SP3SPL.EXE to capture information from an application into spool files. However, after capturing the information,

the information is not sent on to the printer. Rather, the information resides in the spool files.

The syntax for SP3SPL.EXE is as follows:

```
SP3SPL /pn: DIR=splpath\ /BUFF=buffsize /T
```

The following table lists the parameters you can use with SP3SPL and a description of each one.

Parameter	*Description*
/*pn:*	Enables you to specify the port and port number to use. Replace ***p*** with L for parallel or C for serial, and ***n*** with the numbers from 1 through 4. /L3: means to use parallel port 3. The default is /L1:.
DIR=*splpath*	Indicates the directory to store the temporary spool files. You must include the final backslash character. **DIR=C:\TEMP** means to use the directory C:\TEMP to store the spool files. The default is C:\DS4\SPOOL\L1\ for L1, C:\DS4\SPOOL\L2 for L2, etc.
/BUFF=*buffsize*	Enables you to indicate how much memory to use as buffer space to speed up information sent to the printer. *buffsize* is a number from 01 to 64, representing 1K buffer space. **/BUFF=10** means to use 10K memory as buffer space. The default is **/BUFF=2.**
/T	Removes SP3SPDSP from memory. Use this parameter by itself.

Now that you've captured the files, you can despool them at a later time.

USING DESPOOLER

A despooling program called SP3DSP.EXE comes with Spooling/Despooling Software. You can use this program to despool spooled files. The syntax for SP3DSP.EXE is as follows:

```
SP3DSP /pn: DIR=splpath\ /KEEPFILES=Y/N BIOS=Y/N /T
```

The following table lists the parameters you can use with SP3SPDSP and a description of each one.

Parameter	*Description*
/pn:	Enables you to specify the port and port number to use. Replace ***p*** with L for parallel or C for serial, and ***n*** with the numbers from 1 through 4. /L3: means to use parallel port 3. The default is /L1:.
DIR=*splpath*	Indicates the directory to look for temporary spool files. You must include the final backslash character. **DIR=C:\TEMP** means to look in the directory C:\TEMP for the spool files. The default is C:\DS4\SPOOL\L1\ for L1, C:\DS4\SPOOL\L2 for L2, and so on.
KEEPFILES=*Y*/*N*	Indicates whether to keep the temporary spool files or delete them when the document prints. **KEEPFILES=N** means to delete the spool files after they print. The default is **KEEPFILES=Y.**
BIOS=*Y*/*N*	Indicates whether or not to use the BIOS as the printing mechanism. The default is **BIOS=N.**
/T	Removes SP3SPDSP from memory. Use this parameter by itself.

To start a file despooling, use the program SP3PRT.COM, described later in this chapter.

USING MONITOR

You use Monitor when you are spooling and despooling files in a network. Monitor ensures that all files have a different file name. The program for Monitor is SP3MON.EXE. The syntax is as follows:

```
SP3MON /pn: DIR=splpath\name.ext MC=s
```

The following table lists the parameters you can use with SP3MON and a description of each one.

Parameter	*Description*
/pn:	Enables you to specify the port and port number to use. Replace ***p*** with L for parallel or C for serial, and ***n*** with the numbers from 1 through 4. /L3: means to use parallel port 3. The default is /L1:.
DIR=*splpath\file.ext*	Indicates the directory and file skeleton for the temporary spool files. The skeleton is a file specification using wildcards. For example, **DIR=C:\TEMP*.*** means to look to the **C:\TEMP** directory and rename any spool file there to a sequential name. The default is C:\DS4\SPOOL\L1*.SPL for L1, C:\DS4\SPOOL\L2*.SPL for L2, and so on.
MC=*S*	Indicates the number of seconds that Monitor should wait after a spool file is completed before it should rename the file. s is the number of seconds from 1 to 9. For example, **MC=5** means to wait for 5 seconds. The default is **MC=3.**
/T	Removes SP3SPDSP from memory. Use this parameter by itself.

When spooling and despooling files using a network, it is possible that two different people could print their first file of the day at the same time to the same printer. Since the spool directory is the same (the two users are sharing the same printer attached to a computer on the network), both users are creating a file called L1000001.SPL. Since two files by the same name cannot reside in the same directory, Monitor steps in and renames the files using different file extension. One file is renamed L1000001._01. Monitor renames the second file L1000001._02. Both files despool correctly.

USING DESPOOLER/MONITOR

Often, when you monitor files, you will also despool them. This section shows how to use the combined Despooler/Monitor program called SP3DSPMN.EXE. The syntax is as follows:

```
SP3DSPMN /pn: DIR=splpath\name.ext /KEEPFILES=Y/N BIOS=Y/N /T
```

The following table lists the parameters you can use with SP3SPDSP and a description of each one.

Parameter	*Description*
/*pn*:	Enables you to specify the port and port number to use. Replace ***p*** with L for parallel or C for serial, and ***n*** with the numbers from 1 through 4. /L3: means to use parallel port 3. The default is /L1:.
DIR=*splpath\file.ext*	Indicates the directory and file skeleton for the temporary spool files. The skeleton is a file specification using wildcards. For example, **DIR=C:\TEMP*.*** means to look to the C:\TEMP directory and rename any spool file there to a sequential name. The default is C:\DS4\SPOOL\L1*.SPL for L1, C:\DS4\SPOOL\L2*.SPL for L2, and so on.

continues

Parameter	*Description*
KEEPFILES=*Y*/*N*	Indicates whether to keep the temporary spool files or delete them when the document prints. **KEEPFILES=N** means to delete the spool files after they print. The default is **KEEPFILES=Y.**
BIOS=*Y*/*N*	Indicates whether or not to use the BIOS as the printing mechanism. The default is **BIOS=N.**
/T	Removes SP3SPDSP from memory. Use this parameter by itself.

This program works just like the Monitor and Despooler programs described above.

USING SPOOLER/MONITOR/DESPOOLER

The most complete program incorporates spooling, monitoring, and despooling. This program is called SP3ALL.EXE. The syntax for SP3ALL is as follows:

```
SP3ALL /pn: DIR=splpath\ KEEPFILES=Y/N BIOS=Y/N
/BUFF=buffsize /T
```

The following table lists each parameter that you can use with SP3SPDSP and a description of the parameter.

Parameter	*Description*
/pn:	Enables you to specify the port and port number to use. Replace ***p*** with L for parallel or C for serial, and ***n*** with the numbers from 1 through 4. /L3: means to use parallel port 3. The default is /L1:.
DIR=*splpath*	Indicates the directory to store the temporary spool files. You must include the final backslash character. **DIR=C:\TEMP** means to use the directory

Parameter	*Description*
	C:\TEMP to store the spool files. The default is C:\DS4\SPOOL\L1\ for L1, C:\DS4\SPOOL\L2 for L2, and so on.
KEEPFILES=*Y*/*N*	Indicates whether to keep the temporary spool files, or delete them when the document prints. **KEEPFILES=N** means to delete the spool files after they print. The default is **KEEPFILES=Y.**
BIOS=*Y*/*N*	Indicates whether or not to use the BIOS as the printing mechanism. The default is **BIOS=N**.
/BUFF=*buffsize*	Enables you to indicate how much memory to use as buffer space to speed up information sent to the printer. ***buffsize*** is a number from 01 to 64, representing 1K buffer space. **/BUFF=10** means to use 10K memory as buffer space. The default is **/BUFF=2**.
/T	Removes SP3SPDSP from memory. Use this parameter by itself.

QUEUEING DOS FILES

A program is enclosed with Spooling/Despooling Software that sends a spool file to the printer: SP3PRT.COM. The syntax for this program is as follows:

```
SP3PRT filename port
```

Where ***filename*** is the complete file name of the spool file to print including the path, and ***port*** is the port to print to. For example, to send the spool file L1000001.SPL from the C:\SPOOL directory to the printer attached to LPT2, use the following command:

```
SP3PRT C:\SPOOL\L1000001.SPL LPT2:
```

SUMMARY

Printing doesn't have to be a hassle. There is no need to wait around for your printer to print your document when you really need to finish other work before you can get out of the office. Using the print spooler shown in this chapter, you can print to your heart's content. While the spooler sends your print jobs to the printer, you can continue to use your computer for other applications. Spooling/Despooling software will make you as productive as DOS should have let you be to begin with.

ENHANCING COMMUNICATIONS WITH DOS

4

Communication is a necessity of modern life. We have desk phones, pay phones, car phones, portable phones. We have devices for when we are away from the phones—beepers, answering machines, voice mail. One day, should technology keep advancing, maybe we'll have electronic implants in our heads so we can talk to people simply by thinking. Just think—what power, to be able to contact anyone anywhere for absolutely *no reason at all*!

(My idea of communication fun: Set up two phones on automatic redial so that they call each other at alternate cycles until their little ringers smoke. Heh, heh, heh. The ultimate in phone-frenzy.)

This chapter enables you to use your computer to communicate better. Included with this book is a program that dials a phone

MICROSOFT FORGOT:

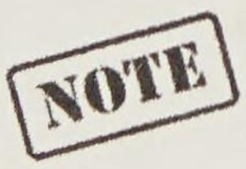
- A way for your messages to come to you.
- A means of direct communication between computers.
- A translator for hard-to-fathom DOS error messages.

number when you have messages on your answering machine; a program that enables you to communicate files between computers without using diskettes; and a program that displays an easily understood English-language message instead of a cryptic DOS error code.

OK, OK, the last program is a bit different from the first two. But what good is communication if you can't understand what is being said? So, the third program translates *unintelligible* error codes into intelligible messages.

GETTING MESSAGES WITH PORTACALL

Do you have an answering machine in your home or at your business? When you're away from home, how do you know you have messages? Well, you call home and check your messages (if your answering machine is the beeperless remote type). That's fine if you're calling from a local phone number. But what if you use a portable phone? Or what if it's long distance to call your answering machine? Calling to check for messages can be costly. Wouldn't it be great if your computer called *you* when you have a message? Using PortaCall, it can!



> In order to use PortaCall, your computer must be equipped with a modem.

PortaCall is a program you run on your computer to monitor your phone line. Using the modem in your computer, PortaCall *listens* for your phone to ring, answer, and hang up. Once the caller hangs up, PortaCall dials your phone number, rings your phone a couple of times, then hangs up. That's your cue that you have a message. Pretty neat, huh? You can find PortaCall on your hard disk in the directory \DSMF\PORTCALL.

To register PortaCall 2.0:

Bill Dineen
455 Hope Street #3F
Stamford, CT 06906

Complete registration and price information is available in the documents included with this utility. If you're going to keep it and use it, please register it!

To use PortaCall, use the following syntax:

PCALL phoneno mintowait numring C#

The parameters are as follows:

phoneno The phone number to call when you have a message. Generally, PCALL dials the phone using tones. To dial using pulses, precede the phone number with the letter *P* (P555-1234 instead of 555-1234).

mintowait The number of minutes that PCALL should wait after receiving a message before it dials the number **phoneno**.

numring The number of times that PCALL rings your phone to notify you that you have a message.

C# The communications port that your modem is attached to. If your modem is attached to COM2, type **C2**.

Before you can use PCALL, make sure your phone jack is plugged into the *line* connector of your modem, and your answering machine plugged into the *phone* connector of your modem. Then, start PCALL. For example, suppose your portable phone number is 555-1234, you want PCALL to wait 2 minutes after you get a message, you want PCALL to let your phone ring only 2 times, and your modem is attached to serial port COM2. Enter the following command:

PCALL 555-1234 2 2 C2

When you do, PCALL displays the information on your screen shown in figure 4.1.

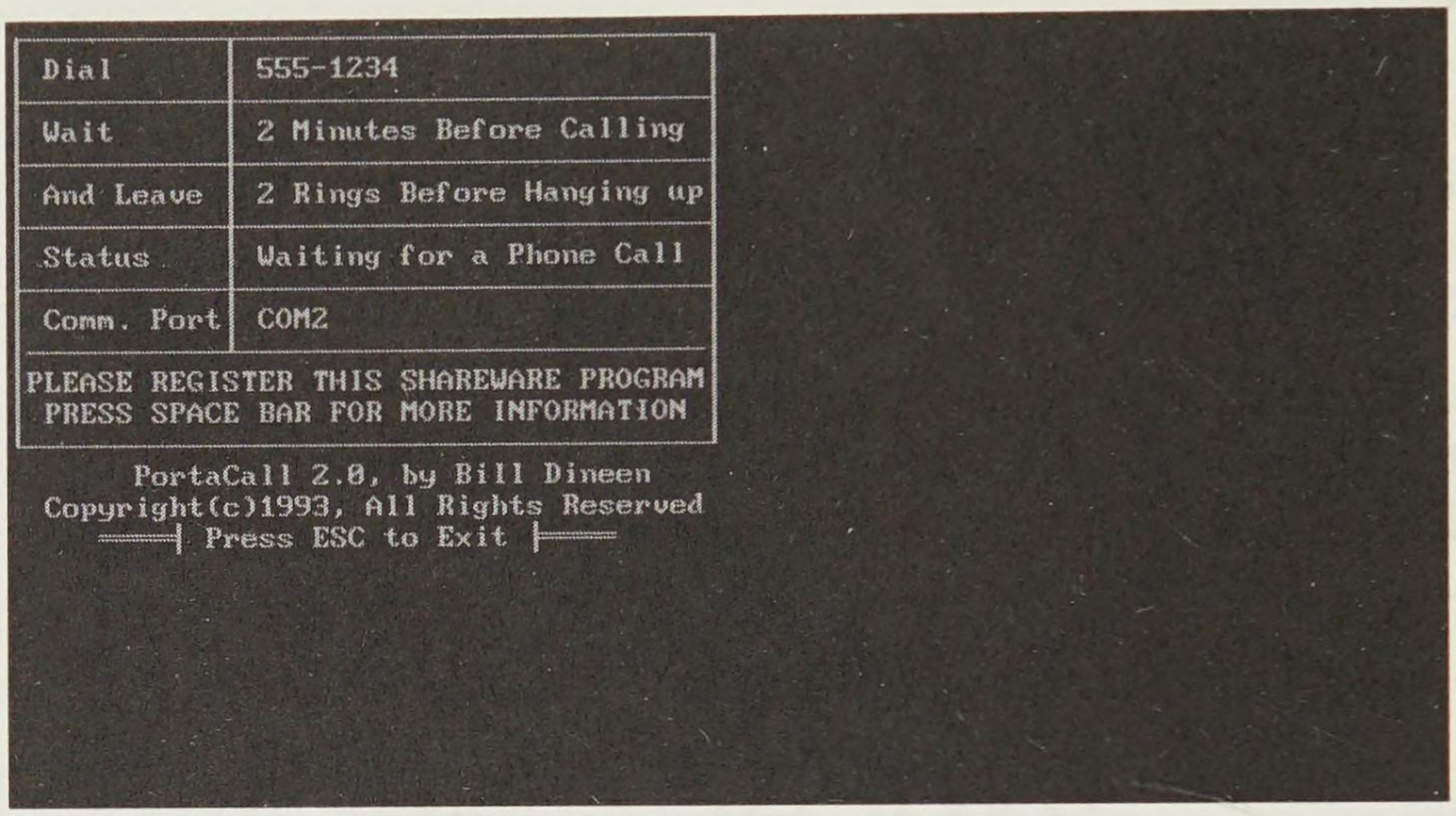


Fig. 4.1
PortaCall displays the "waiting statistics" on the screen.

Don't worry about hang-ups. PortaCall dials your phone number only when you actually receive messages. PortaCall simply ignores calls that hang up before your answering machine picks up.

Sure, PortaCall was designed with the portable phone in mind. But don't limit yourself to only a portable phone. Use PortaCall anytime you are away from your answering machine. Expecting an important call at home, but have to run to the office? Use PortaCall and specify your office number. Have to go next door to the neighbors? Use PortaCall specifying the neighbors' number. You will find many uses for PortaCall.

TRANSFERRING FILES USING ZIP

Do you have two computers and spend time transferring files between them? Maybe you have a desktop and a laptop computer, and you generally use your desktop computer. When you have to go on the road, however, you need to copy files to your laptop. You could use diskettes—but what a hassle. All the files you want won't fit on one diskette. Don't you wish you could just hook up your computers and transfer the files from one computer to the other? Well, you can.

ZIP File Transfer is a fast file transfer program. Using ZIP, you can transfer files from one computer to another without using diskettes. You just plug the two computers together using serial ports and a standard null-modem serial cable. You can find ZIP in the directory \DSMF\ZIP.

To register ZIP File Transfer :

Eric Meyer
3541 Smuggler Way
Boulder, CO 80303

Eric Meyer offers single-system and multi-system site licenses.

Complete registration and price information is available in the documents included with this utility. If you're going to use it, please register it!

GETTING READY TO USE ZIP

Before you can use ZIP, you must ensure that your computers are equipped to use ZIP. ZIP requires that a computer have one free serial port (also called a COM port). Most computers have at least one serial port. However, many computers have a mouse attached to the serial port. So be sure you have at least one free serial port on your computer. You can detach your mouse while you use ZIP.

The second thing you must have is a cable. The cable is important to using ZIP. (Trying to use ZIP without a cable is like trying to water the flowers without a hose.) You must use a null-modem cable to attach one computer to the other. You can purchase a null-modem cable at most stores that sell computers (and some stores that don't).

NOTE

A *null-modem* cable is a special type of serial cable. It takes the place of modems, which you usually use for communications between two computers. You can build a null-modem cable yourself, or purchase one. The next section discusses building and using a null-modem cable.

Before you run out and buy a null-modem cable, take a look at the serial ports on the computers. Some serial port connectors have 25 pins. Others have 9 pins. Check to see what type of serial ports you will connect to. If one computer has a 9-pin and the other has a 25-pin serial port, then you need a cable with a 9-pin connector on one end and a 25-pin connector on the other. (Kind of a no-brainer, huh?)

When you are purchasing a null-modem cable, see if you can find one that has both a 9-pin and a 25-pin connector at each end of the cable. If you purchase a cable like this, you have a very flexible cable. No matter what type of serial ports you have, you can connect the cable to them. This is especially useful if you use the cable on several different computers.

Finally, before you use ZIP, you must have a copy of ZIP on each computer. You can copy ZIP to each computer using diskettes. You also can transfer the ZIP program files from one computer to the other over the serial ports using a program that comes with ZIP (called ZIPDUP). For more information about using ZIPDUP, see the section "Transferring ZIP to Another Computer."

As a recap, check the serial ports on your computer. Make sure you know which ones you'll use. Plug the cable to the serial port on each computer. Note whether you are plugging the cable onto COM1, COM2, COM3, or COM4.

TRANSFERRING FILES

You can use ZIP in two ways. One way is by using ZIP's menus. The second is by using ZIP commands from the command line. This section, and the section titled "Using Server Mode," show you how to use ZIP menus. To issue ZIP commands from the command line, refer to the section "Using the Command Line."

STARTING ZIP

You can choose from many parameters when starting ZIP. However, when starting ZIP to transmit files using the menus, you can really only use two parameters to set up serial communications. The following syntax line shows the parameters to use when starting ZIP for menu use:

ZIP */p /Br*

where */p* is the communications port, 1 through 4, and */Br* is the baud rate for communications. Valid baud rates for *r* are: 2400, 4800, 9600, 19200, 38400, 57600, and 115200. You can specify the baud rates using the first two digits of the baud rate. To indicate 9600 baud, for example, use the parameter /B96.

By default, ZIP starts using COM1 and a baud rate of 115200. Suppose, however, that you want to start ZIP using COM2 and 19200 baud. Use the following syntax line:

ZIP /2 /B19

When you start ZIP, the ZIP menu displays on the screen. Figure 4.2 shows the ZIP menu when ZIP was started using the command **ZIP /2**.

```
ZIP 1.63 (01 Jun 1993) Fast File Transfer
(c)1988-93 E. Meyer, all rights reserved

Speed=115200 bps    Port=02F8(COM2)

Send        Fetch      Directory   Log        serVer     Parameters
Receive     Options    Compare     dElete     Unlink     Quit       :
```

Fig. 4.2
You can use the ZIP menu to help transfer files from one computer to another.

To use ZIP to transfer files, you must start ZIP on both computers. You must specify the same baud rate for both computers. However, you do not have to use the same communications port.

Before you continue using ZIP, you should be familiar with two terms. They are *host* and *guest*. These aren't tough terms. If you have been to anyone's home, or had anyone visiting you, you can relate to these terms. The *host* is the computer containing the files that you want. The *guest* is the computer receiving the files. The best part about these is that you can terminate the relationship between the host and guest—that's kind of hard to do with people, especially if the guest brought the wine.

SENDING FILES

Sending files requires menu commands from both the guest and the host computer. On the host computer, you must change to the drive and directory containing the files to send. Then you choose to send files. On the guest computer, you simply choose to receive files. Then you specify the guest directory that ZIP copies the files to. The following steps show you how to send files from one computer to another:

1. On the host computer, choose **S**end. You are prompted as follows:

   ```
   SEND file(s)
   ```

2. At this prompt, type the name of the file or files to send and press Enter.

   Use standard DOS conventions for the file names. You can use the DOS wildcards * and ? in the file names. You can also specify the path of the file.

   You can type more than one file name at this prompt. If you do, separate the file names with commas. For example, to send all files with the .TXT and .DOC extensions in the C:\DATA directory, and a file named FRED.EXE in the current directory, enter the following:

   C:\DATA*.TXT, C:\DATA*.DOC, FRED.EXE

   You are prompted `Send to directory`.

3. At the prompt, type the name of the directory on the *guest* computer to send the files to, then press Enter.

For example, type **C:\DATA** and press Enter. The screen displays `Ready` on the host computer.

4. On the guest computer, choose **R**eceive. You are prompted as follows:

```
RECEIVE into directory
```

Since you specified the directory on the guest computer already, ignore this prompt and press Enter.

ZIP now sends the file or files that you indicated from the host computer to the guest computer. When ZIP is finished, the menus are again displayed on both computers.

TIP

When you specify files, you may include any of several option parameters. These are:

/T Only overwrite older files

/E Send only files that exist on the guest

/N Send only files that do not exist on the guest

/P Prompt before overwriting existing files

CAUTION

By default, ZIP overwrites files on the destination disk. Use the /P parameter when indicating the file to send so that ZIP prompts you before overwriting existing files.

CHANGING DIRECTORIES

You can change the current directory that ZIP uses to send files. For example, if you started ZIP from the C:\ZIP directory, you can change the current directory to C:\DATA.

Even though you can specify a path for files that you are sending, you will find times when changing the current directory is more convenient. If you are sending files from the same directory, you should change to the directory containing the files to send before sending the files. To do this, follow these steps:

1. On the host computer, choose **L**og. ZIP prompt as follows:

```
LOG directory (">"=server)
```

2. Type the name of the directory on the host computer containing the files to send, then press Enter. For example, type **C:\DATA\LETTER** and press Enter.

You can ensure that you have changed to the correct directory by choosing the command **D**irectory and pressing Enter. When you do, ZIP displays the contents of the current directory.

USING SERVER MODE

Sending and receiving files, as described above, is fine if the computers are within close range to one another—if they are next to one another, for example. However, if one computer is on the next desk, or across the office, running back and forth between the two computers can be a hassle (unless you are into the jogging thing). You will find *Server mode* to be much more handy.

"So, what is Server mode?" you ask, intrigued. When you use Server mode, you place the host computer in *automatic pilot* and control all file transfer from the guest computer. That is, you place the host in Server mode, so that it services all your requests.

(Actually, this mode makes more sense than the send and receive commands shown above. Server mode is much more convenient than typing commands at two different computers.)

STARTING SERVER MODE

To put a computer in Server mode from the menu, start ZIP as shown above. Then, from the menu on the host computer, choose Ser**V**er. The computer displays the message:

```
SERVER

[Press ESC to exit server mode]

Waiting
```

CONTROLLING THE SERVER FROM THE GUEST COMPUTER

Once the host computer is in Server mode, you can start controlling the host from the guest computer. You can fetch files from the host, and you can send files to the host. You can compare files and change the directory on the host. Additionally, you can delete files on either the host or the guest computers.

To fetch files from the host, follow these steps:

1. Start Server mode on the host computer by choosing SerVer.
2. Choose Fetch on the guest computer. The following prompt displays on the guest computer:

```
FETCH file(s)
```

3. At the prompt, type the file or files to fetch, then press Enter. You can type a single file name, a file name using wildcards, or a group of files, delimiting each with a comma. You also can specify the path to the file on the server. When you press Enter, the following prompt appears on the screen:

```
Fetch into directory
```

4. At the prompt, type the directory on the guest computer where you want to copy the files to, then press Enter. ZIP begins transferring the files.

To send files, follow these steps:

1. Start server mode on the host computer by choosing SerVer.
2. On the guest computer, choose Send. You are prompted as follows:

```
SEND file(s)
```

3. At this prompt, type the name of the file or files to send and press Enter.

Use standard DOS conventions for the file names. You can use the DOS wildcards * and ? in the file names. You can also specify the path of the file.

You can type more than one file name at this prompt. If you do, separate the file names with commas. For example, to send all files with the .TXT and .DOC extensions in the C:\DATA directory, and a file named FRED.EXE in the current directory, enter the following:

C:\DATA*.TXT, C:\DATA*.DOC, FRED.EXE

You are prompted `Send to directory`.

4. At the prompt, type the name of the directory on the host computer to send the files to, then press Enter. For example, type **C:\DATA** and press Enter. ZIP transfers the files.

To compare files, follow these steps:

1. Start Server mode on the host computer by choosing Ser**V**er.

2. On the guest computer, choose **C**ompare. You then see the following prompt:

```
COMPARE files (">"=server)
```

3. Type the name of the files to compare, including the path to the files, then press Enter. To use the files on the host computer as the original files, precede the file path with the greater-than sign. For example, to indicate all DOC files in the \TEMP directory on the host as the originals, type **>C:\TEMP*.DOC**. If you do not include the greater-than sign, the guest computer is the original.

You use pretty much the same steps to delete files as you did to compare files. To delete files, follow these steps:

1. Start Server mode on the host computer by choosing Ser**V**er.

2. On the guest computer, choose d**E**lete. You see the following prompt:

```
DELETE files (">"=server)
```

3. Type the name of the files to delete, including the path to the files, then press Enter. To delete files on the host computer, precede the file path with the greater-than sign. For example, to delete all DOC files in the \TEMP directory on the host, type **>C:\TEMP*.DOC**. If you do not include the greater-than sign, ZIP deletes files on the guest computer.

You can stop Server mode from the guest computer. Choose **U**nlink from the guest computer to exit Server mode on the host computer. The host returns to the ZIP menu.

USING THE COMMAND LINE

Any of the commands you perform from the ZIP menu can also be performed from the command line. When you issue the ZIP command, you add option parameters to send and receive files, and place computers in server mode. This section lists commands you use to perform command line options.

NOTE

When using ZIP from the command line, you can set the communications port and baud rate using the /1, /2, /3, or /4 option and the /B*r* option shown above. Additionally, you can prompt for files being overwritten with /P, copy existing files with /E, copy non-existing files with /N, and check that you are only sending the newest files using /T.

SENDING FILES FROM THE COMMAND LINE

The syntax that you use to send files using ZIP from the command line is as follows:

ZIP *d:\path\filename.ext /[dir]*

In the syntax, ***d:\path\filename.ext*** is the file name or file names, optionally including the location of the file or files, to

send from the host computer. The option ***/[dir]*** is the directory on the guest computer to send the files. When you specify the destination directory, you must include the directory in square brackets. To send all files with the extension DOC from C:\DATA on the host computer to C:\WORK on the guest computer, issue the following command:

ZIP C:\DATA*.DOC /[C:\WORK]

RECEIVING FILES FROM THE COMMAND LINE

The syntax that you use to receive files using ZIP from the command line is as follows:

ZIP /R[*dir*]

The option **/R[*dir*]** indicates that ZIP should receive files. You can indicate the directory on the guest computer where ZIP copies the files. If you indicate the destination directory with the send command, you do not need to indicate the destination directory using this command. To receive files being sent from the host computer, issue the following command on the guest computer:

ZIP /R

STARTING AND USING SERVER MODE FROM THE COMMAND LINE

From the command line, you can start and stop Server mode, enabling you to control both the host and guest computers from the guest computer. Use the following syntax from the host computer to start Server mode:

ZIP /S[dir]

The /S options initiate Server mode on the host computer. You can set the default directory by using the **[*dir*]** option when starting Server mode. To start Server mode using C:\TEMP as the default directory, for example, type the following command on the host computer:

ZIP /S[C:\TEMP]

Once you start Server mode on the host computer, you can issue commands on the guest computer to fetch files from and send files to the host computer. In order to fetch files from the host computer running in Server mode, use the following syntax:

ZIP *filename.ext* /F[*dir*]

In this syntax, ***filename.ext*** is the file or group of files to fetch from the host computer to the guest computer. The **/F[*dir*]** option indicates that ZIP should fetch the files to the directory on the guest computer indicated by **[*dir*]**. To fetch the file FRED.TXT from the C:\LETTERS directory on the host computer, and place the file in the C:\WORK directory on the guest computer, issue the following command:

ZIP C:\LETTERS\FRED.TXT /F[C:\WORK]

Keep in mind that you can use more than one file specification by separating each with a comma. For example, to fetch the files FRED.TXT and IRS.WK1 from the C:\TAX directory, use the following command from the guest computer:

ZIP C:\TAX\FRED.TXT, C:\TAX\IRS.WK1 /F

This command places the indicated files in the default directory on the guest computer.

There is no special command for sending files to a host in Server mode. To send files using the Server mode, use the same commands as shown above in the section "Sending Files."

Once you fetch or send files, you may want to compare the files with the original files. You can do this using the following command:

ZIP *filename.ext* /C[*dir*]

In this command, ***filename.ext*** is the file or group of files to compare. The parameter **[*dir*]** is the directory containing the files on the host computer. To compare the files *.DOC on the guest computer with the files in the C:\TEMP directory on the host computer, enter the following command:

ZIP *.DOC /C[C:\TEMP]

You can change to a different directory and display the contents of the current directory from the command line. To change the directory, use the following syntax:

ZIP /L[dir]

In this syntax, the [***dir***] parameter indicates the directory on the host computer to change to.

To display the contents of a directory on the host computer, use the following syntax:

ZIP /D[*filename.ext*]

In this command, you can indicate the file or group of files to display using the [***filename.ext***] parameter. If you only use the /D parameter, ZIP displays the entire contents of the current directory on the host computer.

You can delete files located on the host computer from the command line of the guest computer using the following command:

ZIP /K[*filename.ext*]

You indicate the file or group of files to delete by including the file specification in square brackets. To delete all files with the TXT extension, for example, enter the following command:

ZIP /K[*.TXT]

TRANSFERRING ZIP TO ANOTHER COMPUTER

It is quite possible that you are using ZIP because your two computers have incompatible disk drives. If that's the case, you might wonder how you're supposed to get ZIP from one computer to the other. "I need ZIP to transfer the files to the other computer," you reason, "but I can't, since I don't have ZIP *on* the other computer." There is only one word to use here: ZIPDUP.

You use the program ZIPDUP to duplicate ZIP, copying the ZIP files to the other computer. Follow these steps to use ZIPDUP:

1. Plug your trusty serial cable between the two serial ports on your computers. One catch—you can only use COM1 or COM2 for ZIPDUP. DEBUG.EXE must also be available on both computers.

> DEBUG.EXE is a utility that comes with MS-DOS from the original version through MS-DOS 5. Microsoft stopped including DEBUG with MS-DOS 6. (Why, oh why?) You still can obtain DEBUG from Microsoft, however, by ordering the MS-DOS 6 Supplemental Disk from Microsoft.

NOTE

2. On the computer that does not have ZIP, type the following commands, pressing Enter after each one:

   COM1:9600,N,8,1
   CTTY COM1

   The above commands assume you have the serial cable plugged into COM1. If you are using COM2, then replace COM1 with COM2 in the above commands.

3. On the computer that contains ZIP, type **ZIPDUP** and press Enter. The following prompt displays on the screen:

```
Are you ready to proceed with the Transfer? (Y/N)
```

4. Type **Y** and press Enter. Another prompt appears on the screen:

```
Which COM port is the cable connected to here? (1/2)
```

5. Type the number of the COM port that the cable is connected to—either **1** or **2**, then press Enter. ZIPDUP sends the files required to the other computer.

CONFIGURING ZIP

You can configure ZIP for use with your computer. For example, you can specify the communications ports to use, and the baud rate, just to name a couple of configuration options. You can set

the parameters when you are using the ZIP menu. Or you can change the default settings of ZIP. The next two sections show you how to change the configuration.

CHANGING THE ZIP CONFIGURATION FROM THE MENU

You can set several parameters from the ZIP menu. Choose **P**arameters to change the serial port and file attribute settings. When you choose **P**arameters, ZIP displays the menu shown in figure 4.3. The following table lists the settings that you can choose from the Parameters menu.

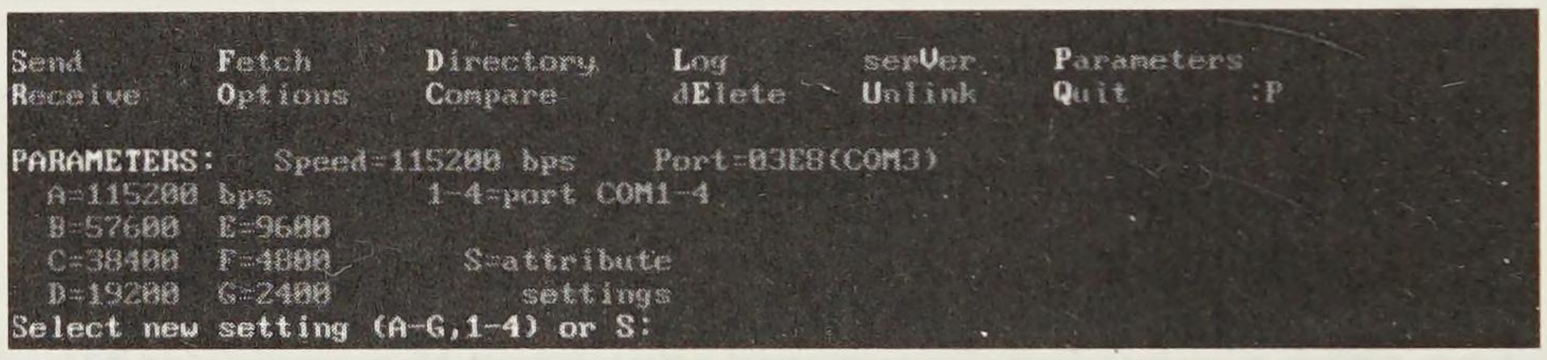


Fig. 4.3
The Parameters menu.

Parameter	*Description*
A	Set the serial port to 115200 baud.
B	Set the serial port to 57600 baud.
C	Set the serial port to 38400 baud.
D	Set the serial port to 19200 baud.
E	Set the serial port to 9600 baud.
F	Set the serial port to 4800 baud.
G	Set the serial port to 2400 baud.
1	Choose COM1.
2	Choose COM2.
3	Choose COM3.
4	Choose COM4.

Parameter	*Description*
SH	Toggle between displaying and not displaying files with the hidden attribute.
SS	Toggle between displaying and not displaying files with the system attribute.
SA	Toggle between displaying and not displaying files with the archive attribute.
SR	Toggle between displaying and not displaying files with the read-only attribute.

CHANGING THE DEFAULT ZIP CONFIGURATION

ZIP comes with a program that enables you to change the default parameters that ZIP uses. The program is called ZIPCFG.COM. To use ZIPCFG.COM, type **ZIPCFG** and press Enter. When you do, the menu shown in figure 4.4 appears on the screen.

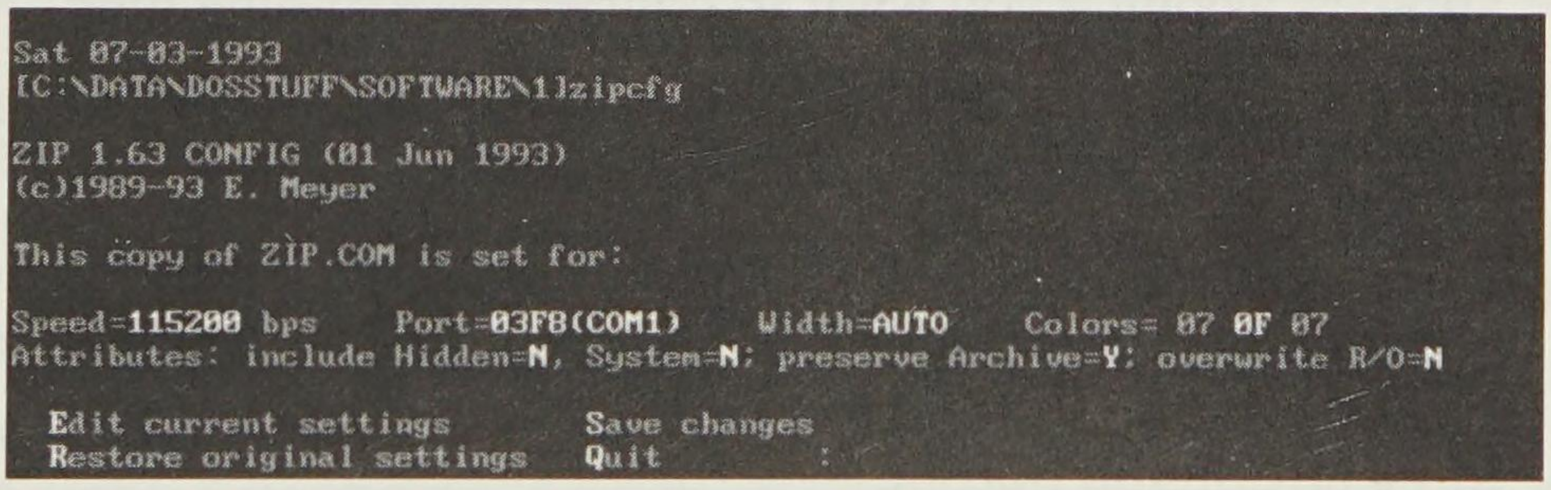


Fig. 4.4
Use ZIPCFG to change the default ZIP configuration.

To change the current ZIP configuration, follow these steps:

1. From the ZIPCFG menu, choose **E**dit current settings. You are prompted to enter the default communications port.
2. Type the communications port to use. For example, type **COM3** to use communications port 3. Then press Enter. A prompt displays asking for the default baud rate.

3. Choose the default baud rate by pressing the letter associated with the correct speed. For example, to choose 115200, type **A**. A prompt appears asking if you want to include hidden files.

4. Type **Y** to include hidden files when you transfer files. Type **N** to ignore hidden files. Unless you have some compelling reason, you should type **N**. A prompt appears asking if you want to include system files.

5. Type **Y** to include system files when you transfer files. Type **N** to ignore system files. Unless you have some compelling reason, you should type **N**. A prompt appears asking if you want to preserve archive status.

6. Type **Y** to preserve archive status when you transfer files. Type **N** to not preserve archive status.

   The archive attribute is used to indicate that a file is new or has been changed. Many backup programs use this attribute to back up changed or new files. Unless you have some compelling reason, you should type **Y**. A prompt appears asking if you want to overwrite read-only files.

7. Type **Y** to overwrite read-only files when you transfer files. Type **N** to not overwrite read-only files.

   Files are generally marked read-only when they should not be changed. Unless you have some compelling reason, you should type **N**. A prompt appears asking if you want to force 40-column output.

8. Type **Y** to display ZIP in 40-column mode, or **N** to enable ZIP to use the complete width of your display. Press **N**. A prompt appears asking if you want to display ZIP in color.

9. Press **Y** to display ZIP in color, or **N** to display ZIP in a single color.

   If you choose to display in color, ZIPCFG enables you to set the color for message color, highlight color, and active color. For each color setting, type the number associated with the foreground and background colors that you desire.

10. The ZIPCFG menu returns to the screen. Choose **S**ave changes to save all the settings that you made. ZIPCFG saves the changes that you made and returns the DOS prompt.

    If you don't want to save the changes you made, choose **Q**uit. ZIPCFG abandons all the changes and returns the DOS prompt.

You can return ZIP back to its original setting using ZIPCFG. Start ZIPCFG and choose **R**estore original settings. Then choose **S**ave changes.

DEFINING DOS ERROR MESSAGES

Many people have scoffed at DOS because of the cryptic C> prompt and the lack of English-language messages. You have to admit that the C> prompt doesn't really tell you what to do. However, after a few minutes of using DOS, you come to realize that the C> prompt is just that, a prompt for you to enter a command—a real no-brainer.

On the other hand, you must admit that DOS is a little lacking in its error-message display, although it's better than it used to be. From time-to-time, you still might see a message that says `Error 100`. Well, just what does that mean? To find out, use the ERROR program found in the \DSMF\ERROR directory on your hard disk.

ERROR is not shareware, but public domain software. That means you do not have to register the software in order to use it. However, if you want to send a "thank you" note to the author of ERROR, his address is as follows:

SL-EASY Software
5 West Canal Street
Navarre, OH 44662

ERROR is a little program that accepts the error code as a parameter, and in turn, displays the meaning of the message. To use ERROR, use the following syntax:

ERROR *nnn*

where ***nnn*** is a numeric value from 2 to 214.

So, what does `Error 100` mean? Type the command **ERROR 100** and press Enter. Error displays the message `Error code 100 is an I/O error - Disk read error`. Makes a lot more sense than Error 100, right? Now that you know what the error is, you have a fighting chance of determining what caused the error, and fixing the problem.

SUMMARY

Is communication a snap, or what? And just think, Microsoft forgot this stuff!

How are you going to better manage your phone messages? By using PortaCall, of course. And you can use ZIP to keep files up-to-date between two computer systems. Finally, use Error to get information about those pesky DOS error numbers. These three programs will help your computer help you become as good a communicator as you can be.

DON'T JUST FOLLOW THE PATH— MAKE IT!

5

One of the best DOS commands, but the least thought about, is the PATH command. After reading that comment, you might cock your head, wrinkle your eyebrows, and think to yourself, "The PATH command? Oh, the *PATH* command!" That's right, the old "use it in the AUTOEXEC.BAT file, set it once, and forget it" PATH command.

Just think about how critical that little fellow is to DOS! You use the PATH command to tell DOS which directories it can search through to find commands. For example, nearly everyone sets a search path to their DOS directory. Then, no matter what your current directory is, when you type CHKDSK and press Enter, the little utility that checks your disk (CHKDSK.EXE) runs. Is it a miracle that the CHKDSK.EXE program starts when you

MICROSOFT FORGOT:

- A flexible way to set and change search paths.

type the command? Not on your life! When you type the name of a DOS command, and press Enter, the command starts because DOS can find the command. PATH enabled DOS to find the command in the DOS directory.

Although the PATH command is a hoot, it could be better. The PATH command could provide you with more flexibility. For example, it doesn't enable you to edit the series of paths that you set. Rather, to change your search paths, you must retype the entire command. Additionally, you are limited as to how many paths that you can set. You will find two utilities on the disk, however, that give you flexibility you ordinarily would not have.

USING PATH

OK, maybe when you read the opening sentence you cocked your head, wrinkled your eyebrows, and thought, "The *PATH* command? What's this guy talking about?" A little reminder information about the PATH command can't hurt anyone.

In the introduction to this chapter, you read that PATH enables DOS to find commands. How does PATH help DOS to find commands? What would life be like without PATH?

HOW DOS LOOKS FOR COMMANDS

Generally, you create many different directories on a hard disk. When you enter a command at the DOS command line, DOS has a set of rules that it follows to run the command. First, DOS looks in memory to see if the command is an internal command (that is, a command already in memory) and runs the command if it is internal. If the command is not internal, however, DOS assumes it must be external (good guess, huh?) and looks to the disk to find the command. But here's the catch—DOS looks only in the disk's current directory for the command. If DOS does not find the command in the current directory, it spews that lovable message, `Bad command or filename`.

How about an example? Suppose you have four directories on your hard disk. (I know, you have more—just humor me.) Those directories are as follows:

C:\

C:\DOS

C:\WP

C:\WP\LETTERS

You currently are in the C:\WP\LETTERS directory. You want to copy the three new letters you just wrote to a diskette in drive A. Oops—the disk in drive A is not formatted. So, you type the command **FORMAT A:** on the command line and press Enter. DOS looks to see if FORMAT is a command already in memory. (You know FORMAT is not in memory, but DOS loves routine tasks.) Since FORMAT is not an internal command, DOS looks to the current directory on drive C for the command. However, since FORMAT.EXE is in the C:\DOS directory (remember, FORMAT is a DOS command), and not the C:\WP\LETTERS directory, DOS comes back and says (you guessed it), `Bad command or filename`. "Oh, pshaw!" you say. "How I long for something to help DOS find commands." Enter PATH.

HOW PATH HELPS DOS LOOK FOR COMMANDS

Using the PATH command is quite simple. At a command line, or in a batch file, simply enter the PATH command as follows:

PATH=*searchpath*

or

SET PATH=*searchpath*

The parameter ***searchpath*** is simply a list of directories through which you want PATH to search for commands. You separate each search directory with a semicolon. For example,

to set up DOS so that it searches in the directories C:\DOS, C:\UTILS, and C:\DOS\BATCH, enter the command as follows:

PATH=C:\DOS;C:\UTILS;C:\DOS\BATCH

Now, when you type a command, DOS looks internally, then externally—in the current directory, the C:\DOS directory, the C:\UTILS directory, and the C:\DOS\BATCH directory. If, after looking in all those places, DOS still does not find the command, it justifiably displays `Bad command or filename`. Oh, yeah—when DOS finds a command before reaching the last search directory, it leaves off searching the rest of the directories and runs the command. I mean, come on, even DOS has its limits on repetitive commands.

So how do you know what *your* search path is? At a DOS command line, type **SET** and press Enter. The DOS search path is stored in the *environment*, an area in memory. When you Enter the SET command, you might see something like the following:

```
COMSPEC=C:\DOS\COMMAND.COM
PROMPT=$P$G
PATH=C:\DOS;C:\WORD;C:\123;C:\
TEMP=C:\TEMP
```

Check out the third line. DOS stores the PATH *variable* in memory for reference. A variable is an element of memory for storing information, in this case the search path. There are four directories in the sample search path shown.

PATHS NEVER ETERNAL

Although PATH is better'n beans, it does have limitations, as mentioned above. Just a moment on those limitations will help you appreciate why the Patha utility is so great. There are two limitations to mention.

The first limitation that PATH has is that you cannot edit a search path. The only way you can change the search path is to enter the complete command again. That could be a pain in the wazoo if you have several directories in your search path.

The second limitation is the length of your search path. The length of your search path must be shorter than 127 characters—the length of a DOS command line. If you want to list several subdirectories in your search path, you could run out of command line space before you list all your search paths.

Do you want a better path? Then get busy with Patha.

EXPANDING YOUR PATH USING PATHA

So you want to edit your path and expand it, too. With Patha, you can...and more. Using Patha, you can add directories to your search path, delete directories from it, and completely replace your current search path with another search path.

> *To register PATHA 2.2:*
>
> HGB Enterprises
> P.O. Box 150706
> Ft. Worth, TX 76108
>
> Complete registration and price information is available in the documents included with this utility. If you're going to keep it and use it, please register it!

You use Patha from the command line. The following shows you the syntax to use with Patha:

PATHA *searchpath* /B /O /D /F /E /N /L

or

PATHA /C /B /O /D /F /E /N /L

Notice that there are several parameters that you can use with Patha. Table 5.1 lists each parameter, along with a description.

Table 5.1. Patha Parameters

Parameter	*Description*
searchpath	Directories to include in the path. Separate each with a semicolon. ***searchpath*** can also be a file name when used with /F.
/C	Uses the current directory.
/B	Adds the directory or directories to the beginning of the search path.
/O	Does not check for an existing directory or directories.
/D	Deletes the directory or directories from the search path.
/F	Uses the path listed in a text file.
/E	Enables you to edit the path.
/N	Replaces existing search path with one in ***searchpath***.
/L	Enables the search path to be as long as environment space allows, up to 1,024 characters long.

Now you know a little about Patha, what the syntax is, and what the parameters mean. You're ready to give the baby a whirl. (Not literally—you should always treat children with respect, the way they treat you.)

CHANGING YOUR SEARCH PATH

Basically, Patha offers two ways to edit your search path. The first way is using the /E switch. Using the /E switch, Patha displays the current search path on the screen enabling you to edit the search path as it exists. The editing keys for Patha are listed in Table 5.2.

Table 5.2. Patha Editing Keys

Editing Key	*Description*
Home	Moves the cursor to the beginning of the path.
End	Moves the cursor to the end of the path.
←	Moves the cursor left one character.
→	Moves the cursor right one character.
Alt-R	Restores the original search path line.
Alt-C	Clears the search path line.
Del	Deletes one character above the cursor.
Backspace	Deletes one character to the left of the cursor.
Ins	Toggles between Insert and Overwrite modes.
Enter	Stops editing the path, accepting all changes that you made to the search path.
Esc	Stops editing the path, abandoning any change that you made to the search path.

To edit your search path interactively, type the following and press Enter:

PATHA /E

The current search path displays on the screen, enabling you to make changes. Change the search path as you desire and press Enter to accept the changes or press Esc to abandon the changes.

You can use several switches with Patha to change the search path from the command line. Some of these switches you can use together. In the following examples, assume that the current directory for each example is C:\DOS\BATCH and the current search path for each example is C:\DOS;C:\UTILS;C:\.

In each of the following examples, you will see the command to use, and the results to the current search path:

- To add the current directory to the search path, type the following:

  PATHA /C

  Result: `C:\DOS;C:\UTILS;C:\;C:\DOS\BATCH`

- To add the current directory to the beginning of the search path, type the following:

  PATHA /C /B

  Result: `C:\DOS\BATCH;C:\DOS;C:\UTILS;C:\`

- To add the directory C:\WORD to the search path, type the following:

  PATHA C:\WORD /B

  Result: `C:\WORD;C:\DOS;C:\UTILS;C:\`

- To add the directories C:\WORD and C:\123 to your search path, type the following:

  PATHA C:\WORD;C:\123

  Result: `C:\DOS;C:\UTILS;C:\;C:\WORD;C:\123`

- To delete the directory C:\DOS from the search path, type the following:

  PATHA C:\DOS

  Result: `C:\UTILS;C:\`

- To add the current directory to the beginning of the search path, then interactively edit the search path, type the following:

  PATHA /C /E

  Result: `C:\DOS\BATCH;C:\DOS;C:\UTILS;C:\`

As you can see from the examples, even using Patha without the interactive edit feature (/E) is quite easy. Take note of a few things from the examples: you can specify more that one directory to add or delete to the search path if you separate each

directory with a semicolon; you can add the current directory to the search path; and you can delete a single or multiple directories from the search path just as you add them.

One switch you didn't see in the examples is the /O switch. This enables you to add a directory to the search path without checking to see if it exists. That means, if you do not use the /O switch, Patha generally checks to see if a directory exists before adding the directory to your search path—something that the PATH command does not do on its own.

While the examples you have seen might sell you on the necessity of Patha, you ain't seen nothin' yet! The next section shows some of the more advanced features of Patha.

POWERING WITH PATHA

So far you have seen that you can edit the search path. But what if you want to extend the length of your search path, or want to change the search path altogether? You will see that Patha enables you to completely change the search path in the following examples.

For these examples, assume the search path and the current directory is the same as above. Add to these assumptions that you have a file, SRCHPATH.TXT, which contains the search path C:\UTILS;C:\123;C:\DOS;C:\DBASE. Additionally, you have an *environment variable* in the DOS environment called DIFFPATH. (An environment variable is an element in memory for storing information, for example, the search path.) This variable contains the path C:\DOS. Notice the examples:

- To enable Patha to create a long path of up to 1,024 characters, type the following:

  PATHA /L

  C:\DOS;C:\UTILS;C:\

- To completely replace the search path with the search path contained in the file SRCHPATH.TXT, type the following:

  PATHA SRCHPATH.TXT /F /N

  C:\UTILS;C:\123;C:\DOS;C:\DBASE

- To completely replace the search path with the search path contained in the environment variable DIFFPATH, type the following:

  PATHA %DIFFPATH% /N

  C:\DOS

  (Just to remind you, you can specify that you want to use the contents of an environment variable by placing a percent sign on each side of the variable name. This is what makes example 3 work.)

Of course, in the second example listed above, you can simply add the search path from the text file to the current search path. You do not have to completely replace the search path. Likewise, in the third example, you do not have to completely replace the current search path with the search path in DIFFPATH. You can add the directory from DIFFPATH to the current search path.

WHERE TO USE PATHA

Where do you use Patha? You can use Patha from the DOS command line. You issue the Patha command just like you do any other DOS command. You, no doubt, will most often use Patha /E from the command line, enabling you to edit the current search path.

Besides using Patha from the command line, you can issue the Patha command from a batch file. Using Patha from a batch file can be very effective. For example, if you use batch files to start programs from your hard disk, you can add the Patha command to the batch file to change the search path before the program starts.

If you ordinarily want to use a long search path, then use PATHA /L from your AUTOEXEC.BAT. Doing so enables you to set the long path each time you start your computer.

SUMMARY

Setting search paths is a necessity when using DOS on a hard disk. Without a search path, you must remember the location of all your commands, and specify the location of the command before using it.

Patha stretches the PATH command. This utility gives you benefits to keep you from pulling out your hair. No longer do you have to type a complete new search path; you can edit the search path already in memory. No longer do you have to put up with the limited space for the search path; you can make your search path as long as an old computer's entire memory.

MEMORY: DON'T LOSE IT, USE IT

6

Many personal computer old-timers will never forget a phrase echoing though the computer industry at the introduction of the IBM Personal Computer: "You'll never use 256K!" Put that in perspective. Most computers of that time used between 48K and 64K of RAM. (Just remarkable.)

Time passed, and software became memory-hungry. The standard system came with 640K of memory. Then, oh-my, expanded memory is on the scene. Finally, the 640K *memory-barrier* is broken.

Today, extended memory is the *slop that happifies the pig*. How much total memory is enough? Brother, there ain't enough! What do you do with the memory you have, then? Make the most out of it, that's what.

This chapter provides you with three shareware utilities to make the most out of the memory that your computer *does* have. DOSMax helps you to manage the memory in

MICROSOFT FORGOT:

- Effective memory-management tools.
- Faster memory loading.
- A way to diagnose memory problems.

your computer. Fast Load shows you how you can speed up your programs by loading them directly in memory. Finally, there is the System Capture Utility Package, a program that helps you diagnose memory problems (and some disk problems, for good measure).

MANAGING MEMORY

You know, when it comes to memory management, DOS is doing a pretty good job. You can make pretty good use of memory using DOS 5 and even better with DOS 6. Of course, to make the best use of memory with DOS 5, you have to *love* to tinker. To find just the right configuration, you have to change your CONFIG.SYS and AUTOEXEC.BAT files till the cows come home.

> *To register DOSMAX 2.1, STOPMAX 2.1, FREELOAD 2.1, SHELLMAX 2.1, and ENVIMAX 2.1:*
>
> Philip B. Gardner
> 10461 Lever St.
> Circle Pines, MN 55014
>
> Complete registration and price information is available in the documents included with this utility. If you're going to keep it and use it, please register it!

DOS 6 is a bit better. MemMaker, a DOS 6 utility for optimizing your computer's memory, takes a lot of the guesswork out of configuring your memory. MemMaker does a dandy job of configuring memory.

One of the ways DOS 5 and DOS 6 manage memory is by loading much of the operating system into the High Memory Area (HMA). That's fine. However, some programs make better use of the HMA—for example, Microsoft Windows and Quarterdeck DESQView. The problem is, if you enable these programs to use the HMA, DOS loads in conventional memory, taking up space. You really gain nothing.

Wouldn't it be nice, however, to load DOS in upper memory without using the HMA? Forget it with plain DOS. But, with DOSMAX, a DOS memory manager, you can load DOS in upper memory blocks (UMB), freeing the HMA, yet not wasting precious conventional memory that you use to run programs. (You will find DOSMAX in the \DSMF\DOSMAX directory after you install the diskette contained in this book.) "How does DOSMAX work?" you ask. Read on.

UNDERSTANDING DOSMAX

When you use DOS on a 386 or 486 computer, DOS manages memory using two different programs—HIMEM.SYS and EMM386.EXE. HIMEM.SYS enables DOS to manage extended memory and to load portions of DOS into the HMA.

EMM386.EXE works in conjunction with HIMEM.SYS. EMM386.EXE provides expanded memory and can create UMB—normally unused memory located above conventional memory (640K) and the HMA (extended memory). Using EMM386.EXE, you can instruct DOS to load device drivers and memory-resident programs into the UMBs, freeing up conventional memory for program use. Using UMBs, you can gain, on the average, 70K conventional memory space—quite a lot, considering that the maximum conventional memory is 640K.

The purpose of DOSMAX is really twofold. First, with DOSMAX, you can load DOS into UMBs rather than the HMA. Doing so frees up conventional memory, and frees up the HMA for programs that make better use of it. Second, DOSMAX can load portions of DOS in UMBs that DOS, by itself, can't. You can benefit from more of DOS being loaded high even if you let DOS use the HMA.

What part of DOS does DOSMAX load into upper memory? Besides loading part of the DOS kernel, DOSMAX loads data space used by the FILES, FCBS, LASTDRIVE, STACKS, and INSTALL commands from CONFIG.SYS. This space can amount to around 10K, depending on how you configure DOS on your computer.

MANAGING MEMORY WITH DOSMAX

DOSMAX is made up of five different programs. The five programs, and a definition of each, are listed in the following table:

DOSMAX Program	*Description*
DOSMAX.EXE	The device driver which loads DOS into upper memory blocks.
STOPMAX.SYS	The device driver which prevents DOS from loading into the HMA.
SHELLMAX.COM	The device driver which loads the shell (COMMAND.COM) into upper memory.
ENVIMAX.COM	The device driver which loads the DOS environment into upper memory.
FREELOAD.SYS	A device driver that maximizes memory used by the Quarterdeck Office Systems' LOADHI.SYS device driver.

The following sections list the syntax for each device driver. The section that follows, "Sample DOSMAX Configurations," shows you how you can practically use DOSMAX to manage your memory.

USING DOSMAX.EXE

DOSMAX.EXE is the main device driver that you use to allocate UMBs. DOSMAX enables you to load DOS into upper memory blocks. You load DOSMAX.EXE from the CONFIG.SYS file using a DEVICE statement, placed after loading the XMS manager and the UMB manager (DOS' HIMEM.SYS and EMM386.EXE, respectively). The syntax for DOSMAX.EXE is as follows:

DEVICE=*d:\path***DOSMAX.EXE** */Rt /Pt /Nt /Mm /Fss /Ht /Lt /Ct /Abbbbb /Qt /St /Uhh /Xt @response.fil*

In the above syntax *d:\path*\ indicates the location where DOSMAX.EXE resides. For example, DEVICE=C:\BOOT\DOSMAX.EXE.

The *t* in many of the above options represents a *toggle*, that is, you turn the option on or off. You replace the *t* with a + or - to enable the option or to disable the option, respectively. Generally, all the options either have default values, or DOSMAX.EXE determines what the value should be when it loads into memory. The following table lists the options, along with a description of each option:

DOSMAX.EXE Option	*Description*
/Rt	Reports DOSMAX.EXE status messages on the screen. The reports display helpful information while you are testing DOSMAX installation. /R+ enables the reports and /R- disables the reports. The default is /R-.
/Pt	Booting pauses, enabling you to read messages displayed by DOSMAX. /P+ enables pausing and /P- disables pausing. The default is /P-.
/Nt	Disables pausing, even if DOSMAX displays errors. Use /N+ to disable pausing, or /N- to enable pausing. The default is /N-.
/Mm	Controls how DOSMAX allocates UMBs, where *m* is 0, 1, or 2. Use /M0 to assign the first available UMB. Use /M1 to assign the best fitting UMB. /M2 assigns the last UMB that is large enough. The default is /M0.
/Fss	Indicates the DOS system file size, where *ss* is a number from 40 to 61,

continues

DOSMAX.EXE Option	Description
	indicating the number of 1K bytes of memory. If you use DOSMAX from the command line, *ss* can be a number from 40 to 82. By default, DOSMAX determines the correct DOS system file size.
/Ct	Enables splitting COMMAND.COM, and loading part of COMMAND.COM into upper memory. Use /C+ to enable splitting COMMAND.COM and /C- to disable splitting COMMAND.COM. The default is /C-.
/Abbbbb	Indicates the amount of UMBs to use for loading part of COMMAND.COM into UMBs, where *bbbbb* is the number of bytes. To use this option, you must use /C+. The default is /A2080.
/Qt	Disables DOSMAX from relocating to EMS memory while initializing memory. Use /Q- to disable using EMS memory or /Q+ to enable using EMS memory. The default is /Q-.
/St	Controls whether DOSMAX moves DOS' data segments to UMBs. Use /S+ to enable moving data segments and /S- to disable moving the data segments. The default is /S-.
/Uhh	Controls which hardware interrupts are routed through the DOS STACKS, where *hh* can be 02, 08, 09, 0A, 0B, 0C, 0D, 0E, 70, 72, 73, 74, 76, or 77. Use this switch when STACKS

DOSMAX.EXE Option	*Description*
	are loaded into UMBs. By default, /U is not used. If you are using EMM386.EXE, you should include /U09 to prevent system lockup when you warm-boot the computer.
/Xt	Controls whether DOSMAX.EXE should be loaded into UMBs. Use /X+ to enable loading DOSMAX.EXE into UMBs, or /X- to disable loading DOSMAX.EXE into UMBs. The default is /X+.
@response.fil	Enables you to indicate a text file containing the parameters for DOSMAX.EXE. For example, if you create a file called DOSMAX.CLO, containing the options that you want to use, indicate to use DOSMAX.CLO by placing the option **@DOSMAX.CLO** on the DOSMAX.EXE CONFIG.SYS line.

Place DOSMAX.EXE in the CONFIG.SYS file after the lines starting the XMS and UMB memory managers. For example, include the DEVICE statement to start DOSMAX.EXE after the two DEVICE statements starting HIMEM.SYS and EMM386.EXE.

USING STOPMAX.SYS

The whole purpose of STOPMAX.SYS is to stop DOS from loading into the high memory area, enabling DOSMAX to load DOS into UMBs. You must be using DOS 5.0 or later to use STOPMAX.SYS.

STOPMAX.SYS includes three different methods to stop DOS from loading into high memory area. Two of the methods require no additional memory.

To use STOPMAX.SYS, you must load it as a device driver from CONFIG.SYS. The following syntax shows how to install STOPMAX.SYS:

DEVICE=*d:\path***STOPMAX.SYS** ***/Rt /Pt /Nt /Bt /It /Vt***

In the above syntax, *d:\path*\ indicates the location where STOPMAX.SYS resides. For example, DEVICE=C:\BOOT\STOPMAX.SYS.

The *t* in the above options represents a *toggle*, that is, you turn the option on or off. You replace the *t* with a + or - to enable the option or to disable the option, respectively. Generally, all the options have default values, if you do not indicate the option in the syntax line. The following table lists the options, along with a description of each option:

STOPMAX.SYS Option	*Description*
/Rt	Reports STOPMAX.SYS status messages on the screen. The reports display helpful information while you are testing STOPMAX.SYS installation. /R+ enables the reports and /R- disables the reports. The default is /R-.
/Pt	Booting pauses, enabling you to read messages displayed by STOPMAX.SYS. /P+ enables pausing and /P- disables pausing. The default is /P-.
/Nt	Disables pausing, even if STOPMAX.SYS displays errors. Use /N+ to disable pausing, or /N- to not disable pausing. The default is /N-.
/Bt	Stops DOS from loading into the HMA by patching DOS 5 or DOS 6

STOPMAX.SYS Option	*Description*
	code used to load DOS into the HMA. Use /B+ to enable overwriting code and /B- to disable overwriting the code. The default is /B-.
/It	Stops DOS from loading into the HMA on systems which load HIMEM.SYS into memory after the UMBs are already created. Use /I+ to enable this option and /I- to disable this option. The default is /I-.
/Vt	Use this option to stop DOS from loading into the HMA if STOPMAX.SYS, on your system, reports that the DOS version is incompatible with STOPMAX.SYS. Use /V+ to enable this option and /V- to disable this option. The default is /V-.

You should place the DEVICE statement that starts STOPMAX.SYS in your CONFIG.SYS file before statements that start any other memory managers.

USING SHELLMAX.COM

With SHELLMAX.COM, you can load COMMAND.COM (or another command processor) into UMBs, freeing conventional memory for your programs to use. Because some programs may not work correctly with the command processor loaded outside of conventional memory, SHELLMAX.COM includes an option to place a command processor marker in conventional memory. This marker fools programs into thinking the command processor is in conventional memory which ensures compatibility.

You load SHELLMAX.COM from the CONFIG.SYS file in the line with the command processor. The following syntax shows how to install SHELLMAX.COM:

SHELL=*d:\path***SHELLMAX.COM** ***/Rt /Pt /Nt /St***
*@response.fil d:\path***COMMAND.COM** *comspec options*

In the above syntax, *d:\path*\ indicates the location where SHELLMAX.COM and COMMAND.COM reside. For example, C:\BOOT\SHELLMAX.COM.

The *t* in the above options represents a *toggle*, that is, you turn the option on or off. You replace the *t* with a + or - to enable the option or to disable the option, respectively. Generally, all the options have default values, if you do not indicate the option in the syntax line. The following table lists the options, along with a description of each option:

SHELLMAX.COM Option	*Description*
/Rt	Reports SHELLMAX.COM status messages on the screen. The reports display helpful information while you are testing SHELLMAX.COM installation. /R+ enables the reports and /R- disables the reports. The default is /R-.
/Pt	Booting pauses, enabling you to read messages displayed by SHELLMAX.COM. /P+ enables pausing and /P- disables pausing. The default is /P-.
/Nt	Disables pausing, even if SHELLMAX.COM displays errors. Use /N+ to disable pausing, or /N- to not disable pausing. The default is /N-.

SHELLMAX.COM Option	*Description*
/St	Enables placing a command processor stub in conventional memory for compatibility. Use /S+ to place the stub in conventional memory, and /S- to not place the stub in conventional memory. The default is /S+.
@response.fil	Enables you to indicate a text file containing the parameters for SHELLMAX.COM. For example, if you create a file called SHELLMAX.CLO, containing the options that you want to use, indicate to use SHELLMAX.CLO by placing the option **@SHELLMAX.CLO** on the SHELL CONFIG.SYS line.

To load COMMAND.COM from C:\ using SHELLMAX, place a syntax line similar to the following in your CONFIG.SYS file:

SHELL=C:\SHELLMAX.COM C:\COMMAND.COM C:\ /E:512 /P

USING ENVIMAX.COM

DOS creates an environment which maintains your search path, environment variables, and other information necessary to DOS. This environment takes space in conventional memory. However, you can use ENVIMAX.COM to move DOS' environment to UMBs.

Unlike the other utilities that come with DOSMAX, you start ENVIMAX.COM from the AUTOEXEC.BAT file. The following syntax shows how to install ENVIMAX.COM:

*d:\path***ENVIMAX.COM** ***/Rt /Pt /Nt /Et*** *@response.fil*

In the above syntax, *d:\path* indicates the location where ENVIMAX.COM resides. For example, C:\BOOT\ENVIMAX.COM.

The *t* in the above options represents a *toggle*, that is, you turn the option on or off. You replace the *t* with a + or - to enable the option or to disable the option, respectively. Generally, all the options have default values, if you do not indicate the option in the syntax line. The following table lists the options, along with a description of each one:

ENVIMAX.COM Option	*Description*
/Rt	Reports ENVIMAX.COM status messages on the screen. The reports display helpful information while you are testing ENVIMAX.COM installation. /R+ enables the reports and /R- disables the reports. The default is /R-.
/Pt	Booting pauses, enabling you to read messages displayed by ENVIMAX.COM. /P+ enables pausing and /P- disables pausing. The default is /P-.
/Nt	Disables pausing, even if ENVIMAX.COM displays errors. Use /N+ to disable pausing, or /N- to not disable pausing. The default is /N-.
/Et	Enables you to place the environment into upper memory blocks. /E+ enables placing the environment into UMBs, while /E- disables placing the environment into UMBs. The default is /E+.

ENVIMAX.COM Option	*Description*
@response.fil	Enables you to indicate a text file containing the parameters for ENVIMAX.COM. For example, if you create a file called ENVIMAX.CLO, containing the options that you want to use, indicate to use ENVIMAX.CLO by placing the option **@ENVIMAX.CLO** on the ENVIMAX.COM line in the AUTOEXEC.BAT file.

USING FREELOAD.SYS

FREELOAD.SYS is a device driver that loads into your memory thoughts of recent visits by your relatives. Just kidding.

Actually, FREELOAD.SYS is a device driver that releases memory used by LOADHI.SYS or 386LOAD.SYS included in QEMM386 and 386MAX, respectively. The drivers LOADHI.SYS and 386LOAD.SYS place TSRs and device drivers in upper memory when you are using these third party memory managers. However, these two drivers use conventional memory.

You load FREELOAD.SYS as a device driver from the CONFIG.SYS file. The following syntax shows how to use FREELOAD.SYS:

DEVICE=*d:**path***FREELOAD.SYS** */Rt /Pt /Nt /Mn /Dnn /An /St @response.fil* **/Fd:\path\loader.ext**

In the above syntax, *d:**path*\ indicates the location where FREELOAD.SYS and loader.ext resides. For example, C:\BOOT\FREELOAD.SYS.

The *t* in some of the above options represents a *toggle*, that is, you turn the option on or off. You replace the *t* with a + or - to enable the option or to disable the option, respectively. Generally, all the options have default values, if you do not indicate

the option in the syntax line. The following table lists the options, along with a description of each option:

FREELOAD.SYS Option	*Description*
/Rt	Reports FREELOAD.SYS status messages on the screen. The reports display helpful information while you are testing FREELOAD.SYS installation. /R+ enables the reports and /R- disables the reports. The default is /R-.
/Pt	Booting pauses, enabling you to read messages displayed by FREELOAD.SYS. /P+ enables pausing and /P- disables pausing. The default is /P-.
/Nt	Disables pausing, even if FREELOAD.SYS displays errors. Use /N+ to disable pausing, or /N- to not disable pausing. The default is /N-.
/Mm	Controls how FREELOAD.SYS allocates UMBs, where *m* is 0, 1, or 2. Use /M0 to assign the first available UMB. Use /M1 to assign the best fitting UMB. /M2 assigns the last UMB that is large enough. The default is /M0.
/Ddd	Enables you to indicate the drive table size. The values for *dd* are 81 for DOS 3.1, and 88 for DOS 4.0 through DOS 6.0. However, if you are using DOS 5.0 or DOS 6.0, you can indicate a larger number. FREELOAD.SYS will determine the correct drive table size.

FREELOAD.SYS Option	*Description*
/Aa	Enables you to indicate the high device loader. Replace *a* with 0 to enable FREELOAD.SYS to determine the high device loader, 1 to indicate that LOADHI.SYS is the high device loader, and 2 to indicate that 386LOAD.SYS is the high device loader. The default is /A0.
/St	Enables you to shrink the size of FREELOAD.SYS in memory. Use /S+ to shrink FREELOAD.SYS and /S- to disable shrinking. The default is /S-.
@response.fil	Enables you to indicate a text file containing the parameters for FREELOAD.SYS. For example, if you create a file called FREELOAD.CLO containing the options that you want to use, indicate to use FREELOAD.CLO by placing the option **@FREELOAD.CLO** on a DEVICE= CONFIG.SYS line.
/Fd:\path\loader.ext	Specifies the location and name of the third party driver enabling TSRs and device drivers to load high. For example, /FC:\QEMM\LOADHI.SYS.

SAMPLE DOSMAX CONFIGURATIONS

Enough of that boring syntax stuff. On with the good stuff—how to gain back that otherwise lost memory.

The following configurations are based on a 386 computer running MS-DOS 6.0, using DoubleSpace (the disk-compression utility), and SMARTDrive 4.0 for disk caching. As a point of reference, the following lists the CONFIG.SYS and AUTOEXEC.BAT files before modifying them.

CONFIG.SYS

```
DOS=HIGH,UMB
DEVICE=C:\DOS\HIMEM.SYS /machine:13
DEVICE=C:\DOS\EMM386.EXE NOEMS HIGHSCAN
BUFFERS=15,0
FILES=60
LASTDRIVE=Z
FCBS=16,0
DEVICEHIGH=C:\DOS\SETVER.EXE
BREAK=ON
STACKS=9,256
SHELL=C:\DOS\COMMAND.COM C:\DOS\ /P
DEVICEHIGH=C:\DOS\DBLSPACE.SYS /MOVE
DEVICEHIGH=C:\UTILS\MOUSE.SYS /Y
DEVICEHIGH=C:\DOS\ANSI.SYS
```

AUTOEXEC.BAT

```
@ECHO OFF
LH C:\DOS\SMARTDRV 2048 2048
LH C:\UTILS\CENTER
C:\UTILS\VMODE 35k
VERIFY OFF
PROMPT $D$_[$p]
```

```
PATH C:\WINDOWS;C:\EXCEL;C:\WINWORD;C:\DOS;C:\UTILS
SET TEMP=C:\WINDOWS\TEMP
LH C:\DOS\SHARE.EXE
```

Using the original configuration, the computer has 613K conventional memory remaining and 59K upper memory blocks available—a total of 672K DOS memory. Suppose you want to load as much of DOS into upper memory blocks, including loading DOS into the high memory area. To do so, modify the above files to include DOSMAX.EXE, SHELLMAX.COM, and ENVIMAX.COM. The modified files are listed below with the added lines appearing bold. The DOSMAX utilities are stored in the C:\BOOT directory:

CONFIG.SYS

```
DOS=HIGH,UMB
DEVICE=C:\DOS\HIMEM.SYS /machine:13
DEVICE=C:\DOS\EMM386.EXE NOEMS HIGHSCAN
DEVICE=C:\BOOT\DOSMAX.EXE /M1 /U09 /Q-
BUFFERS=15,0
FILES=60
LASTDRIVE=Z
FCBS=16,0
DEVICEHIGH=C:\DOS\SETVER.EXE
BREAK=ON
STACKS=9,256
SHELL=C:\BOOT\SHELLMAX.COM C:\DOS\COMMAND.COM C:\DOS\ /P
DEVICEHIGH=C:\DOS\DBLSPACE.SYS /MOVE
DEVICEHIGH=C:\UTILS\MOUSE.SYS /Y
DEVICEHIGH=C:\DOS\ANSI.SYS
```

AUTOEXEC.BAT

```
@ECHO OFF
C:\BOOT\ENVIMAX
LH C:\DOS\SMARTDRV 2048 2048
LH C:\UTILS\CENTER
C:\UTILS\VMODE 35k
VERIFY OFF
PROMPT $D$_[$p]
PATH C:\WINDOWS;C:\EXCEL;C:\WINWORD;C:\DOS;C:\UTILS
SET TEMP=C:\WINDOWS\TEMP
LH C:\DOS\SHARE.EXE
```

After changing the CONFIG.SYS and AUTOEXEC.BAT files, reboot the computer. Now, the computer has 626K conventional memory remaining and 46K upper memory blocks remaining—a total of 672K DOS memory. Now, you might exclaim, "Why, that's the same DOS memory as before!" This is true, but take a look at the conventional memory. You see an increase of 13K. The largest contiguous area of DOS memory into which you can load a program is 13K larger than before.

You can change the configuration so that you load DOS into upper memory blocks, leaving the high memory area free for other programs to use. Use the following files for this configuration:

CONFIG.SYS

```
DOS=HIGH,UMB
DEVICE=C:\BOOT\STOPMAX.SYS /B+
DEVICE=C:\DOS\HIMEM.SYS /machine:13
DEVICE=C:\DOS\EMM386.EXE NOEMS HIGHSCAN
DEVICE=C:\BOOT\DOSMAX.EXE /M1 /U09 /Q-
```

```
BUFFERS=15,0
FILES=60
LASTDRIVE=Z
FCBS=16,0
DEVICEHIGH=C:\DOS\SETVER.EXE
BREAK=ON
STACKS=9,256
SHELL=C:\BOOT\SHELLMAX.COM
C:\DOS\COMMAND.COM C:\DOS\ /P
DEVICEHIGH=C:\DOS\DBLSPACE.SYS /MOVE
DEVICEHIGH=C:\UTILS\MOUSE.SYS /Y
DEVICEHIGH=C:\DOS\ANSI.SYS
```

AUTOEXEC.BAT

```
@ECHO OFF
C:\BOOT\ENVIMAX
LH C:\DOS\SMARTDRV 2048 2048
LH C:\UTILS\CENTER
C:\UTILS\VMODE 35k
VERIFY OFF
PROMPT $D$_[$p]
PATH C:\WINDOWS;C:\EXCEL;C:\WINWORD;C:\DOS;C:\UTILS
SET TEMP=C:\WINDOWS\TEMP
LH C:\DOS\SHARE.EXE
```

Using this configuration, the computer has remaining 577K conventional memory and 16K upper memory blocks. You do have less DOS memory (593K compared to 672K), but remember

that you have available the high memory area—64K additional memory. Some programs make better use of the HMA than DOS does.

NOTE

> The information in this section deals only with DOS 6. If you are using an earlier version of DOS, you may realize an even greater benefit of extra memory, since earlier versions of DOS used less memory.

True, these configurations work with one computer. You might have to work with your computer to get the best results. Be sure to enable each utility to report information when it loads into memory. This is important, as the reports help you to refine your memory configuration.

LOADING PROGRAMS IN MEMORY FOR SPEED

How fast is fast? Maybe you've got a fast hard disk. When you type a command, your hard disk simply kicks, loading that program into memory immediately. Within a few seconds, boom—the program is running. Zowee, that's fast!

Do you want to make your program even faster? You can install a disk cache, a program that maintains often-used information in memory, freeing the computer from constantly reading the disk. For example, you can use SmartDrive, which comes with some versions of DOS. Yes, the disk cache speeds up your programs. It keeps information in the cache that you previously used. And if the program is still in the disk cache when you execute the command again, DOS loads the program from the cache (without accessing your hard drive again). Wow, that's really fast! But what if the program is not still in the disk cache? Back to the hard disk for you, mister.

Do you want to guarantee that often-used programs load just as quickly as possible? Load the program into RAM so that every time you start the program, it starts from RAM rather than from

your hard disk. "But that requires a RAM disk (a program that makes memory act like a disk drive), right?" you ask. No, no, no—it doesn't. What it requires is Fast Load. (Look for Fast Load after you install this book's disk in the \DSMF\FASTLOAD directory.)

Fast Load is a memory-resident program which loads often-used programs or files into extended or expanded memory. For example, Fast Load can load your word processing program and the spelling dictionary into memory. When you use the program or file (in this case, the word processing program or the spelling dictionary), DOS reads the file from memory rather than reading the file from the disk. Using the file directly from memory is a whole lot faster.

> *To register Fast Load 1.25:*
>
> Microsystems Software, Inc.
> 600 Worcester Road
> Framingham, MA 01701
>
> Complete registration and price information is available in the documents included with this utility. If you're going to keep it and use it, please register it!

USING FAST LOAD

In order to use Fast Load, you must have available extended or expanded memory. That is because Fast Load loads programs or files that you indicate into available extended or expanded memory. When you use the program, DOS loads the program from memory, rather than spending the time reading the disk.

You use Fast Load by starting the program from the command line. You can also add a line to your AUTOEXEC.BAT file to start Fast Load each time you boot your computer. You will find Fast Load in the \DOSSTUFF\FL directory.

When you start Fast Load, you must indicate the files that you want to load into memory. You can indicate the files directly from the command line, or you can create a file called FL.DAT

containing files to load into memory. The syntax to use for Fast Load is as follows:

FL */P progs /F datfile /B /C /D /E /H /I /K /NS /NV /NW /Q /R /T /V /W /X /Y mm-dd-yyyy,cnt /U*

Of the nineteen options shown in the syntax line, you can only use six when you start Fast Load. The following table lists the Fast Load options, giving a definition of the options:

Fast Load Option	*Description*
/P progs	Enables you to indicate the programs to load into memory. *progs* is all the programs to load, including the path to the program. Separate programs using commas; for example **C:\WORD55\WORD.EXE,C:\UTILS\PKZIP.EXE**.
/F datfile	Enables you to indicate the datafile (*datfile*) that contains a list of programs that Fast Load loads into memory. *datfile* is a text file, listing each program on a separate line in the file. Use this switch when running, or after loading, Fast Load.
/B	Disables monitoring for batch files loaded by Fast Load.
/C	Displays the extended or expanded memory used by each file Fast Load loads into memory. Also displays the total expanded and extended memory used, and how much is free for use. You can use this switch without Fast Load in memory, and after running Fast Load.
/D	Enables the program in memory to only be run, and not opened for data access.

Fast Load Option	*Description*
/E	Indicates that Fast Load should not load files into expanded memory, but only extended memory.
/H	Stops Fast Load from checking High memory when loading.
/I	Displays Fast Load information. Similar to using /C, but does not show the path or size of each file loaded into memory. You use this switch after running Fast Load.
/K	Enables you to run a program in a different directory from the disk when it has the same name as a program loaded into memory.
/NS	Disables the Scroll Lock key from controlling Fast Load.
/NV	Disables the video speedup module.
/NW	Indicates that Fast Load should not wait for a key to be pressed when an error occurs while using a program loaded from memory.
/Q	Keeps Fast Load from displaying registration information when you start Fast Load. (This does not work in the shareware version.)
/R	Closes all files opened by Fast Load. Use this option after running Fast Load.
/T	Enables files loaded with Fast Load to be truncated or deleted.
/V	Loads the video speedup module.

continues

Fast Load Option	*Description*
/W	Allows files loaded by Fast Load to be changed, as long as the size of the file does not change.
/X	Indicates that Fast Load should not load files into extended memory, but only expanded memory.
/Y mm-dd-yyyy,cnt	Fast Load responds with the date *mm-dd-yyyy* when DOS uses the Get_Date call. *cnt* is a number from 0 to 255 indicating the number of times that Fast Load responds to the Get_Date call with the date *mm-dd-yyyy*. Using the value 0 indicates an infinite number of times.
/U	Removes Fast Load from memory. Use this option after running Fast Load.

To start Fast Load, type **FL** and press Enter. Fast Load loads into memory, and looks for the file FL.DAT. FL.DAT is a file that you create, containing the names of programs that Fast Load should load into memory. In fact, if FL.DAT does not exist, then Fast Load will not load into memory, unless you specify the program or programs to load using the /P parameter.

What does the file FL.DAT look like? This file contains a list of programs that you want Fast Load to load into extended or expanded memory. You enter each file on a separate line, including the path where the file can be found. For example, the following lists the contents of one example FL.DAT file:

```
C:\WORD55\WORD.EXE
C:\UTILS\PKZIP.EXE
C:\UTILS\PKUNZIP.EXE
C:\COMMAND.COM
```

You can include some of the options listed in the table above with each command in FL.DAT. When you do, the option affects only that file. The options that you can use are: /K, /Y, /B, /T,

and /W. Include the options on the same line as the file to affect, directly after the file name.

LOADING PROGRAMS FROM THE COMMAND LINE

You can use the /P parameter to load additional programs when you first run Fast Load, or after Fast Load is in memory. To load C:\WORD55\WORD.EXE in memory using the /P parameter, enter the following command:

```
FL /P C:\WORD55\WORD.EXE
```

To load more than one program in memory with the /P parameter, separate the programs with a comma. For example, to load C:\WORD55\WORD.EXE and C:\UTILS\PKZIP.EXE in memory, enter the following command:

```
FL /P C:\WORD55\WORD.EXE,C:\UTILS\PKZIP.EXE
```

INDICATING A DIFFERENT PROGRAM DATA FILE

While Fast Load looks for FL.DAT by default, you can substitute a different file name. Use the /F switch to substitute a file other than FL.DAT. For example, to use the file located in C:\UTILS called LOADPROG.TXT, enter the following command:

```
FL /F C:\UTILS\LOADPROG.TXT
```

If you do not want the Scroll Lock key to control Fast Load, use the /NS parameter when starting Fast Load.

> The Shareware version of Fast Load enables you to load as many as seven different programs into memory. When you register Fast Load, you can load up to 250 files in memory (depending, of course, on available memory).

CONTROLLING FAST LOAD ACTIVITY

Fast Load is active while it is in memory. When you type a command that Fast Load loads in memory, the file executes from memory. You can disable Fast Load, however, so that you can run a program from the disk. When you turn on the Scroll Lock key, you disable Fast Load. Turning off Scroll Lock enables Fast Load again.

Why disable Fast Load? Suppose you want to try a new version of a program. The program name is likely the same. When you type the new program name, Fast Load intercepts and loads the old program loaded into memory. However, if you disable Fast Load, you can run the new version of the program from disk.

You can use an auxiliary program that accompanies Fast Load, called SETFL.COM, to disable and enable Fast Load. Type **SETFL -** to disable Fast Load, and **SETFL +** to enable Fast Load. The SETFL.COM program enables you to control Fast Load from a batch file.

MONITORING PROGRAMS YOU USE

You can probably think of several programs that you use during the day that you would like to load into memory. But what are the names of those programs? Without knowing the names of the programs, you can't load them into memory.

Maybe you know the names of the programs you use. But do you know whether those programs, in turn, start other programs? You probably don't.

Load Watch, a utility that comes with Fast Load, keeps track of programs that you load into memory. Load Watch, like Fast Load, is a memory-resident program. When in memory, Load Watch tracks programs that execute in memory, and records those programs in a file called LWATCH.DAT.

You can find Load Watch in the \DOSSTUFF\FL directory. The syntax for Load Watch is as follows:

LWATCH */D /H /U*

The following table lists the options you can use with Load Watch, and each option's description:

Option	*Description*
/D	Load Watch records descriptive information, including the contents of some CPU registers and the program that started the listed program.

Option	*Description*
/H	Stops Load Watch from checking High memory when loading.
/U	Unloads Load Watch from memory.

When started, Load Watch records executable programs, including the path to the program, in the file LWATCH.DAT. You can view the LWATCH.DAT file to see which programs to include in the FL.DAT file. If you use the /D option, you not only discover executable programs that you use during the day, but you also learn whether certain programs start other programs.

PROTECTING YOUR SYSTEM

DOS really offers no utilities to protect your system, or let you diagnose problems—short of trying to figure out how to use DEBUG, that is. There is no real way to maintain configuration information for your computer system.

Suppose you want to maintain a copy of your hard disk's boot areas to recover from a virus. Or suppose you want to maintain a copy of your CMOS memory, so you can recover from a damaged setup. How can you perform these somewhat technical tasks with DOS? You can't. But with SYS_CAP, the System Capture Utility Package, you can. (SYS_CAP is placed in the \DSMF\SYSCAP directory when you install the disk contained in this book.)

To register SYS_CAP 1.0:

Leslie E. Geddes
1125 Edinboro Drive
Boulder, CO 80303

Complete registration and price information is available in the documents included with this utility. If you're going to keep it and use it, please register it!

SYS_CAP consists of several utilities that enable you to record information about your computer, examine that information, and change certain information on your computer. The utilities that make up the System Capture Utility Package are described in the following table:

Utility	*Description*
RAMCAM.COM	Captures the first 1M of RAM and stores the information in 16 different 64K image files.
DISKDATA.COM	Captures the master boot records and boot sectors for each hard disk attached to the computer.
GETCMOS.COM	Records a copy of your CMOS setup into a disk file.
CAPTURE.BAT	A batch file which executes RAMCAM.COM, DISKDATA.COM and GETCMOS.COM, recording this vital information for a single computer system.
MEMDUMP.COM	Displays the contents of memory or of a memory image file.
EXAMINE.COM	Enables you to view or change the contents of a binary file.
MEMALOC.EXE	Displays a computer's current memory allocation.
CMOS.EXE	Enables you to view and to change the contents of CMOS memory.

The following sections display information for using the utilities described in the table above.

CAUTION

> Before using any of these utilities, realize that they are not for the technically timid. The SYS_CAP utilities are powerful diagnostic and system maintenance tools. Use them only if you feel completely confident of your technical expertise. (In layman's terms: If you can't pass the propeller-head test, you probably shouldn't use these utilities.)

CAPTURING RAM

From time to time, you may experience system lock-ups. Or you may obtain unusual results when running a program. To aid you in your diagnosis, you can use RAMCAM to capture the contents of the first 1M of memory for examination.

RAMCAM.COM is a utility that captures all of the first 1M of memory. The program stores what it captures in 16 different files, each being 64K in length. To run RAMCAM.COM, use the following syntax:

RAMCAM */B*

You use the */B* parameter only when running RAMCAM in batch mode.

When you start RAMCAM, the following prompt appears on the screen:

```
Enter destination for capture:
```

Type the location in which you want to store the memory files. Type, for example, **A:** to store the memory files on a high-density disk in drive A. Then, place the diskette in drive A and press Enter. Another prompt appears, as follows (Yes, "Procede" is misspelled. However, that's how it appears on the screen.):

```
Procede with capture? (y/N)
```

To continue, press **Y**. To abort RAMCAM, press **N** or the Esc key.

As mentioned above, RAMCAM creates 16 different files which, when taken together, contain the entire contents of the first 1M of system memory. The files are named as follows: MEMSEG00.DAT, MEMSEG01.DAT, MEMSET02.DAT,...,MEMSEG0D.DAT, MEMSEG0E.DAT, MEMSEG0F.DAT. (RAMCAM uses hexidecimal numbers 00-0F, that is, 0-15, to create the file names.) If the files already exist, RAMCAM prompts you to overwrite the files. If you use the /B parameter, RAMCAM does *not* prompt you; it goes ahead and overwrites the files.

Once you capture system memory, you can view the files at your leisure, comparing segments of memory to files captured earlier from that computer. Or you can compare the files to results obtained on another computer using the same configuration.

To view the contents of these files, use the MEMDUMP.COM utility. See the section "Examining Memory and Files" for information about MEMDUMP.COM.

RECORDING VITAL HARD DISK INFORMATION

Your hard disk contains vital information. You are well aware of that. No doubt, your back up your hard disk regularly, so you'll have a copy of your files to fall back on in case something drastic happens. But there are areas of your hard disk that you do not back up. These areas are absolutely essential to your data storage. Unfortunately, these areas are often the target of viruses. These areas are the Master Boot Record and the Partition Boot Sector.

The utility DISKDATA.COM enables you to capture copies of the Master Boot Record and of the Partition Boot Sectors (there can be more than one Partition Boot Sector on a disk) into files. You can examine these files to make sure they are free from damage and keep copies to compare with the actual Master Boot Record and Partition Boot Sectors in case you suspect they have been damaged.

To capture these areas of each hard disk on your system, use the following syntax:

DISKDATA

When you run DISKDATA, the program creates at least two files for each hard disk on your system. The file containing the Master Boot Record is called MASTBOOT.D1 and the file containing the Partition Boot Sector is called BSECTR_1.D1. If there is more than one Partition Boot Sector on a hard disk, DISKDAT sequentially numbers each file—for example, BSECTR_1.D1 and BSECTR_2.D1.

If you have more than one hard disk, DISKDATA sequentially numbers the files using the file extension. The extension D1 is for the first hard disk, D2 is for the second hard disk, and so on.

Use EXAMINE.COM to view the contents of the files created by DISKDATA.COM. See the section "Examining Memory and Files" for information about using EXAMINE.COM.

STORING YOUR CMOS SETUP IN A FILE

Your computer stores setup information in CMOS memory—memory that is backed up with a battery. This information is necessary for your computer to start. Because this information is so important, you should keep a copy of CMOS memory in a disk file. Then, should your CMOS memory be lost, you can restore the contents of CMOS by reviewing the information in this file.

To store CMOS memory contents in a file, use the utility GETCMOS.COM. To use GETCMOS, type the following and press Enter:

GETCMOS

When you use this command, GETCMOS creates a file called CMOSRAM.DAT. Use the utility EXAMINE.COM to view the contents of this file. See the section "Examining Memory and Files" for information on using EXAMINE.COM.

Suppose you need to correct your CMOS. You can do this using CMOS.EXE. To use this utility, type **CMOS** and press Enter. A message displays on the screen telling you to press a key to continue; when you press a key, a window like the one shown in figure 6.1 appears on the screen.

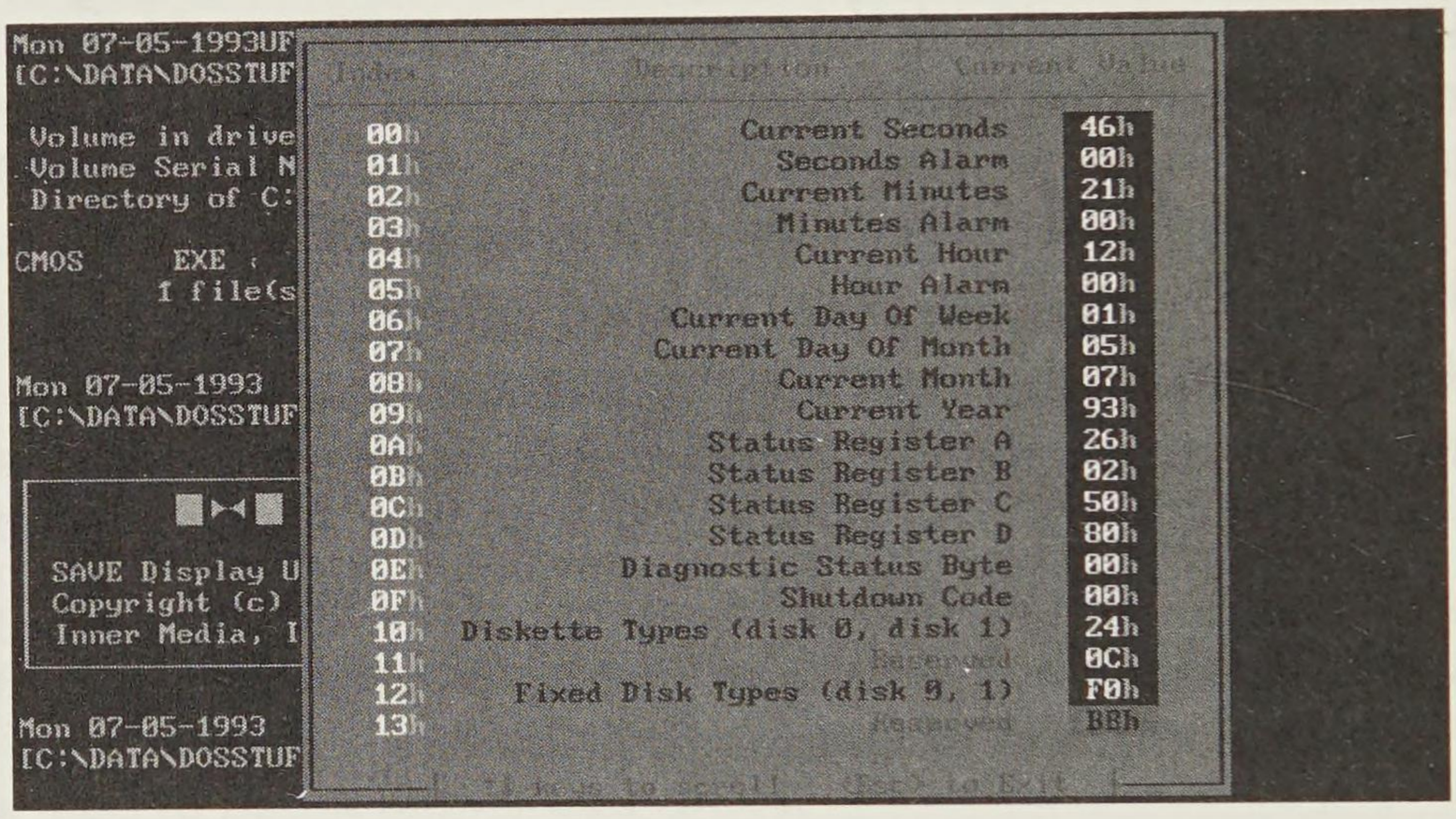


Fig. 6.1
Use CMOS.EXE to correct your computer's CMOS setup.

Use the up- and down-arrow keys to move to the different CMOS entries. Type the correct value for each CMOS entry. When you are finished, press Esc. A message appears on the screen as follows:

```
Please Confirm Changes to CMOS RAM
'Y' to write changes (else 'N')
```

Press **Y** to save your changes, or press **N** to abandon CMOS, not saving changes to your CMOS RAM.

PRESERVING SYSTEM INFORMATION

The System Capture Utilities comes with a batch file that runs the three capture programs—RAMCAM.COM, DISKDATA.COM and GETCMOS.COM. The name of this batch file is CAPTURE.BAT.

To use CAPTURE, you first must format a high-density diskette. Then, copy these four files (RAMCAM.COM, DISKDATA.COM,

GETCMOS.COM and CAPTURE.BAT) to the diskette. Change to the drive containing the diskette to which you just copied the files, then type **CAPTURE** and press Enter.

CAPTURE runs the three programs, capturing system information, and writing that information to the diskette. Label the diskette, and keep the diskette with the computer. For informational purposes, run CAPTURE on each computer you have.

EXAMINING MEMORY AND FILES

Two programs that come with System Capture Utilities enable you to view the contents of memory and the contents of memory-image and binary files. These utilities are: MEMDUMP.COM and EXAMINE.COM.

To use MEMDUMP.COM to display the contents of memory, type **MEMDUMP /M** and press Enter. MEMDUMP loads into memory and asks for a starting address to begin the display. Type a starting segment and offset and press Enter—for example, type **F000:000** and press Enter. MEMDUMP asks for an ending address. Type an ending segment and offset, or simply press Enter to display through the end of the first megabyte of memory. When you do, a screen similar to the one in figure 6.2 displays on the screen containing the memory.

```
F000:0170  2A 88 3A 3D 2A F5 24 1C -- 22 C0 3A F5 24 3D 2A 3D  *.:=*.....:..=*=
F000:0160  2A 00 00 00 00 00 00 3D -- 2A 3D 2A 00 00 08 19 3D  *.......=*=*....=
F000:0150  2A 5E 3A 3D 2A F5 24 1C -- 22 73 3A F5 24 3D 2A 3D  *^:=*....s:..=*=
F000:0140  2A 00 00 00 00 00 00 3D -- 2A 3D 2A 00 00 07 1F 3D  *.......=*=*....=
F000:0130  2A 02 3A 3D 2A F5 24 1C -- 22 2F 3A F5 24 3D 2A 3D  *.:=*..../:..=*=
F000:0120  2A 00 00 00 00 00 00 3D -- 2A 3D 2A 00 00 07 1C 3D  *.......=*=*....=
F000:0110  2A A6 39 3D 2A F5 24 1C -- 22 D3 39 F5 24 3D 2A 3D  *.9=*.....9..=*=
F000:0100  E9 9C 17 53 6C 6F 77 00 -- 46 61 73 74 00 07 19 3D  ...Slow.Fast...=
F000:00F0  20 20 20 20 20 20 20 20 -- 20 20 20 20 20 20 20 20  ................
F000:00E0  20 20 20 20 20 20 20 20 -- 20 20 20 20 20 20 20 20  ................
F000:00D0  20 20 20 20 20 20 20 20 -- 20 20 20 20 20 20 20 20  ................
F000:00C0  20 20 20 20 20 20 20 20 -- 20 20 20 20 20 20 20 20  ................
F000:00B0  20 20 20 20 20 20 20 20 -- 20 20 20 20 20 20 20 20  ................
F000:00A0  20 20 20 20 20 20 20 20 -- 20 20 20 20 20 20 20 20  ................
F000:0090  20 20 20 20 20 20 20 20 -- 20 20 20 20 20 20 20 20  ................
F000:0080  20 20 20 20 20 20 20 20 -- 20 20 20 20 20 20 20 20  ................
F000:0070  20 20 20 20 20 20 20 20 -- 20 20 20 20 20 20 20 20  ................
F000:0060  20 20 20 20 20 20 20 20 -- 20 20 20 20 20 20 20 20  ................
F000:0050  20 20 4C 4C 74 74 64 64 -- 2E 2E 20 20 20 20 20 20  ..LLttdd........
F000:0040  6E 6E 6F 6F 6C 6C 6F 6F -- 67 67 69 69 65 65 73 73  nnoolloogiieess
F000:0030  6E 6E 69 69 78 78 20 20 -- 54 54 65 65 63 63 68 68  nniixx..TTeecchh
F000:0020  39 39 39 39 31 31 20 20 -- 50 50 68 68 6F 6F 65 65  999911..PPhhooee
F000:0010  74 74 20 20 31 31 39 39 -- 38 38 35 35 2D 2D 31 31  tt..11998855--11
F000:0000  43 43 6F 6F 70 70 79 79 -- 72 72 69 69 67 67 68 68  CCooppyyrriigghh
Start>                    Pause: ON   Printer Output: Off   Help  Esc  Exit
```

Fig. 6.2
Displaying memory with MEMDUMP.COM.

You can press **P** to toggle **P**ause on and off; that is, you can scroll through memory. Press **O** to toggle printer **O**utput on and off. Press Esc to exit MEMDUMP.

To view the contents of a file created with RAMCAM, type **MEMDUMP /F** and press Enter. MEMDUMP works identically whether you are viewing the contents of a memory file or of actual memory.

Another viewing utility is EXAMINE.COM. This utility enables you to view the contents of a binary file, including searching for, marking, copying, and changing information. The syntax for EXAMINE.COM is as follows:

EXAMINE *filename*

To view the contents of CMOSRAM.DAT, type **EXAMINE CMOSRAM.DAT** and press Enter. The screen displays similar to the one shown in figure 6.3.

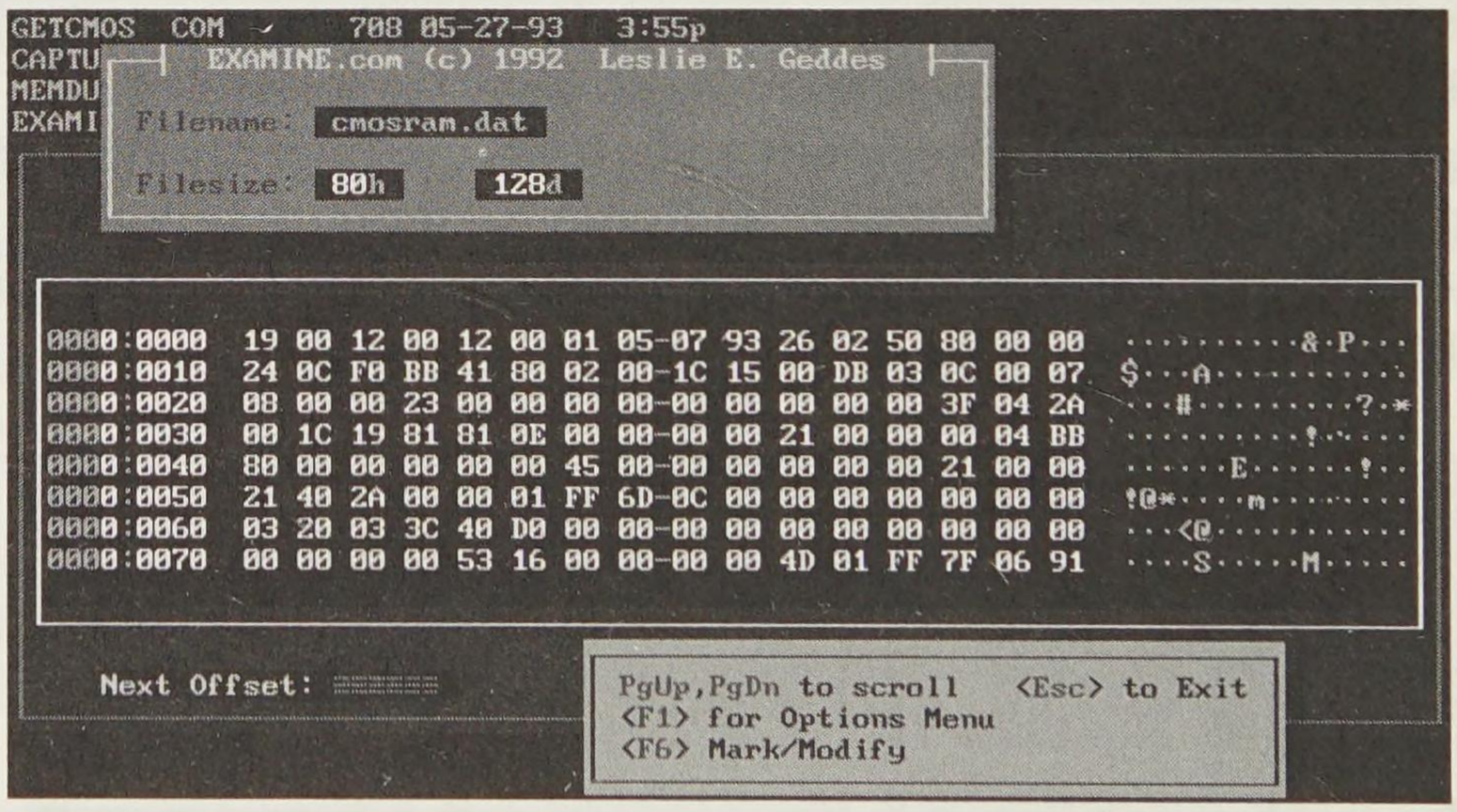


Fig. 6.3
Examining a binary file with EXAMINE.COM.

To scroll through a binary file which you are examining, press the up- and down-arrow keys to move line by line through the file. Use the PgUp and PgDn keys to scroll page by page through the file.

You can change information found in the file. To change the file, press F6 to enter the Mark/Modify menu. Press the arrow keys

to move to the byte in the file to change, then type the new byte (a 2-digit, hexadecimal number). Press F6 again to exit the Mark/Modify menu.

While you are in the Mark/Modify menu, you can mark information to copy. Press **M** to toggle **M**arking on and off. When **M**arking is on, use the arrow keys to mark information from the file. Press F1 to choose the Options menu, then choose **C**opy marked to copy the marked information to a **F**ile, or to an **O**ffset in the current file. (See the screen shown in fig. 6.4.)

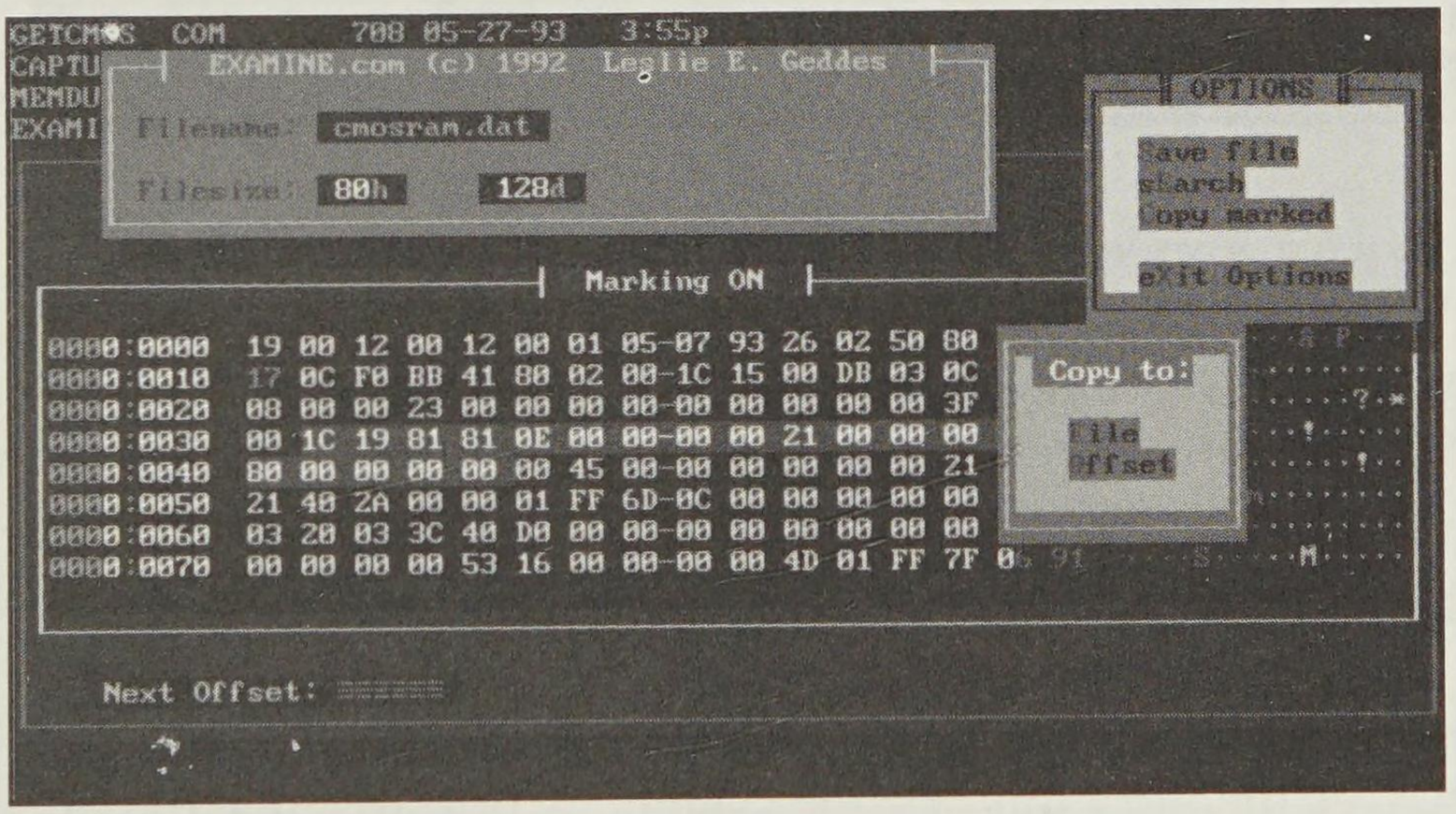


Fig. 6.4
You can copy marked information to a file or to another location in the current file.

Other actions you can choose in the Options menu include s**E**arch and **S**ave file. S**E**arch enables you to find specific information in the file, while **S**ave file enables you to save any changes that you have made back to the file. When you are finished with EXAMINE, press Esc.

DISPLAYING MEMORY ALLOCATION

The final program to discuss is MEMALOC.EXE. This program displays the current memory allocation about the Memory Control Block. To display this information, type **MEMALOC** and press Enter. The screen displays information similar to that shown in figure 6.5.

When you are finished viewing the memory control block display, press Esc.

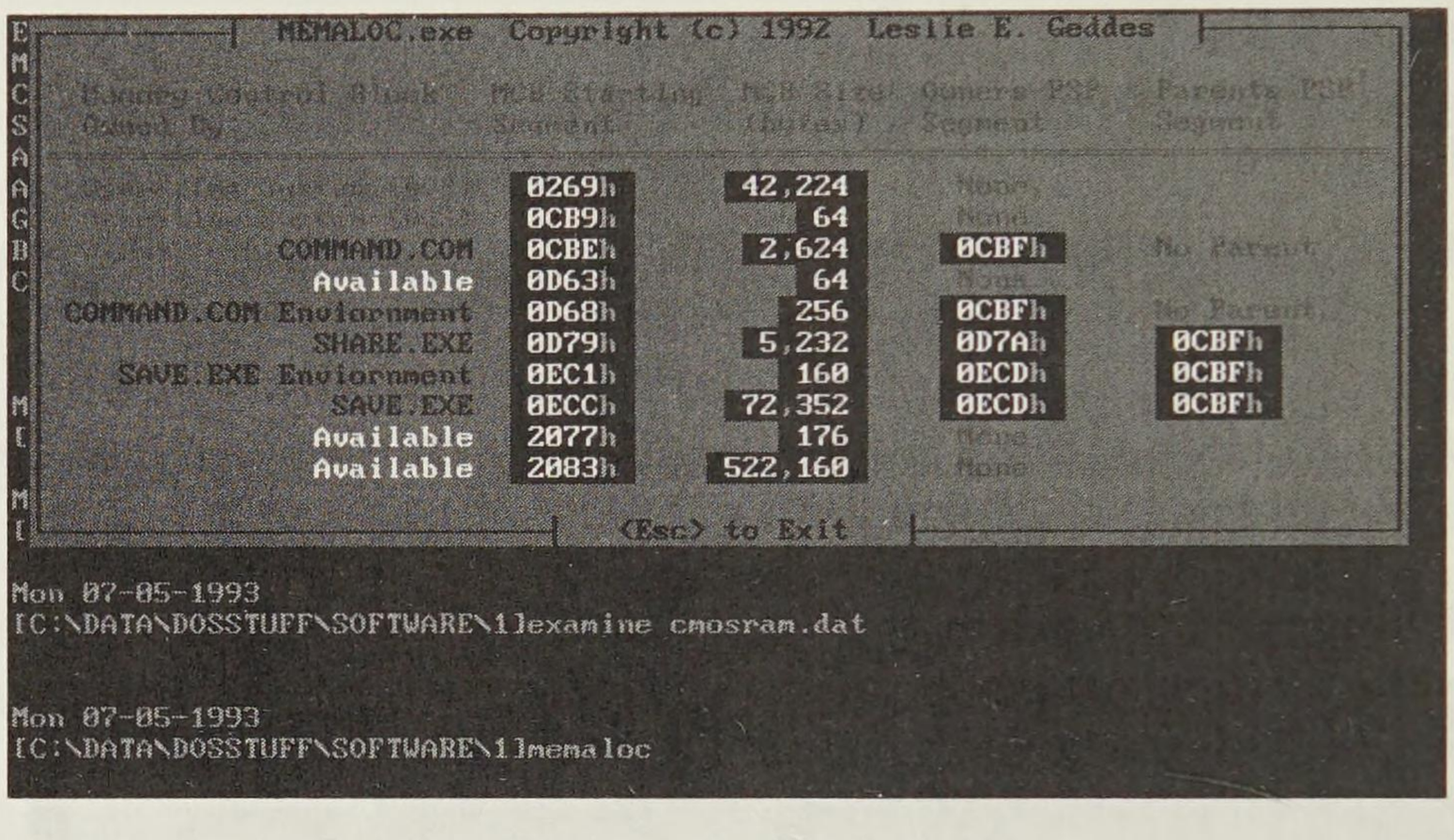


Fig. 6.5
Displaying the Memory Control Block information.

SUMMARY

Memory lends itself to good, old-fashioned technical discussion. (If you call something technical old-fashioned, that is.) Anyway, there is some power associated with managing memory.

In this chapter, you found three groups of utilities which enable you to make the best use of your memory and to record features of your memory. The DOSMAX utilities enable you to gain as much free memory out of your system as possible. Fast Load enables you to load often-used programs into memory for quick access to the programs. And, SYS_CAP enables you to protect your setup, and analyze your memory contents for excellent troubleshooting capabilities. By putting these groups of utilities to work on your computer, you won't lose your memory, you'll use it more efficiently.

7

AUTOMATING YOUR WORK

Do you use batch files? You probably do. With a batch file, you can automate many tasks. For example, you can create a batch file to copy specific files from different directories on your hard disk to a diskette. You can perform many copy commands by entering the name of the batch file. Most people create batch files to start programs. That way, by entering the name of the batch file, you instruct DOS to change directories and start the program. The idea is to decrease the amount of work *you* have to do by automating tasks.

Batch files are great, but they are also limited. For example, when you start a program using a batch file, the batch file halts—you can perform no other batch processing until you quit the program. A batch file cannot

MICROSOFT FORGOT:

- A scripting program to perform complex tasks that a batch file can't handle.

react to changed information on your screen. A batch file cannot automatically start in the middle of the night. Heck, the list of what a batch file *can't* do goes on and on.

Now, you ask, "Just why would I want a batch file to do any of those things listed above?" Good question! Why would you? Of course, if life (and DOS) were filled with only batch files, you would have no reason even to ask the question. But ask you did.

The answer is, *control*. You deserve to have more control over your computer than a batch file offers. You deserve to *automate* your computer in the truest meaning of the word. By golly, if you want to start a process when certain information displays on the screen, you should be able to! If you want to run an automated task in the middle of the night, you should be able to! Why should you be limited simply because *DOS won't let you*?

EXC is a program that enables you to *truly* automate your computer. With EXC, you can automate many tasks that batch files can't even think of performing. EXC is not a *batch file enhancer*, a program that enhances a batch file. Rather, EXC is completely separate from batch files.

This chapter shows you EXC, and gives you information with which you can CONTROL THE WORLD—AH, HA, HA, HA! ...Well, said the author sheepishly, at least control your computer.

Batch Files Revisited

A batch file is a text file containing commands you normally type from the command line. Although you generally include DOS commands in a batch file, there are a few specific commands that you only use in a batch file.

You create a batch file to perform repetitive commands. The most common use of a batch file is to start a program. Suppose you want to start a program called WP.EXE. That program resides in the \WORDP\ directory on drive C. Suppose you start the program each day, then when you finish the program, you copy any new or changed files to a

diskette. You would type the following commands at the command line, pressing Enter after each command:

```
CD \WORDP
WP
CD \
XCOPY \DATA\LETTERS A: /M
```

Rather than type the commands listed, create a batch file called WP.BAT. This file contains the four commands listed above.

To start the batch file, type the name of the batch file, **WP**, and press Enter. The batch file carries out the first two commands, changing the directory and starting WP.EXE. While WP.EXE is running, the batch file is inactive. However, when you exit WP.EXE, the batch file activates again, changing the directory and copying the new, changed files to a diskette.

Why is EXC superior to the batch file? Because the batch file halts while WP.EXE runs. But EXC can continue running, controlling WP.EXE with a script that you create. EXC can do everything that a batch file can do, such as running repetitious commands; plus, it can perform repetitious acts while other programs are running.

UNDERSTANDING EXC

So far, all you know is that EXC enables you to automate tasks. Well, how? With an EXC program, you can search for information to appear on the screen, and alter how the program operates based on what appears on the screen. You can enter information on the screen, and control the keyboard automatically. You can even create an EXC program that waits until a specific time to start.

EXC is not a TSR program. That is, EXC only occupies memory when you load it into memory. When you finish running a script, EXC leaves memory.

To register EXC 2.6a:

Parity Solutions
1105 Burgess Court
Arlington, TX 76015

After 10/31/93 use the following address:

1903 Pavia Ct.
Arlington, TX 76006

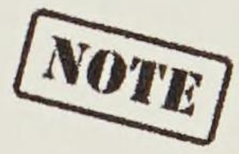
Complete registration and price information is available in the documents included with this utility. If you're going to keep it and use it, please register it!

WHAT ARE SCRIPTS?

A script is actually an EXC program. The EXC program contains EXC commands that enable you to perform many varied tasks. With an EXC script, you can

- Place text information onto the screen just as if you were typing from the keyboard.
- Search for information on the screen and alter the way the script works based on the information that appears on the screen.
- Control the position of the cursor.
- Highlight information on the screen using different colors.
- Place the contents of a file onto the screen.

You create a script using a text editor—EDIT, which comes with MS-DOS 5 and MS-DOS 6, for example. With the text editor running, type the EXC commands you need to control your script. You can include comments in your script by starting a comment line with a semicolon (;).


NOTE

> EXC ignores tabs and spaces at the beginning or the end of a line. This enables you to add spacing to your scripts. Spacing in a script is useful in helping you understand program flow, much as indents in an outline enable you to see main points and sub-points.

When you save the script, give the file the extension SCR. For example, SCRIPT.SCR. If you've indicated the script to run with EXC, you only have to indicate the script name, not the script extension.

STARTING EXC

You start a script from the DOS command line. EXC enables you to start just a script, or both a script and a program. The syntax for EXC is as follows:

EXC script.*scr program programparams*

In the syntax, **script**.*scr* is the script to run. You must indicate at least the name of the script. If the script file contains an extension other than SCR, then you must indicate the extension. *program* is the program to start after loading the script. *programparams* are any parameters that affect *program*.

For example, suppose you want to start the script NAME.SCR. Start Microsoft Word 5.5 and load the document LETTER.DOC. Type the following command and press Enter:

EXC NAME WORD LETTER.DOC

In this case, EXC loads into memory and reads NAME.SCR. Then, Word 5.5 starts and loads the file LETTER.DOC. Finally, the script begins its operations.

MEMORY USAGE

Since EXC is not a TSR, EXC requires no memory when a script is not active. However, you don't get off scot-free. When running a script, EXC *does* require some memory—but not a bunch.

When running, EXC requires about 40K of memory. Additionally, EXC creates two buffers—one for the script and one for the screen. When you run a script, EXC must load the entire script into memory. Because some EXC commands search for information on the screen, or determine if a screen has changed, EXC must buffer the screen. EXC keeps a copy of the screen in the buffer for comparison with what actually appears on the screen.

The buffer size for the script depends on the contents of the script. EXC allocates buffer memory for commands and for one space between commands—at 1 byte per character. EXC ignores spaces in your script file, other than the single space between commands. So, a script file with two commands and one hundred spaces takes just as much buffer space as a script file with only two commands.

Screen buffer size varies depending on how much information displays on your screen. EXC assigns one byte of memory for each character position on the screen, plus 4 bytes. For example, an 80×25 text screen requires a buffer of 2004 bytes; an 80×50 text screen requires 4004 bytes.

All in all, you can run EXC in 80×25 screen mode with a decent-sized script (4,000 characters, including spaces between commands) in less than 50K of memory.

CREATING A SCRIPT

Writing a script is simple—just type the script into a text editor like EDIT and save it with some descriptive file name and the extension SCR. Although writing a script is simple, *creating* a script is not so simple. (There's always a catch, right?)

Creating a script requires that you know what commands you need to use in your script in order to make your script do what you want it to do. Creating a script requires that you *plan* your script. Planning your script means you have to think through the steps you want your script to perform. Write down each step. Then apply EXC commands to perform each of those steps. This section will show you how to plan, then write, a simple script.

PLANNING A SCRIPT

What is a good topic for a script? Hmmm... it needs to be simple. OK, here goes! From time to time, you probably create a document that uses some long, hard-to-type name. Something like the *International Widget Corporation, Atlantic Division*. You have to type that name bunches of times in the document. Isn't there an easy way to automate typing that long name? There sure is—EXC. (Why else would I go through this except to show how EXC works?)

NOTE

In order to create a script, you must use EXC commands. The section "EXC Commands," found later in this chapter, lists the different EXC commands you can use to create scripts. The sample script in this section uses the following commands: CASE OFF, IFY, SEARCH, LOCATE, KEY, TYPE, GOTO, QUIT, WAITFOR. You might want to reference "EXC Commands" for a full explanation of what these commands do.

Typing a long name is only one task this sample script can perform. Suppose you want to add another task to the script? For example, you also want your script to automatically spell-check the document, save it, then print it when you're finished typing the document. No problem. You can include commands that control your program.

Let's recap. You want a script that performs two separate tasks. The first task is to type a long name on the screen. The second is to spell-check, print, and save the document.

How can you automate these two tasks? Well, remember, EXC is good at looking for information on the screen and reacting based on that information. You can create a script that looks for a certain code that you type on the screen. You can assign one code to represent that long name. For example, determine that ~~ is the code to represent the long name. Then, assign another code that means you are finished typing, for example **XXX**. When EXC finds one of the codes, the script reacts—either replacing that code with the long name, or deleting the code,

spell-checking, printing, and saving the document. The flow chart shown in figure 7.1 shows the tasks that the script should perform:

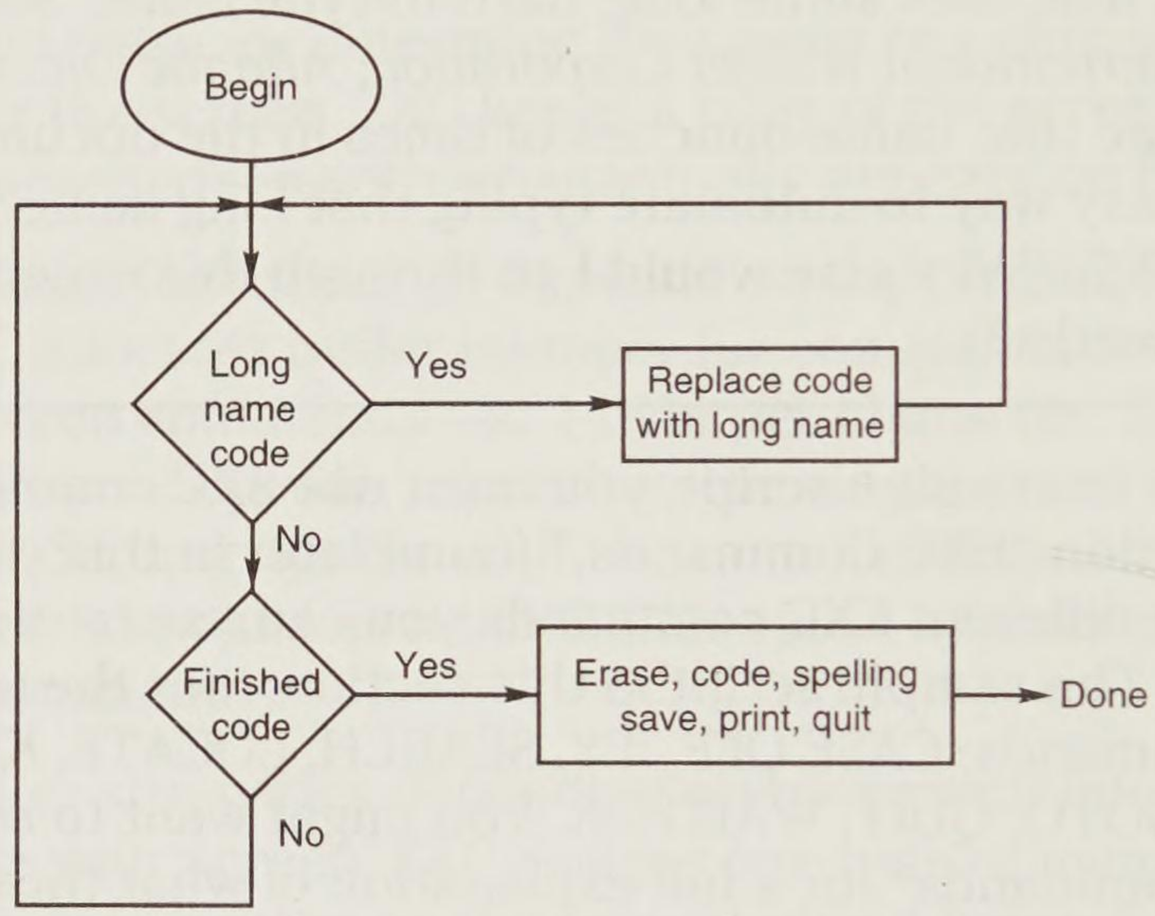


Fig. 7.1
A flow chart of the simple script.

Take a close look at the flow chart. There are two test boxes, each testing for a different code to appear on the screen. Notice that if both of the tests answer no, the script just loops back around to perform the tests again. From the flow chart, then, you can see that your script must have one main loop.

Another thing you can see from the flow chart: there are two different action boxes. One action box replaces a code with the long name, then returns back to the main loop. The second action box deletes the code you typed, checks the spelling, saves and prints the document, then ends the script. You could say your script contains one main loop and two subroutines. (By the way, a subroutine is just a small set of commands that a script uses over and over again—like replacing code with a long name.)

Before you continue to write your script, you need to do a little research. You need to find out what keystrokes you must press to check the spelling of your document, save the document, and print the document. You also must record the information that displays on your screen when each of these tasks completes.

I decided to use Q&A as the word processor. To spell check a document in Q&A, press Shift+F1. When spelling completes, Q&A displays the message `Spelling check completed.` at the bottom of the screen. To print a document, you press F2 followed by F10. When printing completes, Q&A displays one of two messages. Either `1 page printed.` or *x* `pages printed.` where *x* is the number of pages that Q&A printed. To save a document in Q&A, press Shift+F8, type the name, and press F10.

You have now fully planned your script. Here comes the tough part—writing the script.

WRITING A SCRIPT

Remember high school composition class? You had to write a paper. You have your subject, but where do you begin? Writing a script may seem a bit like that high school paper. Where do you begin? "Why, at the beginning, of course!" Pretty plain and simple, huh? But for now, let's work backwards. I'll list the completed script, then describe how I arrived at the script.

NOTE

> Keep in mind that EXC is not meant to be used with Microsoft Windows. In fact, EXC will not work with Windows (though you can use EXC from a DOS window in Windows). You can use EXC, however, with DOS programs, including DESQView and VMiX (a multitasking program included in this book).

The following lists the script to perform the tasks planned. Each line is numbered for reference only. When you write scripts, however, do not include line numbers.

```
1   ;The beginning of the script
2   CASE OFF
3
4   ;The main loop that searches for codes
5   :LOOP
6     SEARCH "~~"
7           IFY REPLACE
```

```
 8     SEARCH "XXX"
 9           IFY END
10     GOTO LOOP
11
12   ;Subroutine to replace a code with a long name
13   :REPLACE
14     LOCATE "~~"
15     KEY <DEL> 2
16     TYPE "International Widget Corporation, Atlantic Division"
17     GOTO LOOP
18
19   ;Subroutine to check spelling, save and print the document
20   :END
21     ;Delete the code
22      KEY <BKSP> 3
23     ;Check spelling
24      KEY <SHF+F1>
25      WAITFOR "Spelling check completed."
26     ;Print document
27      KEY <F2>
28      KEY <F10>
29      WAITFOR "printed."
30     ;Save document
31      KEY <SHF+F8>
32      QUIT
```

Before you begin analyzing the script, take a look at lines 1, 4, 12, 19, 21, 23, 26, and 30. These eight lines begin with a semicolon. Remember what that means? That means the lines are simply comments. These comments are reminders as to what that portion of the script does.

You might ask, "Why waste your time with comments in a script?" I used to ask that question about adding comments in dBASE code. Then one day I had to modify a dBASE program I had written several months before. After looking through the program for a while, I began asking myself what I was thinking when I wrote the code. I couldn't remember what parts of the code were for. "Oh, how I wish I would have added comments," said I! Comments serve as a reminder of just what was running

through your head when you wrote the script. You should get used to them and use them often.

Notice another point about the script. The script contains spaces to make it more readable. There are blank lines between groups of commands. Some code is indented to show the relationship of that code to the code group. Remember, EXC ignores the extra spaces. But, like the comments, the spaces are there for your benefit. Enough said. On with the script discussion.

STARTING YOUR SCRIPT

Before you can use your script, you must start the script, called REPLACE.SCR. Because Q&A was used as the word processor, you must start Q&A, also. Remember, EXC enables you to start your script and the program on the same line. For example, type the following command:

```
EXC C:\SCRIPTS\REPLACE QA
```

This command assumes a few things. First, that EXC is in a directory on the DOS search path. Second, that you have a directory called C:\SCRIPTS where you keep your script files. Third, that your script is called REPLACE.SCR. And, finally, that the current directory is the directory containing the Q&A program. When Q&A starts, you begin typing your document as usual. EXC runs in the background, so to speak, enabling the script to work while you work.

As the script begins, it does a little preparation. Notice line 2—the command CASE OFF. This command tells EXC not to be case-sensitive. So, when EXC scans the screen for the text **XXX**, it finds any combination of upper- or lowercase X's, as long as there are three X's in a row.

> Even though the comments are important, and appear to be part of the code, they won't be part of the discussion. Remember, EXC ignores comments.

NOTE

Line 5 begins the main loop. Remember, our flow chart in figure 7.1 showed us we needed a main loop? Line 5 contains a label.

A label is nothing more than a word beginning with a colon. You use labels in scripts as the point where you can alter the flow of the script. For example, notice line 10—GOTO LOOP. When EXC processes this line, EXC changes processing of the script to continue, again, with line 5.

Line 6 contains the first SEARCH command. Remember, your script is searching the screen for you to type in one of two codes. So, line 6 looks for the first of these codes, two tildes (~~).

In the SEARCH command, the two tildes make up the parameter for SEARCH.

Note that in line 6 the two tildes are shown with quotation marks around them. This is important in EXC. Parameters that are literal text you are searching for, or that you want to display on the screen, must be enclosed in quotes.

You must enclose literal text to search for or to display on the screen in quotes. But suppose you want to search for text that contains quotes—for example **"Hello"**. Use the code **\q** to represent a quotation mark. For example, to search for the text containing quotes, use the following:

SEARCH "\qHELLO\q"

EXC maintains an internal *yes/no* variable. That is, this mechanism changes between yes and no depending on the success of the SEARCH command (or other commands that look for information on the screen). For example, if the screen does not contain two tildes, the internal variable is set to **NO**. However, when the screen contains two tildes, the internal variable is set to **YES**. This variable is very integral to EXC, as you will see in lines 7 and 9 of the script.

Line 7 acts on the outcome of line 6, and on the contents of the internal yes/no variable. The IFY command changes the direction of script processing if the internal variable is **YES**. If the internal variable is YES, then EXC changes script processing to the label following the IFY command—in this case, the label is

REPLACE (found on line 13). However, if the internal variable is NO, script processing continues with the next line.

Line 8 contains the second search. This time, the script searches the screen for **XXX**. Line 9, then, tests the internal Yes/No variable again. If EXC found **XXX** on the screen, EXC changes the direction of script processing to the label END. If the internal variable is NO, EXC continues with the next line, line 10.

Line 10 contains the GOTO command mentioned earlier. This command creates the looping action. When EXC reads this command, it changes script processing back to line 5.

EXC happily runs this loop forever. EXC keeps searching for ~~ and **XXX** until it finds one of those codes on the screen. When EXC finds one of the codes on the screen, it breaks out of the loop and starts processing one of the subroutines.

THE REPLACE SUBROUTINE

Suppose, as you are typing on your word processor, you need to type that long name, *International Widget Corporation, Atlantic Division*. Instead of typing the long name, type two tildes. When EXC processes line 6, finding the two tildes, it sets the internal variable to **YES**. Line 7 discovers that the internal variable is **YES** and changes script processing to the label REPLACE, line 13.

(By the way, when you type either of the two codes, stop typing. Allow EXC time to process your script.)

Line 14 contains the LOCATE command. The purpose of this command is to locate text on the screen and place the cursor at the beginning of this text. In this case, LOCATE moves your cursor to the first of the two tildes.

Once EXC moves the cursor to the beginning of the two tildes, it continues with line 15 of the script. This line contains the KEY command. The purpose of this command is to simulate pressing keys on the keyboard. Line 15 of the script is the same as pressing the Del key 2 times. So this line of the script eliminates the two tildes.

Line 16 places that long name on the screen, in place of the two tildes. The TYPE command inserts the text contained in quotes on the screen at the position of the cursor just as if you typed from the keyboard. (OK, maybe faster than you type.) Once the name is placed on the screen, line 17 redirects script processing back to the loop with the GOTO command.

THE END SUBROUTINE

So, you've finished typing your document, having used that long name the required one hundred times. Now it's time to spell-check the document, print it, and save it. So, at the end of the document you type **XXX**. Of course, your script is just merrily running through the loop to beat the band. Lines 8 and 9 of the script find the end code and change script processing to line 20, the END label.

The purpose of the end subroutine is to perform specific functions of your word processor, so the KEY command is used quite a bit in this subroutine to control your word processor.

Line 22 is the first KEY command. This command simulates pressing the Backspace key 3 times. This deletes the **XXX** code.

Line 24 is the second KEY command. This command simulates pressing a key combination, Shift+F1. As noted when planning the script, Shift+F1 makes Q&A spell-check the document.

Notice line 25, the WAITFOR command. This command monitors information that appears on the screen. The command in line 25 looks for the text `Spelling check completed.` to appear on the screen. At this point, the script is in a holding pattern. Of course, this message does not display on the screen until the spelling check completes. So, you continue checking the spelling, replacing misspelled words until the spell-check is complete. When the message appears on the screen, the script continues its work.

Lines 27 and 28 contain two more KEY commands. These commands sequentially press F2 and F10, keys which instruct Q&A to print the document.

Line 29 is another WAITFOR command. This time, the script waits for the message `printed.` to appear on the screen. Another holding pattern. (Q&A displays a message at the bottom of the screen telling you how many pages were printed.) Once this message appears on the screen, the script continues.

Line 31 contains the final KEY command, pressing the key combination to save the document. Line 32 contains the QUIT command. With the QUIT command, the script halts processing, and EXC is finished. However, EXC does not leave memory just yet. EXC remains in memory until you quit the program that you started.

WORKING WITH A SCRIPT

You can alter how a script works in several ways. You can pause a script while it is running. You can abort a script altogether. You can also change information that a script uses by using DOS environment variables.

You may have occasion to pause a script while it is running. Pressing the Scroll Lock key while a script is running pauses the script. Pressing Scroll Lock a second time enables the script to continue. The Scroll Lock key acts as a toggle.

You might have to stop a script from processing. To stop a script from processing, press the right and left shift keys simultaneously.

You can use DOS environment variables in a script in order to change the way that the script functions. (You create a DOS environment variable by using the SET command at the DOS command line.) For example, the sample script shown before is pretty set. This script can only replace the two-tilde code with the one long name, *International Widget Corporation, Atlantic Division*. However, using a variable in place of this long name makes the script more versatile. You can change the variable at the DOS command line so that the script can replace the two-tilde code with any name you prefer.

To use a DOS environment variable in a script, place the variable name in the script in place of the text. Include a percent

sign on both sides of the variable. For example, to use the environment variable LONGNAME in the sample script, change line 16 as follows:

```
TYPE "%LONGNAME%"
```

Notice that %LONGNAME% is placed in quotes, just as *International Widget Corporation, Atlantic Division* in the original script was enclosed in quotes. Then, before you run the script, create the DOS environment variable. For example, at the DOS command line, type the following and press Enter:

```
SET LONGNAME=International Widget Corporation,
Atlantic Division
```

When you run the script, EXC replaces the variable %LONGNAME% with the contents of the variable.

EXC COMMANDS

You can choose from 33 commands to create an EXC script. You learned about nine of these commands in the sample script above. However, you deserve a complete list of the commands, showing syntax and a description of each command. So that's what you'll get. The commands in this section are listed by function, showing the syntax to use, and a description of the command.

CONTROLLING THE SCRIPT

The commands in this section enable you to control how the script operates. With these commands, you can emit a sound from the speaker, suspend the script, test for information to appear on the screen, and stop the script, just to name a few.

BEEP

Use the BEEP command to emit a tone from the speaker. Use the following syntax:

```
BEEP
```

Generally, you will not use BEEP in a script. However, you will find this command useful when testing a script. You can insert the beep command in your script at strategic locations to audibly follow script processing.

DELAY

Use DELAY to pause the script for an indicated number of seconds. The syntax for DELAY is:

DELAY s

You can only use one parameter with DELAY. This parameter, **s**, indicates the number of seconds that the script should pause before continuing processing. For example, DELAY 5 causes a script to pause processing for 5 seconds.

GOTO

Redirect script processing with GOTO. The syntax for GOTO is as follows:

GOTO label

In the syntax line, **label** is a text label that appears on a separate line in the script. The text label must begin with a colon. The GOTO command enables you to skip from procedure to procedure in the script. Most often, you use GOTO to set up a loop, repeating steps over and over again.

IFN

Changes the direction of the script when a "no" condition is met. To use IFN, follow this syntax:

IFN label

In the syntax line, **label** is a text label that appears on a separate line in the script. The text label must begin with a colon. The IFN command enables you to skip to a different procedure in the script when there is a "no" condition. Generally, you use the IFN command after you search the screen for text. IFN transfers script control to **label** if the text did not appear on the screen.

IFY

Changes the direction of the script when a "yes" condition is met. To use IFY, follow this syntax:

IFY label

In the syntax line, **label** is a text label that appears on a separate line in the script. The text label must begin with a colon. The IFY command enables you to skip to a different procedure in the script when there is a "yes" condition. Generally, you use the IFY command after you search the screen for text. IFY transfers script control to **label** if the text appears on the screen.

KEY

The KEY command enables EXC to simulate pressing a key or key combination from the keyboard. The syntax for KEY is:

KEY <keyname> *n*

You can indicate two parameters with KEY. The first is **keyname**—in which you indicate the key or key combination to press. You must include the angle brackets around the keyname. Optionally, you can indicate *n*, the number of times you press the key.

You must choose from specific keynames. The following is a list of acceptable keynames:

<BKSP> <DEL> <DOWN> <END> <ENTER> <ESC>
<F1>-<F12> <HOME> <INS> <LEFT> <PGDN> <PGUP>
<PRTSC> <RIGHT> <TAB> <UP> <GRAY5> <GRAY+>
<GRAY-> <GRAY*> <SPACE>

Additionally, you may indicate any alpha-numeric key by placing that character in angle brackets, for example, <D>.

For key combinations, use the following keys (where *x* is any non-reserved key):

<SHF+*x*> <CTL+*x*> <ALT+*x*>

For example, to use the Shift+F1 key combination, use <SHF+F1>.

ON

Enables the script to break out of a loop after a specific number of repetitions. The syntax for ON is:

ON r label

In the syntax line, **r** is the number of repetitions that must pass before breaking out of a loop. **label** is a text label that appears on a separate line in the script. The text label must begin with a colon.

If, for example, you place the command **ON 5 PRINT** in a script, EXC will pass the command 5 times. On the sixth time, however, the ON command changes the script processing control to the label **PRINT**.

PAUSE

Enables you to pause a script. The syntax to use is:

PAUSE

This command in a script performs the same function as you pressing the Scroll Lock key. To continue the script, press the Scroll Lock key.

QUIT

Stops script processing. The syntax for QUIT is:

QUIT

When processing this command, EXC aborts script execution. The same thing happens when EXC reaches the end of the script file.

SETWAIT

Enables you to indicate the number of seconds that EXC should wait when processing the WAITFOR or WAITSCR commands. The syntax for SETWAIT is as follows:

SETWAIT s

The parameter **s** indicates the number of seconds that EXC should wait before continuing on with the script. If you indicate **SETWAIT 0**, then EXC does not time out—EXC will wait forever.

TICK

Enables you to pause the script for an indicated time. The syntax for TICK is as follows:

TICK n

In the syntax, **n** indicates the time that EXC pauses, in increments of 1/18th of a second. So the command **TICK 18** indicates that EXC should pause the script for one second.

TIME

Use the TIME command to pause a script until a specific time during the day. The syntax to use is:

TIME hhmmss

In the syntax line, the one parameter you indicate specifies the hour, minute, and second of the day when the script should activate. This command uses military time. For example, to pause a script until 8:30 p.m., insert the command **TIME 203000** in the script.

TIMEOUT

Enables EXC to change script processing when experiencing a timeout from a WAITFOR or WAITSCR command. The syntax is as follows:

TIMEOUT label

If you specified a timeout length with SETWAIT, EXC monitors the WAITFOR and WAITSCR commands. If the set length of time passes while waiting for text to appear on the screen or waiting for the screen to change, EXC causes the script to continue processing. You can use the TIMEOUT command to indicate *where* the script should continue processing.

DISPLAYING AND FINDING TEXT

The commands in this section enable you to display text on the screen, display the contents of a file on the screen, and find text that appears on the screen.

LOCATE

The LOCATE command finds indicated text and positions the cursor at the beginning of that text. The syntax for LOCATE is as follows:

LOCATE "text"

You include the one parameter **text** in quotation marks. **text** is any information that you want EXC to find on the screen. If you want to locate text that includes quotation marks, represent quotation marks with **\q**. For example, **LOCATE "\qInformation\q"**. If EXC finds **text**, then it sets the internal variable to YES. Otherwise, EXC sets the internal variable to NO.

LOOK

The LOOK command looks for information on the screen the last time that the screen was read into the screen buffer. The syntax for LOOK is:

LOOK "text"

EXC reads the screen into the screen buffer only with the commands READ, SEARCH, WAITSCR, and WAITFOR. LOOK searches through the screen buffer, rather than the current contents of the screen, to find **text**. If you want to locate text that includes quotation marks, represent quotation marks with **\q**. For example, **LOOK "\qInformation\q"**. If EXC finds the text, it sets the internal variable to YES. Otherwise, EXC sets the internal variable to NO.

SEARCH

EXC reads the screen into the screen buffer and searches for specific information. The syntax for SEARCH is:

SEARCH "text"

SEARCH reads the screen into the screen buffer, then searches through the screen buffer to find **text**. Again, use **\q** to represent quotation marks in the text to search for. If EXC finds the text, it sets the internal variable to YES. Otherwise, EXC sets the internal variable to NO.

SHOW

You display information on the screen with SHOW. The syntax for SHOW is:

SHOW attrib "text"

There are two parameters you can use with SHOW. The first, **attrib**, is the attribute that you want to use to display **text**. The **attrib** designates the background and foreground colors. Although you are designating two colors, there is only one **attrib** parameter. These attributes and associated colors are listed below:

Background color attributes

Attribute	*Color*
0	Black
16	Blue
32	Green
48	Cyan
64	Red
80	Magenta
96	Brown
112	White

Foreground color attributes

Attribute	*Color*
0	Black
1	Blue
2	Green
3	Cyan
4	Red
5	Magenta
6	Brown
7	White
8	Gray
9	Light Blue
10	Light Green
11	Light Cyan
12	Light Red
13	Light Magenta
14	Yellow
15	Bright White

To indicate an attribute with a foreground and background color, add together the attributes for the foreground and background color, and use that as the attribute. For example, to display Blue on white, add 112 and 1 together, getting the attribute 113.

The second parameter, **text**, is the information to display on the screen. You must include **text** in quotation marks. Again, use **\q** to represent quotation marks within **text**. The information displays on the screen indicated by the SHOWAT command.

TYPE

Displays information on the screen. The syntax for TYPE is:

TYPE "text"

The TYPE command simulates you typing information from the keyboard. **text** is the information that you want to appear on the screen. You must include **text** in quotation marks. Again, use **\q** to represent quotation marks within **text**.

TYPFILE

The TYPFILE command displays the contents of a text file on the screen. The syntax for TYPFILE is:

TYPFILE *d:\path***filename** *cr*

When you use TYPFILE, you must indicate the name of the text file, including the path of the file. Additionally, you can indicate the character that EXC should use to end each line. The end-of-line character is indicated by *cr*. Indicate *cr* using the key designations—for example, <SPACE> or <DEL>—shown in the KEY command.

Most text files end each line with a carriage return and a line feed. This is fine for displaying text on the screen and for displaying text on a printer. However, if you want to include the information from the text file in a word processor, the carriage return and line feed might get in the way.

Generally, when you type in a word processor, you don't press Enter at the end of every line. Rather, you allow the word processor to automatically *word-wrap* at the end of the line. When you indicate <SPACE> for *cr*, the text typed from the text file onto the screen can word-wrap just as if you were typing from the keyboard. For example, to include text from the file C:\TEMP\TEXT.TXT into a letter using a script, include the line **TYPFILE C:\TEMP\TEXT.TXT <SPACE>** in your script. If you do not indicate *cr*, EXC enters nothing at the end of each line, making the entire text file just one long line.

WAITFOR

The WAITFOR command causes EXC to wait for indicated information to appear on the screen before continuing script processing. The ability to wait for information to appear on the screen, then change the actions the script takes based on that information, is what makes EXC unusual from batch files. The syntax for WAITFOR is:

WAITFOR *NOT* **"text"**

The **text** parameter is the information the script waits for to appear on-screen. You must include **text** in quotation marks. Again, you can represent quotation marks within **text** as **/q**.

WAITFOR pauses the script until the indicated information displays on the screen. When **text** does display, the processing continues. The amount of time that EXC waits for the text depends on the setting of SETWAIT. (For more information about SETWAIT see the section, "Controlling the Script.") If you include the *NOT* parameter, then EXC waits for the indicated information to *not* appear on the screen.

WAITSCR

WAITSCR causes EXC to wait until the screen changes. The syntax for WAITSCR is:

WAITSCR

WAITSCR reads the current screen into the screen buffer, then pauses the script until something on the screen changes. EXC compares the active screen with the screen buffer to determine when the screen changes. The amount of time that EXC waits for the text depends on the setting of SETWAIT. (For more information about SETWAIT see the section, "Controlling the Script.")

CONTROLLING THE SCREEN

The commands in this section enable you to control the screen. With these commands, you can perform commands that alter searching for text, clear the screen, and control the area of the screen that EXC affects.

CASE ON AND CASE OFF

Use CASE ON and CASE OFF to control the case-sensitivity of text searches. Use the following syntax:

CASE ON

CASE OFF

When case is on, EXC makes case-sensitive searches. Text strings must match identically with upper- and lowercase letters. When case is off, however, EXC only matches like characters, no matter whether the characters are upper- or lowercase. By default, the case is on.

CLEAR

You use CLEAR to clear the screen. The syntax to use is as follows:

CLEAR

Use the CLEAR command with care. When you use CLEAR, the entire screen clears. When running some programs, you may clear information off the screen that you need to operate the program. Before using CLEAR with a program, be sure to test how CLEAR affects the program.

CURSOR

Use CURSOR to move the cursor to a different location on the screen. The syntax for CURSOR is as follows:

CURSOR c r

You use two parameters with CURSOR. The **c** parameter indicates the column to move the cursor to, and the **r** parameter indicates the row to move the cursor to. The way CURSOR works is by issuing a series of cursor key movements to reposition the cursor at the desired location. For example, if the cursor currently is in column 12, row 10, and your script contains the command CURSOR 20 20, then EXC issues 8 right-arrow keypresses and 10 down-arrow keypresses.

HIDEWIN

Removes the attributes set to the screen by the SHOWIN command. (See "SHOWIN," later in this section.) The syntax for HIDEWIN is:

HIDEWIN

You can highlight areas of the screen with SHOWIN. To remove the screen highlight, use the HIDEWIN command.

READ

The READ command reads the screen, placing a copy of the current screen into the screen buffer. To use READ, use the following syntax:

READ

When you issue the READ command, EXC reads a copy of the screen into the screen buffer. You can then use the LOOK command to find information on-screen. Using READ to read the screen, then using LOOK to search for text on a screen that does not change, is much quicker than using SEARCH several times in a row. SEARCH reads the screen with each search, while LOOK does not read the screen.

SCRMAX

Use SCRMAX to allocate the screen buffer. The syntax for SCRMAX is:

SCRMAX c r

There are two parameters you use with SCRMAX. The **c** parameter indicates the number of screen columns, and the **r** parameter indicates the number or screen rows. Using the command **SCRMAX 80 25** indicates that you want a screen buffer large enough to store a screen 80 columns by 25 rows.

SHOWAT

Indicates the location on the screen to display information using the SHOW command. The syntax for SHOWAT is:

SHOWAT c r

The two parameters indicate the location on the screen that SHOW displays information.The **c** parameter is the column and the **r** parameter is the row. To display information at column 5 and row 10, use the command **SHOWAT 5 10**.

SHOWIN

The SHOWIN command highlights the contents of the indicated area of the screen. The syntax for SHOWIN is:

SHOWIN attrib

The **attrib** parameter for SHOWIN is the same as for the SHOW command. See the tables listed with the SHOW command above to determine the attribute.

You must use SHOWIN after you indicate the area of the screen to highlight. You indicate the area of the screen using the WINCOLS and WINROWS commands.

WINCOLS

Enables you to restrict the area of the screen that the script searches for information. To use WINCOLS, follow this syntax:

WINCOLS l r

The parameters you use with WINCOLS, **l** and **r**, enable you to indicate the left- and right-most columns of the screen, respectively. The command **WINCOLS 10 20** restricts EXC to searching the screen from columns 10 to 20 only. Use this command in conjunction with WINROWS for the most flexibility. Use the WINMAX command to override the WINCOLS settings, removing the restriction.

WINMAX

This command sets the search area of the screen to its maximum size. To use WINMAX, follow this syntax:

WINMAX

If you restrict the search area using the WINCOLS and WINROWS commands, use WINMAX to remove the search restriction.

WINROWS

Enables you to restrict the area of the screen that the script searches for information. To use WINROWS, follow this syntax:

WINROWS t b

The parameters you use with WINROWS, **t** and **r**, enable you to indicate the top- and bottom-most rows of the screen, respectively. The command **WINROWS 10 20** restricts EXC to searching the screen from rows 10 to 20 only. Use this command in conjunction with WINCOLS for the most flexibility. Use the WINMAX command to override the WINROWS settings, removing the restriction.

SUMMARY

Automation is one of the best things to use computers for. There are many things you do from day to day that are *manual labor* tasks. You shouldn't have to perform manual labor—you can afford a computer! EXC is just the program to fit the bill.

You will find many uses for EXC. You can create scripts that back up your computer at night, running a virus check beforehand. You can create scripts that work with communications software and log onto BBSs automatically, reading any mail that you have. There are many uses for EXC. Just think of the extra time you'll have when you let EXC automate your tasks. You'll have more time to play with the stuff that Microsoft forgot!

MAKING DOS MULTITASK

8

Could you imagine trying to run a grocery store by yourself? Just think, you'd have to check customers out, butcher meat, stock shelves, provide customer service, weigh produce, receive new inventory.... I don't think so! Face it, a grocery store runs nicely when you let each person in each department perform their own task. What you can do is *manage* all those people performing their different tasks quite nicely.

So maybe running several programs on your computer at once sounds silly—just as you can't run a grocery store by yourself, you can't run many programs by yourself. Or, can you? No, you can't actually type information into every program running on your computer. But you can *manage* all those programs.

MICROSOFT FORGOT:

- A way to multitask without Windows.

Start that spreadsheet program, load the worksheet, and start that long calculation process. Start your database program and begin sorting the names of your customers. Start that word processing program and print that long report you wrote the day before. While all those things are running, start your personal information manager program and plan your day. All the time, everything else is running just fine.

"But," you interject, "DOS doesn't multitask!" *Oh*—more DOS stuff Microsoft forgot! (OK, OK; you can run that Windows thing and multitask. But you gotta admit—since Microsoft didn't bundle Windows with DOS, they forgot it. Or was that just another marketing opportunity?)

In this chapter, you learn about a program called VMiX. VMiX is a multitasking program with a special purpose. You can not only run multiple programs at once, but you also can attach a terminal to your computer, enabling more than one person to use your computer.

MULTITASKING DOS WITH VMiX

VMiX is a Shareware multitasking, multiuser operating environment that works with MS-DOS. That means you can run several programs at once on your computer, and attach other computers to your computer so that other people can run programs on your computer. And, you can link two computers running VMiX so that they can communicate with one another, sharing computing responsibility.

VMiX is not an uppity program, as some multitasking programs are. VMiX will run on an 8088 computer, as well as a 286, 386, or 486 computer. On an 8088 computer, you can run multiple tasks in conventional memory. On a 286, you can use VMiX in 8086 mode, multitasking programs in conventional memory, or you can use 286 mode, swapping programs to and from extended memory.

> The lowest processor mode in which VMiX operates is 8086 mode. You use 8086-mode on computers using the 8086 and 8088 microprocessor. However, you can use 8086 mode on a 286, 386, or 486 computer as well.

Using VMiX's 386 mode (on 386 and 486 computers) gives you the greatest power. This mode takes advantage of the virtual 8086 capabilities of the 386 and 486 microprocessors. Each task that you start works just as if it is running on its own computer—with its own memory and video.

> *To register VMiX 2.87:*
>
> Commercial Software Associates
> P.O. Box 36
> Corona del Mar, CA 92625
>
> Complete registration and price information is available in the documents included with this utility. If you're going to keep it and use it, please register it!

The next section shows you how to configure and run VMiX, and how to use remote access to attach a second computer to the VMiX computer for multiuser capabilities.

CONFIGURING DOS FOR VMiX

Before you start VMiX, you must configure DOS. Many multitasking programs require you to add a bunch of device drivers and memory managers to your computer. VMiX doesn't require that. In fact, VMiX is happy with a very minimal configuration. That's because VMiX manages memory itself.

VMiX in 386 mode manages all the memory. VMiX provides EMS memory to programs requiring that kind of memory. And VMiX enables you to use upper memory blocks. With upper memory blocks, you can load TSRs high, freeing VMiX conventional memory for the application you run in VMiX to use. You can even load VMiX into upper memory blocks.

The configuration (that is, CONFIG.SYS) that Commercial Software Associates recommends for use with VMiX is as follows:

```
SHELL=C:\COMMAND.COM C:\ /P /E:512
FCBS=0,0
STACKS=9,192
DOS=HIGH
BUFFERS=15
FILES=20
LASTDRIVE=H
DEVICE=HIMEM.SYS
DEVICE=MOUSE.SYS
```

You have some leeway in the file listed above, of course. For example, you can change BUFFERS, FILES, and LASTDRIVE to meet your needs. If HIMEM.SYS requires a particular parameter (such as /MACHINE:11) you can add that parameter. Also, if the HIMEM.SYS or the MOUSE.SYS resides in a directory, add the path to the device drivers (C:\DOS\HIMEM.SYS, for example). If you don't have a mouse, don't add MOUSE.SYS as a device driver. If your computer requires some other device driver (such as DBLSPACE.SYS), by all means, add that device driver to the configuration.

"How should I configure my AUTOEXEC.BAT file?" you might ask. That's up to you. Fix that little feller any way you like. However, you should add the directory where VMiX is located to your PATH statement.

When you install the utilities from the disk, Install places VMiX in the \DOSSTUFF\VMIX directory. You can run VMiX from that directory, or you can copy the files to a new directory, \VMIX. There is no special setup program that you must run. VMiX is ready when you are.

STARTING VMiX

When you boot your computer, change to the directory containing VMiX, unless that directory is part of your search path. Then, type **VM_BOOT** and press Enter.

When VMiX starts, it makes several decisions based on the type of computer that you have. For example, VMiX determines the mode that it must operate in—8086 mode, 286 mode, or 386 mode. If VMiX operates in 286 or 386 mode, then VMiX determines whether it can load itself into extended (XMS) memory.

Though VMiX makes many decisions while starting, you can alter those decisions by including command-line options. And, you can enter VMiX commands as you start VMiX. The complete syntax for VMiX is as follows:

VM_BOOT *8086 286 LOW HIGH QUIET TEXT EMS=eeee*
EXCLUDE=xxxx /vmixcommand

The following table lists the purpose of each command-line option:

VMiX Option	*Description*
8086	Forces VMiX to start in 8086 mode. Enables you to allocate up to 224K for each task.
286	Forces VMiX to start in 286 mode. Loads VMiX in extended memory, and enables you to use Extended memory to swap tasks.
LOW	Stops VMiX from relocating its code into extended memory.
HIGH	Forces VMiX to relocate its code into extended memory.
QUIET	Stops VMiX from displaying startup information and from changing the look of the screen.
TEXT	Prevents you from loading graphics fonts in VMiX while in 8086 mode. This option saves memory.

continues

VMiX Option	*Description*
EMS=eeee	Indicates the EMS memory segment, indicated by *eeee* where VMiX can load its code. Using with an 8086 or 286 computer requires an EMS memory board installed in the computer. When in 386 mode, using EMS=FFFF disables using EMS memory so you can load more VMiX code into upper memory.
EXCLUDE=xxxx	Indicates a block of XMS memory to exclude from VMiX's use, starting at the beginning of extended memory. Indicate the number of 1K blocks with *xxxx*.
/vmixcommand	Enables you to specify one or more VMiX commands to perform after starting VMiX. You may specify more than one command by preceding each with the forward-slash (/).

After you start the program, VMiX begins displaying information on the screen. This information indicates that VMiX is starting, and tells you each of the processes it performs as it starts. When VMiX finally starts, you see a command line prompt on the screen, as shown in figure 8.1.

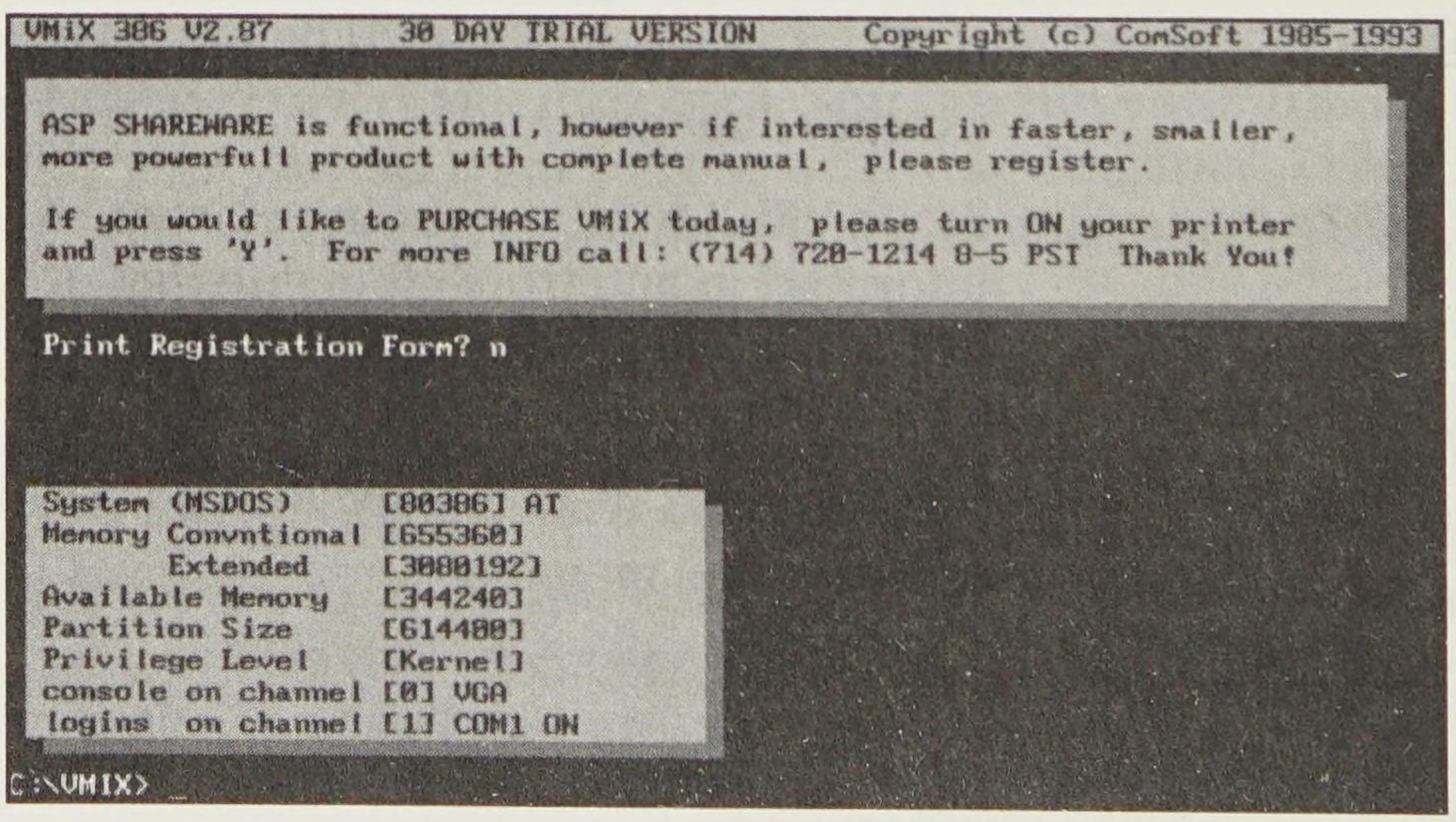


Fig. 8.1
The VMiX startup screen.

This screen is called the *console*. From the console, you can issue VMiX commands, besides DOS commands. For example, if you type **DO /RSHELL** and press Enter, you start another VMiX task. (See fig. 8.2.) After you log in by typing a name of up to eight characters, you see the VMiX command line. From the new task command line, you can start other DOS commands, VMiX commands, or applications. To exit the new task, type **QUIT** and press Enter. You return to the VMiX console screen.

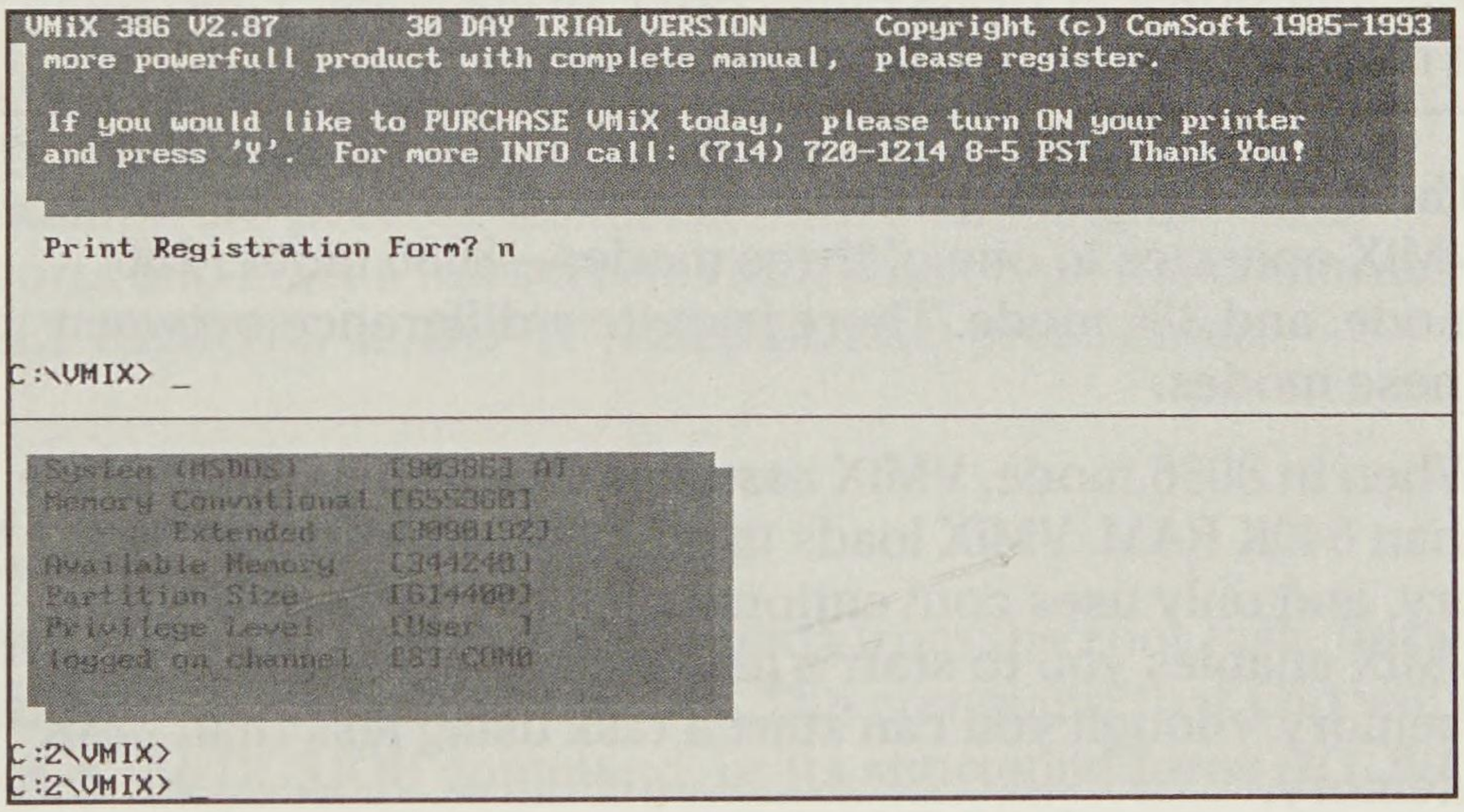


Fig. 8.2
VMiX after starting a new task.

NOTE

When you start VMiX on a 386 computer that has extended memory, VMiX loads part of itself into XMS memory. However, VMiX still uses about 125K of conventional memory. You can decrease the amount of conventional memory that VMiX uses to 38K by loading VMiX into upper memory blocks (the area in memory between 640K and 1M that generally is unused). The following command loads VMiX into UMBS on a 386 computer:

VM_BOOT EMS=FFFF /LOADUP VM_BOOT

LOADUP is a utility that comes with VMiX that loads TSR programs and VMiX into upper memory blocks.

You don't have to do anything special to your programs to use them with VMiX. If the program runs at the DOS command line,

it will run just fine in VMiX. You can start applications from the console. However, instead of starting applications from the console, start another task. Then start the application from the task. Keep the console free for issuing VMiX commands. Always start programs from other tasks.

When you are finished using VMiX, close all applications and tasks. Then, from the original console command line, type **QUIT** and press Enter. VMiX displays a message on-screen telling you it will shut down in 10 seconds. After 10 seconds, VMiX quits and returns you to the DOS command line.

The VMiX Operating Modes

VMiX operates in one of three modes—8086 mode, 286 mode, and 386 mode. There is quite a difference between these modes.

When in 8086 mode, VMiX assumes that you have no more than 640K RAM. VMiX loads itself into conventional memory, and only uses conventional memory for multitasking. VMiX enables you to start a new task using up to 224K of memory, though you can start a task using less than 224K memory.

286 mode provides more memory relief than does 8086 mode. If you have extended memory, and run VMiX in 286 mode, VMiX loads part of itself into XMS extended memory. Loading part of VMiX into XMS memory frees up more conventional memory for you to run programs. In 286 mode, you are not limited to 224K tasks; you can start a task as large as you have conventional memory for. You can swap tasks to XMS memory, enabling you to start more than one task at a time.

386 mode is the ultimate mode to run VMiX in. With 386 mode, VMiX operates the computer in virtual 386 mode. Each task you run works just as if it had its own computer. Each task is completely protected from the other. Each task has virtualized video memory, keeping one program's video separate from another—you can run two graphics programs at once without one program affecting the other.

OPERATING VMiX

When you first start VMiX, you work in the console or root task. This is the main task where you can issue VMiX commands and DOS commands. From the console, you can change the way that VMiX works. For example, you can control memory allocation, change the video attributes, and control remote connections that others use to log into your computer, besides starting other tasks.

TIP

You can enter more than one VMiX command at a time by separating each command with the forward slash. For example, to prevent terminals from accessing the serial ports and start a full-screen VMiX shell, type the command **SET REMOTE -x /DO -W /RSHELL** then press Enter.

STARTING PROCESSES

There are many different commands from the root task, but you probably will use only a handful. The command that you will use most is the DOSJOB command, or its shortened form, DO. With the DO command, you start new VMiX tasks. For example, typing **DO /RSHELL** enables you to start another command prompt. From this command prompt, you can issue DOS commands, VMiX commands, or start an application.

When you use the DOSJOB command, VMiX starts the task in a window on the screen. You can add the -W option to start a full screen task rather than a windowed task. For example, to start another VMiX command line in full-screen mode, type **DO -W /RSHELL** and press Enter.

When you start RSHELL, you are starting a new process that inherits the settings of the root shell. When RSHELL starts, you are prompted to enter a name with the prompt `login:`. Type a name of up to eight characters and press Enter. Then, you view a prompt. From the prompt, start a program, or run a command. When you are finished with the process, type **QUIT** and press Enter. You return to root shell.

VMiX assigns each new process a process number. VMiX manages up to eight processes at a time.

When you have several processes running, you can switch from process to process. Do this by pressing Shift-Tab. When you press Shift-Tab, the next process in line becomes the active process. That means you view the process window or screen and can affect the process from the keyboard. Continuing to press Shift-Tab takes you from process to process in a *round-robin* fashion. (Where in the world did the term "round-robin" come from anyway?)

VMiX enables multitasking up to eight different processes. However, VMiX starts 2 processes itself. So you actually can start six additional processes or tasks. What does that mean to you? This: You can start and run six different programs with VMiX.

> VMiX does not load a new copy of DOS for each task that you start. Rather, VMiX only uses one copy of DOS. That means each process has the full amount of memory allocated to the process—you do not have to subtract DOS allocation from each new process.

QUITTING A PROCESS

When you finish using a task, you should close the task by using the QUIT command. Press Shift-Tab to make the task you want to quit the current task. From the command prompt, type **QUIT** and press Enter. VMiX quits the task and makes the previous task the current task.

You can quit a task in another way. The KILL command enables you to stop a task other than the current task. However, in order to use the KILL command, you must be in the console—the root shell.

To use the Kill command, switch to the console shell, type **KILL** and press Enter. When you see `process id`, type the process number and press Enter. VMiX removes the process from

memory. Before killing a process, ensure that no one is using the process. Generally, you only kill a process that the user cannot quit.

CONTROLLING REMOTE ACCESS

A powerful feature of VMiX, one that few other multitasking environments offer, is that it is a multiuser environment. You can attach a computer terminal, or a computer running a terminal program, to a serial port on the VMiX computer, and operate programs from the VMiX computer.

Suppose you are running VMiX on a 386 computer with two serial ports. You can attach a *remote* computer (one running a terminal program such as Procomm, Telex, or another communications program) to each serial port, and run a program from each remote computer. You can, for example, run a database program from each remote computer, enabling each computer user (including a user on the VMiX computer) to update a different portion of the database. All the programs are actually running on the 386 computer. However, each program is operated from the remote computer attached to the serial port.

From the VMiX computer, you can control how each computer attaches to the VMiX computer. You can select the serial ports that you use for connection and the serial port settings. You control remote computer connections using the **SET REMOTE** and the **SET BAUD** command.

By default, when you start VMiX, COM1 is the default remote communications port that you can attach a terminal to. You may be using COM1 for another purpose, such as a modem or a mouse. In this case, you attach the remote computer to another serial port. However, you must indicate to VMiX the serial ports used for remote computer. To indicate using COM2 for a remote connection, issue the VMiX command **SET REMOTE -l2**. (You must use a lowercase l with the SET REMOTE command.) Notice that -l2 indicates the serial port to configure. Likewise, setting COM3 as your remote serial port requires the command **SET REMOTE -l3**.

NOTE

To indicate that VMiX check all serial ports, use the command **SET REMOTE -a**. Be sure to set the correct baud rate for each serial port, however.

The default settings for the serial port are 9600 baud, no parity, 8 data bits, and 1 stop bit. That means if you attach a remote computer to your VMiX computer, you must set the remote computer's serial port settings to match the VMiX serial port. To change the speed of a serial port, COM3 for example, to 19200 baud from 9600 baud, type the command **SET BAUD -c3 19200** and press Enter. In this case, -c3 indicates the serial port to set.

You can change more than the speed of the serial port with the SET BAUD command. To change COM3 to 19200 baud, even parity, 7 data bits, and 2 stop bits, enter the command **SET BAUD -c3 19200 E 7 2** and press Enter.

Several types of terminals have been developed over the years. VMiX supports seven different terminal types. You must select the terminal type—whether you attach an actual terminal to the VMiX computer or attach a computer that is running a terminal emulation program.

In order to select the type of terminal, type the command **SET TERMINAL -t** and press Enter. A list of terminal types appears on the screen, and you are prompted `terminal type`. Type the number of the terminal and press Enter. The following table lists the terminal types that you may select from. Type 5, ANSI, is the default:

Terminal Code	*Terminal Type*
0	TV1920
1	Adm3A
2	VT-52
3	Zenith H19
4	VT-100
5	ANSI
9	tty

You can select the terminal type from the command line without waiting for a prompt. Use the command shown above, including the terminal code. For example, to select the VT-100 terminal type, type the command **SET TERMINAL -t4** and press Enter.

You can use two other VMiX commands to control VMiX's use of serial ports. When you run VMiX, it looks to each communications port to see if a remote computer needs servicing. However, if you use a communications program and a modem with one communications port, VMiX may actually interfere with communications. You can control how VMiX manages the communications port by entering the command **SET REMOTE -i**.

Suppose you only want to use VMiX as a multitasking environment; you don't want to attach remote computers to the VMiX system. Issue the command **SET REMOTE -x** and press Enter to shut off remote computer servicing to the serial ports altogether. This command does not prevent you from using the serial ports for a communications program or for a printer attached to the serial ports. Rather, this command disables remote computers logging into the VMiX computer.

So how does a remote user log into VMiX? Simple. Plug a serial *null-modem* cable (null-modem cables are discussed in Chapter 4, "Enhancing Communications with DOS") from the remote computer's serial port to the VMiX computer's serial port. Start the terminal emulation program on the remote computer—Procomm, for example. Set the VMiX computer and the remote computer so they both have the same settings—terminal emulation, baud rate, and the like. Then, press the Enter key a couple of times on the keyboard. When you see the `login` prompt on the remote computer, type a user name and press Enter. You're off and running.

CONTROLLING THE SCREEN

VMiX is not a graphical user interface like Windows. Rather, it is a text-oriented environment. However, you still can control the screen by changing colors, selecting the video mode, and sizing windows.

You can control the colors of the entire screen, or of individual windows. To control the colors of the entire screen, use the command **SET VIDEO -c 0X*bf*** where ***b*** is the background color and ***f*** is the foreground color. To set the colors for the current window, use the command **SET VIDEO -C 0X*bf***. Notice that the command for setting window colors is similar to that for setting screen colors. The difference is the **-c** vs **-C**. The following table lists the color codes and colors:

Background colors

Color Code	*Color*
0	Black
1	Blue
2	Green
3	Cyan
4	Red
5	Magenta
6	Brown
7	White

Foreground colors

Color Code	*Color*
0	Black
1	Blue
2	Green
3	Cyan
4	Red
5	Magenta
6	Brown

Color Code	*Color*
7	White
8	Gray
9	Light Blue
A	Light Green
B	Light Cyan
C	Light Red
D	Light Magenta
E	Yellow
F	Bright White

To change the screen color to Blue on White, type the command **SET VIDEO -c 0X71** and press Enter. To set the screen color to Yellow on Red (Yech!), enter the command **SET VIDEO -c 0X4E**. To set the current window to the colors Light Cyan on Blue, type the command **SET VIDEO -C 0X1B** and press Enter.

When you start a new process, you can control the size of the window that the process uses on the screen. Do this with the **SET VIDEO -w** command. To size the window, you indicate the top corner of the window and the bottom corner. The screen is set up with 25 rows (labelled 0 - 24) and 80 columns (labelled 0 - 79). The actual syntax to set the window size is

SET VIDEO -w *toprow, topcol, botrow, botcol*

For example, suppose you want to size a process from row 2, column 5 to row 20, column 60. First, press Shift-Tab to activate the process. Then, type the command **SET VIDEO -w 2,5,20,60** and press Enter. The screen appears like the one in figure 8.3.

You can set VMiX to automatically partition windows, rather than setting window size yourself. Type the command **SET VIDEO -W** and press Enter. (Yes, that is a capital W.)

At the bottom of the screen, you can display the VMiX status line. This line displays the current time, the current console

channel and task name, the current video mode, and the status of the keyboard toggle keys Ins, Caps Lock, Num Lock, and Scroll Lock. You can toggle the status line on and off by typing the command **SET VIDEO -s** and pressing Enter.

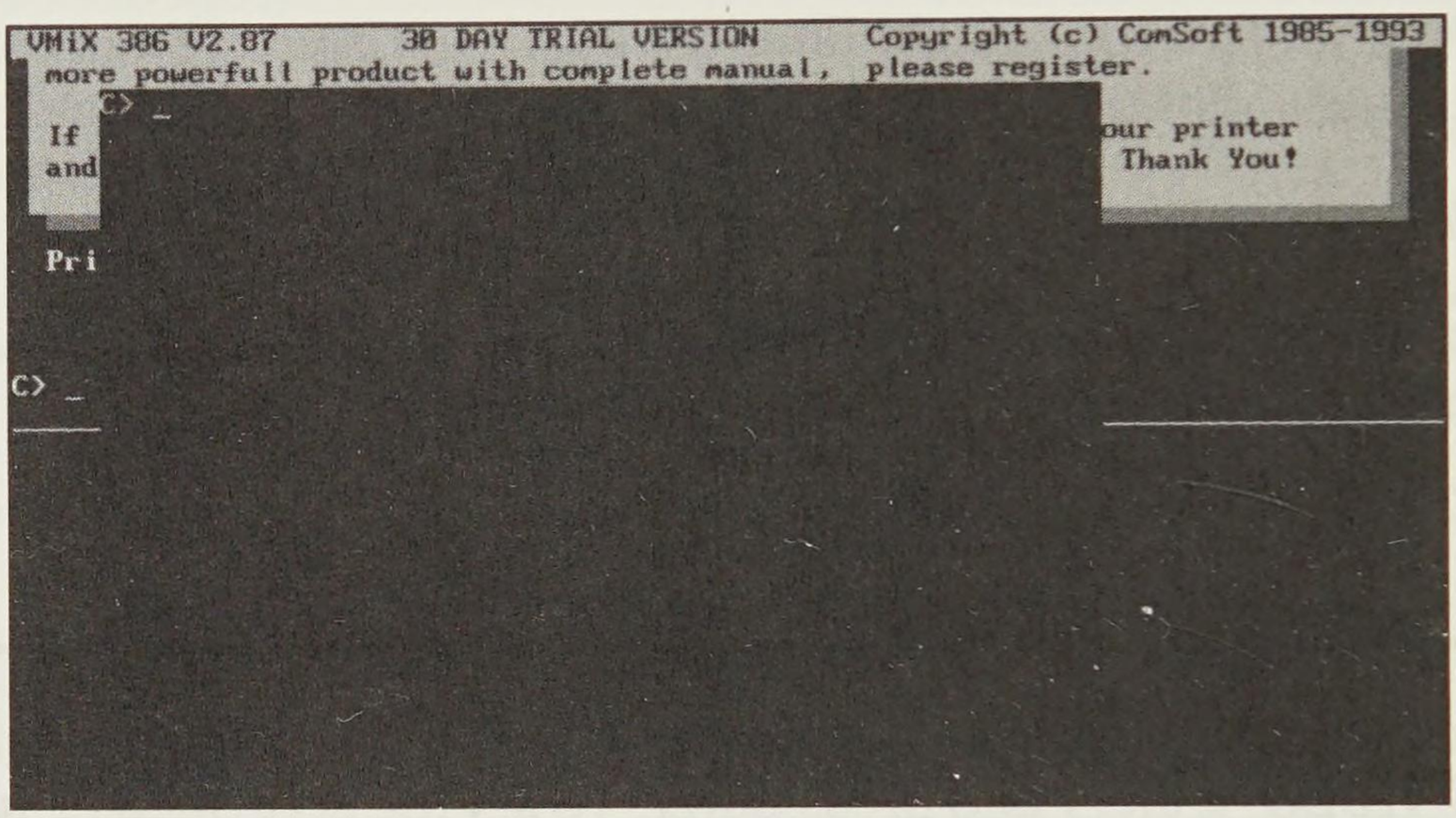


Fig. 8.3
Sizing a process window over an application running in the background.

VMiX enables you to set the video mode that the screen displays in. For example, you can display 80×25 text mode or 640×480 VGA graphics mode. You change the video screen mode by using the command **SET VIDEO -m**. This command displays a list of video modes and the prompt `video mode`. Type one of the video mode codes, listed in the following table, and press Enter.

Mode Code	*Video Mode*	*Text/Graphics*
0	40×25Black and White	Text
1	40×25 Color	Text
2	80×25 Black and White	Text
3	80×25 Color	Text
4	320×200 Black and White	Graphics
5	320×200 Color	Graphics
6	640×200 Black and White	Graphics

Video Mode Code	*Video Mode*	*Text/Graphics*
7	Monochrome	Text
13	320×200	Graphics
14	620×200	Graphics
15	640×350 Black and White	Graphics
16	640×350 Color	Graphics
17	640×480 Black and White	Graphics
18	640×480 Color	Graphics

As with other commands, there is a quicker way to select the video mode to use. Add the video mode code shown in the table above to the command. For example, to choose video mode 18, type the command **SET VIDEO -m 18** and press Enter.

In 386 mode, VMiX controls each application's use of the screen. VMiX virtualizes the screen; that is, VMiX provides each application with its own video memory. That way, one program's video information does not affect, or *bleed over*, the video information of another program. VMiX's 8086 mode and 286 mode do not have that feature, however.

What VMiX does enable you to do is select a different video page to store video information on. Each video page is a different part of video memory. To change the video page, type the command **SET VIDEO -p** and press Enter. At the prompt `video page`, enter the video page to use, as shown in the following table:

Video Page Code	*Video Page*
0	Page 0
1	Page 1
2	Page 2
3	Page 3

As with the other commands you have experienced with VMiX, type the video page code with the command. For example, enter the command **SET VIDEO -p 1** to choose video page 1.

CAUTION

Not all programs enable you to select the video page to use. Before you start working with your programs, experiment to see which programs work on alternate video pages.

LOADING PROGRAMS INTO UMBS

VMiX in 386 mode enables you to load drivers and TSRs into upper memory blocks, saving conventional memory for your applications' use. You can even load VMiX into UMBs. Use the program LOADUP.EXE to load TSRs and VMiX into UMBs.

The syntax for LOADUP is as follows:

LOADUP */SEGMENT=ssss /SIZE=zzzz* **filename**.*ext*

In the syntax, **filename**.*ext* is the TSR to load into UMBs. */SEGMENT* is the area of UMB in which to load the TSR, indicated by *ssss*, where *ssss* is the memory segment in which to load the TSR. */SIZE* is the amount of memory to use indicated by *zzzz*, where *zzzz* is the number of 1K blocks. For example, to load a TSR called READ, type the command **LOADUP READ** and press Enter.

VMiX loads a small section of its code into the upper memory blocks. However, you can load more of VMiX into upper memory using LOADUP. Enter the following command from the DOS command line to start VMiX and load VMiX into the upper memory blocks:

VM_BOOT EMS=FFFF /LOADUP VM_BOOT

In this syntax line, the option **EMS=FFFF** indicates that VMiX should not initially load any of itself into UMBs. However, the subsequent command automatically loads a second copy of VMiX into UMBs, and shuts down the first copy.

ISSUING COMMANDS IN THE VMIX.INI FILE

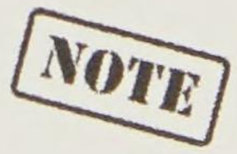
There is a text file in the VMiX directory that you can use to configure VMiX when you start VMiX from the DOS command line. The file is called VMIX.INI. This file contains three areas: access passwords, remote startup, and startup. The following list is a sample VMIX.INI file:

```
;--- VMiX.INI  SAMPLE CONFIGURATION FILE  (Ver 2.87)
;--- This file is optional and, if found, it will be processed by
VM_BOOT.EXE
;--- (everything after an ';' is a comment)
[access passwords]
;----------------- sample access passwords
guest
superuser
password

[remote startup]
VSETUP.EXE
rshell

[startup]
se re -l1 vsetup/se re -a/se ba -c1 24/se ba -c2 24/se ba -c3 24/se
ba -c4 24
```

Each time you start VMiX or log into VMiX from a remote computer, VMiX looks for this file. If VMiX finds this file, it begins processing the file. When you are starting VMiX, VMiX processes the commands in the startup section. In the sample file above, VMiX enables four serial ports for remote terminals. You can place any valid VMiX commands in the startup section that you like. You can gang commands on a line using the forward slash, or place each VMiX command on a separate line in the file.



The first time VMiX reads VMIX.INI, VMiX searches for the access passwords section. If you have passwords listed, VMiX copies VMIX.INI to VMIX.BAK, then encrypts the

continues

continued

passwords, renaming the section to encrypted passwords, and resaving the file as VMIX.INI. You can then delete VMIX.BAK to keep others from determining your passwords, but be sure to keep a copy of the passwords somewhere in case you forget them.

The first two sections, access passwords and remote startup, contain commands for remote computers using VMiX. The access passwords section contains valid passwords that remote users must type to access VMiX. When a remote terminal attaches to VMiX, VMiX processes the commands in the remote startup section. You can place any valid VMiX commands in this section, including programs to start for the remote user.

USING VMiX WITH VW MENUS

While you are running VMiX, you can issue VMiX commands from the VMiX command line. Additionally, you can issue more than one command at a time by separating each with the forward slash (/). You can issue commands another way, however—using the VMiX interactive menu system. This system, called VW, is discussed in the next section. (I'm not sure what VW actually stands for. But who knows—maybe it will bring back fond memories of driving that little round car to college.)

You use VW, as you do the VMiX console, to control VMiX. From the menu, for example, you can start processes, set up serial ports for remote access, and change the video mode.

You can start VW after you load VMiX. Or, you can start VW *as* you start VMiX. To start VW while starting VMiX, type the following at the command line and press Enter:

VM_BOOT VW

Figure 8.4 shows the VW menu when started in VMiX.

To use the menu, select a menu with the arrow keys, then press the Down-arrow key to pull the menu down; or you can press a menu's capitalized letter. For example, to pull down the Options

menu, press the right-arrow key to highlight Options and press the down-arrow key, or press **O**. Figure 8.5 shows the Options menu pulled down.

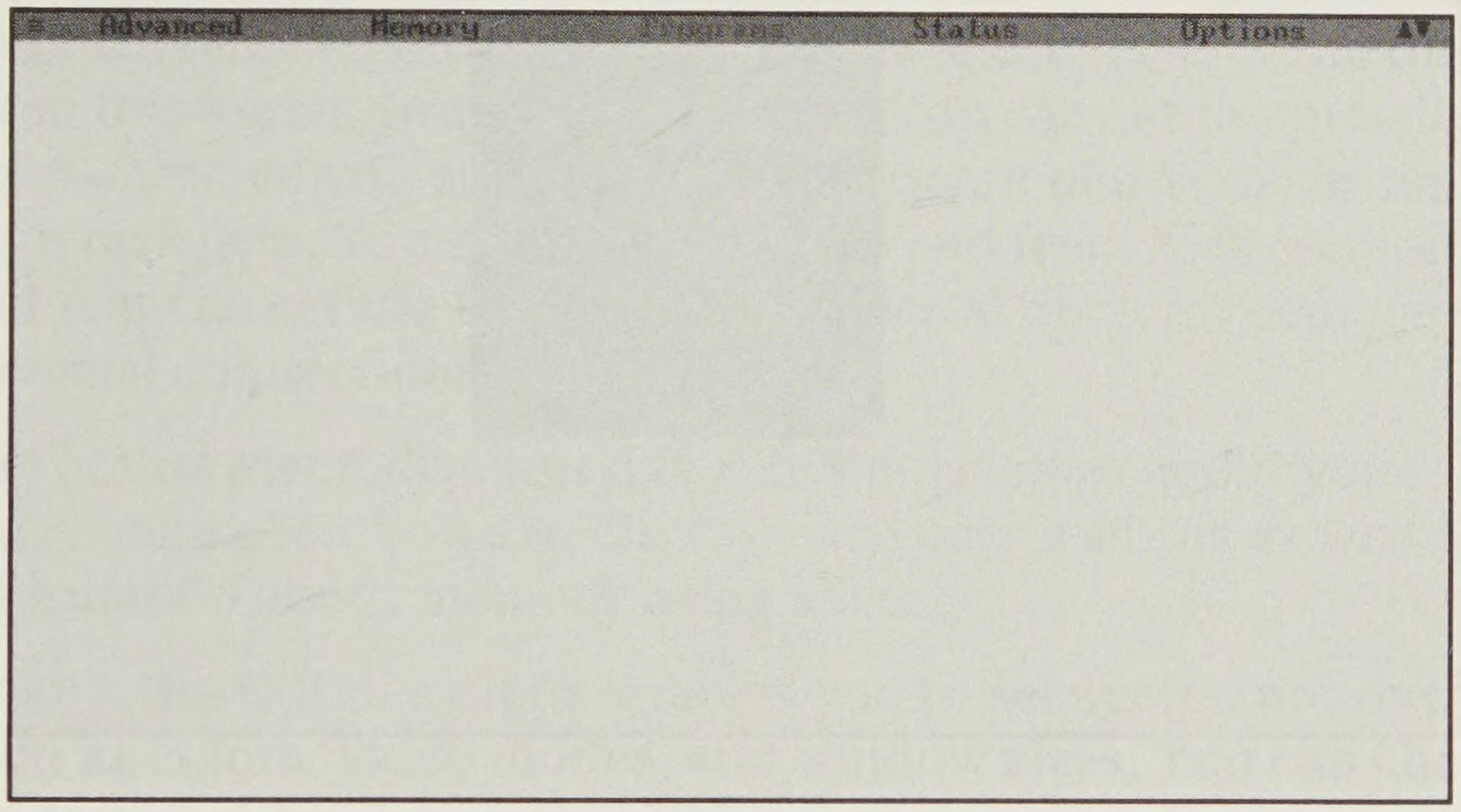


Fig. 8.4
Use the VW menu to control VMiX.

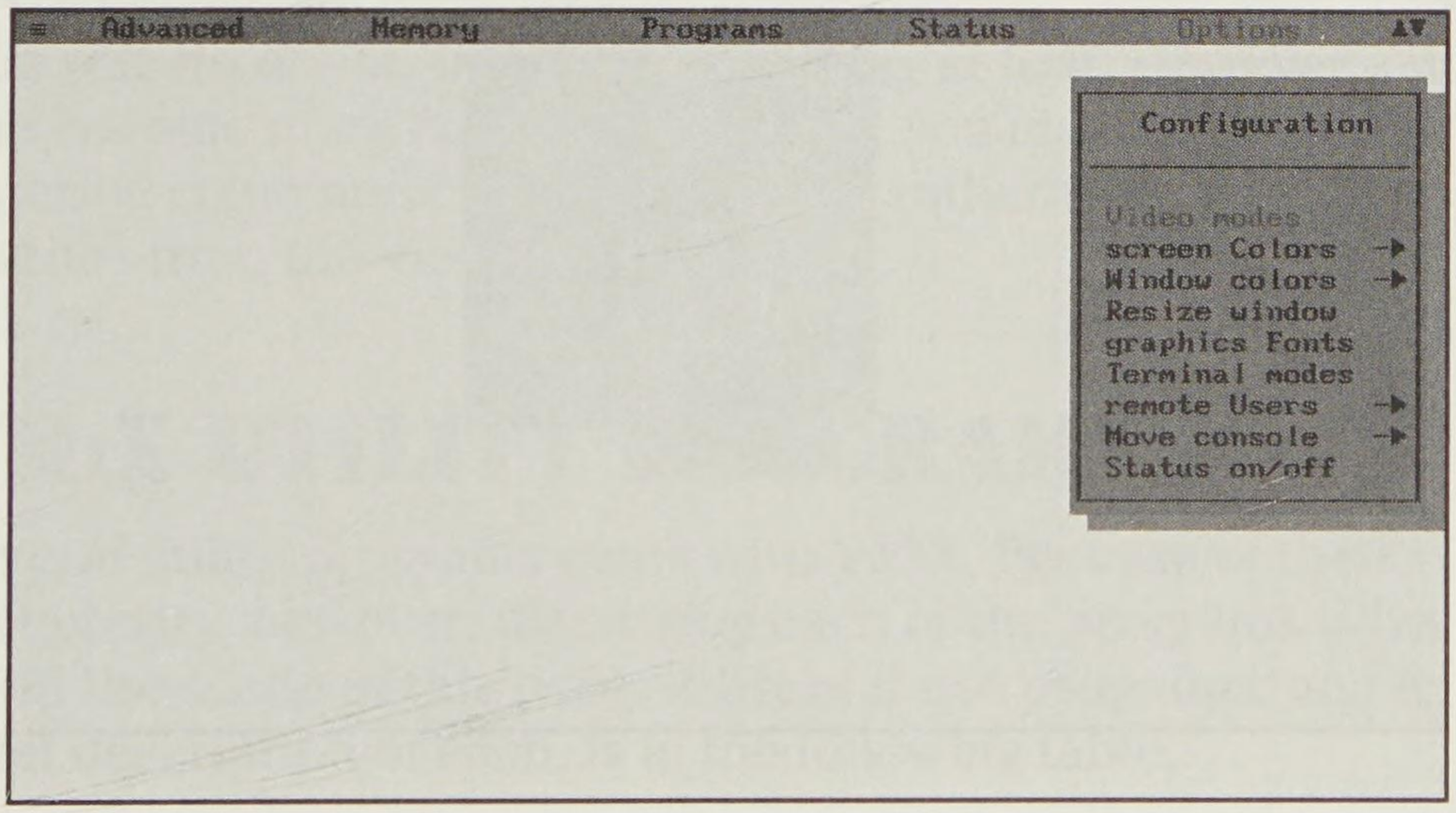


Fig. 8.5
Pulling down the Options menu.

You can start a new process from the **P**rograms menu. Pull down the **P**rograms menu, then use the down-arrow key to select start **V**mix shell (see fig. 8.6). Notice that this option has an arrow adjacent to the option. That means the menu item contains a submenu. When you choose start **V**mix shell, its submenu displays, as shown in figure 8.7. Select **W**indow to start a new VMiX process in a window, **F**ull to start a new VMiX

process running full-screen, or **R**emote to connect a remote terminal to another computer running VMiX.

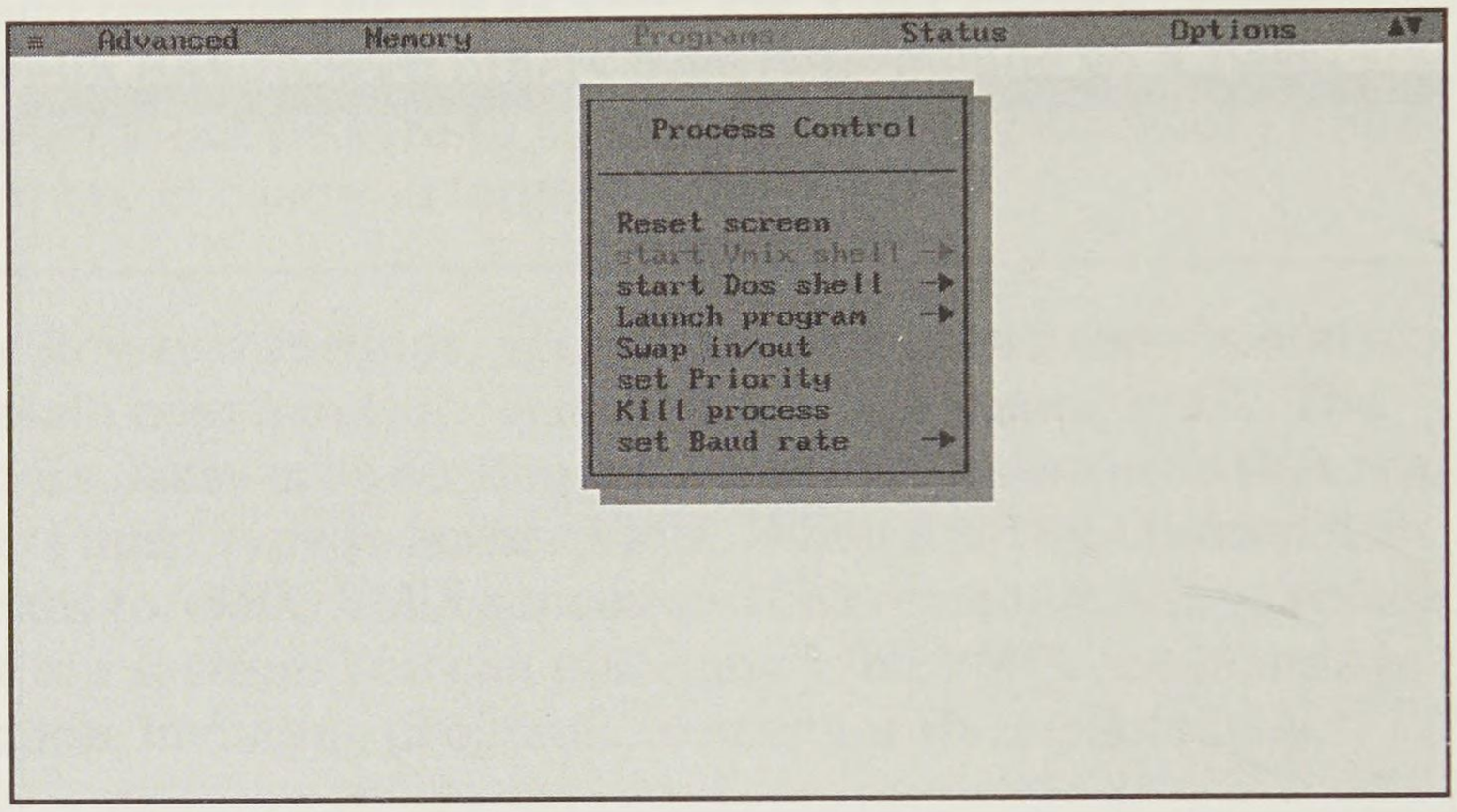


Fig. 8.6
The Programs menu.

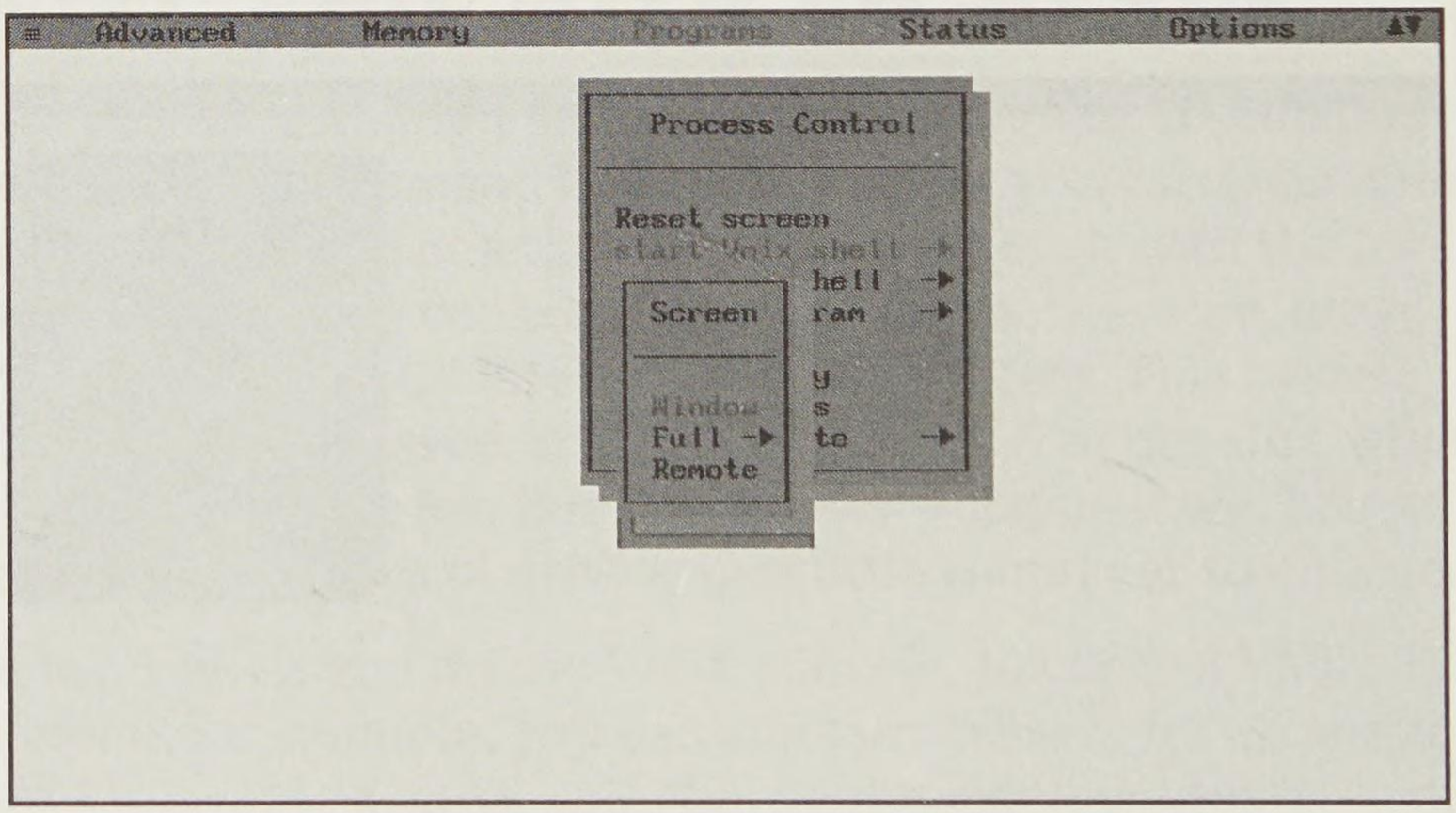


Fig. 8.7
The start Vmix shell submenu.

If you choose **F**ull from the start VMiX shell submenu, another submenu displays on the screen. This submenu enables you to select the screen mode for the program you are starting. Select the correct video mode and press Enter. VMiX starts a new process.

What are the different menus for? The Advanced menu enables you to perform tasks such as passing console control to a

remote terminal or kill processes. The Memory menu enables you to manage memory such as setting the program space and enabling use of video memory when you are using text-only programs.

The Programs menu is the menu you will use most of the time. From this menu, you start new tasks. You can set the priority of tasks—that is, you can indicate the amount of processor time each task gets. You can swap tasks to and from XMS memory and you can set the baud rate of the serial ports for remote terminal connections.

The Status menu displays different information about your VMiX computer. You can display each user's status as well as information about memory being used.

Finally, the Options menu enables you to set screen options such as colors, video modes, and window sizes. You can change the terminal emulation that VMiX performs and display or shut off the status bar at the bottom of the screen.

You will, no doubt, find VW very handy at first. However, once you become more familiar with VMiX, you may find yourself entering commands from the console rather than selecting options from the menus.

VMiX UTILITY PROGRAMS

Several utility programs come with VMiX. Because of their complexity, however, discussing each of the programs is beyond the scope of this book. A list of these programs, and a brief description of each, is in the following table:

Utility	*Description*
PS.EXE	Displays the status of all running processes.
DSTAT.EXE	Displays information about each VMiX object and channel.

continues

Utility	*Description*
DPATH.EXE	Enables you to set a search path for data files. Compliments the DOS PATH command.
LOADUP.EXE	Enables you to load TSRs into upper memory blocks.
MONITOR.EXE	Monitors the actions of each process or the entire system.
OUTSIDE.EXE	A utility for bulletin board system (BBS) users that enables you to display a DOS prompt from the bulletin board. This redirects DOS information normally displayed on the screen to the communications port.
VSETUP.EXE	A utility that processes the VMiX configuration file.
VT.EXE	A terminal emulation program that enables you to attach a VMiX computer as a remote computer to another VMiX computer.

For complete information about the utilities that come with VMiX, and some advanced commands not discussed in this chapter, see the file TECHREF.DOC in the VMiX directory (\DOSSTUFF\VMIX).

SUMMARY

Multitasking is a great thing. You can start applications and switch back and forth between them. No more wasted time while you wait for one program to process. And how about that multiuser feature? Even if you never have another person attach to your computer, you still might find multiuser capabilities useful. Just think, you can attach that crazy old XT to your

spanking new 486. Then you can give that XT 486 some power. You can run an application from the 486 screen and keyboard, and run another application from the XT, using the 486's processor, memory, and speed.

Did Microsoft forget multitasking? Maybe. Do you have to do without? Not anymore.

INDEX

SYMBOLS

\DSMF\PORTCALL directory, PortaCall, 70
286 mode, VMiX, 176
386 processor mode, VMiX, 171, 176, 186
8086 processor mode, VMiX, 171, 176

A

/AC command parameters, SCROLLit colors, 26
Advanced menu, VMiX, 190
altering scripts, 153-154
/AM command parameters, SCROLLit colors, 26
attributes, color, (EXC program), 160-161
AUTOEXEC.BAT file
  DOSMAX 2.1, 118-119
  ShowMe Saver addition, 11
automating operations, 139-144

B

background colors
  attributes, SHOW command (EXC program), 160
  VMiX, 182
batch files, 100, 140-141
BEEP command (EXC program), 154-155
binary files, EXAMINE.COM utility, 135-137
boot sectors, reading with Con>Format, 34
Bromfield Software, SCROLLit, 19
BSECTR_1.D1 file, 133
Budget Software Company, Spooling/Despooling Software, 57

C

c parameter
  CURSOR command (EXC program), 164
  SHOWAT command (EXC program), 166
C:\BOOT directory, DOSMAX utilities, 119-120
C:\DSMF\DCF directory, Disk Copy Fast, 36
C:\SHOW directory, ShowMe Saver installation, 10
cables, null-modem, logging into VMiX, 181
Campbell, Guy, Find Directory, 51-52
CAPTURE.BAT file, 134-135
CASE OFF command (EXC program), 149, 164
CASE ON command (EXC program), 164
CFCONFIG command (Con>Format), 33
CFCONFIG file, 33-34
changing directories with ZIP File Transfer, 77

CHKDSK.EXE program, 91
CLEAR command
(EXC program), 164
CMOS command
(CMOS.EXE utility), 134
CMOS setup, storing in files, 133-134
CMOSRAM.DAT file, 133
color codes, SCROLLit, 26
colors
background, VMiX, 182
changing, SCROLLit, 26-27
defaults, SCROLLit, 26-27
foreground, VMiX, 182-183
SCROLLit, 26
command parameters, /AC, /AM, SCROLLit colors, 26
command keys, SCROLLit, 20-21
command line
DOS, using Patha, 100
loading programs with Fast Load, 127
options
Disk Copy Fast, 41-44
VMiX, 173
running SHREDFILE, 49
SCROLLit parameters, 23-27
ZIP File Transfer, 81-84
command parameters
/H=hotkey, 25
ShowMe Saver, 15-17
Command-Date (RebelWare), 52-53
\DSMF\CMDDATE
directory, 52
commands
BEEP (EXC program), 154-155
CAPTURE, 135
CASE OFF (EXC program), 149, 164
CASE ON (EXC program), 164
CFCONFIG (Con>Format), 33
CLEAR (EXC program), 164
CMOS, 134
Compare
(ZIP File Transfer), 80
CONFMT (Con>Format), 30
CONFMT R (Con>Format), 32
COPY, 35
CURSOR (EXC program), 164
DCF, 36
DEL, 44
DELAY (EXC program), 155
Delete (ZIP File Transfer), 81
Directory
(ZIP File Transfer), 78
DISKCOPY, 35
DO (VMiX), 177-178
DOSJOB (VMiX), 177-178
ERASE, 44
EXC, 154-167
FD SPRDSHT
(Find Directory), 51
Fetch (ZIP File Transfer), 79
GETCMOS, 133
GOTO (EXC program), 151-152, 155
HIDEWIN (EXC program), 165
IFN (EXC program), 155
IFY (EXC program), 150, 156-167
KEY (EXC program), 151-152, 156
Kill (VMiX), 178-179
LOADUP READ (VMiX), 186
LOCATE (EXC program), 151, 159
LOOK (EXC program), 159
MEMDUMP /M, 135
ON (EXC program), 157
PATH, 91, 92-95
PATHA /L, long search paths, 100
PAUSE (EXC program), 157
QUIT
EXC program, 153, 157
VMiX, 178-179
READ (EXC program), 165
SCRMAX (EXC program), 165
SCROLLIT /+
(SCROLLit program), 25

SCROLLIT /-
(SCROLLit program), 25
SEARCH (EXC program), 150, 159-160
Send (ZIP File Transfer), 79
SerVer (ZIP File Transfer), 78-79
SET (EXC program), 94, 153
SET BAUD (VMiX), 179
SET REMOTE (VMiX), 179
SET REMOTE -i (VMiX), 181
SET REMOTE -x (VMiX), 181
SET TERMINAL -t (VMiX), 180
SET VIDEO -c (VMiX), 183
SET VIDEO -m (VMiX), 184
SET VIDEO -p (VMIX), 185-186
SETWAIT (EXC program), 157-158
SHOW (EXC program), 160-161
SHOWAT (EXC program), 165-166
SHOWIN (EXC program), 166
SHRED, 45
TICK (EXC program), 158
TIME (EXC program), 158
TIMEOUT (EXC program), 158
TYPE (EXC program), 152, 162
TYPEFILE (EXC program), 162
Unlink (ZIP File Transfer), 81
VM_BOOT
(VMiX program), 172
WAITFOR (EXC program), 152, 163
WAITSCR (EXC program), 163
WINCOLS (EXC program), 166
WINMAX (EXC program), 166
WINROWS
(EXC program), 167
ZIP (ZIP File Transfer), 81
ZIPCFG (ZIP File Transfer), 87
Commercial Software Associates, VMiX, 171
Compare command (ZIP File Transfer), 80
comparing files, ZIP File Transfer, 80-81
computer phone-line monitoring, PortaCall, 70-72
computers, automating operations, 139-141
Comstock, Seth W., SHREDFILE, 45
Con>Format (SYDEX), 30
CFCONFIG command, 33
concurrent diskette-formatting program, 30
configuring, 33-35
CONFMT command, 30
CONFMT R command, 32
\DSMF\CONFMT directory, 30
formatting disks, 31
hot-keys, 31
loading, 30-32
reading boot sectors, 34
registering, 30
Con>Format window, 31
CONFIG.SYS file
DOSMAX 2.1, 118
DOSMAX.EXE placement, 109
configuration example, DOSMAX 2.1, 117-122
configuration information, maintaining, 129-137
configuration menu, ShowMe Saver, 13-15
configuring
Con>Format, 33-35
DOS for VMiX, 171-172
upper memory blocks, DOSMAX utilities, 120-121
VMIX.INI file, 187-188
ZIP File Transfer, 85-89
CONFMT command (Con>Format), 30
CONFMT R command (Con>Format), 32
CONFMT.COM file, 33
CONFMT.EXE file, 30
console, VMiX, 175

controlling screens with VMiX, 181-186
COPY command, 35
copying disks, Disk Copy Fast, 38-40
CRCONFIG file, 33
CURSOR command (EXC program), 164
customizing
screens, SHREDFILE, 50
ShowMe Saver, 13-15

D

/D parameter, ZIP File Transfer, 84
DCF command, 36
DCF Software, Disk Copy Fast, 35-36
DEBUG.EXE utility, 85
defaults
colors, SCROLLit, 26-27
remote communications ports, VMiX, 179
DEL command, 44
DELAY command (EXC program), 155
delayed writes, disabling to use SHREDFILE, 48
Delete command (ZIP File Transfer), 81
deleting files, ZIP File Transfer, 80-81
despooler/monitor programs, SP3DSPMN.EXE, 65-66
despooling
on networks, Monitor, 64-65
files, SP3DSP.EXE program, 63
programs, 61-67
DESQView, EXC compatibility, 147
Dineen, Bill, PortaCall, 71
directories
\DSMF\PORTCALL (PortaCall), 70
C:\BOOT (DOSMAX utilities), 119-120
C:\DSMF\DCF (Disk Copy Fast), 36
C:\SHOW (ShowMe Saver), 10
changing, ZIP File Transfer, 77
\DOSSTUFF\FL
Fast Load, 123
Load Watch, 128
\DOSSTUFF\VMIX (VMiX), 172, 192
\DSMF\CMDDATE (Command-Date), 52
\DSMF\CONFMT (Con>Format), 30
\DSMF\DOSMAX (DOSMAX 2.1), 105
\DSMF\ERROR (ERROR program), 89
\DSMF\FASTLOAD (Fast Load), 123
\DSMF\FINDIR (Find Directory), 51
\DSMF\SCROLLIT (SCROLLit), 19
\DSMF\SHOWME (ShowMe Saver), 10
\DSMF\SHRED (SHREDFILE), 45
\DSMF\SYSCAP (SYS_CAP program), 129
\DSMF\ZIP (ZIP File Transfer), 73
\DSMK\SP3 (Spooling/Despooling Software program), 57
locating, 51-52
SCROLLIT.DOC (SCROLLit), 25
TECHREF.DOC (VMiX), 192
Directory command (ZIP File Transfer), 78
disabling Fast Load, 127-128
disk caching, SmartDrive 4.0, 118, 122

Disk Copy Fast (DCF Software), 35-41
  C:\DSMF\DCF directory, 36
  command-line options, 41-44
disk space, shredding with SHREDFILE, 49-50
DISKCOPY command, 35
DISKDATA.COM utility, 132-133
disks
  copying, Disk Copy Fast, 38-40
  formatting, Con>Format, 31
displaying
  files, SHREDFILE, 47
  text, EXC commands, 159-163
DO command (VMiX), 177-178
DOS (Disk Operating System), 29
  command line, using Patha, 100
  command searching, 92-93
  configuring for VMiX, 171-172
  error messages, 89-90
DOS 6, memory configuration/management, 104-122
DOSJOB command (VMiX), 177-178
DOSMAX 2.1
  AUTOEXEC.BAT file, 118-119
  CONFIG.SYS file, 118
  DOS memory loading, 105
  \DSMF\DOSMAX directory, 105
  memory management, 106-117
  registering, 104
DOSMAX utilities
  C:\BOOT directory, 119-120
  upper memory block configuration, 120-121
DOSMAX.EXE program, 106-109
\DOSSTUFF\FL directory
  Fast Load, 123
  Load Watch, 128
\DOSSTUFF\VMIX directory (VMiX), 172, 192
DoubleSpace (disk-compression utility), 118
\DSMF\CMDDATE directory (Command-Date), 52
\DSMF\CONFMT directory (Con>Format), 30
\DSMF\DOSMAX directory (DOSMAX 2.1), 105
\DSMF\ERROR directory (ERROR program), 89
\DSMF\FASTLOAD directory (Fast Load), 123
\DSMF\FINDIR directory (Find Directory), 51
\DSMF\SCROLLIT directory (SCROLLIT), 19
\DSMF\SHOWME directory (ShowMe Saver), 10
\DSMF\SHRED directory (SHREDFILE), 45
\DSMF\SYSCAP directory (SYS_CAP program), 129
\DSMF\ZIP directory (ZIP File Transfer), 73
\DSMK\SP3 directory (Spooling/Despooling Software program), 57

E

/E parameter, ShowMe Saver, XMS memory residence, 17
/E switch, Patha, 96
ECX program, IFY command, 156-167
editing paths, Patha (HGB Enterprises), 95-100
electronic shredders, 45
eliminating files, 44-50
EMM386.EXE program, expanded memory, 105
EMS (expanded memory), ShowMe Saver, 13, 17
enabling Fast Load, SETFL.COM program, 128
END subroutine, EXC program, 152-153

ENVIMAX 2.1, registering, 104
ENVIMAX.COM program, 113-115
environments
  multiuser, VMiX, 179-181
  variables, 99
ERASE command, 44
error messages, DOS, 89-90
ERROR program (SL-EASY Software), 89
EXAMINE.COM utility
  Mark/Modify menu, 136
  Options menu, 137
  viewing binary files, 135-137
EXC 2.6a program (Parity Solutions), 142
EXC program
  automating tasks, 141-144
  BEEP command, 154-155
  CASE OFF command, 149, 164
  CASE ON command, 164
  CLEAR command, 164
  commands, 154-167
  CURSOR command, 164
  DELAY command, 155
  DESQView compatibility, 147
  END subroutine, 152-153
  GOTO command, 151-152, 155
  HIDEWIN command, 165
  IFN command, 155
  IFY command, 150, 156
  KEY command, 151-152, 156
  LOCATE command, 151, 159
  LOOK command, 159
  ON command, 157
  PAUSE command, 157
  QUIT command, 153, 157
  READ command, 165
  REPLACE subroutine, 151-152
  SCR file extensions, 143
  screen buffers, 144
  scripts, 142-143
  SCRMAX command, 165
  SEARCH command, 150, 159-160
  SET command, 153
  SETWAIT command, 157-158
  SHOW command, 160-161
  SHOWAT command, 165-166
  SHOWIN command, 166
  starting, 143
  TICK command, 158
  TIME command, 158
  TIMEOUT command, 158
  TYPE command, 152, 162
  TYPEFILE command, 162
  VMiX compatibility, 147
  WAITFOR command, 152, 163
  WAITSCR command, 163
  WINCOLS command, 166
  WINMAX command, 166
  WINROWS command, 167
  yes/no variable, 150
expanded memory
  EMM386.EXE program, 105
  Fast Load, 123-128
extended memory
  Fast Load, 123-128
  HIMEM.SYS program, 105

F

/F parameter, ShowMe Saver, 17
/F switches, Fast Load, 127
Fast Load 1.25 (Microsystems Software, Inc.)
  disabling, 127-128
  \DOSSTUFF\FL directory, 123
  \DSMF\FASTLOAD directory, 123
expanded memory, 123-128
extended memory, 123-128
loading programs
  from command line, 127
  into memory with /P parameters, 126
SETFL.COM program, 128
starting from AUTOEXEC.BAT file, 123
substituting program data files with /F switches, 127

FD SPRDSHT command
(Find Directory), 51
FD.EXE filem, Find Directory, 52
Fetch command
(ZIP File Transfer), 79
file extensions, SCR
(EXC program), 143
file transfer programs, ZIP File
Transfer, 72-89
files
AUTOEXEC.BAT
DOSMAX 2.1, 118-119
ShowMe Saver addition, 11
starting Fast Load, 123
starting SCROLLit, 19
batch, 100, 140-141
binary, EXAMINE.COM utility,
135-137
BSECTR_1.D1, 133
CAPTURE.BAT, 134-135
CFCONFIG, 33-34
CMOSRAM.DAT, 133
comparing with ZIP File
Transfer, 80-81
CONFIG.SYS
DOSMAX 2.1, 118
DOSMAX.EXE placement,
109
CONFMT.COM, 33
CONFMT.EXE, 30
CRCONFIG, 33
deleting with ZIP File
Transfer, 80-81
despooling with SP3DSP.EXE
program, 63
displaying with
SHREDFILE, 47
eliminating, 44-50
FD.EXE, 52
FL.DAT, 123
grouping by date, 52-54
LWATCH.DAT, 128
MASTBOOT.D1, 133
memory, MEMDUMP.COM
utility, 136
multiple, shredding, 48
program data, substituting
with /F switches
(Fast Load), 127
RAMCAM.COM utility, 132
receiving from ZIP File
Transfer command line, 82
sending
from ZIP File Transfer
command line, 81-82
ZIP File Transfer, 76-77
shredding individual with
SHREDFILE, 47-48
sorting with SHREDFILE, 46-47
spool
SP3PRT.COM program, 67
SP3SPL.EXE program, 61-62
storing CMOS setup, 133-134
tagging to delete with
SHREDFILE, 46
TECHREF.DOC, VMiX
directory, 192
transferring, ZIP File
Transfer, 74-78
VMIX.BAK, VMiX
passwords, 187
VMIX.INI, configuring
VMiX, 187-188
Find Directory (Guy Campbell),
51-52
\DSMF\FINDIR directory, 51
FD SPRDSHT command, 51
FD.EXE file, 52
finding text with EXC
commands, 159-163
FL.DAT file, 123
foreground colors
attributes, SHOW command
(EXC program), 161
VMiX, 182-183
formatting disks with
Con>Format, 31
FREELOAD 2.1, 104
FREELOAD.SYS program, 115-117
Fspec menu, SHREDFILE, 47

G

Gardner, Philip B., 104
Geddes, Leslie E., SYS_CAP program, 129
GETCMOS command, 133
GETCMOS.COM utility, 133-134
GOTO command (EXC program), 151-152, 155
grouping files by date, 52-54
guests, ZIP File Transfer, 76

H

/H=hotkey command parameter, 25
hard disks recording information, DISKDATA.COM program, 132-133
HGB Enterprises, Patha, 95-100
HIDEWIN command (EXC program), 165
HIMEM.SYS program, extended memory management, 105
HMA (High Memory Area), 104
hosts, ZIP File Transfer, 76
hot-keys
  Con>Format, 31
  SCROLLit, 25

I-J

IFN command (EXC program), 155
IFY command (EXC program), 150, 156
information, printing, saving, searching with SCROLLit, 21-23
Information window, Disk Copy Fast, 37-38
interactive use, Disk Copy Fast, 36-41

K

KEY command (EXC program), 151-152, 156
keynames, EXC program, 156
Kill command (VMiX), 178-179

L

Load Watch, 128-129
  \DOSSTUFF\FL directory, 128
  LWATCH.DAT file, 128
loading
  Con>Format, 30-32
  programs into RAM, 122
  TSRs and VMiX into UMBs, LOADUP.EXE program, 186
LOADUP READ command (VMiX), 186
LOADUP utility, VMiX program, 175
LOADUP.EXE program, 186
LOCATE command (EXC program), 151, 159
locating directories, 51-52
long search paths, PATHA /L command, 100
LOOK command (EXC program), 159
LWATCH.DAT file, Load Watch, 128

M

maintaining information
  configuration, 129-137
  screen, 18-27
Mark/Modify menu, EXAMINE.COM utility, 136
MASTBOOT.D1 file, 133
Master Boot Record, DISKDATA.COM utility, 132-133
MEMALOC.EXE program, 137
MEMDUMP /M command, 135
MEMDUMP.COM utility, 135-137

memory
allocation, MEMALOC.EXE program, 137
conservation with ShowMe Saver, 17-18
DOS 6 configuration, 104-122
EXC program requirements, 143-144
expanded
EMM386.EXE program, 105
Fast Load, 123-128
extended
Fast Load, 123-128
HIMEM.SYS program, 105
loading
DOS with DOSMAX 2.1, 105
programs for speed, 122-129
programs with Fast Load /P parameters, 126
management, 104-122
DOS 6, 104
DOSMAX 2.1 program, 106-117
removing Con>Format with CONFMT R command, 32
system, capturing with RAMCAM.COM utility, 132
upper, loading
Con>Format, 31
viewing with MEMDUMP.COM utility, 135-137
Memory menu (VMiX), 191
menus
ShowMe Saver, configuration, 13-15
SHREDFILE
Fspec, 47
Shred, 46
Sort, 47
message display, ShowMe Saver, 9
Meyer, Eric, ZIP File Transfer, 73
Microsystems Software, Inc., Fast Load 1.25, 123
Monitor, spooling/despooling on networks, 64-65
monitoring programs, 128-129
multi-task scripts, 145-147
multiple files, shredding with SHREDFILE, 48
multitasking, VMiX, 169-171, 178
multiuser environment, VMiX, 179-181

N

/N parameter, ShowMe Saver, 18
networks, despooling/spooling with Monitor, 64-65
/NS parameter, Fast Load, 127
null-modem cables, logging into VMiX, 73, 181

O

/O switches, Patha, 99
ON command (EXC program), 157
operations of computer, automating, 139-141
Options menu
Disk Copy Fast, 40
EXAMINE.COM utility, 137
VMiX, 191

P

parameters
/AC command, SCROLLit, 26
/AM command, SCROLLit, 26
c
CURSOR command (EXC program), 164
SHOWAT command (EXC program), 166
command
/H=hotkey, 25
ShowMe Saver, 15-17

/D, ZIP File Transfer, 84
/E, ShowMe Saver, 17
/F, ShowMe Saver, 17
/N, ShowMe Saver, 18
/NS, Fast Load, 127
/P,Fast Load, 126
Patha, 96
PortaCall, 71
/Ppath, ShowMe Saver, 18
r
  CURSOR command (EXC program), 164
  SHOWAT command (EXC program), 166
SCROLLit command line, 23-27
searchpath, 93
SP3DSP.EXE program, 63
SP3SPL.EXE program, 62
/U, ShowMe Saver, 18
ZIP FIle Transfer, 77, 86-87
Parameters menu, ZIP File Transfer, 86
Parity Solutions, EXC 2.6a program, 142
partition boot sector, DISKDATA.COM utility, 132-133
passwords, VMIX.BAK file, 187
PATH command, 91-95
PATH variable, 94
Patha (HGB Enterprises)
  batch files, 100
  /E switches, 96
  editing keys, 97
  /O switches, 99
  PATHA /L command, 100
  parameters, 96
  paths
    changing, 96-100
    editing, 95-100
  registering, 95
  using from DOS command line, 100
PAUSE command (EXC program), 157
performing tasks with VMiX, 177-188
planning scripts, 145-147
PortaCall (Bill Dineen), 71
  \DSMF\PORTCALL directory, 70
  parameters, 71
  phone-line monitoring, 70-72
ports, remote communications defaults (VMiX), 179
/Ppath parameter, ShowMe Saver, 18
preparations for ZIP File Transfer operation, 73-74
print spooling, 56
printing
  information with SCROLLit, 22-23
  Spooler/Despooler program, 60-61
program data files, substituting with /F switches (Fast Load), 127
programs
  CAPTURE.BAT file, 134-135
  CHKDSK.EXE, 91
  Con>Format, 30
  despooling/monitor, SP3DSPMN.EXE, 65-67
  DOSMAX 2.1, memory management, 106-117
  DOSMAX.EXE, 106-109
  EMM386.EXE, expanded memory, 105
  ENVIMAX.COM, 113-115
  ERROR, \DSMF\ERROR directory, 89
  EXC, automating tasks, 141-144
  file transfer, ZIP File Transfer, 72-89
  Fast Load, 126-128
  FREELOAD.SYS, 115-117
  HIMEM.SYS, extended memory, 105
  LOADUP.EXE, 186

Load Watch, 128-129
MEMALOC.EXE, memory allocation, 137
memory loading for speed, 122-129
Monitor, spool/despool monitor, 64-65
SETFL.COM, 128
SHELLMAX.COM, 111-113
SP3ALL.EXE, 66-67
SP3DSP.EXE, 63
SP3DSPMN.EXE, 65-66
SP3MON.EXE, 64-65
SP3PRT.COM, 67
SP3SPDSP.EXE, 57-59
SP3SPL.EXE, 61-62
Spooler/Despooler, 58-61
Spooling/Despooling Software, \DSMK\SP3 directory, 57
STOPMAX.SYS, 109-111
SYS_CAP, 129
TSR
  Con>Format, 30
  SCROLLit, 18
  ShowMe Saver, 9
  Spooler/Despooler, 59
VMiX, 170-171, 186, 189-192
ZIP File Transfer
  ZIPCFG.COM, default parameters, 87-89
  ZIPDUP, 84-85

Q

Quarterdeck DESQView, 104
QUIT command
  EXC program, 153, 157
  VMiX, 178-179
quitting tasks with VMiX, 178-179

R

r parameter
  CURSOR command (EXC program), 164
  SHOWAT command (EXC program), 166
RAMCAM.COM utility, 131-132
READ command (EXC program), 165
Read menu, Disk Copy Fast, 38
RebelWare, Command-Date, 52
recalling scrolled screens, 18-27
receiving files from ZIP File Transfer command line, 82
registering
  Command-Date, 53
  Con>Format, 30
  Disk Copy Fast, 36
  DOSMAX 2.1, 104
  ENVIMAX 2.1, 104
  EXC 2.6a, 142
  Fast Load 1.25, 123
  Find Directory, 51
  FREELOAD 2.1, 104
  Patha, 95
  PortaCall, 71
  SCROLLit, 19
  SHELLMAX 2.1, 104
  ShowMe Saver, 9
  SHREDFILE, 45
  Spooling/Despooling Software, 57
  STOPMAX 2.1, 104
  SYS_CAP, 129
  VMiX 2.87, 171
  ZIP File Transfer, 73
remote accessing, VMiX, 179-181
remote communications port defaults, VMiX, 179
REPLACE subroutine, EXC program, 151-152
REPLACE.SCR script, 149-151
Resource window, Disk Copy Fast, 37

S

/S options, ZIP File Transfer, 82
saving information with SCROLLit, 22-23
SCR file extensions, EXC program, 143
screen buffers
  EXC program, 144
  SCROLLit, 19
screens
  control, EXC commands, 163-167
  controlling with VMiX, 181-186
  Disk Copy Fast main, 36
  information, maintaining, 18-27
  savers, 8-18
  scrolled, recalling, 18-27
  SHREDFILE, 46
scripts
  altering, 153-154
  controlling with EXC commands, 154-158
  creating, 144-154
  EXC program, 142-143
  multiple task, 145-147
  planning, 145-147
  REPLACE.SCR, 149-151
  writing, 147-153
SCRMAX command (EXC program), 165
Scroll Lock key, disabling Fast Load control with /NS parameter, 127
SCROLLit
  (Bromfield Software), 18
  20K screen buffer, 19
  color codes, 26
  colors
    /AC, /AM command parameters, 26
    changing and default, 26-27
  command keys, 20-21
  command line parameters, 23-27
  \DSMF\SCROLLIT directory location, 19
  hot-keys, 25
  printing/saving information, 22-23
  SCROLLIT /+ command, 25
  SCROLLIT /- command, 25
  searching for information, 21-22
  starting from AUTOEXEC.BAT file, 19
  TSR program, 18
SEARCH command (EXC program), 150, 159-160
search paths
  changing with Patha, 96-100
  PATH command, 93-94
searching for information with SCROLLit, 21-22
searchpath parameters, 93
securing system, 129-137
Send command (ZIP File Transfer), 79
sending files, ZIP File Transfer, 76-77, 81-82
SerVer command (ZIP File Transfer), 78-79
Server mode, ZIP File Transfer command line, 78-84
SET BAUD command (VMiX), 179
SET command (EXC program), 94, 153
SET REMOTE -i command (VMiX), 181
SET REMOTE -x command (VMiX), 181
SET REMOTE command (VMiX), 179
SET TERMINAL -t command (VMiX), 180
SET VIDEO -c command (VMiX), 183
SET VIDEO -m command (VMiX), 184

SET VIDEO -p command (VMIX), 185-186
SETFL.COM program, disabling/enabling Fast Load, 128
setUp menu (SHREDFILE), 50
SETWAIT command (EXC program), 157-158
shareware, 8
SHELLMAX 2.1, 104
SHELLMAX.COM program, 111-113
SHOW command (EXC program), 160-161
SHOWAT command (EXC program), 165-166
SHOWIN command (EXC program), 166
ShowMe Saver, 9, 12-18
  addition to AUTOEXEC.BAT file, 11
  C:\SHOW directory, 10-12
  \DSMF\SHOWME directory, 10
  /E parameter, 17
  /F parameter, 17
  /N parameter, 18
  /Ppath parameter, 17-18
  TSR program, 9
  /U parameter, 18
  XMS (extended memory) residence, 13
shredders, electronic, 45
SHREDFILE (Seth W. Comstock), 45-50
  disabling delayed writes, 48
  \DSMF\SHRED directory, 45
  Fspec menu, 47
  setUp menu, 50
  SHRED command, 45
  Shred menu, 46
  shredding
    disk space, 49-50
    individual files, 47-48
    multiple files, 48
  Sort menu, 47
  sorting files, 46-47
SL-EASY Software, ERROR program, 89
SMARTDrive 4.0, disk caching, 118, 122
SP3ALL.EXE program, spooling/despooling/monitoring, 66-67
SP3DSP.EXE program, despooling, 63
SP3DSPMN.EXE program, despooler/monitor, 65-66
SP3MON.EXE program, spool/despool monitor, 64-65
SP3PRT.COM program, 67
SP3SPDSP.EXE program, 57-59
SP3SPL.EXE program, 61-62
spooling
  on networks with Monitor, 64-65
  programs, 61-67
Spooling/Despooling Software (Budget Software Company), 57-61
Status menu, VMiX, 191
STOPMAX 2.1, 104
STOPMAX.SYS program, 109-111
subroutines
  END (EXC program), 152-153
  REPLACE (EXC program), 151-152
switches
  /E, Patha, 96
  /O, Patha, 99
  /X, ShowMe Saver, 17
SYDEX, Con>Format, 30
SYS_CAP (System Capture Utility Package), 129-137
system memory, RAMCAM.COM utility, 132

T

tagging files to delete with SHREDFILE, 46
tasks, quitting, VMiX, 178-179
TECHREF.DOC file, VMiX directory, 192

text
displaying, EXC commands, 159-163
finding, EXC commands, 159-163
TICK command (EXC program), 158
TIME command (EXC program), 158
TIMEOUT command (EXC program), 158
toggling options, 107
transferring
files, ZIP File Transfer, 74-78
ZIP File Transfer to other computers, 84-85
TSR programs
Con>Format, 30
SCROLLit, 18
ShowMe Saver, 9
Spooler/Despooler, 59
TYPE command (EXC program), 152, 162
TYPEFILE command (EXC program), 162

U

/U parameter, unloading ShowMe Saver, 18
UMB (upper memory blocks), 105
Unlink command (ZIP File Transfer), 81
upper memory blocks
configuration, DOSMAX utilities, 120-121
loading Con>Format, 31
loading programs with VMiX 386 mode, 186
utilities
CMOS.EXE, 134
DEBUG.EXE, 85
DISKDATA.COM, 132-133
DOSMAX, 119-121
EXAMINE.COM, 135-137
GETCMOS.COM, 133-134
Load Watch, 128
LOADUP, 175
MEMDUMP.COM, 135-137
Patha, 94-95
RAMCAM.COM, 131-132
SCROLLit, 18
SHREDFILE, 45
SYS_CAP, 130-131
VMiX programs, 191-192

V

variables
environment, 99
PATH, 94
yes/no, EXC program, 150
video flashing, controlling in ShowMe Saver, 17
video mode codes, VMiX, 184
viewing memory, MEMDUMP.COM utility, 135-137
VMiX 2.87 (Commercial Software Associates), 170-188
286 mode/386 mode, 171, 176
8086 mode, 171, 176
Advanced menu, 190
colors, 182-183
commands
DO, 177-178
DOSJOB, 177-178
KILL, 178-179
LOADUP READ, 186
QUIT, 178-179
SET BAUD, 179
SET REMOTE -i, 181
SET REMOTE -x, 181
SET REMOTE, 179
SET TERMINAL -t, 180
SET VIDEO -c, 183
SET VIDEO -m, 184
SET VIDEO -p, 185-186
VM_BOOT, 172

\DOSSTUFF\VMIX directory, 172, 192
LOADUP utility, 175
logging into null-modem cables, 181
Memory menu, 191
Options menu, 191
passwords, VMIX.BAK file, 187
Programs menu, 189
Status menu, 191
TECHREF.DOC file, 192
utility programs, 191-192
VMIX.INI file, 187-188
Vmix shell, 189
VW menus, 188-191

W

WAITFOR command (EXC program), 152, 163
WAITSCR command (EXC program), 163
WINCOLS command (EXC program), 166
windows
  Con>Format, 31
  Information (Disk Copy Fast), 37-38
  Resource (Disk Copy Fast), 37
WINMAX command (EXC program), 166
WINROWS command (EXC program), 167
Write menu, Disk Copy Fast, 39

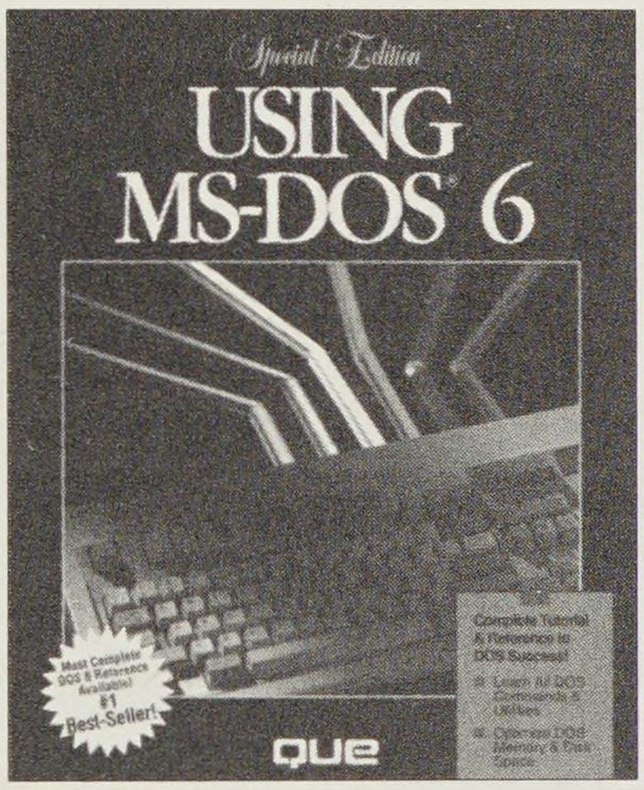
X-Y-Z

/X switch, ShowMe Saver, 17
XMS (extended memory), ShowMe Saver residence, 13

yes/no variable, EXC program, 150

ZIP File Transfer (Eric Meyer), 73-77, 85-89
  changing default parameters ZIPCFG.COM program, 87-89
  commands
    Compare, 80
    Delete, 81
    Directory, 78
    Fetch, 79
    Send, 79
    SerVer, 78-79
    Unlink, 81
    ZIP, 81
    ZIPCFG, 87
controlling host from guest computer with server mode, 79-81
/D parameter, 84
\DSMF\ZIP directory, 73
files
  comparing, 80-81
  deleting, 80-81
  receiving from command line, 82
  transfer program, 72-89
guests and hosts, 76
Parameters menu, 86
Server mode, 82-84
ZIP menu, 75, 86-87
ZIPCFG.COM program, 87-89
ZIPCFG menu, 87
ZIPDUP program, 84-85

Count on Que for the Latest in DOS Information!



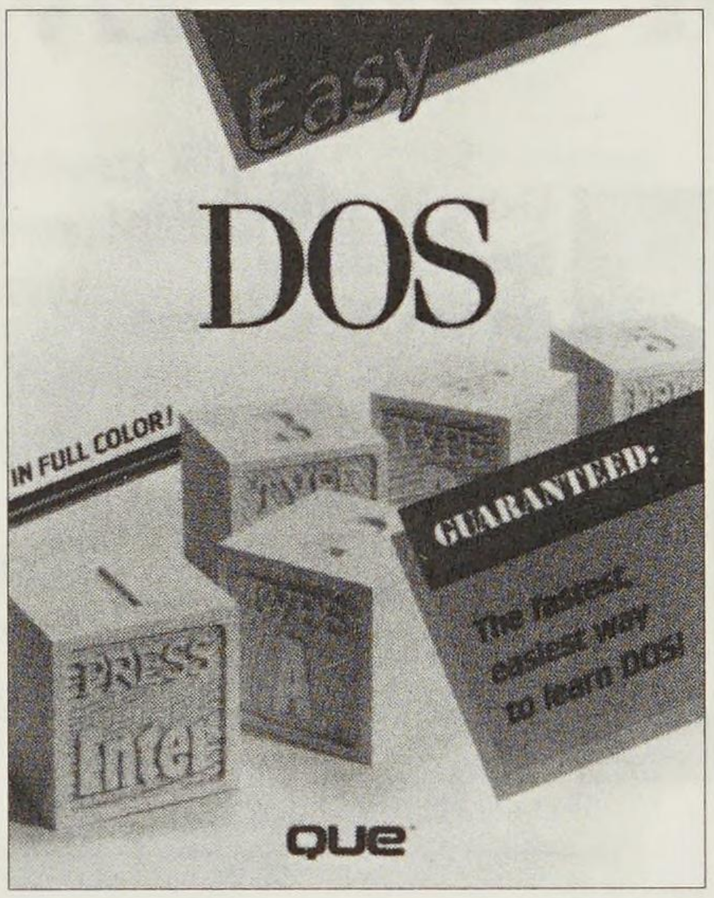
Using MS-DOS 6, Special Edition

Que Development Group

Version 6

$29.95 USA

1-56529-020-8, 1,000 pp., 7 3/8 x 9 1/4



Easy DOS, 2nd Edition

Shelley O'Hara

Through Version 6

$16.95 USA

1-56529-095-x, 200 pp., 8 x 10

More DOS Titles from Que

I Hate DOS

Bryan Pfaffenberger

Version 6

$16.95 USA

1-56529-215-4, 384 pp., 7 3/8 x 9 1/4

MS-DOS 6 QuickStart

Que Development Group

Version 6

$21.95 USA

1-56529-096-8, 420 pp., 7 3/8 x 9 1/4

MS-DOS 6 Quick Reference

Que Development Group

Version 6

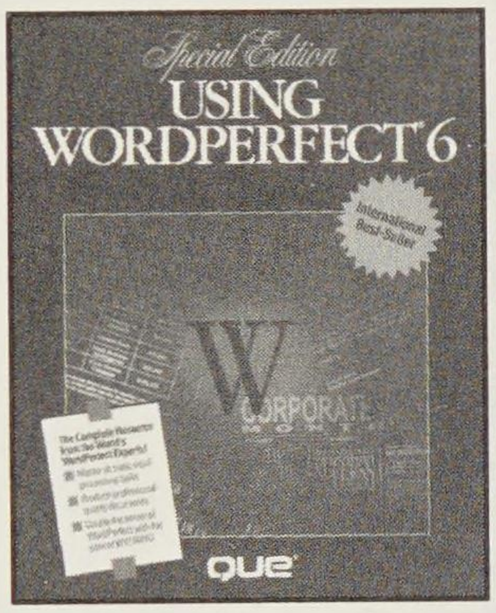
$9.95 USA

1-56529-137-9, 160 pp., 4 3/4 x 8

que

To Order, Call:(800) 428-5331 OR (317) 573-2500

A Powerful Dose of DOS for the Everyday User!

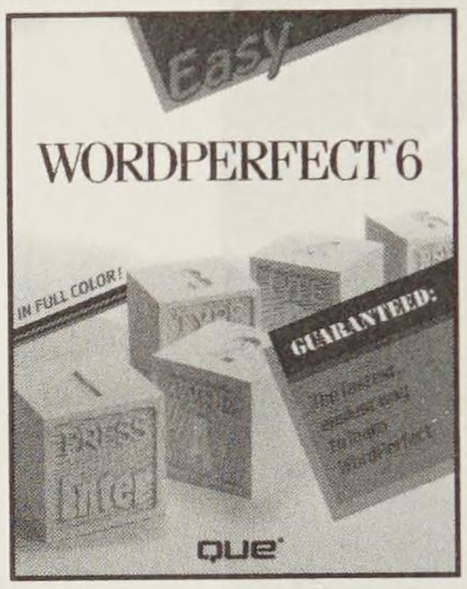


Using WordPerfect 6, Special Edition
Que Development Group

The complete tutorial and reference guide…ideal for users of all skill levels!

Version 6.0

$29.95 USA
1-56529-077-1, 1,200 pp., 7 3/8 X 9 1/8

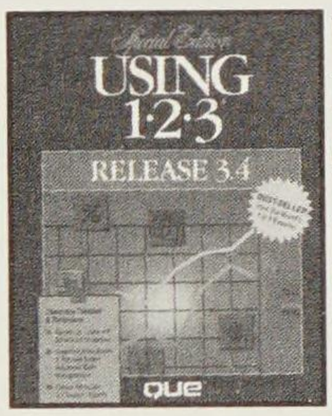


Easy WordPerfect 6
Que Development Group

The revolutionary, 4-color introduction to the latest version of WordPerfect!

Version 6.0

$16.95 USA
1-56529-087-9, 256 pp., 8 X 10

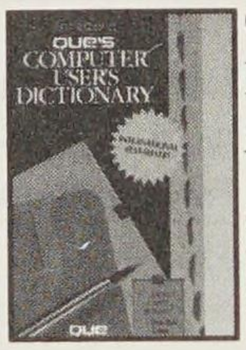


Using 1-2-3, Release 3.4
Que Development Group

This comprehensive tutorial/ reference covers all readers need to know to use 1-2-3!

Release 3.4

$29.95 USA
1-56529-004-6, 1,100 pp., 7 3/8 X 9 1/8

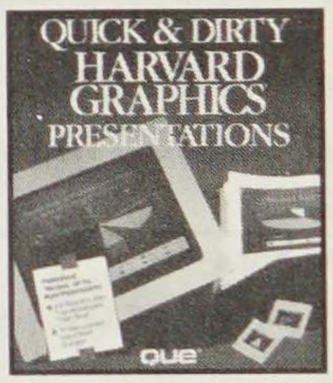


Que's Computer User's Dictionary, 3rd Edition
Bryan Pfaffenberger

Completely revised and updated, with more terms than ever before!

All Computer Users

$12.95 USA
1-56529-023-2, 600 pp., 4 3/4 X 8



Quick & Dirty Harvard Graphics Presentations
Bob Benedict

A complete recipe book for a dazzling array of Harvard Graphics charts…plus advice on presentation!

Version 3

$24.95 USA
1-56529-089-5, 320 pp., 8 X 10

Upgrading & Repairing PCs, 2nd Edition
Scott Mueller, et al.

0-88022-856-3, 1,000 pp., 7 3/8 X 9 1/8
$34.95 USA

IBM PCs-Compatibles

Using Novell NetWare, 2nd Edition
Bill Lawrence

0-88022-756-7, 700 pp., 7 3/8 X 9 1/8
$29.95 USA

Through Version 3.11

Using dBASE IV 2.0, Special Edition
Steve Davis

1-56529-153-0, 1,100 pp., 7 3/8 X 9 1/8
$29.95 USA

Version 2.0

Using 1-2-3, Release 2.4, Special Edition
Que Development Group

0-88022-988-8, 1,000 pp., 7 3/8 X 9 1/8
$29.95 USA

Release 2.4

Using Paradox 4, Special Edition
Walter R. Bruce, III

0-88022-822-9, 900 pp., 7 3/8 X 9 1/8
$29.95 USA

Version 4

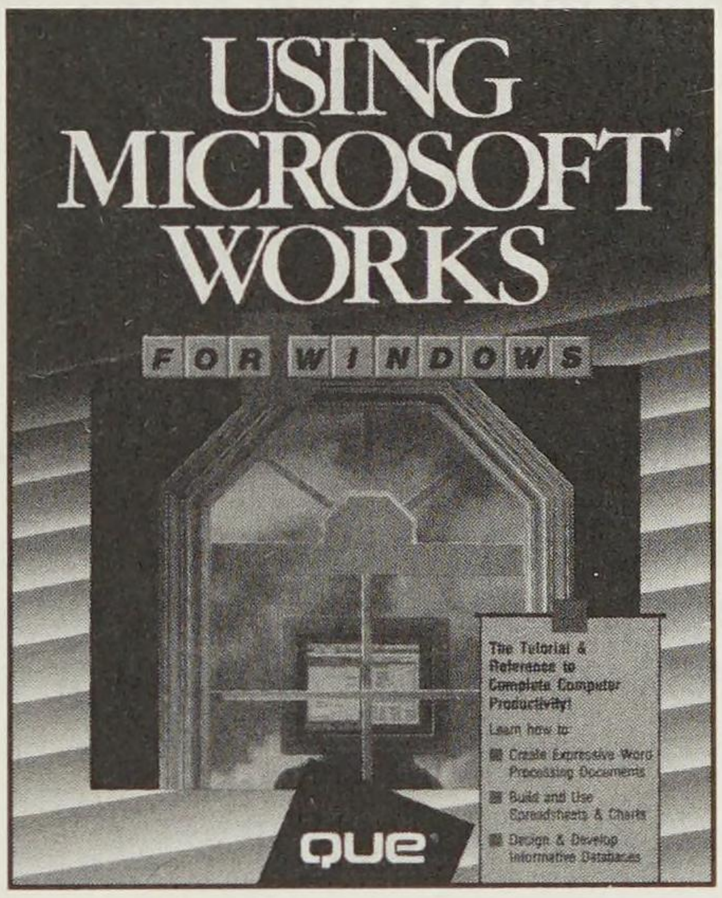
Using Quicken 6
Linda Flanders

1-56529-071-2, 750 pp., 7 3/8 X 9 1/8
$22.95 USA

Version 6

For more information or to place an order, call: 1-800-428-5331

Count on Que for the Most Up-to-Date Information on Integrated Packages

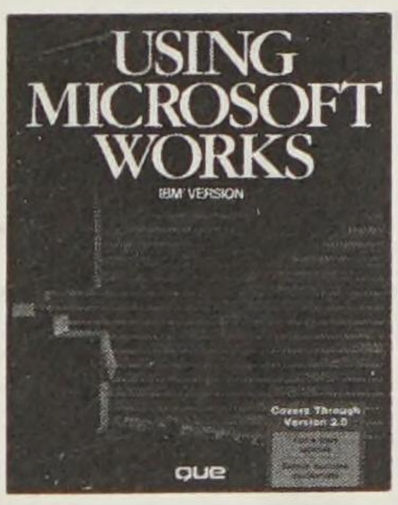


Using Microsoft Works for Windows, Special Edition
Douglas Wolf

This book provides plenty of timesaving tips and hints on sharing information between Works' modules. A comprehensive reference and tutorial, this book will give you everything you need to know about Microsoft Works.

Version 1

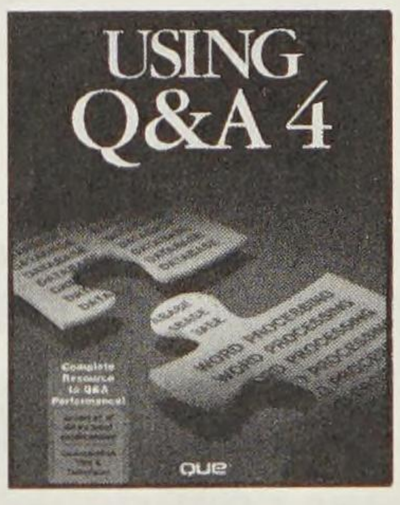
$24.95 USA
0-88022-757-5, 550 pp., 7 3/8 x 9 1/4



Using Microsoft Works: IBM Version
Douglas Wolf

Through Version 2

$22.95 USA
0-88022-467-3, 444 pp., 7 3/8 x 9 1/4



Using Q&A 4
David Ewing & Bill Langenes

Version 4

$27.95 USA
0-88022-643-9, 550 pp., 7 3/8 x 9 1/4

Q&A 4 Quick Reference
Que Development Group

Latest Version

$9.95 USA
0-88022-828-8, 160 pp., 4 3/4 x 8

Q&A 4 QuickStart
Que Development Group

Version 4

$19.95 USA
0-88022-653-6, 400 pp., 7 3/8 x 9 1/4

Que's Using Enable
Walter Bruce

All Versions Through Enable/OA4

$29.95 USA
0-88022-701-X, 700 pp., 7 3/8 x 9 1/4

Using Lotus Works 3.0
J. D. Watson

Release 3.0

$24.95 USA
0-88022-771-0, 500 pp., 7 3/8 x 9 1/4

Using PFS: First Choice 4.0
Katherine Murray

Through Version 4.0

$27.95 USA
0-88022-969-1, 600 pp., 7 3/8 x 9 1/4

Using PFS: WindowsWorks 2
Deanna Bebb

Through Version 2

$29.95 USA
1-56529-073-9, 500 pp., 7 3/8 x 9 1/4

que

To Order, Call: (800) 428-5331
OR (317) 573-2500

DISK INSTALLATION

To install the *DOS Stuff Microsoft Forgot* disk, follow these simple steps:

1. Log onto the disk drive where the *DOS Stuff Microsoft Forgot* disk is. For example:

   A: (then press the Enter key)

2. Type

   MAKEDSMF *x*

   where *x* is the letter of the hard disk drive to install to.

The MAKEDSMF batch file creates a directory on the specified disk named DSMF. The individual programs are installed in directories under the DSMF directory. You need at least 4 megabytes of free space.

DISCLAIMER

By opening this package, you are agreeing to be bound by the following:

This software product is copyrighted, and all rights are reserved by the publisher. You are licensed to use this software on a single computer. You mark, copy, and/or modify the software as needed to facilitate your use of it on a single computer. Making copies of the software for any other purpose is a violation of United States copyright laws.

This software is sold *as is*, without warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Neither Que nor its dealers or distributors assume any liability for any alleged or actual damages arising from the use of this program. (Some states do not allow for the exclusion of implied warranties, so the exclusion may not apply to you.)

OOPS!

Microsoft isn't the only one who forgot something. Step 2 of the disk installation instructions should read:

2. Type

DSMF *x*:

Where *x* is the letter of the hard disk drive to install to.

DOS STUFF

MICROSOFT FORGOT!

DOS Stuff Microsoft® Forgot!

Unleash the Power of DOS!

We found the DOS stuff that Microsoft forgot—and put it all in one package. Now—without delay—you can make your DOS complete.

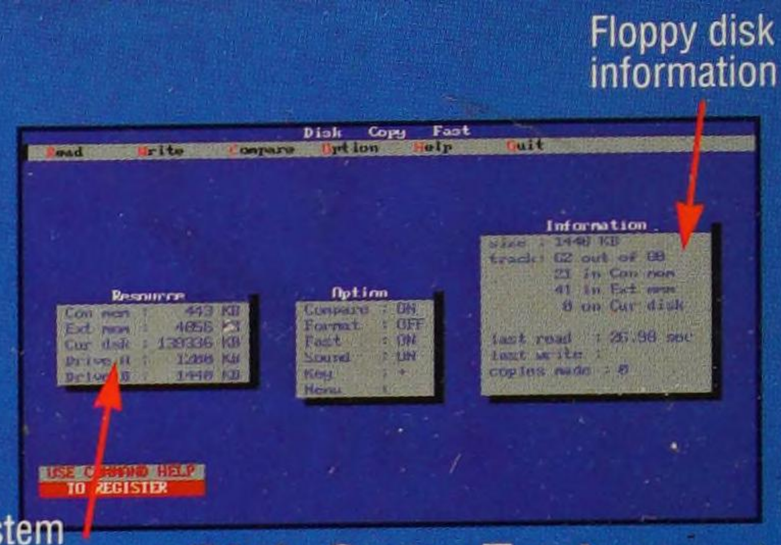
More! More! More!

More power! More control! More simplicity! This handy book includes 18 indispensable DOS applications that give you more. You'll find great utilities for managing your desktop, working with and manipulating disks, enhancing communications, finding files fast, and automating just about any DOS procedure.

Plus, the book includes lots of hard-to-forget techniques and expert performance tips that make your work something to remember.

Some of the Stuff Microsoft Forgot:

- VMiX—an ingenious, multitasking, multi-user interface for DOS
- Spooling/Despooling Software—enables you to continue working while printing larger documents
- ShowMe Saver—displays great-looking graphics while protecting your screen from image burn-in
- Con>Format—formats disks in background while you keep working on any software program
- FD—a file finder that works hard to find what you can't
- PortaCall—notifies you when you have messages on your answering machine
- DOSMAX—helps you squeeze the maximum conventional memory out of your PC
- ERROR—provi[illegible] the English-language meanin[illegible] cryptic DOS error codes



SHREDFILE
Provides a menu-driven environment for shredding files

SYSCAP
Provides a break-down of all programs using system memory.

VMiX
Provides a multi-tasking environment in DOS. Here, three tasks are open.

$19.95 USA / $24.95 CAN
£18.77 (Incl VAT) Net U.K.

INFORMATION CENTER
All User Levels
How To
IBM & Compatibles
Operating Environment
Covers Version 6.0

DISK INCLUDES
18 Powerful Tools & Toys Microsoft Forgot!

que®

ISBN 1-56529-475-0
9 781565 294752 90000