Brady UTILITY SOFTWARE

MACWORLD

THE FULLY POWERED MAC®

Everything You Need to Make Your Mac Faster, More Efficient, and Tailored to Your Needs

Robert C. Eckhardt

CONTAINS A DISK

The Fully Powered Mac®

Robert C. Eckhardt

MACWORLD

BRADY
New York

Copyright © 1988 by Robert C. Eckhardt
All rights reserved.

All rights reserved,
including the right of reproduction
in whole or in part, or in any form

BRADY

Simon & Schuster, Inc.
Gulf +Western Building
One Gulf +Western Plaza
New York, New York 10023

Distributed by Prentice Hall Trade

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Library of Congress Cataloging-in-Publication Data

Eckhardt, Robert C.
The Fully Powered Mac [computer file].

1 computer disk; 3 1/2 in. 1 v.—(Brady utility software)
System requirements: Macintosh Plus, SE, or II; 1 Mb; 2 800K floppy disks or 1 hard disk plus 1 800K floppy disk; monochrome monitor; Apple printer.
Title from title screen.
Not copy protected.
Audience: Users of Macintosh computers.
Summary: Presents strategies and utilities software for making the Macintosh computer faster, more efficient, and better tailored to individual user's needs.
1. Utilities (Computer programs)—Software.
I. Title. II. Series.
QA76.76.U84E27 1988 005.4'3 88-8620
ISBN 0-13-332230-0

In memory of
Larry, Jess, Brent,
Robert, Hugh,
Joe...

Contents

Acknowledgments xi

Introduction **Getting Started** xiii
Macintosh Basics: What You Need to Know before You Begin xiii
Software: What's Included on the Disk and What Isn't xv
Compatibility, or How This Book Was Written xvi
Conventions Used in *The Fully Powered Mac* xvii
A Quick Overview of What's Ahead xviii
A Fully Powered Mac Roadmap xxi

Chapter 1 **Hardware and The Fully Powered Mac** 1
The Minimum Hardware Setup 2
The Ideal Hardware Setup 5
Additional RAM 5
Is One Megabyte Enough? 6
Adding RAM: How and What Kind 8
Hard Disk Drives 9
Hard Disks: How Big Is Big Enough? 10
What Kind of Hard Disk to Buy 12
More RAM versus A Hard Disk 15
Getting a Mac II without Buying One 15
Summary 17

Chapter 2 **Files, Folders and Organization** 19
The System Folder 21
Inside the System Folder 22
The System File 22
The Finder 23
Printer Drivers 23
Control Panel Files 25
Init Resources 25

Downloadable Fonts 26
Other Files 27
Version Numbers And Keeping Up-to-date 28
When To Use Out-of-date System Files 30
Working With A Master System Folder 30
And A Master System File 31
Organizing Your Files 33
Organizing Files on Floppy Disks 35
A Dedicated System Folder Disk 36
System Folder And Application(s) On A Single Disk 38
Organizing Files On A Hard Disk 40
Sorting by Application 40
Sorting By File Type 43
A Compromise Style of Organization 44
Putting Support Files Out of the Way 46
Dividing a Hard Disk Into Subvolumes 48
Making Your Organizational Style Work On Screen 51
Windows And Where To Put Them 51
Viewing Files and Folders in Windows 56
Naming Files And Folders 59
Summary 66

Chapter 3 **The System File, Part 1: FKeys and Desk Accessories** **69**
FKeys: Why, How, and Which Ones 72
An Annotated FKey Sampler 73
Putting Your FKeys To Work 80
Managing Your FKeys With FKey Manager 81
Some FKey Words Of Wisdom 97
Other Ways To Work With FKeys 98
Home-made FKeys With FKey Maker 98
FKeys (And More) From The Keyboard, Using *QuicKeys* 99
Desk Accessories: Why, How, and Which Ones 103
An Annotated Desk Accessory Catalog 104
Putting Your Desk Accessories Under The Apple Menu 123
The Desk Accessories In Your System File 124
Installing Desk Accessories In Applications 126
Summary 136

Chapter 4 **The System File, Part 2: Fonts** 139
Which Fonts For What Purposes 141
The Kinds Of Fonts 142
ImageWriter Fonts 144
On-screen versus On the Page 145
Choosing ImageWriter Fonts 147
LaserWriter Fonts 148
On-screen versus On the Page 149
Resident versus Downloadable Fonts 151
Choosing LaserWriter Fonts 154
Troubleshooting: When Bad Things Happen To Good Fonts 158
Fitting All Your Fonts In The Font Menu 161
Installing Only the Screen Sizes and Styles You Need 162
ImageWriter Fonts: Which Sizes to Install 162
LaserWriter Fonts: Which Sizes and Styles to Install 167
Installing Fonts In Applications and Documents 172
Adding More Fonts With A Font/Desk Accessory Extender 176
Using *Fontsie* 176
Using a Commercial Font/Desk Accessory Extender 180
Space-Saving Strategies for Downloadable LaserWriter Fonts 182
On Disk 182
In the LaserWriter 184
Customizing Fonts 185
Changing Screen Font Names—and Why 186
Reordering Fonts with *FONTastic Plus* 188
Hiding Fonts with *ResEdit* 192
Adding Fractions To A PostScript Font 198
Summary 215

Chapter 5 **Customizing the Desktop (and Other Things) to Fit the Way You Work** 219
Custom Windows 221
Icon Views 223
Standard Icon View 223
Small Icon View 225
List Views 226
Font And Font Size 227

Other Window Options 229
The Default Window 230
Custom Menus And Menu Command Shortcuts 232
Customizing the Desktop Menu Bar 232
EZ-Menu 232
JClock 233
AutoBlack And *Pyro!* 233
New Menu Command Shortcuts: Getting Ready 236
Drawing a Menu Map 236
Deciding on Appropriate Keyboard Combinations 240
ResEdit: Some Comments for the Uninitiated 242
Assigning New Shortcuts to Existing Menu Commands 245
Shortcuts for Standard Menu Items 245
Shortcuts for Desk Accessories and Fonts 252
Creating New Commands and Their Shortcuts 255
Using *QuicKeys* 256
Using *Tempo II* 258
Custom Keyboards 264
Using *MacKeymeleon* 264
Using *QuicKeys* 270
Making a Custom Keyboard Map 274
Custom Dialog Boxes 276
Custom Icons 278
Other Customizations You Can Do 286
Custom Command and Menu Names 286
The Apple Menu 286
Font Menus 290
Standard Menus 291
Custom Desktop Patterns 295
Custom Scroll Bars 297
Custom Startups 299
Creating a Startup Screen 299
Startup Sounds, Beeps, and Surprises 305
Summary 310

Chapter 6 **Creating Your Own Integrated Program** **313**
Standardizing Your Applications 315
Order out of Chaos 316

Assigning Standard Menu Shortcuts 316
A General Macro and Menu Shortcut Strategy 322
Assigning Shortcuts and Creating Macros 324
Menu Shortcuts, a Reprise 324
Macros: Some Examples 327
Changing the Application Font 334
Integrating Your Applications 335
Limited-Memory Integration 336
Using Direct-transfer Utilities 337
Using Finder Substitutes 345
Using *HyperCard* 364
Ample-Memory Integration 377
Using MultiFinder 379
Using *Servant* 385
Using *Switcher* 389
Summary 397

Chapter 7 **More Speed and How to Get It** **401**
Automated Routines for Greater Speed 402
Using *StartMac* 403
Using *Tempo II* 410
Keyboard Shortcuts in Dialog and File Selection Boxes 422
Universal Keyboard Shortcuts 423
Application Keyboard Shortcuts 424
Shortcuts for Mouse-oriented People 427
Easier Folder and File Access 427
Creating "Icon Bars" for All Your Applications 433
RAM Caches and RAM Disks 436
RAM Disks 438
RAM Disk Pros and Cons 439
RAM Disk Strategies 442
Using a RAM Disk Program 477
What to Look for in a RAM Disk Program 450
RAM Caches 451
RAM Cache Pros and Cons 453
Optimizing RAM Cache Performance 457
What to Look for in a RAM Cache 465
RAM Disk or RAM Cache? 467

Print Spoolers 467
Summary 473

Chapter 8 **Fragmentation, Copy-protection, Backups, Disk Problems and More** 475
Speed Traps that Slow the Macintosh Down 475
Copy-Protection 476
Fragmented Files 482
Back ups, The Computer Owner's Safety Net 493
A Fully Powered Backup Strategy 494
Hard Disk Backup Programs 500
What to do if Your Safety Net Has a Hole in It 502
Summary 508

Appendix A **The Fully Powered Mac Utilities Disk** 511

Appendix B **Troubleshooting** 523

Appendix C **Products Guide** 551

Appendix D **Further Reading** 567

Index 569

Acknowledgments

Although the general form of this book is a distillation of several years of my own experience working on a Macintosh, many of the details came from conversations, both formal and informal, with friends and colleagues. Since it is the details which form the heart of a work such as this, I am indebted to the many people who have shared with me their good ideas and helpful insights: John Avila, Steven Brecher, Chip Carmen, Brian DeLacey, Edward Dougherty, Nancy Dunn, Dan Farber, Danny Goodman, Michael Mace, Janet McCandless, Adrian Mello, John Mitchell, Erfert Nielsen, Lon Poole, Jay Siegel, Wallace Westfeldt. Although all of these people contributed to the book, each in their own way, all of the errors and inadequacies are mine alone.

For cutting through the red tape at Apple and providing me with early looks at some of the software described in this book, Martha Steffen deserves special mention. John Duhring and Stuart Klein of SuperMac Technology generously loaned me a hard disk for the period during which this book was written, and many people at a variety of software companies kindly provided prerelease and/or review copies of the software described in the following pages.

All Macintosh owners owe a debt of gratitude to the many shareware and freeware programmers, and I am no exception. To the authors of the programs on the disk accompanying this book, to the authors of the programs mentioned in the text, and to the authors of the many other programs which did not make it onto these pages, I offer many thanks.

Finally, I must acknowledge Terry Anderson and Milissa Koloski, my editors, for their constant faith in the project which resulted in this book. And a special thanks to John, for his faith in me.

Limits of Liability and Disclaimer of Warranty

The author and publisher of this book have used their best efforts in preparing this book and the programs contained in it. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The author and publisher make no warranty of any kind, expressed or implied, with regard to these programs or the documentation contained in this book. The author and publisher shall not be liable in any event for incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of these programs.

Trademarks

Apple, Apple Computer, Inc., ImageWriter, LaserWriter, Macworld, Macintosh, Mac, Finder, HyperCard, MultiFinder, and Stackware are trademarks of Apple Computer, Inc.

CLARIS, MacWrite, MacDraw, MacProject, and MacPaint are trademarks of CLARIS Corporation.

Switcher is a common law trademark of Apple Computer, Inc.

PostScript is a registered trademark of Adobe Systems, Inc.

Helvetica, Times, and Palatino are registered trademarks of Linotype Co.

ITC Advant Garde Gothic, ITC Bookman, ITC Zapf Chancery, and ITC Zapf Dingbats are registered trademarks of International Typeface Corporation.

Additional trademarks, service marks, and indication of ownership is marked in sections on software/freeware by initial capitalization and italics. Every effort has been made to fully identify each trademark/service mark ownership.

Introduction

Getting Started

We Macintosh owners are spoiled. Everything is, after all, so easy. Our computer is the epitome of "plug-n-play" technology. Most of the software we use is built on one basic plan and thus requires little additional learning (at least for basic operations) after the first program or two. Indeed, many Mac owners casually flip through their manuals once or twice and then put them away in a closet, never to look at them again. Some Mac owners don't even bother to expend that small amount of effort, and because the Macintosh is so easy, fun, and forgiving, few notice what they're missing.

But the fact is, there is more to it—lots more. Some of it is in the manuals we never fully digest. Most of it, however, is scattered about in newsletters and magazines and talked about at user-group meetings and seminars. I'm not talking about technical stuff—you don't have to be a programmer to go beyond the manual. I'm talking about simple things that can make your Macintosh easier to use, more convenient, and faster.

Indeed, as the Macintosh becomes more powerful and more complex, it becomes easier and easier for you and your Mac to get bogged down, perhaps even overwhelmed, without your even being aware of it. Perhaps your hard disk is disorganized and files get lost, applications are hard to locate; or desk accessories in one System file are not available in another and you can't remember which file has which desk accessories. The keyboard shortcuts you learned for one application do surprising and undesirable things in another application. Switching from application to Finder to application takes much too long. You do a lot of printing, during which time the Macintosh is unavailable for other pursuits. You've run out of room for all the fonts you want to use. You have four megabytes of RAM, but have little or no idea whether you're using it efficiently. You're tired of the fact that Shift-period prints an angle bracket and not a period (like your old typewriter), and so on, and so on.

Whether or not you've begun to notice minor, annoying problems such as these, and whether you own a Macintosh Plus, a Macintosh SE, or a

Macintosh II, there are a myriad of ways in which you can turn your computer into a fully powered Macintosh—a Macintosh that is faster, and more efficient, more convenient, and more in tune with the way you want to work. With the strategies described in the pages that follow and the software on the disk that accompanies this book, you have virtually everything you need to create a fully powered Macintosh.

Macintosh Basics: What You Need to Know before You Begin

You don't need to be a long-time Macintosh owner, a "power user," or a programmer to make use of the techniques, strategies, and tips described on the following pages. All you need is a knowledge of, and some experience with, these basic Macintosh procedures:

- Using the mouse and menus.
- Launching and quitting applications.
- Locating and opening documents with a File Selection Dialog Box.
- Invoking commands from the keyboard.
- Working with windows: dragging, shrinking and enlarging them, making a window active, and so on.
- Working with objects on the Desktop: arranging icons, setting icon or list views, emptying the trash, and so on.
- Copying files on the Desktop, moving files from folder to folder or disk to disk, throwing files in the trash to delete them.
- Creating folders, nesting folders within folders, placing files in folders.
- Responding to dialog boxes.
- Using Font/DA Mover to install desk accessories and fonts in the System file.

These are all skills that are described in detail in the Macintosh owner's manual. If you are uncertain about any of these basic Macintosh techniques, now is the time to pull the Macintosh manual from the closet

and review them. As long as you are familiar with the basic operation of the Macintosh, you need not be concerned about getting in over your head. More advanced techniques and other tricks and tips will be described in detail as we go along.

While this book assumes no more than the basic Macintosh skills, I should point out that even advanced Macintosh users will find a great deal of interest here. Indeed, the number of pages required to bring the less experienced Macintosh owner up to speed is actually quite small. As a result, the bulk of the book is devoted to techniques and strategies that should be useful to long-time Macintosh owners as well as more recent ones.

Software: What's Included on the Disk and What Isn't

To make full use of this book, you do not need all kinds of fancy or expensive software. Many of the techniques described in this book require no special software at all, just a willingness to experiment with a new way of working. For those that do require special software, many of the necessary programs are public-exchange software. By public-exchange, I mean software that may be freely and legally copied and passed from person to person. Some public-exchange programs are free. Sometimes, however, the programmer requests a fee (usually in the $5 to $20 range) if, after you try the program, you decide to make it a permanent part of your software collection. A number of the public-exchange programs described in the following chapters are included on *The Fully Powered Mac* Utilities Disk, which accompanies the book. Those public-exchange programs that are discussed in the book but are not included on the Utilities Disk are widely available from user groups, electronic bulletin boards, and information utilities (such as CompuServe and GEnie).

Public-exchange software offers a unique opportunity to try-before-you-buy. Only by paying for good public-exchange software (when such payment is requested) can we expect to see more and better public-exchange programs in the future. So, please...

> **if you use public-exchange software for which the programmer requests a fee, send him or her a check—today.**

The commercial programs described in this book are ones that you may already have, or ones which you may want to acquire once you have seen

what they can do. Although it can be truthfully said that virtually everything you need to create the fully powered Macintosh is contained within the pages of this book and on the disk that accompanies it, if your software library is limited, you may find yourself wanting to acquire a few additional utilities in order to make your fully powered Mac exactly the way you want it. Fortunately, almost all of the commercial programs recommended in this book are inexpensive (usually somewhere between $50 and $75).

If your budget is tight, however, and you were to ask me to name the one commercial program that can most enhance your Mac's performance and which you should most seriously consider buying, I would have to point to a macro processor such as *Tempo II* (Affinity Microsystems) or *QuicKeys* (CE Software). Macro processors are explained in Chapter 5 and are used in a variety of ways throughout the book, so I won't go into the details here. But I will say this: if you have the money to spend, I suggest you buy *Tempo II*, *QuicKeys*, or a similar program soon, so you can use it in the many strategies requiring a macro processor described in the book. You may find it helpful to review my comments about macro processors in general, and *QuicKeys* and *Tempo II* in particular (starting on page 100), in deciding which one to buy.

Compatibility, or How This Book was Written

This book was written with Microsoft *Word* 3.01 (Microsoft) on a Macintosh Plus with four megabytes of RAM and a 20-megabyte hard disk. The first draft was written using System 4.1/Finder 5.5, and the final revisions were made under System 4.2/Finder 6.0 (otherwise known as MultiFinder). All tips, techniques, and strategies were devised on this computer under System 4.1/Finder 5.5 and then tested on a Mac Plus, Mac SE, and Mac II under System 4.2/Finder 6.0.

You should have no problems (except where explicitly noted in the text) with the suggestions in this book if you own a Mac Plus and are using System 4.2/Finder 6.0. If you own a Mac SE or a Mac II, a few of the programs described in this book may not function properly. Usually, these instances are mentioned in text, although a few may have slipped past the testing process unnoticed. If you encounter problems, first make sure that you are following my instructions carefully and that there are no special comments or caveats concerning your type of Macintosh or System file version.

It is an unfortunate fact, however, that even if you play by the book, you may run into problems. The reason for this is twofold. First, the Macintosh and its software is a complex entity. Second, it is physically impossible to test every conceivable combination of fully powered strategies on every type of machine, using every appropriate System version with every known desk accessory and FKey installed, with all possible Init files in the System folder, and running all the major application programs. It is thus impossible, even with the small universe of programs covered in this book, to guarantee that all possible combinations of software strategies described in the following pages are compatible with all the utilities, desk accessories, and applications you currently use.

Most readers should experience no difficulties if they are careful to follow my directions precisely as written and are mindful of any warnings I make specific mention of. But, much as we have tried to anticipate them, there are times when unusual combinations of predictable and otherwise well-behaved software will fail to cooperate, and a few of you may experience just such a situation. If you do run into problems, turn to Appendix B for recommended solutions.

Conventions Used in *The Fully Powered Mac*

To help the reader (not to mention the author), a number of conventions were adopted concerning the text. As you read, you will want to keep the following in mind:

Software names (such as *MacWrite* and *SuperPaint*), whether they be simple desk accessories or full-sized application programs, are set in italic. The only exceptions to this rule are System file names (such as System, Finder, Laser Prep, Font/DA Mover). The first time a program appears in the text, the name of the company that makes it will be given afterward in parenthesis. If the program is what I call public-exchange software (shareware, freeware, public domain, that sort of thing), a note to that effect appears instead. Names, addresses and phone number of all the software publishers mentioned in the book are listed in Appendix C.

`File names` (such as `87.2.Budget` and `Story.txt`) and text that you should type as part of an example or instruction, are set in Courier, a typewriter-like typeface.

Command names (such as **Save** and **Hyphenate...**), Dialog box and Control Panel options, which can be turned on and off by means of check boxes or radio buttons (such as **Facing Pages** and **US Letter**) are set in boldface when used in instructions.

Menu names (such as the File and Edit menus), the names of keys on the keyboard (such as the A key and the Shift key), the names of dialog box "action buttons" (such as the Okay and Cancel buttons), and the names of text entry boxes (such as Right Margin and Font Size) are capitalized.

In addition, two icons are displayed at appropriate points in the margins.

The disk icon indicates that the program mentioned in the adjacent paragraph can be found on the Fully Powered Mac Utilities Disk that accompanies this book. Sources for programs not so marked and not on the disk can be found in Appendix C.

The Attention! icon indicates that an important piece of information is to be found in the adjacent paragraph or that a list of useful tips begins here. Use this icon to relocate a half-remembered tip or technique or to review the important points contained within a chapter.

A Quick Overview of What's Ahead

I should make clear at the very beginning that not everyone will want to take advantage of every technique that I describe. Indeed, this book is as large as it is because it offers a wide variety of options. Some of these options are mutually exclusive: you might want to use this technique or that technique, this piece of software or that piece of software, but not both of them. Other suggestions may not be to your taste or fit with the way you work. But from among the many strategies described you should be able to create a fully powered Macintosh that conforms to the way you work and enhances your productivity far more than your Macintosh does now.

It is also important to point out that, although it contains many options to choose from, this book is not just a grab bag of tips and techniques. It is, instead, organized around a series of strategies which flow logically from one to the next. Although you could jump around at random from one part of the book to another, if you want to create the fastest and most efficient computer, one that makes the most sense for you and the way you work—a fully powered Mac, in other words—then I recommend that you read the book from front to back. Not all in one sitting, but perhaps a chapter a week. As you read, I also recommend that you experiment on your Macintosh, applying the strategies I have suggested to see how well they work for you.

The book begins with a discussion of hardware in Chapter 1. Although you don't need a Mac II to create a fully powered Mac (and, in fact, some Mac Pluses may be closer to the fully powered ideal than some Mac IIs),

your Mac must meet the minimum hardware requirements outlined in this chapter before it can take advantage of the strategies described later in the book. I also strongly recommend two hardware additions, increased RAM and a hard disk, which will greatly enhance your ability to take advantage of the suggestions in the book, and review the choices that must be made when buying them.

File organization—the nemesis of many Macintosh owners, especially those who use a hard disk—is the logical place to begin construction of a fully powered Mac. Whether you use floppy disks or a hard disk, you won't work efficiently if you can't find files or if you bury applications deep inside a stack of folders. An incoherent jumble of folders and files can slow down not only you, but your Macintosh as well. Chapter 2 thus takes a hard look at folders and files and how to arrange them so that your applications are readily at hand and your files easy to locate. It also covers naming conventions, which can create order out of chaos, using icons views to prioritize your files and folders, window management, and hard disk subdirectories. To help the less experienced Macintosh owner understand the terms and concepts used in this and later chapters, Chapter 2 opens with an explanation of the System folder and the files it contains.

A properly and thoughtfully configured System file is, like good file organization, a basic requirement for the fully powered Mac. Which desk accessories and fonts you choose to install in your System file is especially important because the fully powered Mac always runs from the same System file; it does not change from System file to System file as you switch from application to application. Chapters 3 and 4, therefore, tell you what you need to know to put together the most logical System file for your needs.

For most of us, even with the advent of MultiFinder, desk accessories proliferate like hangers in a closet. They, along with FKeys (function keys), are the topic of Chapter 3. Thus this chapter includes an annotated list of many of the most useful desk accessories, including those that come into play elsewhere in the book. Since the System file allows you to install only a limited number of desk accessories, I describe strategies for getting around this limit. I also look at FKeys, describe many of the most useful ones, and show how you can create a new menu on the menu bar, an FKey menu, which allows you to call up any number of FKeys with the mouse.

Many Macintosh owners take fonts for granted, and as a result, frequently misunderstand and misuse them. Chapter 4 describes many little-known, but nonetheless important, points of font management: how they work, on-screen and on the printed page, on the ImageWriter, and on the LaserWriter, in resident and downloadable form; what kinds to use for

which purposes; how to get as many fonts as possible onto your Font menu and take up the least amount of disk space in the process; how to install fonts in applications and documents, and how to use fonts which are not installed in any file; and how to customize both ImageWriter and LaserWriter fonts.

In real life you arrange the top of your desk the way you want it, so it is only natural that you might want to alter your electronic desktop to suit your taste as well. In Chapter 5, I discuss a wide variety of such alterations, both serious and not-so-serious, useful and useless. Such things as altering the standard arrangement of icons in windows so that long names do not overlap; changing the amount of space provided for file names in list view windows; creating new keyboard shortcuts for often-used commands; putting a clock in the menu bar; and more.

As a group, Chapters 2, 3, 4, and 5 develop a strategy of efficiency and organization in the system architecture that lies behind every task you perform on your Macintosh. With Chapter 6 the emphasis shifts, to focus on the applications you work with, and how you use them. In effect, it shows how to create an integrated megaprogram composed of all your favorite applications. The first part of the chapter covers such things as keyboard commands and the ways in which they can be standardized across all your applications, so that one set of keyboard shortcuts works for all the modules of your megaprogram. The remainder of the chapter discusses the many ways you can integrate your favorite applications, and switch quickly and easily from one to the next and back again when computer memory is in short supply, and when it is not.

Chapter 7 is about the speed and efficiency gains you can realize by making your Macintosh work harder. In it, I show you how to automate the many procedures which most of us still perform manually: procedures such as automatically checking your electronic mail, or downloading fonts to the printer each time you turn on the computer. I describe inexpensive methods (RAM disks and RAM caches) that make the Macintosh quicker on its feet and discuss how to choose between them. I also look at print spoolers, which can shorten the amount of time your Mac is out to lunch during the printing process, and thus give you more work time on the computer.

The final chapter, Chapter 8, takes a look at the problems that can slow you and your Mac down, and shows you how to solve them. Problems such as file fragmentation, copy-protection, and the hardware and software failures that inevitably happen at the worst possible time. I also discuss the importance of backing up all your files—as insurance against hardware and software failures—and consider the fastest and least painful ways of performing this essential, and not especially pleasant, task.

Four appendices bring the book to a close. Appendix A describes the contents of *The Fully Powered Mac Disk*, which accompanies this book, and gives basic instructions for using the programs it contains. For those people who run into problems as they work through the book and, piece by piece, construct their fully powered Mac, Appendix B describes common difficulties and suggests ways to resolve them. Appendix C lists all of the programs mentioned in the book, provides the manufacturer's name and address should the program be unavailable in your area, and gives the retail price at the time this book went to press. Appendix D is a brief listing of useful references for further reading.

A Fully Powered Mac Roadmap

To help you navigate through the book, I've drawn up a list that appears on the following pages. Consider it a kind of roadmap that can help you remember places you've already traveled to but partly forgotten, or show you how to get to places you've yet to visit. Like any map, it lacks the details that make the trip itself so interesting, but I hope it will help those who have already been there relocate important points, and, for those who are just starting out, reveal the width and breadth of the territory.

- Choosing the right hardware for your fully powered Mac—Chapter 1
  - ◊ You can get away with two 800K floppy disk drives and 1 megabyte of RAM with a Macintosh Plus, SE, or II.
  - ◊ But you'll be a lot happier with a hard disk of 20 or 30 megabytes, or more, and a RAM upgrade to 2 or 2.5 megabytes, or more.
- Organizing your files and folders for speed and effficiency—Chapter 2
  - ◊ Use only a single master System file and folder unless your floppy disk setup makes this impractical. Keep your master System file small enough so that you can back it up (frequently) to a single floppy disk.
  - ◊ Group your programs and documents by function, task, application, application type, or whatever method best fits the way you work.

- ◊ Don't nest folders more than three or four deep, and keep your applications near the top of the organizational tree.

- Naming your files so you can find them again—Chapter 2
  - ◊ Name them so they can be easily called up from keyboard.
  - ◊ Use prefixes so the important files float to the top and the unimportant ones sink to the bottom of file selection lists.
  - ◊ Don't abbreviate file names so much that they turn into a kind of code; you might forget the code someday.

- What to do when you have more files than you know what to do with—Chapter 2
  - ◊ If you have a large capacity hard disk and are overwhelmed by files and folders, consider partitioning it into subvolumes and keeping open only the subvolumes you need at the moment.
  - ◊ If you feel overwhelmed by support files (dictionaries, default settings, help files, etc.), put them away in a folder somewhere and use *Set Paths* to tell the Macintosh where to look for them.

- Organizing you windows for speed and efficiency—Chapter 2
  - ◊ Use icon views when you want to arrange files by importance, and list views when you want them ordered by name, by date, and so on.
  - ◊ If you want to be able to find your way around efficiently, organize your windows. It doesn't really matter how; the important thing is that you do, indeed, organize them.

- Putting together the FKey part of your master System file—Chapter 3
  - ◊ Use *FKey Manager* to install FKeys in your System file and create a pop-up menu for them.
  - ◊ If you have installed many FKeys, use *QuicKeys* to assign keyboard commands to the FKeys that lack them.

- Putting together the desk accessory part of your master System file—Chapter 3
  - ◊ Even with MultiFinder, desk accessories make good sense when a small need doesn't require a big application and when memory is tight.

- ◊ To put more than 15 desk accessories under the Apple menu, install special-purpose desk accessories in applications and/or use font/desk accessory extender such as *Suitcase* or *Other....*

- Putting together the font part of your master System file—Chapter 4
  - ◊ Choose the right font for the right purpose. Use serif fonts for text and sans serif fonts for display type. Use decorative typefaces sparingly.
  - ◊ It may sound obvious, but it's true: to save disk space, only install the screen font sizes and styles you actually need and use. For the best quality bold and italic text, LaserWriter owners should use the bold and italic screen fonts (when they are available).
  - ◊ To conserve space, you can eliminate all but a few screen fonts per font family. But if you do this, you pay a price: screen readability and, to a lesser degree, screen accuracy and print quality.
  - ◊ If you have more fonts than Font/DA Mover knows what to do with, use a commercial font/desk accessory extender such as *Suitcase* to send your font menu through the floor.
- Dealing with downloadable PostScript fonts—Chapter 4
  - ◊ Know when the Macintosh should download downloadable PostScript fonts automatically and when you should download them manually.
  - ◊ Downloadable PostScript fonts can cause serious space problems, both on disk and in the LasserWriter. Drastic (or at least creative) measures are sometimes needed to solve them.
- Customizing screen and PostScript fonts—Chapter 4
  - ◊ With the **help** of *FONTastic Plus* or *ResEdit*, you can clean up and organize your font menu to make it easire to navigate.
  - ◊ Use *Fontographer* to add fractions, event a facsimile of your signature, to any existing PostScript font.
- Fine-tuning the Finder to fit your personal working style—Chapter 5
  - ◊ Use *Layout* to change the arangement of icons and lists in all your windows; use the font and font size used to display disk and document information.

- ◊ With *ResEdit* or *MenuMaster*, add keyboard shorcuts to often-used menu commands which lack them, or to change the name of an obscure menu or menu command.
- ◊ Add new keyboard shortcuts, with *QuicKeys* or *Tempo II*, for launching applications and/or specific documents, window management, responding to dialog boxes, and so on.
- ◊ Move characters to other, more convenient keys with *QuicKeys* or a keyboard customization program such as *MacKeymeleon*.
- ◊ Custom desktop patterns, scroll bar patterns, icons, and much more are also possible.

■ Customizing your applications so that they work together better—Chapter 6

- ◊ Develop an overall strategy for keyboard shortcuts and macros—before you become overwhelmed by them.
- ◊ With *ResEdit* or *MenuMaster*, standardize the keyboard shortcuts for the most common commands in all your applications. This eliminates the potential for unpleasant surprises (and makes everything easier to remember).
- ◊ Use *Set Default Font* so that all your applications start with the same font in newly opened documents.
- ◊ Use macros to automate most any task that you perform repeatedly.

■ Using your applications as if they were a single integrated program—Chapter 6

- ◊ If your Macintosh is short on memory, then use a fole transfer utility (such as *Skipfinder* or *SafeLaunch*) or a finder-substitute (such as *Oasis*, *SuperFinder*, *PowerStation*, or *HyperCard*) to speed up the transfer from one applilcation to another.
- ◊ *Switcher* and *Servant* are two alternatives to MultiFinder. Although both have their advantages, neither are recommended for general use because of important shortcomings.

■ Putting your Macintosh on "automatic pilot" and getting it to run faster—Chapter 7

- ◊ Automated startup routines created with *StartMac* or *Tempo II* are a simple and easy way to free yourself from the drudgery of preparing your Macintosh for work every morning.

- ◊ Mouse-oriented Macintosh owners can enhance their speed and efficiency with utility programs such as *HFS Navigator* and *Icon-It!*
- ◊ You can shorten the time your Macintosh is tied up with the printing process by using a printer spooler.
- ◊ By placing your applications and/or System folder in RAM, RAM disks and RAM caches are the most effective way to increase the speed with which the Macintosh works.

■ What to do about flies in the ointment—Chapter 8

- ◊ Copy-protected programs can be a real obstacle to honest attempts toward greater speed and efficiency. Avoid them, or use a program like *Hard Disk Util* or *Copy II Mac* to convert them into a properly-behaved program.
- ◊ Over time, the files on your hard disk can become badly fragmented and greatly slow down the disk's performance. To de-fragment them and free up lost disk space, use a disk optimizer like *DiskExpress*.
- ◊ Because of the fragile nature of magnetic media, backing up your floppy or hard disks is essential. Do it regularly and be organized about it.
- ◊ If a file won't open, a disk becomes unreadable, or your hard disk crashes, recover lost data from your backup set . If you don't have a backup, *1st Aid Kit* may be able to recover the lost file or files.

1

Hardware and the Fully Powered Mac

The Minimum Hardware Setup *2*

The Ideal Hardware Setup *5*
- Additional RAM *5*
- Hard Disk Drives *9*

More RAM versus a Hard Disk *15*

Getting a Mac II without Buying One *15*

Summary *17*

The fully powered Macintosh is a finely tuned combination of software and hardware. The bulk of this book is concerned with the software part of the operation: what you need, and how to use it to make your Macintosh faster, more efficient and easier to work with. This chapter, however, deals with hardware: the configurations of system unit, RAM and disk drives which will allow you to create your own fully powered Macintosh. It is written especially for those people who are intending to buy a Macintosh (their first one or their next one), a hard disk, or a RAM upgrade. But even if you think you have all the hardware you need, there is information in this chapter that may be useful to you.

Although the newer Macintoshes are, in fact, faster than their predecessors, you don't need a Macintosh II or all kinds of exotic paraphernalia to create a fully powered Macintosh; you can, in fact, create one with a basic Macintosh Plus. But to get the most out of this book and to really experience what a fully powered Macintosh can do, your computer should have at least two megabytes of RAM or a hard disk drive or, best of all, both.

Although almost any RAM upgrade or hard disk will do, some RAM additions and hard disks will do better than others. If you are in the market for extra RAM or a hard disk, you may already know that both purchases require that you sort through many options and make a number of decisions. This chapter should help you make the right decisions. If you already have a hard disk, or a RAM upgrade, or both, here's hoping that "read it and weep" won't apply to you.

The Minimum Hardware Setup

At the time this was written, the three basic, unadorned, off-the-shelf Macintosh systems were as follows:

- Macintosh Plus: 1024K (or one megabyte) RAM, one 800K floppy disk drive
- Macintosh SE: one megabyte of RAM, two 800K floppy disk drives
- Macintosh II: one megabyte of RAM, one 800K floppy disk drive

The absolute minimum, bare-bones system for constructing a fully powered Macintosh is:

- Macintosh Plus, SE or II, one megabyte of RAM, two 800K floppy disk drives

Although, as you can see, neither a hard disk nor extra RAM is an absolute necessity, I greatly encourage you to purchase at least a hard disk, for reasons that will become clear shortly.

Of the three basic Apple machines, all have the minimum RAM required for the strategies discussed in this book. All except the Mac SE require the addition of a second floppy disk drive (if a hard disk drive is beyond your means). For the Mac Plus, you must purchase an external floppy disk drive. For the Mac II, the second floppy disk drive may be accommodated internally.

For a variety of reasons, summed up by the sentiment that the old 512K machines are essentially the last remnants of a bygone Macintosh era, I don't recommend that you purchase a (used) 512K or 512K Enhanced Macintosh if you have yet to buy your Mac and are looking for a bargain. But if you already have one, there is no reason to feel you have to throw it out and get a new Plus, SE, or a II. How much RAM is really enough (in the context of an ideal system), is discussed in the following paragraphs, but for the minimum system, you should, at the very least, bring your older Macintosh up to the Mac Plus base line. Upgrades for the Macintosh 512K and 512K Enhanced are widely available from a number of sources. Some do no more than turn your machine into a basic Macintosh Plus and leave it at that, but others provide additional amounts of RAM and sometimes other features as well.

If you cannot afford a hard disk, you must have a second 800K floppy disk drive for your Macintosh Plus or II. At press time, there were few alternatives for internal floppy disk drives for the Mac II other than the one sold by Apple. External floppy disk drives for the Plus are another matter, however. You can choose standard Apple issue, or you can shop around for a better price and a better warranty (some manufacturers give one full year; Apple's is only 90 days). Other points to consider when buying an external floppy disk drive are noise and the size of the case, both of which can vary considerably from one maker to the next. Drive speeds can also differ, but rarely in a significant way. Consult *Macworld* (the November 1986 issue, for example, and more recent issues as well), *MacUser*, and other Macintosh magazines for up-to-the-minute comparisons of the price, size, noise, speed, and warranties of the many floppy disk drives on the market.

Even if money is a serious consideration, do not be tempted to buy a 400K floppy disk drive (for single-sided floppy disks) instead of the 800K drive. Buying one (probably a used one; few, if any, are being manufactured at this point) may save you a little money. But a 400K drive will slow you down (because the drive is slower than an 800K drive), and force you to insert and eject disks twice as often (since 400K disks hold only half as much

as 800K disks). Worse is the fact that you'll be using two different types of disks, double-sided disks for the 800K internal drive and single-sided disks for the 400K external drive. Aside from the confusion two disk formats can cause, your 400K drive will only read from and write to single-sided disks. If you put a double-sided disk in a 400K drive by mistake, you run the risk of replying to the ensuing dialog box ("this disk is unreadable,"etc.) incorrectly and erasing the disk by accident. Don't laugh; such mishaps happen all too easily. Listen to the voice of experience.

The minimum fully powered system — two 800K floppy drives, no hard disk, and one megabyte of RAM — has both advantages and disadvantages. The advantages of this minimal setup boil down to two. One is price; it is decidedly cheaper to go the bare-bones route. The combination of hard disk and additional RAM that are required for an optimal setup will cost you at least an extra $1000, and possibly quite a bit more. The other advantage is that the minimum system puts less stress on the brain as well (at the time of purchase, anyway). Simply walk into any Apple dealer, and walk out with a Mac Plus or II and second floppy disk drive (of one brand or another) or a Mac SE. It's that easy.

The disadvantages of the minimum setup, however, cannot be ignored. Let me put it this way: if you are working with the minimum setup, your Mac is unlikely to be as fully powered as you would like. It will certainly not be as fully powered as it could be. With the bare-bones setup, you will have to spend more time working out disk strategies: deciding which program or file to put on which disk to optimize your work speed. You'll have to swap disks more often. You won't be able to create a master System file of monster proportions; you may, in fact, have to resort to several alternative master System files. You probably won't be able to take full advantage of the speed gains that are possible with a RAM disk, and so on.

Make no mistake; with the minimum setup it is quite possible to make good use of the strategies described in this book. You can still reap increased efficiency, speed, and convenience. But you will find yourself having to compromise. It will require a certain amount of discipline to maintain your software system so that it works at peak efficiency, and following the suggestions in this book will demand a little more creative effort on your part. My recommendation? Shell out the extra money and buy the hardware discussed in the next section. You'll be glad you did.

The Ideal Hardware Setup

The ideal fully powered Macintosh begins, as before, with a Mac Plus, SE, or II. (Actually, if money were no object, the ideal system would begin with a Mac II; but money rarely is no object.) It also includes two hardware options over and above the basic system just described. In addition to the standard one megabyte of memory, the ideal setup has at least one more megabyte of RAM to accommodate a memory-intensive, multi-program environment such as MultiFinder, memory-hog applications such as *HyperCard*, or the speed-enhancing strategies described in Chapter 7. Instead of (or, if you're feeling extravagant, in addition to) the second floppy disk drive, the ideal fully powered Macintosh has a hard disk drive for mass storage. (Some would say that a hard disk is a necessity; I, on the other hand, would call it a near necessity.) Both options come in a variety of forms and confront the purchaser with several important decisions; namely, what kind is best, and how big is big enough? The remainder of this chapter looks at these two hardware options and discusses how much is enough, what kind is best, and other essential questions.

Additional RAM

RAM (for random access memory), commonly referred to simply as memory, is the place where the Mac does its work, where programs actually operate, and where documents under construction are kept until saved to disk. The first Macintoshes came with 128K of RAM, an impossibly small amount by today's standards. It didn't take long for 512K to become the new standard memory size, and then, with the introduction of the Mac Plus, one megabyte (1024K) became the sine qua non. But the Mac Plus was built with four megabytes in mind, and the Macintosh II can contain up to eight megabytes of RAM on the main circuit board, and with the proper software and additional add-on boards, the Macintosh II can utilize as much as four gigabytes (4000 megabytes) of total RAM.

As I pointed out earlier, neither 128K or 512K are adequate memory for a fully powered Macintosh; one megabyte is the minimum. But just how much is really enough?

Is One Megabyte Enough? If you take a quick look at Chapter 7, for example, it demonstrates rather conclusively that one megabyte of RAM severely limits your ability to increase overall speed by placing applications and/or System files in memory and running them from there. Some of the strategies described in Chapter 6, strategies for creating your own integrated programs and for increasing efficiency, require more than one megabyte to attain their full power and flexibility. In addition, an increasing number of large applications (including *HyperCard*), especially when large documents or multiple documents are open, are bumping up against the one megabyte barrier.

Consider Microsoft *Word* 3.0, for example, the program with which this book was written. The program itself is 332K in size, the spelling dictionary is 161K and the help file is 113K. If you put the whole program and a 100K document in memory (an option which *Word* gives you to speed things up), and then select the check spelling command, which loads the dictionary into memory as well, the result is nearly 600K of RAM devoted solely to *Word* files. Add to that some room for the parts of the System file that are kept in memory during normal application operation, some room for *Word* to work in (saving text cut to the Clipboard, and so on), and that leaves precious little RAM left over. Certainly not enough to run another application, even a small one, at the same time under MultiFinder. It's not enough, in fact, for many of the suggestions made in this book.

If one megabyte is not really enough, and four gigabytes is far more than most of us could ever justify, just how much RAM is enough? Now that the Macintosh II has made it possible for us to talk about thousands of megabytes, saying as much RAM as you can get your hands on (which was for many years the standard reply to the question) is no longer a particularly useful answer. But the sentiment is still valid, for no matter how much RAM you get, you always seem to need or want still more.

For the most precise answer to the question, you should read through this entire book, consider the strategies you want to make part of your work routine, and then do the necessary arithmetic. For example, say your everyday work configuration requires room for two large applications under MultiFinder, plus enough room left over to put your System folder in RAM to speed up overall operation. After working through Chapters 2, 3, and 4, you end up with a master System folder of 550K. Chapter 7 tells you that to copy it to RAM you should reserve approximately 600K. After reading Chapter 6, you decide to run *Word* and Microsoft *Excel* under MultiFinder. The absolute bare minimum for *Word* and *Excel* is about 800K (304K for *Excel*, 416K for *Word*, and a little extra for good measure). Add a conservative 300K for MultiFinder and the System file to work in, for a

minimum total of 1.7 megabytes (600K, plus 800K, plus 300K). Playing with the numbers like this is the best way to figure out how much RAM you really need.

A somewhat less precise answer to the question is to buy as much memory as you can afford—and hope that it is more than you currently need. If you determine you still need more, there's always the possibility of buying more sometime in the future (when it is cheaper). If you didn't buy enough, at least you have the satisfaction of knowing that you bought as much as you could. If you bought more than you need now, you'll find a use for it sooner or later (usually sooner).

The most practical answer may be the easiest way to look at the problem. Consider first the options for each machine. The Mac Plus and the SE can accommodate upgrades to 2.5 or 4 megabytes. The Mac II can accommodate upgrades to 2, 5, or 8 megabytes—or more, if it is added by means of add-on boards. (Two other options, 2 megabytes for the Mac Plus and SE, and 4 megabytes for the Mac II, are possible, but not sensible, since they involve the removal of RAM that is already present and paid for.) For reasons detailed below, these are the only options.

Thus the jump from 1 megabyte to 2 or 2.5 megabytes is the minimum upgrade you can, and should, consider. With 2 or 2.5 megabytes of RAM, you can incorporate almost all the strategies outlined in this book into your work routine, without needing to compromise. But if you want real freedom, the ability to use MultiFinder or *HyperCard* without restraint, the ability to do just about anything, including using RAM a little wastefully, if you want enough RAM for the future as well as the present, then only 4 megabytes will do.

Despite the existence of super-powered RAM upgrades well in excess of four megabytes, four megabytes of RAM is probably the practical outer limit for the Mac Plus. Although the Mac SE and the Mac II can handle more than four or five megabytes relatively easily, for most of us and for all but the most high-powered forms of computing, anything over four or five megabytes is likely to see little use. If you do own a Mac SE or II, you know if you are in the high-powered computing bracket (engineering applications, for example), know your programs and their memory requirements, and know whether or not you need more than four or five megabytes. For the rest of us, however, more than four or five megabytes is, at the moment, more of a luxury than anything else.

How much RAM does my own Macintosh have? Well, I've worked with Macs with 128K, 512K, 1 megabyte, 2 megabytes, 2.5 megabytes, and 4 megabytes. My current Macintosh has 4 megabytes. Although I don't use all of that memory all of the time, I do use almost all of it frequently, and I even

bump up against the limit on occasion. And I do so not because I'm some sort of eccentric Macintosh user, but because I practice what I preach in this book. All of which makes me feel that 4 or 5 megabytes is an appropriate amount of RAM for a hard-working fully powered Macintosh which looks to the future—at least for now.

Adding RAM: How and What Kind Adding memory to the Mac Plus, SE and II is easy. Now that the days of home-grown memory upgrades are largely over, you should be sure to use the basic memory modules, called SIMMs (single in-line memory modules), for which these machines were designed. SIMMs come in two sizes, 256K and one megabyte. The basic configurations for all three of these machines include four 256K SIMMs, for a total of one megabyte of memory.

Since the Mac Plus and SE have only four SIMM sockets, you must replace existing 256K SIMMs with one-megabyte SIMMs to increase RAM. (Actually, you should take your computer to your dealer and get them to do the replacing. Fooling around inside a Plus or an SE is not for the uninformed, and RAM upgrades on both a Plus and an SE require a minor hardware modification as well.) SIMMs must be replaced in pairs on the Plus and SE. Pop out one pair of 256K SIMMs and snap in two 1-megabyte SIMMs to get 2.5 megabytes (the sum of the two new 1-megabyte SIMMs plus the two existing 256K SIMMs). Or replace all four original SIMMs to get four megabytes (from four one-megabyte SIMMs). Since the Mac Plus does not have a fan, if you have the choice you should specify CMOS SIMMs; they produce less heat than the others.

The Mac II has two advantages over the Plus and SE in regard to basic memory expansion. One is the fact that you can easily perform the upgrade yourself, unless fussing around with disk drives to get to the SIMM sockets gives you the jitters. (Inserting SIMMs on the Plus and the SE is not all that difficult either; it's getting to the place where the SIMMs go that's the tough part.) The other advantage is that the Mac II has eight SIMM sockets instead of four, which means that you can add four more SIMMs (of either size) before you have to start replacing existing ones. The acceptable combinations of SIMMs result in possible memory configurations of up to eight megabytes.

In the Mac SE and II, you can increase RAM to even greater heights in another way: the insertion of expansion cards in the SE's single SE-Bus slot or one or more of the Mac II's six NuBus slots. By combining additional RAM and special memory management software, such cards can increase the SE's RAM to eight, sixteen, even more megabytes. The Mac II already has one of two types of memory management software and requires only

the insertion of add-on boards containing additional RAM to get into the gigabyte range. As I mentioned earlier, however, expansion over and above four or five megabytes is, for the moment, likely to benefit only the most demanding applications.

Hard Disk Drives

RAM is where programs and documents are held while you work with them, but the so-called mass storage devices are where your files are stored for the long term. Mass storage devices include floppy disks, hard disks, tape drives, CD-ROM, and other, even more esoteric items. The two most widely available and most affordable mass storage devices are floppy disks and hard disks. Floppy disks (which are floppy on the inside but not, in Macintosh disks, on the outside) are used by, and familiar to, everyone; applications software is sold on floppies, files are passed from office to office (for example) on floppies, everybody's got floppies. Hard disks work on the same basic principle that floppies do, except that the disk platter itself is no longer flexible. By sealing the platter (or platters) inside the disk drive, you can increase the speed, precision, *and* delicacy of the whole operation, and end up storing far more data in the same amount of space.

In the years since the Macintosh first appeared, hard disk drives have changed from an expensive status symbol to a near necessity. The reasons for this dramatic about-face are obvious to anyone who has worked with a hard disk. Hard disks can store the equivalent of a shoeboxful of floppy disks (or considerably more), and in the process provide organization and accessibility beyond the means of even the most well-ordered shoe box. They eliminate the need to swap disks when you change applications or documents. They are fast because you can find your files more quickly, and because hard disks can locate and read information far faster than a floppy drive can. In short, they are a miracle of the modern age.

But modern-age miracles have their price, and hard disks are no exception. For one thing, they are expensive. As time passes, they get less and less expensive, but they still cost more than a reasonably priced floppy disk drive and a bulk order of floppy disks. Hard disks also break down (rarely, these days, due to the continual increase in overall hard disk quality; but they still do fail on occasion). Floppy disks break down too, of course. But when floppy disks do go to electronic heaven, they take with them no more than what fits on a single floppy disk. When hard disks die, however, the results are much more dramatic: the contents of the entire disk, which is to say, your entire collection of applications and documents,

perhaps your entire life's work, plus every piece of software you own, can disappear in a flash. Which leads to the third major drawback of hard disks: they must be backed up frequently to avoid the kind of data loss a disk crash can cause. Floppy disks should always be backed up too; it's just that backing up a hard disk is a considerably larger pain in the neck, and it can take a lot of time if you back up to floppy disks.

Despite the drawbacks to hard disk ownership, however, the benefits clearly outweigh them. If you have the money to spend, the real question is not if you should buy a hard disk, but how soon, how big, and what kind. Your frustration with the floppy disk shuffle will dictate the answer to the first question. The following sections will help you answer the other two.

Hard Disks: How Big Is Big Enough? Unlike floppy disks, which come in only two sizes, hard disks are available in a wide range of sizes—and prices. Although the ten-megabyte hard disk was once the standard size (and for a while, the only size) hard disk for the Macintosh, twenty megabytes is about as small as they come these days, and for good reason: ten-megabyte hard disks are just too darn small. After all, it takes only thirteen 800K disks to equal the capacity of a 10 megabyte hard disk, and thirteen disks are not that hard to keep track of (nor are they anywhere near as expensive as a hard disk). On the other end of the scale, you can get a hard disk which holds hundreds of megabytes, or even a gigabyte. Hard disks of this size are big brutes, however, and cost big bucks. Unless you plan to share a hard disk among a number of users or have a mammoth collection of files to store, you can safely ignore the hundreds-of-megabytes set.

So how big is big enough? Consider the machines in my office, for example. I have a ten-megabyte hard disk in an IBM-compatible Compaq, an original 512K Macintosh with a ten-megabyte hard disk, and a Mac Plus with a twenty-megabyte hard disk. I have never run out of room on the hard disk in my Compaq; indeed, I have only recently managed to pass the five megabyte mark. The odd thing is, I *always* run out of room on the ten-megabyte disk in my 512K Mac. Conversations with other computer users indicate that my experience is not an anomaly. One reason for this may be that, with the Macintosh, it is easy to add all sorts of extras to the System folder (see Chapter 2), thus using up as much as two megabytes or more for the System folder alone. The directory (as they are called in the IBM world) that contains the system files on my Compaq isn't anywhere near that size and, just to set the record straight, its capabilities aren't anywhere near as great either. A second reason for the disparity in disk usage may be that Macintosh people tend to use a wider variety of applications, since they are

easy to learn and are (or at least used to be) not as expensive as PC software, but IBM people tend to use only a few.

Whatever the reason, a ten-megabyte hard disk is clearly just not large enough for most people. Twenty megabytes is the minimum size available these days and is probably only large enough for moderate use. I used a twenty-megabyte hard disk while writing this book, and I frequently ran into the limit of its capacity. Thus, if you use your Macintosh seriously, if you run a small business on your Mac, maintain large database files or spreadsheets, mailing lists, inventories, and so on, in addition to everyday applications and files (writing, graphics, telecommunications, that sort of thing), then twenty megabytes is probably not enough. Thirty megabytes may sound like an awful lot, but it is my best guess for the minimum hard disk size for the serious Macintosh owner.

One way to figure out just what size hard disk you need is to add up the disk space requirements of everything you want to store on it. Read over the next three chapters and try to estimate just how large your System folder (System file, plus printer drivers, downloadable font files, and such) will be. Add to that all your applications and their associated files and record the total. Then think about the kind, number, and size of the files you'll want at hand (you can store archive copies of old documents elsewhere). Take that last number, double it—people always underestimate the size and abundance of their files—and then add it to the total.

Think carefully about the number you have come up with. Say, for example, that you arrived at a total of fifteen megabytes for everything: System files, applications, and active documents. Considering the fact that you actually doubled the size required for documents, you might well surmise that you could get away with a twenty-megabyte disk drive fairly easily. However, that's a little like building a library large enough to hold only the books you already own; add a few more books and you'll be thinking about building another library. It makes much more sense to think of a library and a hard disk as something that gets added to continually. Thus, unless the cost is a serious issue, you should plan for the future and consider your fifteen megabyte total as a starting point, not an end point. Thus, rounding up to thirty megabytes, or maybe even forty megabytes, is not unreasonable.

In addition to thinking about the future, you should also consider that, as a rule, the cost per megabyte (the retail price divided by the size of the drive) goes down as the capacity of the drive goes up. For example, when this was written, Apple's smallest external hard disk cost about $65 per megabyte, and the largest model cost about $40 per megabyte. In a sense, you save money by buying a larger hard disk, one which presumably anticipates future needs.

Another point to remember, as you consider what size hard disk to buy, is that if you outgrow your hard disk, you don't have to replace it with a bigger one. You can add a second hard disk to your existing setup and continue to use the first one as well, regardless of whether it is internal or external. If the original disk drive is internal, you just hook up the new one to the SCSI port. If the original is external, you simply chain the new one to the second SCSI port on the original. The advantage to this, of course, is the fact that you do not lose the investment or the storage capacity in the first hard disk drive. The disadvantages are that the two drives will appear as two distinct entities on the desktop, you don't take advantage of the cost per megabyte value of a single, larger hard disk, and you now have yet another box attached to the back of your Mac. Fortunately, none of these disadvantages is especially serious.

Remember, too, that running multiple disk drives from a single Mac requires that each drive have a unique ID number. This is no problem when running a single drive, but a second drive can cause problems if its ID number is the same (and they often are, since many drives are delivered with an ID of zero). Additional drives must be assigned an ID when they are first attached, usually via a switch in the back (either on the case itself, or inside a small door) or a software utility. A few require that you open the case and set a switch on the inside. If you think you might want to chain disk drives sometime in the near or even distant future, do yourself a big favor and be sure to buy a drive that doesn't require a genius (or someone with the right tools) to set the ID number.

What Kind of Hard Disk to Buy As one who has watched several of his own hard disks self-destruct, I now believe that the only criterion worth considering is reliability. The color and size of the case, how quiet the drive is, whether it sits beside, behind, under, or in your Mac these are all valid concerns. But they mean very little if the disk drive can't stand up to the normal wear and tear of everyday usage, or it contains a fatal flaw somewhere. Disk crashes should be avoided like the plague, and the best way to avoid one is to buy a reliable disk drive.

Disk drive technology is getting better all the time, but it pays to research your drive decision carefully. Get as much information as you can from friends, users groups, and others, on which disk drives tend to have problems less frequently and which manufacturers are most responsive to problems when they do occur. Read reviews. Do your homework *before* you go out and buy a disk drive.

Internal versus External Hard Disk Drives No matter which model Macintosh you own, you have the option of fitting it out with either an

external (SCSI) or an internal hard disk drive. Only the Mac SE and the Mac II, with their internal fans, extra vents, and who knows what else, were designed with an internal hard disk drive in mind, however. My experience with internal hard disks in the Mac Plus indicates that retrofitting the Plus with an internal hard disk drive is not a good idea.

If we set aside the retrofit issue for the moment, the external versus internal decision is not an easy one. On the one hand, internal hard disks are, quite simply, more convenient to work with. There is no separate power cord to plug in, no second switch to turn on, no extra box to look at or worry about bumping into. For those who move their Mac around a lot (I'm not referring to the Mac II here), having the disk drive tucked away inside the computer means that the whole setup still fits in a single carrying case —although it is quite a bit heavier. All in all, internal hard disks are elegant and seductive.

External hard disks, on the other hand, have their good points as well. You can share them simply by pulling the SCSI plug on one Mac and sticking it on another one. Even more important, if an external hard disk breaks down, you simply unplug it and take it to your dealer for repair. Your Mac stays on your desk, where you can continue to work with it, a little more slowly perhaps, using floppy disks until your hard disk returns. Internal hard disk failures take your computer with them and put you out of commission entirely. If you can't work without your Macintosh, and if you don't mind an extra box attached to the back of your Mac, this last point is worth considering, even for Macintosh models which were meant to house internal hard disks.

If it were my decision, I would probably select an internal drive for the Mac II and an external one for the SE. I would choose an internal one for the Mac II because it was designed to hold internal hard disks manufactured to certain specifications, because an errant hard disk can be removed with relative ease to take to the repair shop, and because the machine is big and bulky enough already. I would choose an external one for the SE because the machine itself is no bigger than a Plus (and I already know that carrying an external drive with a Plus is not a great inconvenience), because it makes me feel more secure knowing that a hard disk failure won't take my computer with it, and because the number of internal hard disk drives to choose from is much smaller than the number of external drives (as of this writing, at any rate).

For the Mac Plus, there is the added concern that it was never designed to hold an internal hard disk drive. For some, especially those who have undiminished faith in the ability of third-party developers to overcome the shortsightedness of Apple planners, this is an issue of little impor-

tance. For others, who have suffered through the reality of flakey internal disk drives, this is no academic argument. As a member of the latter group, I strongly recommend that you use only external hard disk drives with the Mac Plus.

Removable versus Fixed Hard Disk Drives Most hard disk drives are all alike (from the outside, anyway). There may be minor differences in the placement of the power switch (in external drives) or the presence or absence of power and in-use lights. Otherwise, they are all equally inscrutable and sealed off from the curious. A few hard disk drives, however, (if *hard disk* is the correct term) employ removable disks protected by special cartridges. From the point of view of the user, they are a lot like floppy disk drives, except that the disk cartridges are thicker and heavier, each disk holds ten or twenty megabytes, and one removable cartridge costs a lot more than a floppy disk.

The advantages of removable-media hard disk drives are obvious. With the purchase of a single drive, you get an unlimited storage capacity: to expand the number of files you can access from the drive, all you have to do is pull out the old (and presumably full) cartridge and insert a new one. And if you get a drive with two cartridge slots, you can quickly and easily make a backup of the cartridge in one slot with the cartridge in the other slot, eliminating the need for floppy disk or tape backups.

Although removable-media hard disk drives are about as fast as standard (fixed) hard disk drives, they do have their drawbacks. Discounting their expandability for the moment, the cost per megabyte of removable-media drives is relatively high. The cartridges themselves are also expensive, and they are more likely to wear out over time than a traditional hard disk. In addition, removing and inserting cartridges increases the likelihood of dust getting onto the disk, and dust on a fast-spinning disk is the equivalent of gravel on a phonograph record. The precise placement of the read/write head is also important in disks that pack such a great amount of information into such a small space. Maintaining this precision in removable-media drives may be more difficult, and (for certain types of removable media) more prone to problems.

Many removable-media hard disk drive owners swear by their drives, and there is no reason to feel that their enthusiasm is anything but genuine. Their rarity in the Macintosh world, however, means that the jury is still out on the advantages versus the disadvantages. I, for one, intend to wait for the jury's verdict before I invest in one.

More RAM versus a Hard Disk

Although some have the where-with-all to buy the Macintosh of their choice, a hard disk and lots of RAM all at once, and others barely scrape enough together for the basic setup, there are many whose finances fall somewhere in between. For this latter group, the question is which should I buy first, more RAM or a hard disk?

There are good arguments on both sides, but my instinctive reaction (and my considered opinion) is to buy the hard disk first. Why? Because a hard disk makes software organization much easier; because a hard disk eliminates the floppy disk shuffle; because a hard disk eliminates the 800K limit to the size of your System folder (which must fit on a single floppy disk if you do not have a hard disk); and because, with a hard disk, you will see a dramatic improvement in the time it takes to open, close, and (in many instances) work with your applications. Added RAM may have many benefits, not the least of which is using MultiFinder to good effect, but the benefits of a hard disk are so varied and pervasive that it gets my vote for the first purchase on a limited budget. Just try not to wait too long before you buy more RAM.

Getting a Mac II without Buying One

Although there are many hardware enhancements available in addition to hard disks and RAM upgrades, far too many to discuss here, one in particular deserves brief mention before we move on to the business at hand. If you have a Mac Plus or a Mac SE, and you are looking for major speed improvements (above and beyond the software-mediated speed enhancements described in Chapter 7), you should look into add-on accelerator boards which, in essence, turn your computer into a Macintosh II dressed in Plus or SE clothing.

All accelerator boards, such as the Levco Prodigy 4 for the Plus, and the Levco Prodigy SE and the General Computer HyperCharger 020 (among others) for the SE, replace the Motorola 68000 microprocessor of the Mac Plus and SE with the 68020 microprocessor found on the Mac II. Depending on which board and/or options you select, the board may also contain a Motorola 68881 math coprocessor (for far speedier math-intensive operations), a Motorola 68851 Paged Memory Management Unit (which

permits memory expansion above four megabytes), and additional RAM. Fully loaded, an accelerator board can speed up the performance of your Mac SE or Plus from two to four times what it was before and, in some cases, even outperform the Mac II.

Raw speed isn't everything, of course, but when it comes to redrawing a complex graphic, scrolling a window, recalculating a large spreadsheet, and many other things, I have to admit that it is a lot better than what's in second place. Chopping off a few seconds of wasted wait time during every minute's worth of work can really add up. Accelerator boards offer other advantages as well: the power of a Mac II in a box the size of a Plus, for one thing. Another advantage, if you already own an SE or a Plus, is that it's considerably cheaper to buy an accelerator board than a whole new computer.

Other advantages are perhaps less obvious. If you are using software that is incompatible with the Macintosh II, some boards allow you to switch to standard Mac Plus or SE operation—something a Mac II cannot do. Others have the ability to turn off part of the 68020 (what is called the *instruction cache*), which often causes compatibility problems. The result is a slight performance penalty, but it also means that you can run some applications on an accelerator board that won't run on a Mac II. Several boards also have special versions of Apple's numeric calculation routines (called SANE) that are faster than Apple's own version, and a few of these allow you to switch between the two.

Unfortunately, there are some drawbacks to accelerator boards as well. For example, should you want to add a large-screen display, you may be restricted to a choice of one or two displays (or possibly none at all). If the board does not offer the option of switching back to plain Mac Plus or Mac SE operation, then any software that will not run properly on a Mac II will not run on the board either. And with a computer/board combination whose total cost is not much less than a basic Mac II, you end up with a small screen, no color, and a limited ability to accept additional boards.

If you already own a Mac Plus or SE but hanker for a II, and if the reach to a Macintosh II exceeds your monetary grasp, then you should definitely consider purchasing an accelerator board. But if you have yet to buy a Macintosh, have a budget big enough to accomodate your wildest dreams, and are not overly concerned about portability and disappearing desk space, then I suggest that you forget about accelerator boards and buy the Macintosh II.

Summary

- The absolute minimum, bare-bones system for a fully powered Macintosh is a Macintosh Plus, SE, or II, one megabyte of RAM, and two 800K floppy disk drives.
- You'll be much happier, and the strategies described in this book will be much easier to implement, if your Macintosh is equipped with something more than the basic setup. Especially important are:
  - ◊ A hard disk of 20 or 30 megabytes, or more.
  - ◊ A RAM upgrade to 2 or 2.5 megabytes, or more.
- If money is tight and you only have enough right now for either a hard disk or a RAM upgrade but not both, get the hard disk.

2

Files, Folders, and Organization

The System Folder *21*
- Inside the System Folder *22*
- Version Numbers and Keeping Up-to-date *28*
- Working with a Master System Folder *30*

Organizing Your Files *33*
- Organizing Files on Floppy Disks *35*
- Organizing Files on a Hard Disk *40*
- Making Your Organizational Style Work on Screen *51*
- Naming Files and Folders *59*

Summary *66*

Anyone who spends even a moderate amount of time working at a computer accumulates a lot of documents. Full-time computer users often generate an endless stream of files: letters, reports, data files, spreadsheets, memos, you name it. Add to that applications, utilities, system files, and such, and the total number of files can become enormous. Unfortunately, allowing an enormous number of files to get out of control is relatively easy. Organizing them in a way that allows you to find the files you want quickly and efficiently is far more difficult. But if you want the speed, convenience, and efficiency of the fully powered Mac, an organized set of files is an essential first step.

How often I've seen hard disks in total disarray—files, folders, and applications all crammed together in the same window. Folders with ambiguous names, and files with incomprehensible ones. Documents stacked at random, often covering other files underneath. Narrow windows containing a wide arrangement of files, thus forcing you to scroll the window from side to side. Wide windows with a narrow column of icons, needlessly covering windows underneath. With disorganization on a scale such as this, it's a wonder that more disk drives don't collapse under the sheer burden of inefficiency.

Unfortunately, it's the same old story almost everywhere. Because, given enough time, Macintosh owners can almost always find an errant file, they tend to ignore the fine points (or the basics) of file organization, and overlook the time wasted as a result. You've probably used the electronic equivalent of the classic excuse, "I realize my desk looks like a mess, but I know where everything is, really I do," a few times yourself. True to form, a number of the people who reviewed the manuscript of this book initially commented that a chapter on file organization was, to put it bluntly, a waste of everyone's time. Yet by the time they had finished the book, they had changed their minds completely. Good file organization, they soon discovered, is one of the cheapest and least complicated methods for increasing efficiency and speed. Their only complaint was that other readers might be tempted to skip this chapter and miss out on what is clearly an important part of the fully powered Macintosh.

The truth of the matter is that the advantage of a well–organized filing system is not just that it looks neater. Good organization means that you are less likely to lose a file. It means that files can be located quickly because they are stored where you expect them. If certain file–naming conventions are followed, it means that files can be called up from the keyboard with only a few keystrokes. In general, the better organized your

files are, the more easily and efficiently your work will proceed. Thus the bulk of this chapter deals with the much neglected, frequently maligned, and not-so-obvious subject of folder and file organization.

But before it gets into the organization of your files in general, however, this chapter takes a closer look at a few files in particular: the files in the System folder. The System folder is the place where a lot of work goes on behind the scenes, and as a result, its contents are often completely ignored. Thus the following section considers what files should and should not be in there, and why. It describes briefly what each type of file does, and it takes a look at the System folder's most important file, the System file. You'll see what this file does, and why it makes sense to create a single, master System file for all your applications. More experienced Macintosh owners, those who know what cdev and Init files are, for example, may wish to skim over this material quickly. Since the terms and files discussed in this next section are referred to frequently in the chapters that follow, less experienced readers are encouraged to take a more careful look.

The System Folder

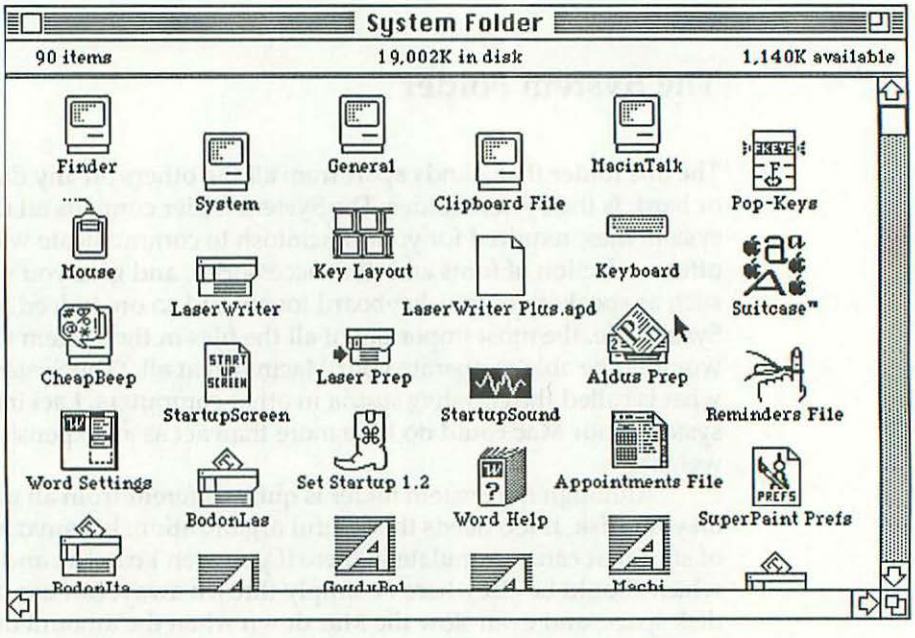
The one folder that stands apart from all the others on any disk, be it floppy or hard, is the System folder. The System folder contains all the files, the system files, required for your Macintosh to communicate with the printer, offer a selection of fonts and desk accessories, and give you various options such as speaker volume, keyboard touch, and so on. Indeed, without the System file, the most important of all the files in the System folder, you wouldn't be able to operate your Macintosh at all. (The System file is part of what is called the *operating system* in other computers. Lacking an operating system, your Mac could do little more than act as an expensive paperweight.)

Although the System folder is quite different from all the other folders on your disk, it too needs thoughtful organization. It's amazing the amount of stuff that can accumulate in there if you aren't careful, and all those files, which should be elsewhere or simply thrown away, can use up precious disk space, and even slow the Mac down when the amount of extraneous matter becomes overwhelming. This section will help you figure out which files should be in your System folder and which should not. For the ones that should be there, it will give you some idea of what they do.

Inside the System Folder

Even the most elementary System folder has a number of files in it, and what they do is not always obvious. To help you organize your System folder, here's a run-down of the files found in an everyday System folder, with a brief explanation of what each one does. You'll find further discussions of many of these files, and the ways in which they fit into fully powered strategies, in later chapters.

The System File This is the disk-based part of the Macintosh's "operating system." (The other part is built into the computer, in its read-only memory, or ROM.) This is the program that makes the Mac work, the one piece of software without which the Mac would simply not function at all, and the one on which all other programs depend. As a result, it is the one



It is easy for the System folder to become choked with System files, Control Panel files, Init resource files, printer drivers, downloadable fonts, and more.

file that must be present when you turn on your computer. It is the System file that contains the fonts and desk accessories installed with Font/DA Mover and the FKeys installed with *FKey Manager*. It contains standard icons, cursors and gray patterns, definitions for windows and menus, the keyboard layout, and more. It is also the most frequently used piece of software on your Macintosh, although virtually everything it does is invisible to you.

Because it is used so frequently, the System file is probably the file most prone to accidental corruption from unexpected power loss, unfriendly bugs in unwitting applications, and other difficulties. Because it is usually modified by the Macintosh owner, with a highly individual set of fonts and desk accessories, to name but the most common additions, a ruined System file can take a considerable amount of time and effort to replace. Thus, backing up your System file is of paramount importance.

The Finder It is the Finder that creates the Desktop, the place you return to after quitting an application. The Finder is really an application, just like any other application, that runs when nothing else is running. Its job is to help you keep track of everything (disks, files, etc.), to give you the necessary housekeeping tools (the ability to copy, delete, rename, and move files, folders, and disks), and to provide a way to find and move to another application.

The Finder offers a convenient, graphical way of keeping track of files, folders, and disks—but it isn't the only way. As you'll see in Chapter 6, the Finder is not a required file. Both hard and floppy disk users may find that they can get along quite well without it by using one of several types of Finder substitutes or by using a desk accessory or FKey to go straight to another application (without an intermediate stop at the Finder). Hard disk owners will want to have it available, even if they don't use it all the time. Floppy disk users, if they decide to use a Finder substitute instead, may find it convenient to delete it from their System folder.

Printer Drivers Printer drivers mediate the communications between the Macintosh and your printer. Without them, you would (in most cases) be unable to use your printer. They translate your application's output into something the printer can understand, and send it to the printer for printing. The ImageWriter files are the drivers for the ImageWriter I and II (the "plain" ImageWriter file), the ImageWriter LQ, and an ImageWriter connected via AppleTalk. Both LaserWriter and Laser Prep files are required by most programs if you want to print with a LaserWriter. Be sure you have

the one appropriate for your printer in the System folder. Also be sure to trash any drivers for printers you do not use; they simply take up valuable disk space.

The existence of two files for the LaserWriter as opposed to one (the norm for most other printers) is somewhat of a mystery to many people. To understand why there are two LaserWriter printer files, you first need to know a little about how both the Mac and the LaserWriter work. On the Mac, most text and graphics on the screen are created by QuickDraw, a sophisticated set of instructions stored in the Macintosh ROM. The LaserWriter, on the other hand, uses a different set of instructions to create its output. Everything you see on a LaserWritten page is the product of a language called PostScript, which is stored in LaserWriter ROM. What the LaserWriter files do is transform QuickDraw information into PostScript.

The reason there are two LaserWriter files is that the translation from QuickDraw to PostScript is a two-step process. First, the LaserWriter file intercepts the printer instructions created by the application when you select the Print command, and it converts them into language only the LaserPrep file can understand. If the Laser Prep file is not yet stored in the LaserWriter's memory, it sends that to the LaserWriter (this is what's happening when you see the message, "Initializing Printer"), and then it sends the translation of the application's printing instructions. The Laser Prep file then completes the translation into PostScript, which is what the LaserWriter needs to go to work.

To make matters more complex, not all programs require Apple's LaserWriter drivers to operate the LaserWriter. There are two ways that an application can get around the need for the LaserWriter files. One is to provide its own PostScript translation process. A good example of this method is *PageMaker* (Aldus). Aldus was not satisfied with a number of limitations imposed by the LaserWriter and Laser Prep files, so they created a substitute. Aldus also programmed *PageMaker* to do the first step in PostScript conversion itself, and to send that partially converted file directly to the Aldus Prep file in the LaserWriter. (Like Laser Prep, Aldus Prep must be sent to and stored in LaserWriter RAM before any printing instructions can be sent). Thus *PageMaker* needs only Aldus Prep, and neither the LaserWriter file nor Laser Prep, to print on the LaserWriter.

The other way to get around the LaserWriter files is to create standard PostScript code which the LaserWriter can understand (without the need for translation), and send that directly to the LaserWriter. The number of programs that can do this is small, but growing. An example is *JustText* (Knowledge Engineering), a program whose power is due, in part, to the fact that it writes PostScript and sends it directly to the LaserWriter.

Hard disk owners should, of course, simply store whatever drivers their applications need in the System folder. Applications will simply locate the one or ones they need, if they need one at all, and go to work. Floppy disk users should take a closer look at the needs of their applications, however. For example, if you have a desktop publishing duo or trio of applications (including *PageMaker*) on a disk, use a specially configured System folder (on another disk, perhaps) with that set, and print only from *PageMaker* when using the set, then save disk space by trashing Apple's LaserWriter files and keep only Aldus Prep in your System folder. Or, if you use a program that produces PostScript code, check the manual or experiment with it to see if you really need the LaserWriter files. Chances are you do not and can thus recover a fair amount of disk space by eliminating the unneeded drivers.

Control Panel Files In its most recent incarnation, the Control Panel provides access to a scrollable list of configuration files, called cdev files by programmer types, each of which contains options concerning the Macintosh itself, or software, or hardware add-ons (mainly for the Mac SE and II) which you may have. The options in each configuration panel are established by the files which must be in your System folder. The main Macintosh configuration files include Keyboard and Key Layout, and Mouse and General, which controls various features of the Macintosh's main unit (the internal clock, the RAM cache, the speaker volume, and so forth). Other configuration files include Sound, Monitors, Startup Device, and Easy Access.

For the major Macintosh configuration files, your decisions about the various options are recorded in battery-backed RAM within the machine. Once set, the keyboard, mouse, and Mac control panel files can be thrown away, as long as you do not intend to change them, that is. In fact, if you have no intention of changing anything in the Control Panel, even the Control Panel desk accessory itself can be removed from the System file. Because you may change your mind about one thing or another later on, however, I don't recommend doing so unless disk space is a serious problem.

Init Resources Init resources are a simple yet powerful way to expand the functions of the System without actually altering the System file on your disk. To install Init files, all you need to do is place them in the System folder. Then, the next time you start up the Macintosh, the System file searches the System folder for any Init files that you may have put there and loads them into memory. Until you remove the Init files and restart your

computer, your Macintosh will act as if the functions contributed by the Init files were built into the System file itself. For example, the Init file *AutoBlack* (public-exchange) is a screen saver which automatically turns the screen black after a few minutes of inactivity (see Chapter 5 for more about *AutoBlack*). As long as *AutoBlack* is in your System folder when you turn on your Macintosh, its screen saver function will always be a part of the way your Macintosh works. *Suitcase* (Software Supply), *Tempo II* (Affinity Microsystems), and a number of other useful programs are Init files as well.

As I mentioned, Init files need only reside in the System folder to become part of the way the Macintosh functions. To prevent them from being loaded by the System file the next time you start the Mac, simply remove them from the System folder, or put them inside a folder within the System folder. You can have as many Init resources in the System folder as you like, but it is a good idea to test combinations of two or more Inits thoroughly before you rely on them in real life situations.

Downloadable Fonts The LaserWriter and LaserWriter Plus contain a number of fonts stored within the printer's ROM. But you are not restricted to using just the fonts which are in ROM; anyone using the LaserWriter or any PostScript printer can also print with other fonts, fonts which are downloaded either automatically or manually from disk files into the printer's RAM. The mechanism by which this is done is described in Chapter 4. Suffice it to say here that you must have a special PostScript font file in the System folder for every downloadable font you want to have downloaded automatically.

Frequently, each major style (roman or plain, bold, italic, and bold-italic) of a given typeface comes in a separate PostScript file, so that a single font family may actually be comprised of four PostScript files. With as many as four files to a font, it's not difficult to accumulate a considerable number of downloadable font files. Serious LaserWriter font collectors be forewarned: System folder obesity is often a product of downloadable font overindulgence. (ImageWriter owners can take heart in the fact that downloadable font files are a characteristic of the LaserWriter, not the ImageWriter.)

If you are one of those serious font collectors who has so many downloadable fonts that it has become difficult to find anything in the System folder, try this trick: simply stack all your downloadable font files one on top of the other (in Icon View). All you'll see is the font file on the top of the stack and perhaps a few bits and pieces of other files sticking out around the edges. The Macintosh will have no difficulty locating the font files it needs, wherever they may be in the stack, and locating your other System files will no longer be like finding a needle in a haystack.

Other Files There is a whole raft of other files which you can reasonably expect to find in a System folder at some time or another. Some are System-related files, such as Backgrounder, DA Helper, MultiFinder, PrintMonitor, and the Scrapbook. (PrintMonitor is actually an application, but since its main purpose is to run automatically, and in the background, with MultiFinder, it makes sense to keep it out of sight in the System folder.)

Many of these additional files, however, are support files related to specific applications and not to the general operation of the Macintosh. Dictionaries, help files, preferences or settings files, spooler files and folders, macro files—all sorts of files are often stored by applications (or by the applications owner) in the System folder, because it is the one folder that is always present. Since that's where the applications want them that's where these files stay (unless you use a program called *Set Paths*, which is described later in this chapter). But not every file that ends up in the System folder belongs there. It's a wise move to look in the folder from time to time for errant files that should have been put in the trash.

When weeding a System folder, however, take care not to throw out any and all files that don't look familiar. Some applications put important files in there, don't let you know that they're doing so, and don't name them clearly. On the other hand, useless files can accumulate in the System folder, and it pays to root them out. One type of file you are likely to run into is the so-called temporary file. These files are used by a program to hold various bits and pieces of your work and are usually removed by the application when you quit. If the program crashes for some reason, however, or if it is poorly behaved, these temporary files can be left behind, sometimes in great numbers. The first version of *Word* 3.0, for example, left dozens of newly abandoned temporary files in my System folder every week (the problem has since been corrected).

My comments in the previous paragraph notwithstanding, do not attempt to remove temporary files (using a file management desk accessory or MultiFinder) while you are at work in an application. You can wreak all kinds of havoc by removing a temporary file which is in use. Only weed old and abandoned temporary files from the Finder, when you can be sure that they are not in use. Although MultiFinder is supposed to be smart enough to prevent you from trashing temporary files which are currently in use, it is probably wise not to tempt fate and to refrain from cleaning house in MultiFinder when one or more applications are open.

Temporary files are usually easy to identify. They are often labeled `temp` or some other simple moniker. Frequently they sport the generic document icon (a blank page with the corner turned down), though some programs go so far as to give them their own special icon. It's the ones with

special icons that are most easily confused with other files, so tread carefully when trashing temporary files with special icons; make sure that you haven't selected an important document by mistake before dropping it in the trash.

Version Numbers and Keeping Up-to-date

Like most software vendors, Apple keeps revising, updating and (one hopes) improving the files and utilities in its standard System folder. If you aren't a CompuServe or GEnie groupie, or don't regularly attend the meetings of your local users group, it is easy to find yourself using old and outdated system software. While there is nothing wrong with being a little out-of-date and using old system files, the newest versions offer not only valuable enhancements, but often improvements in speed and performance as well. Thus it pays, in various tangible and intangible ways, to use the latest versions of the System and Finder and the various files that go with them. (System 4.2 and Finder 6.0 were the current versions when this book went to press.)

The first step in determining where you stand is to figure out the version numbers of the files you are currently working with. The Control Panel and Chooser desk accessories are relatively easy to determine: simply open them and look for a number like 3.0 sitting off by itself. Chooser's version number is in the lower right corner, and the Control Panel's is in the lower left. In Font/DA Mover, the version number is in the far left end of the menu bar. The Finder is also easy. From the desktop, select About the Finder... in the Apple menu. Among other things, the About the Finder... window will tell you the version number of the Finder you are using.

The other system files and utilities, including the all-important System file, are not always so simple. With luck, the version number will be written in the Get Info note box. In the absence of that, your only recourse is to make note of the creation date. You can jot down the modification date as well, but some files, such as the LaserWriter and System files, change their modification date when you use them or work on them with the Font/DA Mover. For such files, the modification date is meaningless. For other files, such as Laser Prep and the files used by the Control Panel (Keyboard, Mouse, and General, for example), either date will do.

Having determined the version numbers (or dates) of all your system software, how do you find out if it is the latest version, and where do you

get it if it isn't? Unfortunately, Apple does not notify all Mac owners of System upgrades. (They may be changing this habit, but had not as of the time this was written.) Rather, Apple has so far depended on various grapevines to spread the word. *Macworld* and other Macintosh magazines regularly inform their readers of system upgrades; *MacUser* prints the current version numbers (but not the dates) for all system files every month in its "Updates" software list. But magazines, due to their lengthy publishing schedules, are inevitably a few months behind the times. To get the most up-to-date information on version numbers, you should check with a users group, bulletin board, CompuServe, GEnie, or your Apple dealer.

Once you have found a knowledgeable soul willing to tell you what you need to know, assuming your system files are out of date, the next step is to get a hold of a set of new system files. The best (though not necessarily the cheapest) way to do this is to go to your Apple dealer and purchase the latest System Upgrade kit. In it you'll find disks with all the newest system files, instructions for installing them in your System folder (or folders), and a manual covering the new features. By purchasing the kit from your dealer, you can be sure that you have the newest versions of everything, that you aren't missing a file or two, and that you have information about the differences between these new files and the older versions.

If your dealer is out of Upgrade kits, if you don't think you need the upgrade manual, or if you don't want to spend the money for the kit ($50 or so), there are other ways you get up-to-date. For example, since system files can be legally copied and traded among Mac owners, you can get the newest versions by duplicating someone else's. CompuServe and some other electronic sources also have them; the size of today's system files makes downloading them from a service that charges you for on-line time an expensive proposition, however. Users groups are another place to look.

If you do get your system file from a source other than an Apple System Upgrade kit, you'll want to be careful to check that you do, in fact, have the correct version of each file. Updates to the various system files, utilities, and desk accessories usually occur in concert, and the revisions from a certain date are meant to be used together. Thus the Finder revision released in January should be used with the System file revision released at the same time and not with the one from two years ago. Unfortunately, the revision numbers of System file, Finder, Font/DA Mover, Chooser, and so on, all run according to different numbering schemes. For example, Finder 5.4 was meant to work with System 4.0, Font/DA Mover 3.2, and Chooser 3.0. Don't assume that files with the same revision number necessarily go together; the fact is, the opposite is usually true. (Don't ask me why. It just is.)

When to Use Out-of-date System Files Having encouraged you to use current versions of the Macintosh system files, I must point out that there are times when using an older version can make sense. The most obvious occasion is when you have a piece of software that will not run under the newest version of the System file. If it is a program which is important to your work, or a game which you can't resist playing, then clearly you'll have to use it with the older System file.

Running different programs under different versions of the System file is an uncomfortable situation at best, a dangerous proposition at worst. If you have a hard disk, you should be sure to run the problem program and the older System from a floppy disk; don't be tempted to put a second System folder on the hard disk. To play it safe, turn off your hard disk before running under the old System (especially if it is a very old System).

The safest bet in a situation like this, however, is to start looking for new software that will fulfill your needs *and* work with the current System.

Working with a Master System Folder

If you have a hard disk, there are many reasons why you should have only one System folder—what I call a master System folder—containing one set of System files on the disk. The most important reason is that multiple System files on a single hard disk can confuse the Macintosh and cause a crash. If that isn't enough to convince you, here are a couple of others: you can get confused as well as the Mac, for it can be hard to tell which System folder is the active one; and if you don't know which System file is in control, you won't know which set of fonts, desk accessories, Init files, and so on is currently in effect. Plus multiple System folders waste space due to unnecessary duplication; if your System folders are large, that can mean a lot of wasted space.

If you are working with a floppy-based setup, it still makes a lot of sense to use only one disk containing a master System file at all times, for all the reasons in the previous paragraph and another important reason as well: desk accessories which require data files, such as phone lists, calendars, scrapbooks, and so on, and applications which store default settings in a file in the System folder work best if you have a single System file. If you have multiple System folders such desk accessories and applications will create a data file for each System folder. If you're not careful, you can end up with slightly or completely different phone books (for example) and default settings in each System folder.

Even if you are working with a floppy-based setup using multiple System disks, you still should try to have duplicates of a master System folder on all your system disks, for all the reasons outlined above. There will be occasions, however, when one or two files in the master System folder do not make sense for a particular application or group of applications. Thus, to save disk space on your floppy disks, you should review what your applications need in terms of system files and alter the corresponding System folder appropriately.

For example, if you have a desktop publishing set of applications running from a single System folder and your page make-up part of the set is *PageMaker*, you can delete the LaserWriter and Laser Prep files and replace them with the Aldus Prep file (assuming that the only printing you will do is with *PageMaker*). Similarly, some programs create PostScript output and send it directly to the LaserWriter, without the help of either the LaserWriter or Laser Prep files. In System folders used exclusively with these programs, both LaserWriter files can be deleted as well. Similarly, if you are not using the Finder's MultiFinder option, you can throw away the PrintMonitor and MultiFinder files.

And a Master System File The most important file in the master System folder is, of course, the System file. Since they have only one System folder, hard disk equipped Macintoshes will have only one System file, what I call the master System file. Similarly, floppy-disk-based Macintosh owners should try to use a single, master System file; whether they have one or a number of System disks and folders.

Indeed, using a single, master System folder and master System file is a central part of the fully powered Macintosh strategy. So important, in fact, that I cannot stress it enough. Although it may not seem like such a big deal at the moment, when we start discussing more advanced topics such as home-made integration and RAM disks, the use of a master System folder is critical. But even if you decide not to take advantage of the integration strategies or to use a RAM disk, the benefits of a master System folder and master System file still far outweigh any disadvantages.

The primary advantages of a master System file are the same as those for a master System folder, including the fact that no matter what you are doing, no matter what application is running, you will always have at hand the same set of desk accessories and the same selection of fonts. The desk accessory you are looking for will always be under the Apple menu, the font you want will always be on the font menu; there are no surprises when you always use the same System file. Finally, one other important advantage of a master System file is that whenever you need to add or delete desk

accessories or fonts, doing so is easier when you only have to change one System file (and copy it to several System folders, if need be).

The disadvantages to a master System file are few. The only one of any magnitude is that with an 800K floppy-disk–based setup, it can be difficult to set up an organizing scheme in which you use only one System file and System folder on a single System disk. Copy-protected programs which refuse to let you make working copies for use on other disks are a particular problem in this regard (as they are for hard disks as well).

Keep in mind that because you will be using your master System file all the time, your choice of which desk accessories and fonts to put in the master System file is worth some extra thought—about two chapters' worth of thought, actually, and in particular, the two chapters that follow this one. Chapter 3 looks at desk accessories (and their cousins, FKeys) and at strategies for choosing and installing them; in the master System file and elsewhere. Chapter 4 covers fonts: what kinds, which ones, what sizes you should use, and how to use them effectively.

Another point to keep in mind is that even though you are working with only a single System file, you can (if you choose to) still organize your fonts and desk accessories according to the applications they are used with. Indeed, in Chapters 3 and 4, you'll see how you can tailor your master System file and all your applications so that a core collection of desk accessories and fonts is available at all times, and application-specific fonts and desk accessories appear only when the appropriate application is running.

Lest your font and desk accessory fever run too high, however, let me point out that, as with any System file, there is a great temptation to stuff the master System file to overflowing with desk accessories, fonts, and FKeys. Those still using single-sided disks may be horrified at the idea of such a huge file, but almost anyone with a hard disk can testify that it is relatively easy to create a System file well over a megabyte in size. But even if there is ample space for a System file well in excess of one megabyte on your hard disk, moderation is a good idea.

In fact, for safety's sake, you should try to keep your master System file small enough to fit on a single disk. This is because, with a System file no bigger than about 780K (until 1600K disks come along), you can easily back it up on a floppy disk. With a larger file, you would have to split the file in half and put it on two disks, an inconvenient process which demands a totally trustworthy file-splitting (and reconstructing) utility if you want the kind of confidence a simple file copy engenders. Since the System file is among the most important and vulnerable files on your hard disk, the ease and assurance with which you can back it up is critically important. Fortu-

nately, as you'll see in the following two chapters, with the advent of font/desk accessory extenders such as *Suitcase*, keeping your System file relatively lean is not all that difficult, even for font and desk accessory addicts.

Now that we've taken a look at the System folder and its contents, it's time to turn to the other folders and files on your hard disk or in your floppy disk collection.

Organizing Your Files

To help you organize your files, Apple developed the Macintosh's hierarchical filing system (HFS). With HFS, files can be stored in folders, and folders can be stored inside of other folders. As with traditional filing cabinets, the net result is a many–branched structure which leads (or should lead) from the general to the particular. For example, in a file cabinet you might look first for the H file, then for Harry Houdini's folder inside the H file, then for the Correspondence folder within Houdini's folder, and then for the letter of September 15, 1912. If the letter were stored under HFS on a well–organized disk, the process of finding the letter would be virtually the same, except that the file cabinet would be your disk, and the folders would be electronic, not physical, constructs.

How you organize your files under HFS, whether you have all your correspondence in one file, with Houdini's letter in the H folder, or whether you put all your files relating to Houdini in a single folder, depends, to some degree, on personal preference. Nonetheless, there are a few simple rules of thumb you should follow no matter what organizational style you adopt. The following are a few basic rules:

- Try, as much as possible, to keep folders grouped with folders and files grouped with files. In other words, don't mix apples and oranges; try to avoid a folder filled half with files and half with other folders. In a well-organized hard disk, folders contain either a clutch of other folders or a collection of files.
- Nest folders no more than three or four deep, so you don't spend all day digging through folders to reach your files. For those of us with a twenty-to-forty-megabyte hard disk containing an average abundance of files, this restriction is not a difficult one to follow. If you have a 120 megabyte hard disk and thousands of files, how-

ever, you may find this rule of thumb difficult, if not impossible, to follow. With disks that large or files that numerous, deeply nested folders or endlessly scrolling lists of folders and files may seem to be the only choice. "Dividing a Hard Disk Into Subvolumes" on page 48, suggests one possible remedy to such a situation.

- Make clear in each folder's name that it is a folder as opposed to a file of some sort. In long lists which do mix folders and files (it is, at times, unavoidable) it is easy to overlook the tiny folder icon and think that you are looking at a file instead. To avoid confusion, always put the word "Folder," or an uppercase F, or a script f (that is, *ƒ*, which is Option-F from the keyboard in most fonts—New York being a notable exception), or some other consistent abbreviation, at the end of the folder's name.

Following these three rules of thumb is relatively easy; figuring out which files should go into which folders and which folders into other folders is the hard part. There are several strategies you can use to organize your files, and which one is best for you depends on the type and number of applications you use, the kind and number of files you generate, and your personal working style. I describe some of these strategies in the sections below. One of them should work well for you, if not straight out of the box, then with some fine-tuning on your part.

Although the organizational principles are the same whether you use floppy disks or a hard disk, the end result is distinctly different. Therefore, floppy disk storage and hard disk storage are discussed in separate sections. People with hard disks who use floppy disks to manually back up their files will want to read the floppy disk section; there are ideas in that section which apply to backup file organization as well. People using a floppy-only system may want to read the hard disk section to see what they are missing.

One other note: virtually all of the strategies described suggest that you combine applications, documents, and/or System files on a single disk. In particular, they require that you copy your application programs to a hard disk or to one or more specially configured floppy disks. This presents no particular problem, as long as your applications are not copy-protected. If any of your applications are copy-protected, however, you should refer to Chapter 8 for information about copying copy-protected programs (legally and for legitimate reasons, of course). If you are in the market for new applications, be sure to avoid copy-protected programs so that you don't have to resort to the fixes described in Chapter 8.

Organizing Files on Floppy Disks

If you use floppy disks as your storage medium, you have probably already developed a scheme for organizing applications, system files, and documents. It's likely, however, that the strategies described in this book will necessitate that you reconsider your current organizational scheme. Thus even if you think you already have an adequate organizational scheme (and who doesn't), be sure to read through what follows.

If your floppy disk collection is not well organized (be honest, now), this is the time to go through those stacks of disks, trash all the files you no longer need, and get a good idea of what you have to deal with. How many applications do you regularly use, and how much disk space do they require? How many and what kinds of documents do you have? With the answers to these questions in mind, read on, decide what kind of organizational scheme best fits you and your disks, and then put that scheme into effect before moving on to the next chapter.

In this book, the recommendation to use only a single, master System file is of significant impact to any floppy disk organizational scheme. It means that the days of using application disks straight out of the box are over. Gone, too, is the notion that the System file (and folder) for one application is distinct and separate from all the System files for all your other applications. It also means that the amount of disk space your master System file and System folder occupy is an essential ingredient in determining your file organization.

As I mentioned, Chapters 3 and 4 are concerned with fonts, desk accessories, and FKeys, the parts of the System file over which you have direct control and which directly (and greatly) affect the System file's size. Unfortunately, until you work through the recommendations in those two chapters and configure a master System file to your individual tastes and needs, you won't know how big your master System file will be. Until you know how big your System file is, you won't have an accurate figure for the size of your System folder.

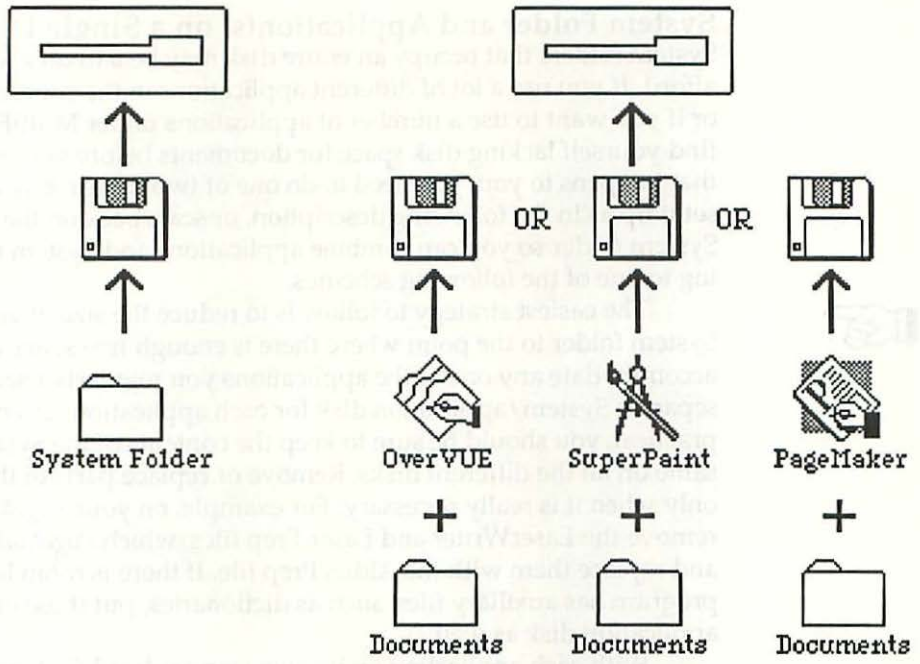
Thus readers using floppy disk storage need to do a little homework before embarking on a major disk and file reorganization. My recommendation is to skim the rest of this section and Chapters 3 and 4 to get an idea of what you should include in your System file and folder. Then copy all the fonts, desk accessories, FKeys, and other System files you are going to use onto a single disk, and check to see how much disk space it all occupies. If you prefer, actually work through Chapters 3 and 4 and create your master System file. Since you may not use all sizes of all the screen fonts you want to use, for example, this will give you a more accurate picture of the size of

your System file. Whether you estimate the size of your System folder or actually create one whose size you can accurately measure, use this number to determine which of the floppy disk strategies that follow will work for you.

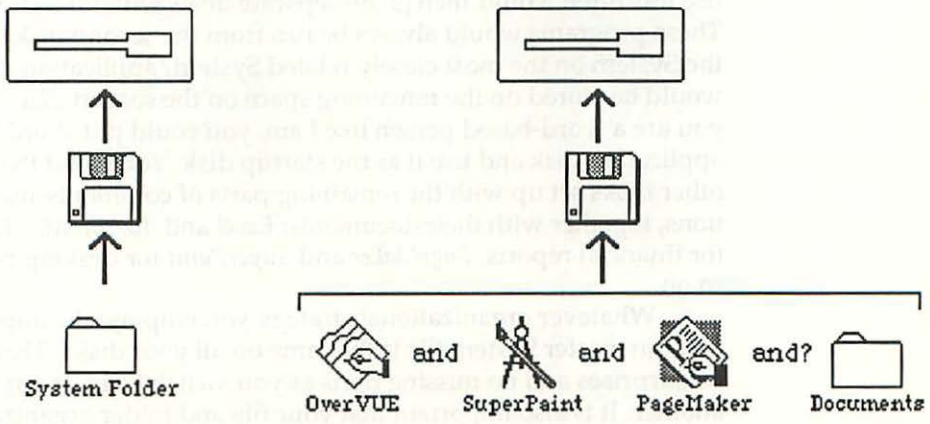
A Dedicated System Folder Disk If your System folder weighs in at 700K or more, then you'll have to dedicate an entire disk and a disk drive as well to a System disk. It's really as simple as it sounds: you put the disk with your master System folder in the disk drive and leave it there (except perhaps when you want to copy some files from one disk to another) until you quit working at the end of the day.

Of course, the difficult part is not what to do with the System folder, but what to do with everything else. In this matter you have several choices, depending in part on how you work. One choice is to have separate disks for each application and its documents. Because you will want sufficient work space on the application disk, only those documents that are really needed should be kept there, however. All other documents should be stored on archive disks and organized according to the application, project, type of document (letter, memo, or report, for example), or some other scheme. The advantage to this scheme is that finding room for files is usually not a problem. The disadvantage is that using several applications under MultiFinder is not practical, switching from one application to another involves swapping application disks, and saving documents may occasionally involve swapping disks as well.

A second choice is to put a number of applications on a single disk. For example, if you have a desktop publishing set of applications, such as *PageMaker*, *Word*, and *SuperPaint* (Silicon Beach Software), you could put them all on one disk. Other examples include a report writing set, such as Microsoft *Word* and *Excel* or a set of the applications you use most often: your word processor, database program, and telecommunications program. (Note that an application that you use for a variety of purposes may well appear on a number of different applications disks.) Although it is a great convenience to have all the applications for a specific job right at hand, if you have too many applications on one disk, you won't have enough room for your documents. If you have any space to spare on your System disk and you have a frequently used application which is small enough, you could put it on the System disk to help save space on your application disks. Overall, this strategy has greater space constraints than the previous one. The only way to find out if it will work for you and your documents is to experiment.



The master System folder goes in one drive, and disks dedicated to individual applications and their documents in the other.



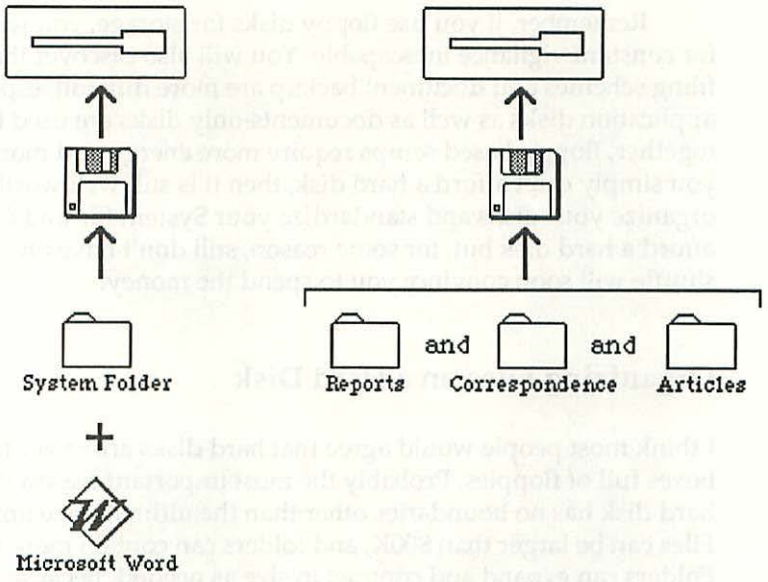
Putting all the applications you need for a specific task on one disk is more convenient, but it leaves little room for documents.

System Folder and Application(s) on a Single Disk Large System folders that occupy an entire disk may be a luxury which you can't afford. If you use a lot of different applications in the course of your work, or if you want to use a number of applications under MultiFinder, you may find yourself lacking disk space for documents before you even begin. If that happens to you, you need to do one of two things: buy a hard disk and set it up as in the following description, or scale back on the size of your System folder so you can combine applications and System folders according to one of the following schemes.

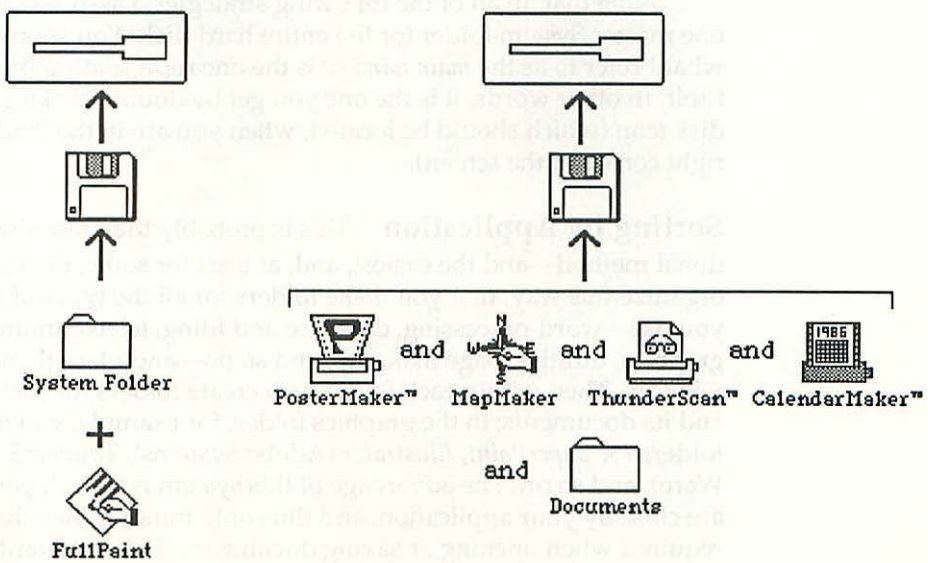
The easiest strategy to follow is to reduce the size of your master System folder to the point where there is enough free space on the disk to accommodate any one of the applications you regularly use. Then make a separate System/application disk for each application. Of course, whenever practical, you should be sure to keep the contents of the System folder the same on all the different disks. Remove or replace parts of the System folder only when it is really necessary. For example, on your *PageMaker* disk, remove the LaserWriter and Laser Prep files, which *PageMaker* does not use, and replace them with the Aldus Prep file. If there is room left over and the program has auxiliary files, such as dictionaries, put these on the System/application disk as well.

With each application on its own standardized System disk, you can work with one application at a time and store your files on separate disks (organized in any way that is appropriate) in the second disk drive; or you could put only your primary applications on System disks. Programs you use less often would then go on separate disks without System folders. These programs would always be run from the second disk drive and use the System on the most closely related System/application disk. Documents would be stored on the remaining space on the second disk. For example, if you are a word-based person like I am, you could put Word on a System/application disk and use it as the startup disk. You could then have several other disks set up with the remaining parts of commonly used combinations, together with their documents: *Excel* and *dBASE Mac* (Ashton-Tate) for financial reports, *PageMaker* and *SuperPaint* for desktop publishing, and so on.

Whatever organizational strategy you employ, the important thing is that the master System file is the same on all your disks. That way there are no surprises and no missing parts as you switch from one application to another. It is also important that your file and folder organization match the way you actually use your applications, whether you run them one at a time or several simultaneously under MultiFinder, for example. (For more on that topic, see Chapter 6.)



If you put the System folder and an application on one disk, you can use the other drive for document disks.



Organization may be easier if you put only major applications on System disks and run minor applications on "System-less" disks from the second disk drive.

Remember, if you use floppy disks for storage, you will find the need for constant vigilance inescapable. You will also discover that document filing schemes and document backup are more difficult, especially if parts of application disks as well as documents-only disks are used for storage. All together, floppy-based setups require more energy and more attention. If you simply can't afford a hard disk, then it is still well worth the effort to organize your disks and standardize your System file and folder. If you can afford a hard disk but, for some reason, still don't have one, the floppy disk shuffle will soon convince you to spend the money.

Organizing Files on a Hard Disk

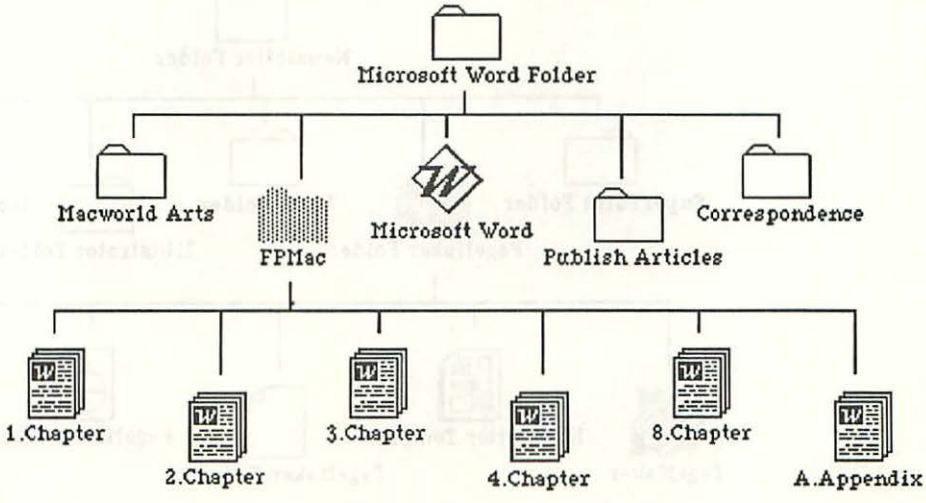
I think most people would agree that hard disks are easier to organize than boxes full of floppies. Probably the most important reason this is so is that a hard disk has no boundaries other than the ultimate size limit of the disk. Files can be larger than 800K, and folders can contain more than 800K. Folders can expand and contract in size as needed. Because you don't have to think in 800K units, you are free to lump or subdivide in any way you wish. The problem is that the freedom to do as you please can lead just as easily to highly inefficient organization or no organization at all.

Note that, in all of the following strategies, I assume that there is only one master System folder for the entire hard disk. You should also note that what I refer to as the *main window* is the one representing the hard disk itself. In other words, it is the one you get by double-clicking on the hard disk icon (which should be located, when you are in the Finder, in the upper right corner of the screen).

Sorting by Application This is probably the most obvious organizational method—and the easiest, and, at least for some, the least efficient. To organize this way, first you make folders for all the types of applications you use—word processing, database and filing, telecommunications, graphics, utilities, page make-up, and so on—and place them in the main window. Then within each folder, you create folders for each application and its documents; in the graphics folder, for example, you might make folders for *SuperPaint*, *Illustrator* (Adobe Systems), *ThunderScan* (ThunderWare), and so on. The advantage of this system is that all your documents are close by your application, and thus only minor folder changes are required when opening or saving documents. The disadvantage is that to get to your application, you have to search around for the right application-type folder, open it, search for the right application folder, and open that.

Inside each application folder you would put the application itself and any of the application's support files (help files, dictionaries, that sort of thing). For *Word*, for example, you would put in *Word* itself, the *Word* Help file, *Word*'s two dictionaries (one for spelling, the other for hyphenation), plus any of your own dictionary files, the *Word* Settings file, and one or more glossary files. *Word* is also just as happy if you put some of these files (though not all of them) in the System folder, where you can completely and safely forget about them. Doing so certainly makes for a much neater application folder. (See the end of the next section for another solution to the problem of overabundant support files.)

Each application's documents are also inside the application's folder. How you organize the documents depends in part on the number of documents you usually work with and in part on how full the application folder is already. If the application folder is filled with all sorts of support files (as in the *Word* example) and if you work with only a handful of documents—two or three database files, for example—then you may want to make a single document folder and put all your documents inside of that. This prevents any confusion which may arise from having support files and documents all mixed together. If the application folder contains only the application and perhaps one or two auxiliary files, and you work with only a few documents, the most logical thing to do is forget about a documents folder and store your documents right alongside the few program files. As

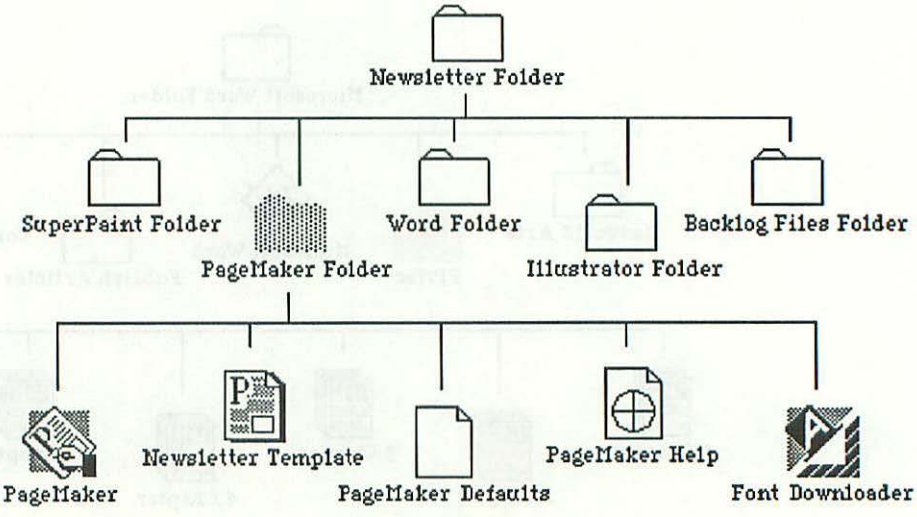


An exploded view of a hypothetical hard disk application folder.

you'll see shortly, there are easy ways you can set up the window in a situation such as this so that the support files stay out of sight.

If you generate a lot of files, however, then it probably makes more sense to divide the files into subgroups and store the subgroups in appropriately identified folders. For example, you could subdivide your spreadsheets according to the projects they were written for, their purpose (forecasts, pricing, quarterly business plan, and so on), or any other method that helps you get to the desired file quickly. All your documents should find a home inside one or another of your subgroup folders. If you have a few misfits that don't seem to fit anywhere, make a miscellaneous folder to hold them.

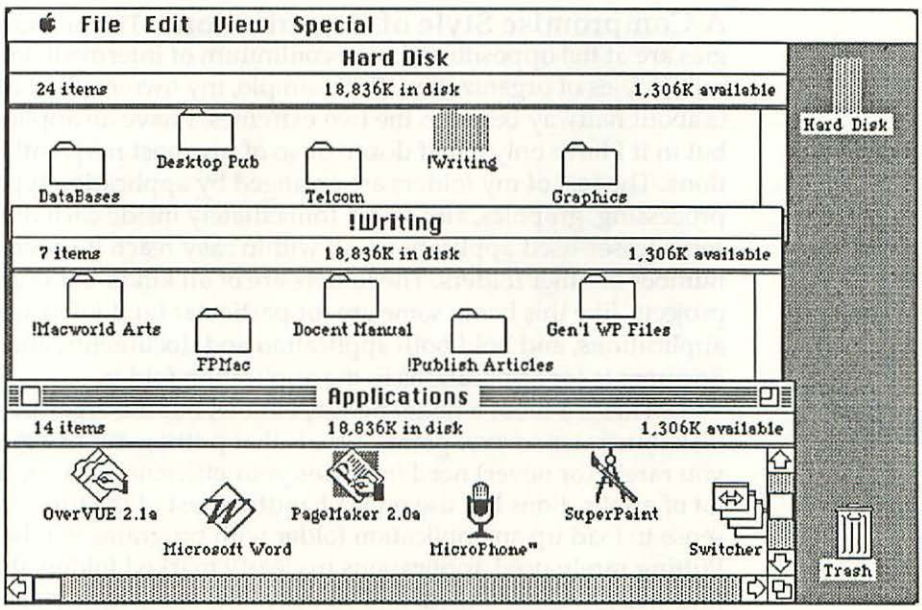
A variation on this scheme is organized at the top level not by application type, but by project or work habits. A newsletter folder would contain graphics, word-processing, and page make-up programs, for example, because these programs are used together when working on a newsletter. Those who tend to use one program for one purpose only may find this a good way to go. Those who, like myself, use individual programs for all sorts of things are likely to be frustrated by this style of organizing. Having several copies of your more widely used applications scattered about your hard disk is one way to make this scheme more workable, but it also wastes a lot of disk space.



An exploded view of a hypothetical hard disk project folder.

Sorting by File Type This style of organization is essentially the opposite of the one described in the previous section. The basic idea is to put all your applications in one folder and all your documents in other folders organized according to application, application type, or project. The advantage to this system is that with all your applications in one folder, all applications are immediately accessible. Keep the application folder window open and, at worst, all you have to do is scroll to find the desired program. The disadvantage is that once in an application, you may have to open and close quite a few folders to get to your documents, which makes saving and opening documents somewhat less convenient than in the previous strategy (unless you use a program such as *HFS Navigator*, which is described in Chapter 7).

To set up this system, first put all your applications in a single folder in the main window. Then divide up your documents much as in the previous strategy: first make folders for each type of application and, within each application-type folder, make folders for each individual program. Thus the main window would have, in addition to the Applications folder, a



With all your applications easily accessible in one folder, you could organize all your documents in other folders by application type, by type of document, or by project.

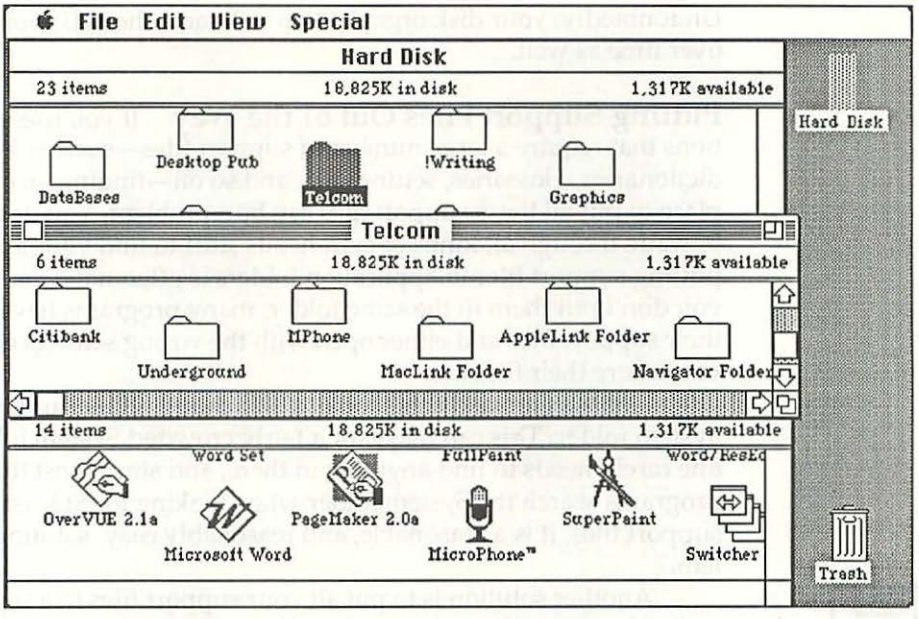
word-processing folder, a database folder, a graphics folder, and so on. Within the word-processing folder, for example, you would have a folder for *Word* documents, a folder for *Acta* (Symmetry Corporation) outlines, and perhaps other folders as well. Within each program folder, you could store all your documents as a single large collection, or you could subdivide them still further into different folders appropriate to the kinds of documents you generate.

Another strategy is to arrange your documents by job, task, project, client, customer, or whatever. For example, you might have a folder for monthly reports and, within that, folders for each month. In each monthly folder you would store whatever data files were used for the report—spreadsheets, database files, that sort of thing—plus the text of the report, graphics files used for the illustrations, and the *PageMaker* document of the finished report. The other types of projects that you work on would be organized in a similar fashion. Those who like to browse quickly through all their word-processing files, for example, would find this organization scheme frustrating. Others may find it convenient to have all the files relating to a particular subject in one place.

A Compromise Style of Organization The previous two strategies are at the opposite ends of a continuum of intermediate, or compromise, styles of organization. For example, my own method of organization is about halfway between the two extremes. I have an applications folder, but in it I have only a half dozen or so of my most frequently used applications. The rest of my folders are arranged by application type: word-processing, graphics, and so on. Immediately inside each of these folders are some lesser-used applications, all within easy reach if I need them, and a number of other folders. The folders are of all kinds: some are for specific projects, like this book; some are for particular (and infrequently used) applications, and hold both application and documents; and some hold documents for applications in the application folder.

This is a bit of a hodgepodge, I know, but the organization of my hard disk illustrates several points. One is that putting out of sight the things that you rarely (or never) need increases your efficiency. If you, like me, have a lot of applications but use only a handful most of the time, it doesn't make sense to load up an application folder with programs you hardly ever use. Putting rarely used applications in clearly marked folders that are easy to find when you need them and sit out of the way when you don't keeps your desktop a little less confusing and cluttered.

A second point is that an organization that works for some people may be a disaster to others, and only you can judge what works best for



In a mixed organization scheme, only the main telecommunications program, MicroPhone *(Software Ventures), is in the application folder. Lesser communications programs and all of* MicroPhone's *documents are in the Telcom folder.*

you. I, in fact, don't recommend to anyone the mixture of folder types exemplified by those on my own hard disk (for example, having project folders, application folders, and document-only folders all mixed together). For me, the system works fine, since it mimics the piles of paper that used to clutter my desk. (My desk is still cluttered, of course, but with computer paraphernalia now, not piles of papers.) For most people, my organization would be too confusing and illogical; but nobody said an organizational scheme has to be perfectly logical—it just has to work.

A final point is that organizational schemes need not be static. My hard disk's organization is a hodgepodge that never stays the same from one month to the next. When projects are started, they may get a folder inside my page make-up or word-processing folder, maybe even inside a folder in that folder. Then, when they become the predominant task of the day, the folder is often moved to the main window, so I can get to it quickly. When the project nears completion and other projects come to the fore, first the folder is moved back to its earlier position, and eventually it is moved to an archive floppy disk to make room for other things on the disk.

Undoubtedly, your disk organization will (and should) evolve and change over time as well.

Putting Support Files Out of the Way If you use lots of applications that require a large number of support files—such as *Word*, with its dictionaries, glossaries, settings file, and so on—finding an out-of-the-way place to put all these support files can be a problem. You don't want to have to wade through all kinds of extraneous stuff to find your applications, so putting support files in application folders is often not a good idea. But if you don't put them in the same folder, many programs have trouble finding their support files and either open with the wrong settings or constantly ask you where their files are.

One way around this problem is to dump all the support files into the System folder. This can make for a fairly crowded System folder, but since one rarely needs to find anything in there, and since most (but not all) programs search the System folder when looking for at least some of their support files, it is a reasonable, and reasonably easy, solution to the problem.

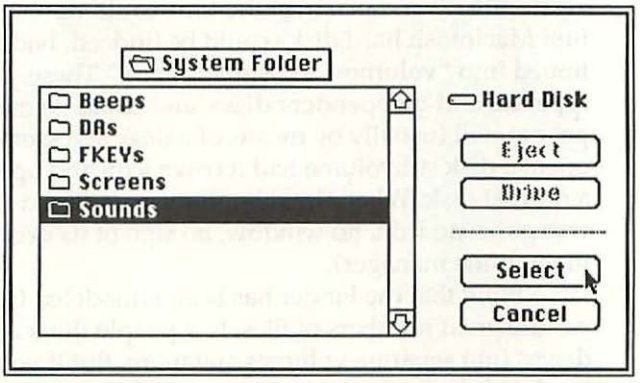
Another solution is to put all your support files in a separate folder (call it Support Files, or something like that) and put that folder either in the main window or in another folder, the most likely candidate again being the System folder. Then use *Set Paths* (public-exchange) to direct your applications to the Support Files folder. *Set Paths* is a desk accessory whose main purpose is to inform the Macintosh where to look, in up to five designated folders, when an application or desk accessory needs to open a support file that is not where it should be. These designated folders lie at the end of a string of other folders, at the end of a "path," which begins with the startup disk's main window.

This technique makes it easy to keep a neat and orderly desktop and puts auxiliary files out of sight and out of mind, right where they should be. The only flies in the ointment are programs that move their support files around without your permission. If you have a program that does that, there's not much you can do but live with it.

The following is what you need to do to get *Set Paths* to tell the Macintosh where to look for your support file folder:

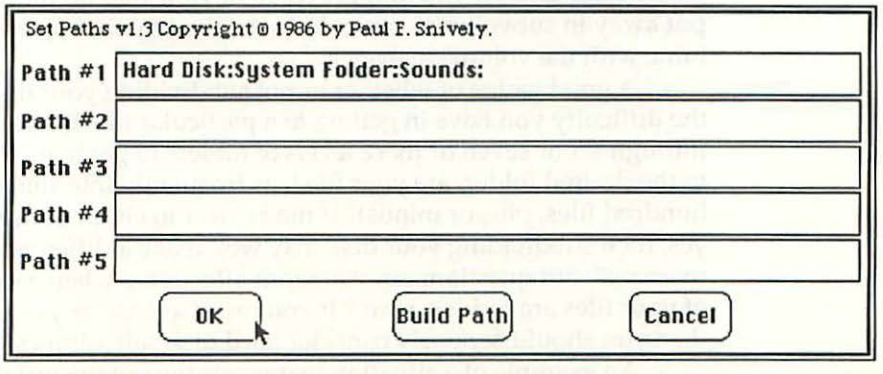
1. Install *Set Paths* with Font/DA Mover (or with *Suitcase*, or whatever you use to fill your Apple menu; see Chapter 3 for more on this subject).
2. Put a copy of the Boot Paths file, which accompanies *Set Paths*, in the System folder.

3. Launch *Set Paths*.
4. Click on the Build Paths button.
5. Use the usual folder selection techniques to open folders as needed to locate your Special Files folder (or whatever folder you want to add to the *Set Paths* list).
6. With the Special Files folder selected (white letters on a black background) in the list window, click on the Select Button.



To delineate a path, highlight the folder at the end of the path and click on the Select button.

7. You should now be back at the main *Set Paths* window, where the path to the Special Files folder will be listed in the Path #1 box (or a later box, if you have previously set one or more paths).



Once the path to the Special Files folder is entered in a Path box, click on the OK button.

8. Click on the OK button to quit *Set Paths*.
9. Restart your Macintosh.

From this point on, your Macintosh will know to look in the Special Files folder for all your applications' support files. As you acquire new applications, you can put their support files in the Special Files folder as well.

Dividing a Hard Disk into Subvolumes Because the Finder version that was current at the time could not handle many files at once, the first Macintosh hard disks could be (indeed, had to be) divided or partitioned into "volumes" or "subvolumes." These subvolumes acted like separate and independent disks, and could be made to appear and disappear at will (usually by means of a desk accessory volume manager). When open, a disk subvolume had its own icon and opened to a window, just like a normal disk. When the subvolume was closed (commanded to go away), it was gone; no icon, no window, no sign of its existence anywhere (except in the volume manager).

Now that the Finder has been remodeled (with HFS) so that it can handle great numbers of files, few people think about dividing their hard drives into separate volumes any more. But if you have a large-capacity hard disk and an overabundance of files, you may want to consider dividing your disk into volumes. Why? Because too many folders and too many files can slow down your Macintosh, and slow you down too, as you slog through nested folders and long lists of files to find the document you need.

Hard disk subvolumes only make sense if the number of documents is really extreme; they aren't for everyone, and just because your disk is getting full doesn't mean that it's time to subdivide the disk. Furthermore, it only makes sense if you don't need all of your files all the time, since files put away in subvolumes can only be reached by first opening the subvolume with the volume manager.

A good gauge of whether or not subdividing your disk makes sense is the difficulty you have in getting to a particular file. Do you have to dig through six or seven or more levels of folders to get to it ? When you do get to the desired folder, are your file lists frequently interminably long (say one hundred files, plus or minus)? If the answer to either of these questions is yes, then subdividing your disk may well make a difference. There is one more relevant question: can you work effectively when a significant number of your files are hidden away? If you can also answer yes to that question, then you should seriously consider hard disk subvolumes.

An example of a situation that meets the criteria just described is a small business which uses the Mac for accounting and inventory. At the

very least, this company's hard disk could be divided into three subvolumes; one for the System folder, applications, and general documents; one for financial data; and the third for inventory data. When working on inventory, for example, you could keep the subvolume containing financial records closed. This would make file management easier and the Mac's performance faster. Or when working on a flier outlining this week's specials, you could keep both the accounting and inventory subvolumes closed. Partitioning your documents in this way also increases security (if the different volumes require passwords to open them) and safety (what you can't see you can't throw in the trash by mistake).

Some disk drives come with software which you can use to create subvolumes on your hard disk, and others do not. If you have yet to buy a hard disk and you think you might need to subdivide it into volumes, add a subvolume utility to your hard disk wish list. If the hard disk you are purchasing (or have already purchased) does not come with a subvolume utility and you want to divide it into volumes, you can buy a program that will do the job. Remember, though, if your hard disk is already rather full, you can't just partition it into subvolumes like slicing up a pie. You'll have to empty the disk at least partway (preferably all the way—and reformat it as well), since new subvolumes can only be created from free disk space. Then once the new volumes are created, you'll have to put the files back in, either by shifting files from other parts of the disk or by restoring files from your backups.

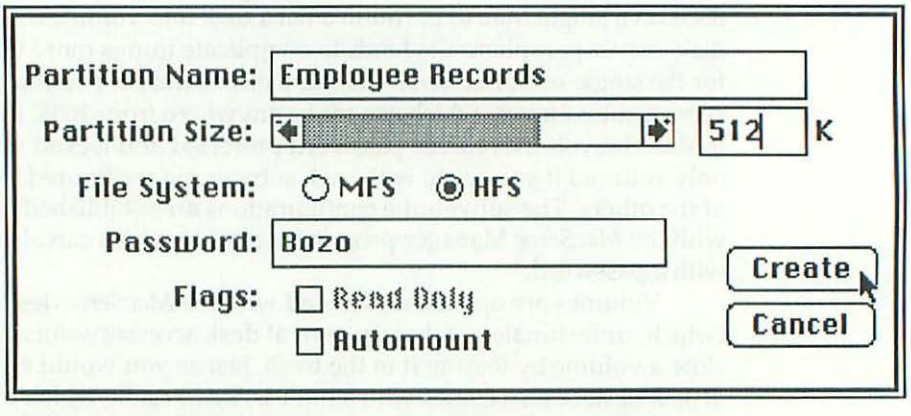
One utility which can subdivide your disk into subvolumes is *MacServe* (Infosphere). *MacServe* is a disk server intended for use by multiple Macintoshes all connected via an AppleTalk network. But it can be used on a single Mac to partition a hard disk into volumes. Although all the disk–server paraphernalia tends to complicate things more than necessary for the single user, *MacServe* has some nice features. You can create up to sixteen subvolumes, which can range anywhere from 100K to 32 megabytes in size. Subvolumes can be password protected and locked (to create a read-only volume) if you wish, with each subvolume configured independently of the others. The subvolume configurations are established (and modified) with the *MacServe* Manager program, access to which can also be restricted with a password.

Volumes are opened and closed with the *MacServe* desk accessory (which, unfortunately, takes up several desk accessory slots). You can also close a volume by tossing it in the trash, just as you would eject a floppy disk. *MacServe* also comes with a built-in RAM cache which cannot be turned off (you can set its size, however; see Chapter 7 for more information on RAM caches), a print spooler for the ImageWriter (but not the LaserWriter; again, see Chapter 7 for more on spoolers), and volume backup and

restore procedures (see Chapter 8 for more on backing up your disks). Each of these three built-in functions is surpassed by software specifically designed for that task alone, but as a package deal it's not bad at all.

In contrast to *MacServe*, *Hard Disk Partition* (FWB Software) is, as its name implies, a single-purpose utility. Semantically speaking, you use it to create partitions rather than subvolumes, and instead of opening and closing partitions, you mount and unmount them. With the *Hard Disk Partition* desk accessory, you can create new partitions as well as mount or unmount existing ones. When you create a new partition, you must give it a name and specify its size (it can be anywhere from 130K to 32 megabytes). You can also assign it a password to prevent unauthorized access, mark it as read only, and specify automatic mounting each time you turn on the computer. Although the size of the partition cannot be changed once the partition is created, you can change the partition's other attributes at any time.

To activate the program's automatic mounting feature, *Hard Disk Partition*'s AutoMounter file must be placed in the System folder. Mounted partitions appear, just like any floppy or hard disk, along the right edge of the screen. You can unmount a partition either with the desk accessory or by throwing the icon in the trash (just as when you eject a floppy disk by putting it in the trash, the files in the partition are not erased when you do this, the partition is merely put out of sight). There is no limit to the number of partitions you can create; the manual, however, suggests that you mount no more than seven at once on a basic Macintosh Plus (due to memory limitations).



Hard Disk Partition *allows you to divide your hard disk into partitions, or subvolumes. Each partition can be password protected and mounted automatically at startup, if you wish.*

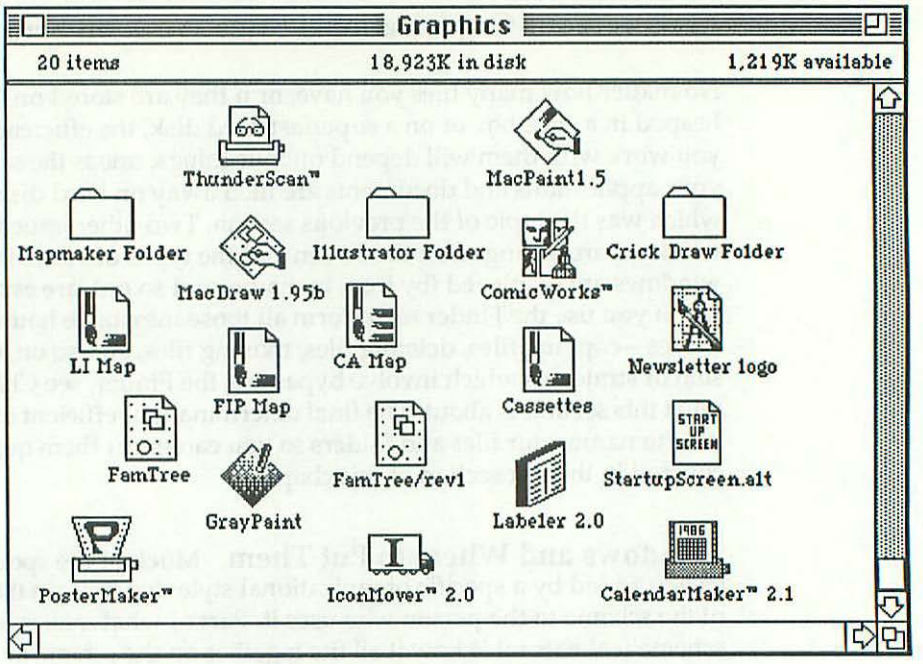
Making Your Organizational Style Work on Screen

No matter how many files you have, or if they are stored on floppy disks heaped in a shoe box or on a superfast hard disk, the efficiency with which you work with them will depend on four things: one is the scheme by which your applications and documents are filed away on hard disk or floppies, which was the topic of the previous section. Two other issues, how your windows are arranged on the screen and the types of views in which your windows are displayed (by icon, by name, and so on), are especially important if you use the Finder to perform all those inevitable housekeeping chores—copying files, deleting files, moving files, and so on. (For a discussion of strategies which involve bypassing the Finder, see Chapter 6.) That's what this section is about. The final determinant of efficient organization, how to name your files and folders so you can get at them quickly, is covered in the last section of this chapter.

Windows and Where to Put Them Much of the speed and efficiency added by a specific organizational style depends on the naturalness of the scheme to the person who uses it. Part of what makes a particular scheme feel natural is how it all fits together on the screen, or more specifically, whether or not the positions of files, folders, and windows "work." Thus the following questions arise: How many windows should be open at once and how should they be organized? Is it better to have large windows that don't need to be scrolled or small ones that do? As you translate an organizational concept to electronic reality, important questions such as these need to be answered.

A desktop randomly littered with odd-sized windows could contain a well-organized set of folders and files, but who would know? For this reason, once you get your files and folders organized, you need to develop a few rules for where folder windows will appear on-screen, and how file and folder icons will be arranged within windows (assuming you use an icon view, a matter we will look into in a moment). Because the Finder remembers where on the screen you put all your files and folders, and also remembers where and how big you want your windows, once you set up the initial arrangement, you can let the Finder maintain the organization from that point on.

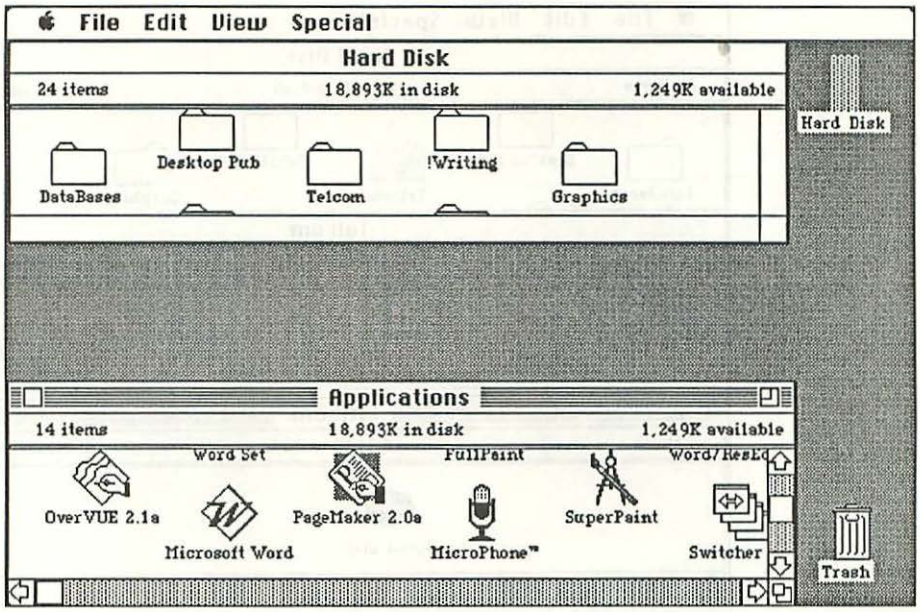
Consider the following example—my own desktop on my Macintosh Plus. When I first turn on my Mac and hard disk, and whenever I return from an application, I am greeted by a desktop displaying two open windows. At the bottom is a narrow window, the Applications folder's window, displaying two offset rows of the applications I spend most of



If there's organization here, I can't find it.

my time with: my word processor, graphics program, telecommunications program, database program, Switcher, and so on. (Why you would want to use offset rows and how you can get the Mac to organize them for you automatically is covered in Chapter 5.) A second double row of applications lies stacked on top of the first and is available with one click on the scroll bar, and the few support files (which insist on being there, unfortunately) are stacked in a third row and largely ignored.

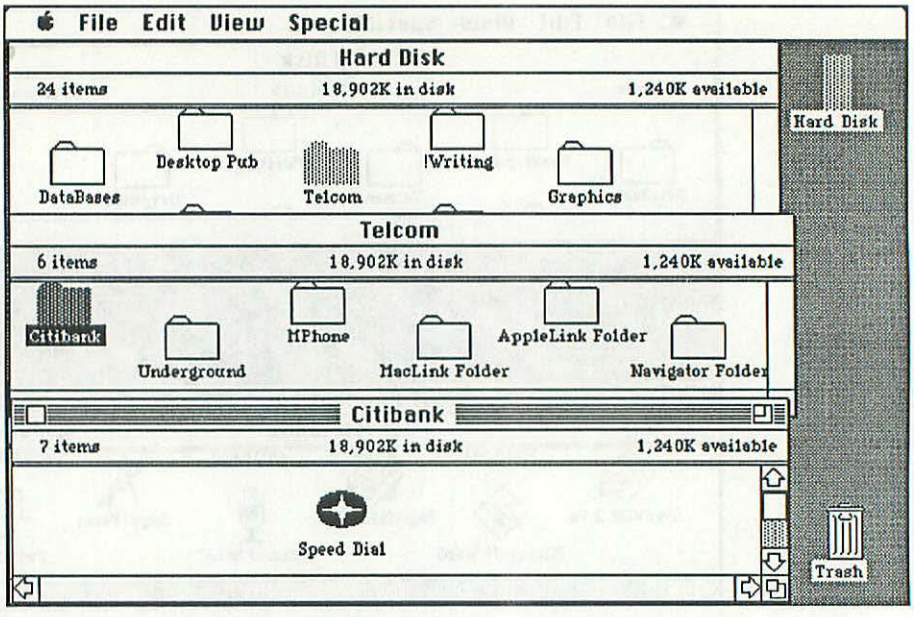
The middle of the screen is blank, and at the top is another long narrow window, the hard disk's main window. It displays two offset rows of folders containing my documents and the most frequently needed of my second-tier applications. Rows of less often used folders are stacked underneath (I can scroll to them quickly if I need them) with the least often used folders at the bottom of the stack, the System folder and the applications folder (whose window is always open and available) among them. Both upper and lower windows stretch from the far left to just before the area on the right used by disk icons and the trash can, and can display five or six full-sized icons at a time.



When I turn on my Mac and hard disk, this is what I see.

So why do I leave a third of the screen empty, and scroll through windows narrower than they need to be? I do this because of what happens when I open a folder. If I need one of my primary applications, I just double-click on the icon in the bottom window and that's that. If I need a less frequently used application or want to go to a specific document, then I open one of the folders in the upper window. Each such folder always opens as a narrow window, just like the first two, in the center of the screen and displays a number of choices, either applications or more folders.

My telecommunications folder is one example. My main telecommunications program, *MicroPhone*, is one of the readily available applications in the Applications folder. But I also use a number of special-purpose telecommunications programs, including *AppleLink*, my home-banking program, *CompuServe Navigator*, and so on. These programs are all stored in folders of their own within my telecommunications folder (always in view in the uppermost window). I put each program in its own folder because most of them have a number of support files which insist on being in the same folder as the application, and dumping them all in one telecommunications folder creates too much of a mess. So when I click on the telecommunications folder in the upper window, its window opens in the middle of the



In my organizational scheme, windows in the middle of the screen always correspond to a folder in the main Hard Disk window, and windows at the bottom of the screen (except for the Applications window) correspond to a folder in one of the middle-level windows.

screen and displays another offset double row of folders, each containing a special-purpose communications program. (There's a *MicroPhone* folder holding my log-on and other files in that window, too.)

By having a narrow telecommunications window situated in the center of the screen, I can keep both the main and application windows in view even when this third window is open. I keep the application window in view so that I can go quickly to one of my main programs if I want to. I keep the main (upper) window in view so that I can open other folders stored there as well—to move a file from one folder to another, for example, or to back up files contained in several folders.

Some center windows display a row of applications or often-used documents. These windows require no further thought, except to order the applications and files in a meaningful way: the most important files in the top row, the row first seen when you open the folder; less important files in the rows hidden underneath. Other center windows, such as the one for my telecommunications folder, display yet another row of folders. All folders in

the center window open to windows which almost completely cover the bottom window.

By placing my home banking window at the bottom of the screen, I keep my telecommunications options open; I can click on another communications folder at any time. Doing so also keeps the screen from getting cluttered with too many windows. In fact, in the home banking example, I cover up almost the entire Application window even though the new window displays only one program. (The support files which my home banking program requires are on a second line and thus out of sight.) Covering the six programs in the application window with a nearly empty window is, in this case, as much a cosmetic confusion-reducer as anything else.

Although well-organized, my window arrangement is not scrupulously neat. In fact, I have it set up so that many of the middle- and lower-level windows are slightly offset in relation to the others which might be open at that level. In this way, no matter which windows are open in the center or at the bottom of the screen, small pieces of each window beneath the active window (including, or perhaps especially, the application window) are almost always exposed; I simply click on the exposed piece to make the hidden window active.

My main Mac Plus windowing principles, then, are that the main window and the application window are always left open, displaying two offset rows of the most important applications or folders. Second-level applications and folders can be scrolled to, and folders and files which are of least interest to me are placed as far as possible from the primary row of folders or files. Folders in the main window open in the center of the screen. Folders in the center windows open to windows covering the bottom application window.

This particular scheme works, for me anyway, because I don't like my Mac Plus's screen cluttered with lots of little square windows, no matter how neatly they are arranged. You may not have the same taste in desktop design, however. In either case, you will probably want to experiment with alternate arrangements. Assume that your files and folders are arranged by application. In the upper left corner of the screen, you could have a small window which displays all your application folders. If you click on an application folder, another small window, displaying the application and its document folders, opens next to it on the right. Clicking on the document folders opens windows arrayed along the bottom half of the screen.

These examples all apply to the smallish Mac Plus screen. You may have a much larger screen, a wider one, or a taller one. Your files may not be organized in the same way, either, and you may work from floppy disks,

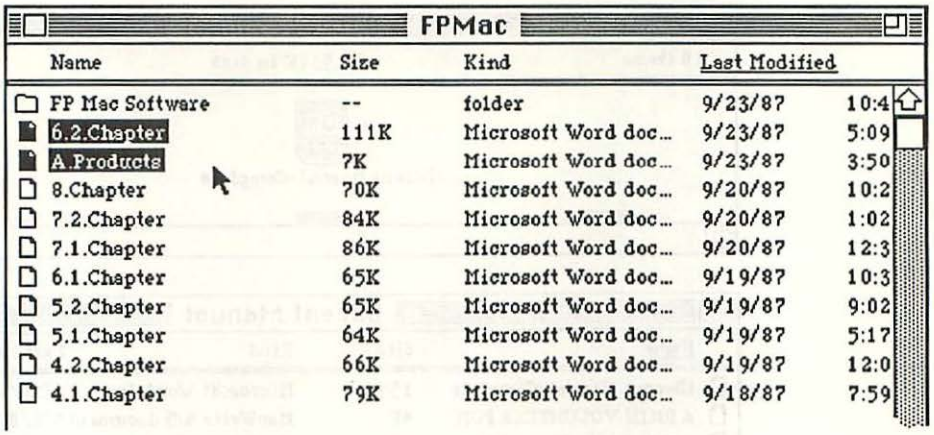
rather than a hard disk. Obviously, your on-screen strategy should be tailored to fit the area you have to deal with and your organizational scheme, and not simply be patterned after mine. There is, of course, no limit to the number of ways you can organize your windows, no matter what size screen you have. The important thing is that you do, in fact, organize them. Unless you devise a setup in which you never, ever, return to the Finder (which is possible, though unlikely; see Chapter 6), chaos on the desktop will only slow you down. The desktop is one place where neatness really does count.

Viewing Files and Folders in Windows The Finder gives you a number of ways to view the files in a window: by icon, by small icon, or in a list sorted by name, date, or file size. Some views are better than others, depending on the circumstances. Using a view which is less than optimal can be little more than a subtle annoyance, or it can be a real drag on your efficiency. Which view makes sense for a particular window depends on a number of factors, the most important ones being why you are searching the window in the first place, whether certain files have priority over other files, and how many files there are in the folder.

Match the View to the Reason You're Viewing the Window If you perform manual backups of your hard disk (without the aid of a backup program), then it would make sense first to sort files by the dates they were last modified. If you are copying a large number of files from a hard disk to a floppy disk, it often helps to sort them out by size so you can find a 12K file to fill a 13K space on a disk that is nearly full. These are but two instances where what you are doing has a direct bearing on what type of view to select.

Not all tasks will dictate that you view a window in a certain way, but many do, so you should be sure to consider the pros and cons of different views as you work with windows. For example, the process of backing up recently changed or new files on your hard disk can be made a lot easier if you first let the Mac sort and display your files by date, and then put the most recent files at the top of the list. Viewing them by name or size means that you'll have to scroll through the list and look at the date of every file to be sure you haven't missed any. Viewing them by icon means that you'll have to open the Get Info box for every single file. Now that's really inefficient.

In general, whenever it is helpful to have the files in a window sorted (even if only alphabetically), you should use one of the three view options which do not use icons. When looking for files of a certain size, use the view-by-size option. When looking for files with a certain name, use the



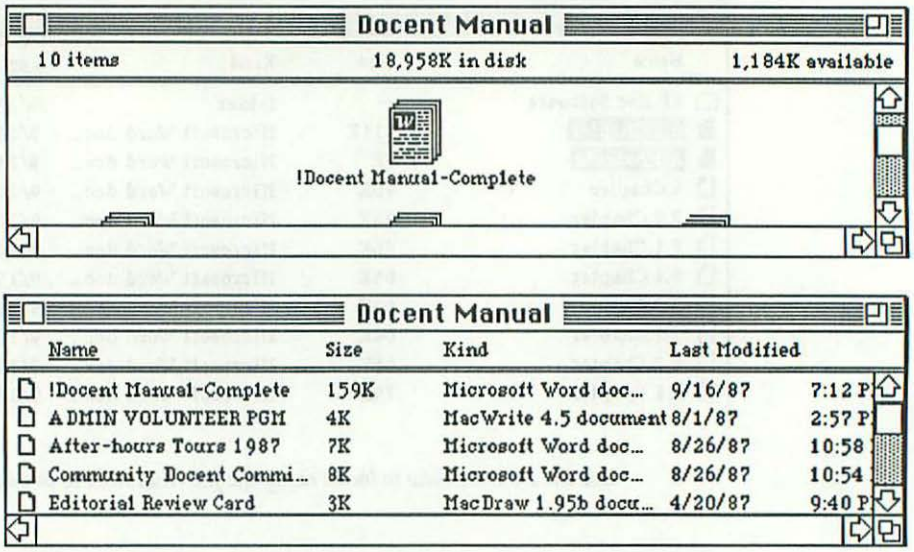
Use the By Date view to locate easily the files that need to be backed up.

view-by-name option. Obvious as these recommendations may sound, many people tend to get stuck in a rut with icon views and forget that the Mac will sort a file list for you, if only you would ask.

If you prefer icon views, by the way, and want your files and folders displayed in alphabetical order, there is a simple technique you can use to avoid sorting your files and folders by hand:

1. Select the window whose files and folders you want to reorder.
2. Select the **By Name** command from the View menu.
3. Choose the **Select All** command from the Edit menu, or press Command-A.
4. Drag all the files and folders onto the desktop. Take care not to deselect the files and folders (by clicking on something other than a selected file) once they are on the desktop.
5. Select the **By Icon** command from the View menu.
6. Drag all the files and folders on the desktop back into the window.
7. Select the **Clean Up** command from the Special menu while holding down the Option key.

Your files and folders should now be arranged in alphabetical (or, more precisely, in alphabetical-plus) order, left to right, from the top row downward.



With the By Icon view, you can easily indicate which file in a folder is the most important.

Use Icon Views to Prioritize Files and Folders In the organizational strategies discussed in the previous sections, I've described a number of situations in which files or folders are placed in specific locations in a window. For example, I mentioned that my hard disk's main window displays two offset rows of icons at a time. The folders I use most often are in the top row, the one which comes into view first, while several rows of less often used folders are stacked underneath (and reached by scrolling). The bottom-most row, the one that takes the most effort to get to, contains the folders I almost never need.

If you want to prioritize your folders or files as I have, by far the easiest way to do this is to put the window in an icon view and move the folders and files around until you have them arranged according to their importance. Icon position quite naturally conveys the idea of importance when viewed on the screen. By preceding file and folder names with special symbols (as described in the next section), you can achieve a similar sorting of files and folders by importance in the By Name view. But conveying a sense of levels of importance is intrinsically more difficult in list views. I, for one, prefer to display files as icons and use the special–symbol technique sparingly, to prevent a select number of much–used files and folders from falling to the bottom of a list, such as a file selection list.

One final icon view issue is the choice between full-sized icons and small icons. I suppose the decision here is really just a matter of taste, but my taste clearly runs towards full-sized icons. I find most small icons hard to figure out, and thus not very effective as icons. In terms of packing a lot of information in a small space, I find the list views more effective. Thus, for me, it's full-sized icons or none at all.

Use List Views When a Folder Has Many Files Although there are times when icon views have clear advantages over list views, there comes a time when the switch to a list view becomes imperative (and not just a matter of preference). That time is when the number of files and folders overwhelms your ability to use the icon view effectively. One way to measure when this more-or-less invisible limit has been passed is to click on the expand box in the window's upper right corner. If your file and folder icons no longer fit comfortably within the window when it is fully expanded, then it is time to think about using the list-by-name view.

As with just about everything, there are exceptions to this guideline. The most significant exception is the System folder. When it gets stuffed to the gills with system files of various sorts, Init resources, downloadable fonts, and application support files (dictionaries, help files, and so on), it can be argued that, no matter how much you have to scroll the window, icons make it much easier to distinguish the different types of files. In a By Name view it is often harder to find files, especially if you tend to cue in on the type of file you are looking for rather than just the name. But the System folder is an unusual folder (for all sorts of reasons, actually), containing as it does such a wide variety of files. Most folders, if organized properly, will contain files of only one or two types, and when these reach a certain abundance, the sensible thing to do is to display the files in a list.

Naming Files and Folders

The names of your files and folders are especially important whenever you use the By Name window view and whenever you select files (with the standard file selection box) from within an application. Of course, everyone uses file selection boxes all the time, so proper file and folder names are crucial. If you organize your hard disk into a nicely logical sequence of nested folders and then proceed to use names which are poorly thought out, obscure, or even downright misleading, you have won only half the battle, and over time you may lose the war. Indeed, rather than thinking of file and folder names as something apart from your overall filing scheme, it makes sense to view names as the last level of a continuous organizational sequence. It may take you a few seconds longer to come up with an appropri-

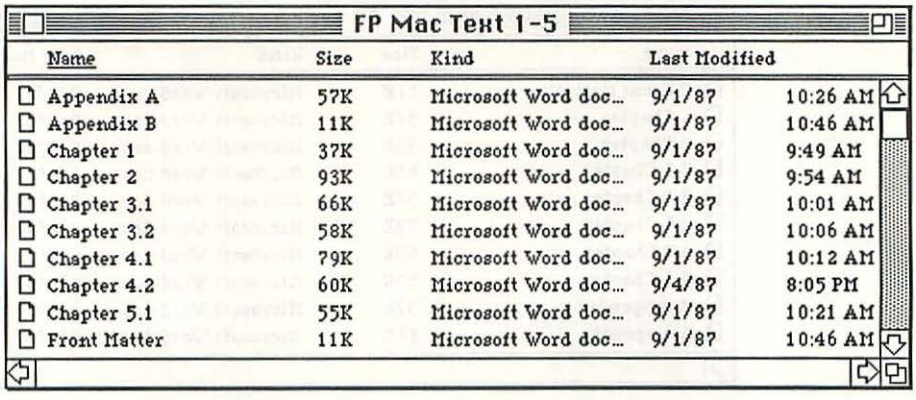
ate name if you have to think about it in this way, but those seconds will be repaid many times over by the ease and speed with which your well-named files can be located.

With the Macintosh, any file-and-folder naming strategy is based on two givens. One is that in a Macintosh file list, files are displayed in what I call alphabetical-plus order. The alphabetical-plus sequence (which is described in detail later in this section) begins with certain symbols, then numbers, then a few more symbols, then the letters of the alphabet, and finally, at the bottom of the list, a number of other symbols. Although this sounds like ASCII order (ASCII is the standard which determines how each symbol on the keyboard is always translated into the same code number for use by the computer), it's not quite; uppercase and lowercase letters are not segregated into separate groups, for example, which they would be if the order were strictly ASCII. This is just as well, for the point is not that you should learn ASCII (heaven forbid), but that letters, numbers, and symbols can be used to sort files in something other than standard alphabetical order when alphabetical order doesn't suit your purposes.

The other given is that, in Macintosh file selection boxes, you can select files by quickly typing the first few characters of the file name. Thus, if all files of a similar type start out with the same few letters—which is a reasonable way to name files on an IBM PC, for example—to specify a particular file, you'll have to type a lot more letters than you really need to. Not only is that a waste of effort, but for poor typists like me it means that typos made while pecking out a file name invariably prevent me from getting to the file I want. If your files are named correctly, however, selecting files from the keyboard is an easy and much-neglected way to speed up your work (see Chapter 7).

For example, consider the folder of files I created when I began my first book. What I did was write a detailed outline for the whole book and then split it up into chapters, putting each piece in a separate file. I named each file for the appropriate chapter `Chapter 1` and so on, and created as well several `Appendix` files and a `Front Matter` file (which contained all that stuff that books have before the first chapter). Of course, the result was neither efficient nor logical. To select a specific chapter from the keyboard, it was necessary to type the entire file name, since the chapter number was at the very end of the name. In addition, the appendices in all file lists were listed ahead of the chapters, and the `Front Matter` file was at the end.

To remedy this rather awkward situation, I ultimately changed the file names to what they should have been to start with. Under the new system, each chapter name began with the chapter number, so that `Chapter 1` became `1.Chapter` and so on down the line. This had the effect of separating

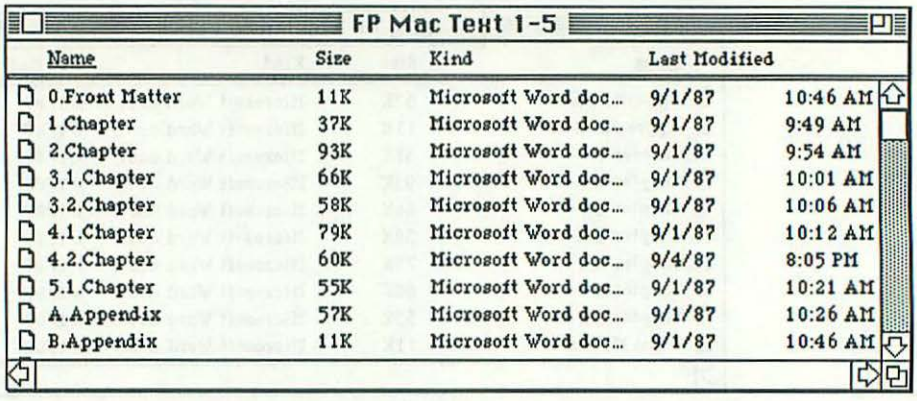


The first file name that comes to mind is not necessarily the best one. For example, in this list of book files, the appendices are at the beginning of the list and the front matter is at the end. Worse, you have to type the whole chapter or appendix name in order to select it from the keyboard in a file selection box.

the chapter files from all the other files in the folder, in correct numerical order. I named the front matter file `0.Front Matter` to put it in correct sequence in relation to the chapter files, and I listed the Appendices by their corresponding letters (as in `A.Appendix`), which thus put them after the numbered chapter files. As a result, the chapters were all listed at the top of the window (and preceded by the front matter), and the appendices followed. If I were using the mouse, I needed to scroll only to get to the very last chapters and the appendices. Even more important, in a file selection list I could simply type a number to select a chapter, zero for the front matter, or a letter for an appendix.

The preceding example illustrates several *important points* about file and folder names (don't forget those folders—they have to be selected as often as, maybe more often than, files). Let's look at each one of these points in more detail.

Name Important Files and Folders So They are Listed First Just as the most important folders and files should appear first when you open an icon view window, whenever possible, the most important and most frequently used files should be listed at the top of a file list (and the least important at the bottom). If reasonable names and standard alphabetical order don't bring your important files to the top of the list (and send the least important at the bottom), make use of the alphabetical-plus sequence. Begin the names of the files or folders you'd like to see at the top of the list



Carefully chosen file names can convey meaning quickly, position files properly in a list, and make it easy to select the file by typing only the first two or three letters of the name.

with a character that is listed ahead of the usual numbers and letters; and begin the names of the files or folders you'd like to see at the bottom of the list with a character that is listed after numbers and letters.

The following summary of the Macintosh's alphabetical-plus order will help you do this. Alphabetical-plus order begins with a number of common symbols and punctuation marks (going from the top to the bottom of the first column and then on to the second column):

! (exclamation point)	) (right parenthesis)
" (double quote)	* (asterisk)
# (pound sign)	+ (plus sign)
$ (dollar sign)	, (comma)
% (percent sign)	- (dash)
& (ampersand)	. (period)
' (single quote)	/ (slash)
((left parenthesis)	

These characters are followed by the numbers. Keep in mind that for whole numbers to sort correctly in a Macintosh file list, they must all possess the same number of digits. To give smaller numbers the same number of digits as the largest numbers in your list, you must add one or more zeros to the left of each number. If you don't add the zeros, all numbers

beginning with a one will appear first (1 and 15 and 156, for example), then all numbers beginning with a two (2 and 24 and 2501), and so on. If you do add the zeros, the numbers will sort as they should: 0001, 0002, 0015, 0024, 0156, and 2501.

After the numbers comes a small group of symbols:

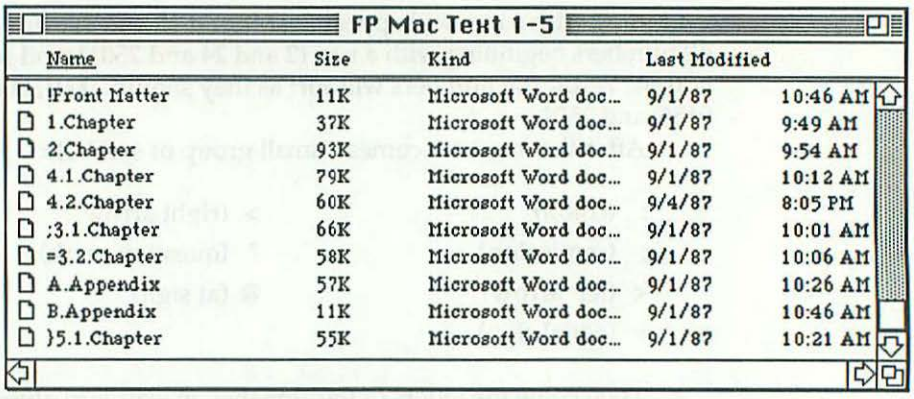
: (colon)
; (semicolon)
< (left arrow)
= (equal sign)
> (right arrow)
? (question mark)
@ (at sign)

Then come the letters of the alphabet, in standard alphabetical order, and finally, a whole raft of symbols, special characters, accented vowels, and so on. Of this rag tag collection of characters, some of the more useful ones include the following (in order):

[(left square bracket)
\ (back slash)
] (right square bracket)
^ (caret)
{ (left curly bracket)
| (vertical bar)
} (right curly bracket)
~ (tilde)
† (dagger)
° (degree)
¢ (cent)
• (bullet)
¶ (paragraph mark)
∞ (infinity)
◊ (diamond)

Thus, to get what I call priority files to the top of the file list, you might start all their names with a slash mark, as in `/Winston Report`, or with an exclamation point, as in `!Zebra Outline`. (Of course, all file names begun with the same symbol would then be ordered alphabetically by the letters following the symbol.) If you want to fine-tune the order of your priority files even further, you can use several different symbols at the beginning of the names of your files, according to the order of the characters in the alphabetical-plus list. If you use commas, periods, and slashes to begin your file names, for example, all file names beginning with a comma will be listed first, then those starting with a period, then those with a slash.

Remember, to make it easier to type the first few characters of the name of a priority file or folder (when you want to select it from the keyboard), use characters which can be typed without pressing the Shift or Option keys (such as a period, comma, or slash).



By using special characters at the beginning of your file names, you can put your files in almost any order you like.

Similarly, you can sandwich certain files between those beginning with numbers and letters, or you can send other files to the bottom of the list by prefixing their names with the appropriate character. For example, if you have a folder called `Anonymous Pranks` which you use only once a year (on April Fool's Day, perhaps), you can make it the last folder on the list, instead of one of the first, by renaming it `}Anonymous Pranks` or `~Anonymous Pranks`.

All this may sound a little complicated and perhaps beyond the realm of the useful, but if you give it a try, you'll see that it can indeed speed up the way you access files and folders, in more ways than one. If you go to your Writing folder more often than any other, for example, renaming it `!Writing` will do more than just move it to the top of the list. Since the topmost name in a file selection list is automatically highlighted when the file selection box first opens, you won't have to scroll or type `w` or `wr` or do anything else to select it. With `!Writing` at the top of the list, all you have to do is press Return to open it.

Put the Variable Part of a File or Folder Name First If your folders are organized properly, you'll find that similar files with similar names are clumped together, to the exclusion of other files (since they're in other folders). The problem is, one is often tempted (as I was, in the book chapter example) to give similar files names in which the part in common is at the beginning and the distinguishing part is at the end: `Chapter 1`, `Chapter 2`, and so on.

If you have a folder full of similar files, putting the distinguishing word number, or date at the end of each name is inefficient, to say the least.

With a folder of book files, typing `1` in a file selection box should be sufficient to select Chapter 1. With a folder of letters, typing the initials of the recipient or perhaps the date on which it was written ought to be enough to select the letter you're looking for.

If at all possible, you should try to name your files so that each one in a given folder can be selected by typing at most three, four, or five characters. This makes for quick access to a particular file or folder, no matter how long the list might be. The rule of thumb here is to relegate the generic parts of a file name to the end of the name and to put the unique parts at the beginning. Don't name all your correspondence files `Letter to Joe` and the like, but `Joe/letter` or `JB.let`. Don't name your monthly status sheets `Status Sheet-May`, or even `SS-May`; try `May.Status` instead. Don't name your daily report memo `Memo-12/03/87`, but `12/03/87.memo` or `12/03.memo`.

For daily memos, make a new folder for each year; otherwise, even if you include the year in the date, as above, memos from different years will be all mixed together (12/03/87 will be listed right after 12/03/86, for example). The only way around this is to write file name dates in the form 87/12/03; this would sort the files in correct chronological order, but then you would have to type the year every time you wanted to select a file. If most of the memos you call up are from the current year, having to type the year every time you want to select a memo file is pretty inefficient.

Of course, if you haven't organized your folders properly in the first place, this sort of naming scheme may not work very well. If you have a file which contains both letters and daily report memos, and both types of file have names starting with a date, the memos and letters will be intermixed, and typing only the date may not be enough to select a specific memo or letter. The best solution to a problem like this is to get better organized and separate memos from letters. If you have a good reason for lumping letters and memos in a single folder, then it may make sense to begin the names of all your letters with an `1`, for example, and then the date. Thus all memos will be listed first (since their names begin with a number) and in chronological order, and all letters will be listed afterward, also in chronological order (by the date which follows the initial `1`).

Make Your File and Folder Names Understandable In making it easy to select files by typing the first few letters of the name, you can go overboard and write the whole file name in a shorthand incomprehensible to all but yourself. This may not seem to be a serious problem, especially if you are the only one who uses your Macintosh or who needs access to the files in question. But cryptic codes tend to proliferate and memories tend to fade, and the combination of the two can lead to chaos.

So although it is a good idea to make your files easily accessible from the keyboard, it is also a good idea to make sure that your file names can be understood without a translator. Ideally, the first four or five characters of each name will be something unique and meaningful—a date, a chapter or quarter number, a correspondent's name or initials—and the remainder of each file name will state what kind of file it is, in English. If you have all your daily reports isolated in a folder of their own, for example, then the date at the front of the file name makes it easy to select files from the keyboard. And the words `Daily Report` after the date make it clear to all concerned (especially when the file is removed from the folder) just which file it is.

The difficulties come when your overall file organization is not well thought out. If you have lots of different kinds of files in one folder, for example, and need to use one- or two-letter codes at the start of each file name just to differentiate types of files, and you use other codes, such as dates, to differentiate individual files, things can get out of hand pretty quickly. Not only is a file name such as `LP.JB.03/08` hard to remember (and even harder to type correctly, if you want to select the file from the keyboard), it is also more or less meaningless to anyone who doesn't know the code it is based on L=letters, P=personal, JB=correspondent Joe Blow, 03/08=the date, of course.

A quick fix for this sort of problem is to write out the translation, in English, at the end of the cryptic description. So the letter to Joe Blow would have the file name `LP.JB.03/08-Joe Blow Letter`. A long name, to be sure, but at least one that most people can understand, and one that can be called up quickly from the keyboard (if you know the code). A name like this also ensures that, when (not *if*, but *when*) you forget the code yourself, you too can figure out what the file is.

A much better solution, of course, is to reorganize your files and folders (as in the previous sections) so that obscure codes are unnecessary.

Summary

- Except in special circumstances, you should always use the most up-to-date version of the Macintosh system files, since they usually offer valuable enhancements and improved speed and performance.
- If you own a hard disk, be sure that it contains only one System folder. Use this one System folder and the one System file it

contains (what I call the master System folder and master System file) with all the applications on your disk drive. Keep your master System file small enough so that it fits on a single floppy disk and is thus easily backed up.

- If you use a floppy-disk-based setup, try as much as is practical to work with a single disk containing only the master System folder in one drive and all your application/document disks in the other drive. If you must have System folders on several different disks, be sure the System files in all the folders are identical (except for a few application-specific fonts and/or desk accessories).
- As much as possible, follow the basic rules of good file organization:
  - ◊ Within folders, group folders with folders and files with files.
  - ◊ Nest folders no more than three or four deep.
  - ◊ Name your folders so that it is immediately obvious in a list box that they are folders and not files.
- Floppy-disk system owners have two major disk organization strategies to choose from:
  - ◊ If the System folder is large (over 700K), then an entire disk should be dedicated to it and used with all applications.
  - ◊ If it is not possible to monopolize one disk drive with a large, master System disk, then System folders and applications should be combined on each of several disks devoted to your most frequently used applications. Lesser applications, stored on separate disks, can run from the System folders on the major application disks.
- Hard disk owners have a variety of file organization schemes to choose from, but three general principles apply, no matter which one is used (and they apply to floppy disk users as well):
  - ◊ Use a scheme that works for you, even if it seems a little messy or illogical.
  - ◊ Keep commonly used applications and files near the top of the organizational tree and less often used files in more deeply nested folders.
  - ◊ Feel free to change your organization as your work needs and habits change.
- If your folders are overwhelmed with support files (help files, dictionaries, etc.), put them out of the way in a special folder all

their own and use *Set Paths* to tell the Macintosh to look for them there.

- If you have a large-capacity hard disk and your folders are nested six or seven deep or are filled with dozens and dozens of files, consider partitioning your hard disk with a program such as *Hard Disk Partition* into subvolumes, which can be put away when not needed. It will make file-finding easier and speed up your Macintosh.
- There is no limit to the number of ways you can organize your windows; the important thing is that you do, indeed, organize them. On the desktop, neatness counts.
- Different window views should be used for different reasons. For example:
  - ◊ To sort your files in a particular way (such as by name), use a list view.
  - ◊ To arrange your files in order of importance, use an icon view.
  - ◊ If a folder contains many files, use the By Name list view.
- If you don't name files so you can easily find and recognize them again, all your other organization is for nought. Good file-naming practices include:
  - ◊ Prefixing the names of important files with special characters that cause them to float to the top of file selection lists and By Name views.
  - ◊ Naming your files so that they can be selected from the keyboard by typing only three or four characters.
  - ◊ Being careful not to abbreviate file names so much that they turn into a kind of code; you might forget the code someday.

3

The System File, Part 1: FKeys and Desk Accessories

FKeys: Why, How, and Which Ones *72*
- An Annotated FKey Sampler *73*
- Putting Your FKeys to Work *80*
- Some FKey Words of Wisdom *97*
- Other Ways to Work with FKeys *98*

Desk Accessories: Why, How, and Which Ones *103*
- An Annotated Desk Accessory Catalog *104*
- Putting Your Desk Accessories under the Apple Menu *123*

Summary *136*

Even with MultiFinder to help us run several programs at once, desk accessories and FKeys (desk-accessory-like programs which are usually called up with Command-Shift-number key combinations) remain important cogs in the machinery that drives the fully powered Macintosh. They can make your daily routine a lot easier, by placing many indispensable utilities at your fingertips. They can add entirely new functions to existing applications, creating a low-powered version of integration when RAM is in short supply. They can also be important in multi-program integration schemes (see Chapter 6), for they provide a way to perform Finder functions (such as deleting files, finding out how large certain files are, or determining how much disk space is available) in the absence of the Finder.

Despite their utility, desk accessories, and FKeys are frequently used ineffectively, or just plain ignored. This is especially true for people who have a system file for every application. The time and effort it takes to assemble and install useful desk accessory and FKey collections in the System file of each and every disk is daunting, and helpful desk accessories often get lost in the floppy disk shuffle. But there are other reasons as well: it is hard to keep up with the proliferation of desk accessories and FKeys, for one thing. For another, commercial desk accessory or FKey collections can seem expensive (since they are, after all, *only* desk accessories or FKeys), unless you know that this one or that one contains something which is especially useful.

On the other extreme, for some people the biggest problem with fonts, desk accessories, and FKeys is that there simply isn't enough room for all of them. With Font/DA Mover, you can install fourteen or fifteen desk accessories, but no more. What with the abundance of useful desk accessories, and desk accessories that take up not just one, but two slots, it is easy to run out of room under the Apple menu long before your appetite for desk accessories is slated. As for FKeys, in addition to the four that are standard with every System file, there are normally only six additional FKey slots that can be filled. Yet there is also an abundance of useful FKeys. As for fonts, it is hard to resist loading up the font menu till it nearly bursts.

It is no surprise then that many attempts to maximize the number of available fonts, FKeys, and desk accessories lead to a bloated System file. Even modest attempts can result in a one-megabyte System file. Although your System file doesn't have to be lean and mean, it is a good idea, as I have mentioned before, to keep it small. For floppy disk based systems, it must be small enough to fit, with all the other System files, on a single floppy disk. For hard disk owners, it should be small enough to fit on a single backup disk (in other words, it should be just under 800K, unless you

have 1600K disk drives). To fit a System file within these size restrictions, it is usually necessary to install fewer desk accessories, FKeys, and fonts in your System file than one would like.

To deal with this problem, you can follow one of at least two possible strategies. One strategy involves careful choices and compromises. It replaces desk accessories with equivalent FKeys as much as possible, creates a special FKey menu, installs specialty desk accessories in applications instead of the System file, and makes it possible to call up desk accessories and FKeys that are not installed in the System file or anywhere else. This strategy may well double the number of fonts, desk accessories, and FKeys you have been accustomed to until now, and can be employed by anyone, no matter what hardware or software they are currently using.

The other strategy requires a hard disk (to provide enough disk space) and a font/desk accessory extender program such as *Suitcase*. With such a combination, you can have an almost unlimited number of fonts and desk accessories (and, with a program like *Suitcase*, FKeys) on their respective menus, all without adding appreciably to the size of your System file.

This chapter and the next provide all the information you need to follow one or the other of these two strategies. They appear at this point in the book because it is important that your master System file, with its special combination of FKeys, desk accessories, and fonts, be in place (and that you put your files and folders in order, as discussed in Chapter 2) before moving on to other aspects of the fully powered Macintosh.

For those who must be concerned about the size of their System file, for anyone who works with a floppy-disk-based system, or for any hard disk owner who does not own a font/desk accessory extender such as *Suitcase*, Chapters 3 and 4 will show you how to create the best possible System file which occupies the smallest possible amount of disk space.

Those with a hard disk and a font/desk accessory extender will be under far fewer constraints and have to compromise in the choice of fonts and desk accessories (and, in some cases, FKeys) far less often. For these people, this chapter and the next offer other important tips and techniques: creating an FKey menu from which you can invoke any number of FKeys; choosing the right desk accessories for your needs and memory requirements; installing fonts and desk accessories in unusual places; calling up FKeys, desk accessories, and fonts which are not on the menu; cutting screen font files down to size; changing font names so that they appear where you want them on the font menu; adding special characters to existing PostScript fonts; making new FKey keyboard commands in addition to the ten Command-Shift-number key combinations; dealing with the space problems created by downloadable font files; and more.

If you plan to follow along on your Mac as I discuss these first, important steps in creating a fully powered Macintosh, then here is the place to begin. To start, you need to locate an unadulterated copy of the most recent version of the System file and copy it to a blank disk. (Even if you have a hard disk, it is easier and safer to construct a new System file from scratch on a floppy disk and then transfer it to the hard disk when you are done.) You are now ready to begin.

FKeys: Why, How, and Which Ones

Although the Macintosh SE and Macintosh II are the first to have keyboards that offer dedicated function keys, the kind that are labeled as such on the keyboard, those of us who do not own an extended keyboard need not feel left out. Those of us with older keyboards, or the so-called standard keyboard, have access to the functional equivalent of function keys: FKeys. FKeys are special functions invoked by pressing Command, Shift, and a number key all at once. All Macintosh System files come already equipped with a number of FKeys: for ejecting disks from floppy disk drives, for example, and for sending a reproduction of the screen to a disk file or to the printer.

The FKeys that are standard in all System files are but the tip of the iceberg. Over the past few years, programmers have been busy creating all manner of FKeys. Some of them provide functions that are otherwise unavailable. Other FKeys are functional equivalents of useful desk accessories that allow you to replace the desk accessory with the FKey and thus make a desk accessory slot available for some other purpose. Still others are just convenient ways of accomplishing a task which could be done in some other, but less convenient, way.

Since quite a few FKeys and desk accessories perform the same functions, the question is, when you run across FKeys and desk accessories that do the same thing, which one should you install? Although there is no hard and fast rule, in general, if you do not own a font/desk accessory extender or don't have the disk space to take advantage of one, it makes sense to choose the FKey over the desk accessory.

The reason for this is simple: the System file limits the number of desk accessories that can be installed to fourteen or fifteen. The number of FKeys you can install (if you use *FKey Manager*'s suite of programs), on the other

hand, has no upper limit, at least in theory. Thus you can use FKeys to decrease the demand for desk accessory space. We will, in fact, look at a number of other ways to get around the limit to the number of desk accessories later in the chapter. But substituting FKeys for desk accessories is an equally valid method and an often effective one.

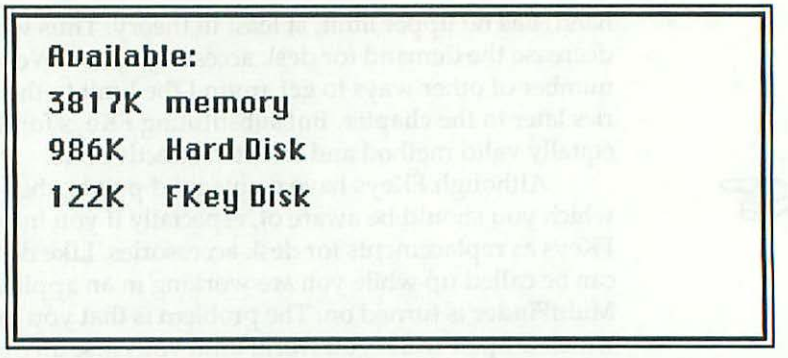
Although FKeys have many good points, they do have one drawback which you should be aware of, especially if you intend to use one or more FKeys as replacements for desk accessories. Like desk accessories, FKeys can be called up while you are working in an application, whether or not MultiFinder is turned on. The problem is that you cannot keep an FKey window open while you work; until you close an open FKey, you cannot return to your application. If you are looking for a desktop utility, such as a notepad or a window which shows free memory or disk space, for example, and you have a choice between an FKey and a desk accessory, remember that only the desk accessory can be kept in view while you work in other windows.

An Annotated FKey Sampler

There must be hundreds of different FKeys by now—far too many to describe here. What follows, then, is an annotated list of some of the FKeys which are widely available (either from public-exchange sources or as parts of commercial software packages) and which I have found useful from time to time. As you browse the list, consider your work habits and think about which FKeys would be useful in your everyday routine.

The FKeys marked with a § symbol are used in techniques and strategies described elsewhere in this book. You'll find more information about them in the discussions of these strategies later on. If the FKey is part of a commercial package sold under a different name, the name of the software package is indicated in parentheses (along with the name of the publisher). For the publisher's name and address, consult Appendix C. If the FKey is public-exchange, the author's name (if known) and a note as to whether there is a shareware fee involved (as of press time and to the best of my knowledge), follows the name of the FKey.

Available Key (Quick & Dirty Utilities, Volume 2, Dreams of the Phoenix) Select this FKey to find out how much memory remains available in RAM and how much unused space there is on all mounted hard and floppy disks.



Available Key *shows free RAM and disk space.*

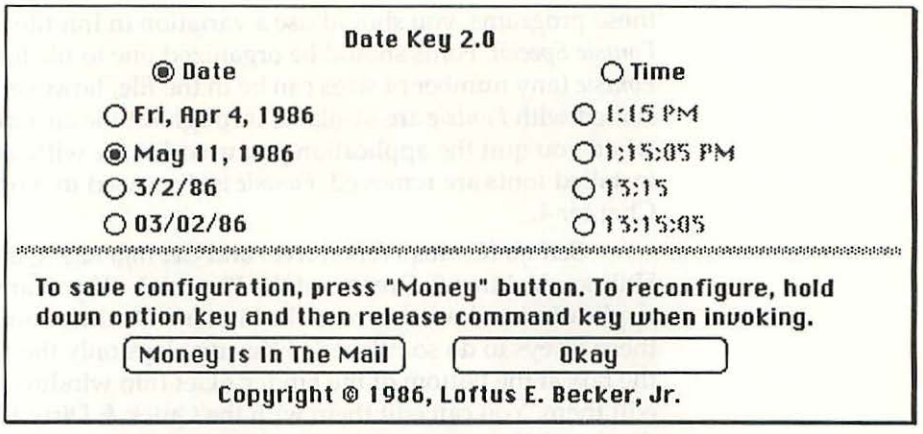
BigCursor (Andy Hertzfeld, free) If you have trouble seeing the Mac's pointer, this FKey doubles the size of the pointer. Select *BigCursor* again to return the pointer to normal size.

Calendar Key (Desk Necessities, MicroSparc) This one displays a small calendar, one month at a time.

Copy Screen (Joe Miller, free) This FKey is a useful variation on the standard screen shot (available with Apple's FKey 3). Instead of converting the entire screen into a Paint document, this FKey puts a copy of a selected portion of the screen into the Clipboard. When you first call up *Copy Screen*, the pointer is replaced with a small, upper-left-corner angle. Drag the pointer down and to the right to create a rectangle (as with MacPaint's rectangle tool), release the mouse button, and a copy of what is inside the rectangle is pasted into the Clipboard.

§ *DA Key* (Loftus E. Becker, Jr., shareware fee) This FKey allows you to use desk accessories which are not installed in the System file. When activated, it will open any desk accessory stored in a Font/DA Mover ("suitcase") file. Since *DA Key* also places the selected desk accessory under the Apple menu, you can close and reopen the desk accessory as often as you like, just like any other desk accessory. When you quit that application, the desk accessory is automatically removed. *DA Key* comes with its own installer program.

Date Key (Loftus E. Becker, Jr., shareware fee) Select *Date Key* to enter either the current date or the current time at the text insertion point. You can choose from four formats for both date and time. Like *DA Key*, *Date Key* comes with its own installer program.



Date Key *offers four formats for date and time.*

Dvorak (free) If you want to convert your keyboard from QWERTY to Dvorak, simply select this FKey. Of course, the key labels stay where they are, so you need to have some idea of Dvorak layout to use this one. Select it again to revert to QWERTY.

FadeKey (Carlos Weber, free) This is a screensaver which, when selected, slowly fades the screen to black. The screen remains black until you click the mouse or press a key, so don't be fooled into thinking the computer is off when it isn't.

FKey Sampler (David Kalin, free) *MasterKey* (Quick & Dirty Utilities, Volume 2, Dreams of the Phoenix). If you use *FKey Manager*'s *Pop-Keys*, you won't need this FKey, since the *Pop-Keys* menu has a similar built-in function. But if you do not use *Pop-Keys*, then FKeys such as these are very useful. With them, you can open and use any FKey which is still in an FKey file (and thus not installed in the System file). *FKey Sampler* and *MasterKey* thus infinitely expand the number of FKeys you can have at your disposal.

FKey-DA Sampler (John Holder, shareware fee) This is essentially a combination of *FKey Sampler* and *DA Key*. If you want both functions, then you might as well use this one FKey to do the job.

§ *Fontsie* (Loftus E. Becker, Jr., shareware fee) *Fontsie* makes it possible to temporarily add fonts not installed in the System file to your application's font menu. *Fontsie* does not work with some programs, such as *MacDraw* (Apple Computer), *Word*, and *SuperPaint*; for

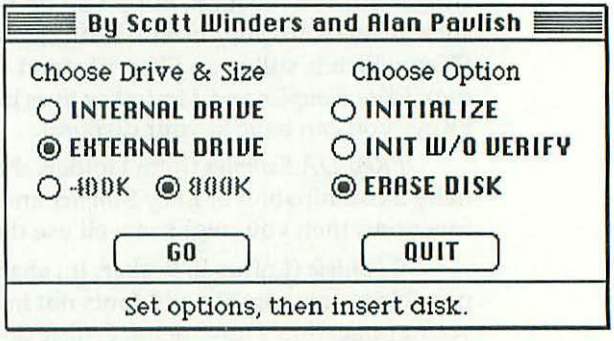
these programs, you should use a variation in Init file format called *Fontsie Special*. Fonts should be organized one to file for use with *Fontsie* (any number of sizes can be in the file, however). Fonts installed with *Fontsie* are available throughout the current work session; when you quit the application you used *Fontsie* with, all the *Fontsie*-installed fonts are removed. *Fontsie* is discussed in more detail in Chapter 4.

GetInfo (Carlos Weber, free) and *Get Info Key* (Quick and Dirty Utilities, Volume 2, Dreams of the Phoenix) If you are running an application and want to read the Finder's Get Info comments, use these FKeys to do so. Weber's FKey displays only the comments from the box at the bottom of the Finder's Get Info window, and you cannot edit them. You can edit them with the Quick & Dirty FKey, which also displays information about the file's size, location, and the creation and last modification dates.

Init (Scott Winders and Alan Pavlish, free) *Init* initializes or erases both 400K and 800K disks from within an application and does so using either the internal or external drive. It comes with its own installer program.

Loadkbd Key (Quick & Dirty Utilities, Volume 2, Dreams of the Phoenix) *Loadkbd* installs keyboard reconfigurations created with the *KeyMap* application also found on Quick & Dirty Utilities, Volume 2. Once installed, the new keyboard assignments will remain in effect until you load another one or reset your computer.

§ *MenuMaster* (Fred Reed, free) This FKey gives you the ability to change menu titles and command names and to assign or change



With Init, *you can erase a disk from within an application.*

command-key shortcuts. You can also use *MenuMaster* to change the style (bold, outline, etc.) of any command as it appears on the menu. *MenuMaster* will only work with standard menus. For more about *MenuMaster*, see Chapter 5.

Moving Lines (John Raymonds, free) This screen saver draws an ever-changing pattern of white lines on a black background. Click on the mouse button or press a key to turn the screen back on.

Password (William Steinberg, free) Once *Password* is invoked, your Macintosh will refuse to do anything until the correct password is entered, or until you reset your computer. To change the default password ("Password" with an uppercase P), you'll need *FEdit+* (MacMaster Systems) and the courage to alter a small amount of code. You should change the password *before* you install the FKey. *Password* is also available in Init resource and application format.

Pathname (free) A path is the route you would take, from disk to folder to file, to locate a given document or application. This FKey is a quick and accurate way to write down the path for any file. *Pathname* places the complete path name, properly punctuated, of the selected file in the Clipboard. Use the Paste command to move the path name into your document.

§ *SafeLaunch* (Fred Reed, shareware fee) *SafeLaunch* moves you quickly from one application to another. When selected, *SafeLaunch* automatically quits your current application and then opens the requested application bypassing the Finder. *SafeLaunch* is discussed in detail in Chapter 6.

ScreenToPaint (Steve Maller, free) This is a variation on Apple's FKey Command-Shift-3. As with that FKey, it saves an image of the standard Mac Plus or SE screen to disk but does so much more elegantly: it centers the image on the page, delineates the boundary of the screen (and rounds the corners), and numbers the Paint files it creates from A to Z.

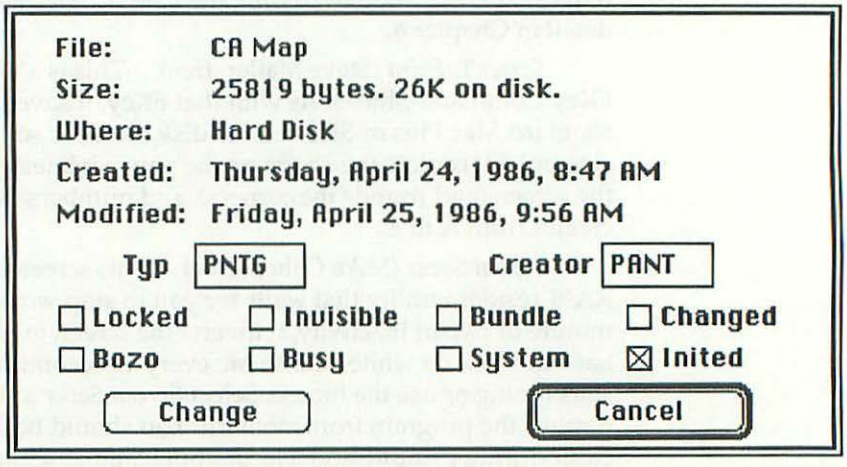
Screen Saver (Mike Cohen, free) This screen saver installs a RAM-resident utility that waits for you to stop working. After a minute or two of inactivity, it inverts the screen to white on black, then back to black on white, and so on, every 15 seconds or so until you start typing or use the mouse. Select *Screen Saver* a second time to remove the program from memory. You should be sure to test *Screen Saver* during printing and telecommunications sessions to be sure it does not adversely affect these two common, time-consuming activities.

SetFile Key (Quick & Dirty Utilities, Volume 2, Dreams of the Phoenix) This FKey displays information about the selected file (size, location, and dates created and last modified) and allows you to change the type and creator of the file and reset its Finder attributes. Be sure you know what you're doing or consult a Macintosh programming manual or a local programming wizard before attempting to change anything that *SetFile Key* allows you to change.

Space Warp (free) This is yet another screen saver. This one puts a stream of fast-moving stars on the otherwise black screen. To return to earth, press a key or click the mouse button.

SS Key (free) Use this FKey to quickly change the volume level of the internal speaker. All you do is call up *SS Key* and press a number key: zero for silent running, or one through seven for ever greater volume levels. *SS Key* responds with a beep of the selected level (or a menu flash for a volume of zero) and then quits.

Time Logger (Loftus E. Becker, Jr., shareware fee) This FKey version of the *Time Logger* desk accessory is a cheap and easy way to keep track of how you use your Mac. *Time Logger* stores, in text-only files, information on each file you open: date and time opened, date and time closed, number of minutes used, and a file description and special code entered by you when the file is opened. You can store everything in a single time log, or maintain separate logs for different



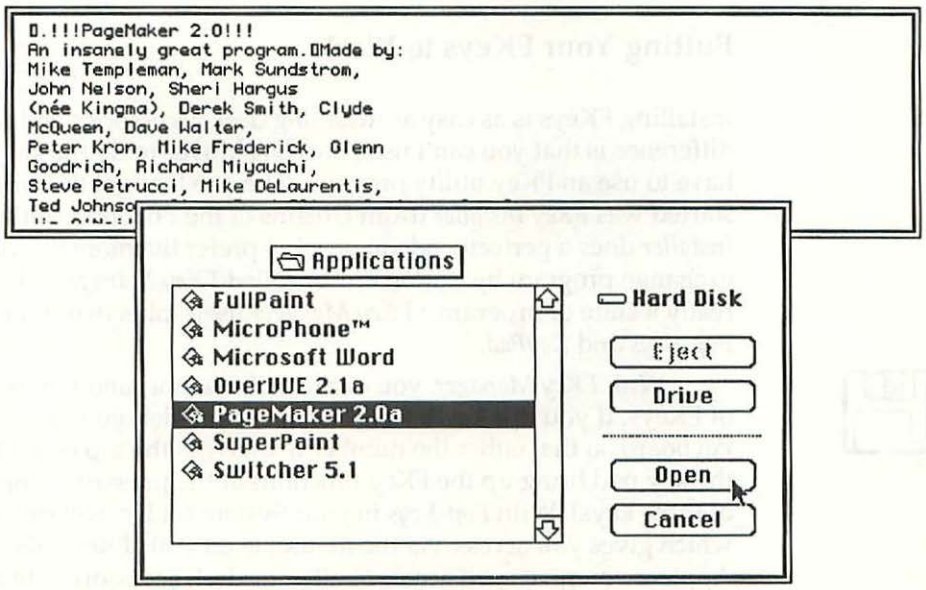
SetFile Key *displays information about a file and lets you change a number of file attributes.*

jobs. Since it is in text-only format, *Time Logger* data is readable by most word processors, spreadsheets, and databases.

Toggle Key (Quick & Dirty Utilities, Volume 2, Dreams of the Phoenix) and *FKToggle* (free) These two FKeys change the right and left angle brackets (< and >) to a comma and a period, so you will get commas and periods even when the Shift key is depressed. Be aware, however, that some programs do not recognize the keyboard changes attempted by either *Toggle Key* or *FKToggle*.

Version Reader (John Holder, free) This FKey displays the contents of the selected application's signature resource, which usually contains the program's version number, date, and sometimes quite a bit more.

Window (Charlie Bounds, free) Window sends the current window to the back and brings all other windows forward one level. It is a simple but useful addition to any multi-window program (including the Finder) which lacks a Window menu. Be sure to close any Alert or Dialog boxes before invoking *Window*.



Version Reader *can often tell you what version of an application you have (if you can't find out any other way). Sometimes it reveals even more.*

Window Manager FKeys (Michael Long, free) This is a set of three FKeys called *Send To Rear, Toggle Window Size,* and *Close Window. Send To Rear* is similar to Charlie Bounds' *Window* FKey; in addition to rotating windows within an application, however, it allows you to rotate the windows of different applications in MultiFinder. *Toggle Window Size* achieves the same effect as clicking on the zoom box of the active window; if the window is full-size, it will shrink the window to its previous size; if the window was previously shrunk, it will zoom it to full-size. *Close Window* closes the window, a handy little item if your application has no keyboard equivalent for closing windows.

Remember, this list is not complete but merely representative of what is available. Keep in mind, as well, an important point about FKeys. Life and Macintosh software being what they are (highly complex), not every FKey will work flawlessly in combination with every other FKey (and desk accessory and Init resource file). Therefore (and this is a common refrain throughout this book), you should be sure to thoroughly test every FKey you intend to use, preferably from a fully configured System disk, before using it during regular working hours.

Putting Your FKeys to Work

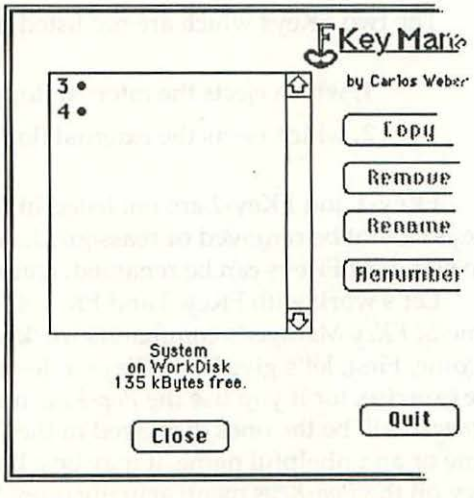
Installing FKeys is as easy as installing desk accessories and fonts. The only difference is that you can't use Font/DA Mover to do the job. Instead you have to use an FKey utility program. The one that got the whole FKey thing started was *FKey Installer* (from Dreams of the Phoenix). Although *FKey Installer* does a perfectly adequate job, I prefer the more versatile public-exchange program by Carlos Weber called *FKey Manager. FKey Manager* is really a suite of programs: *FKey Manager* itself, plus two Init resource files, *Pop-Keys* and *KeyPad.*

With *FKey Manager,* you can install, rename, and remove any number of FKeys. If you put *KeyPad* in your System folder, you can configure the keyboard so that either the number row across the top of the keyboard or the key pad bring up the FKey functions at the press of a single key (instead of three keys). With *Pop-Keys* in your System folder, you can set up a menu which gives you access, via the mouse, to all your FKeys, the same way the Apple menu gives you access to all your desk accessories. In addition, with the *Pop-Keys* menu's Other... command, you can temporarily call up any FKey which is not yet installed.

Managing Your FKeys with *FKey Manager* The *FKey Manager* window looks much like Font/DA Mover's, and functions in much the same way. It opens with a list of all FKeys installed in your System file in the scrolling list on the left, a row of command buttons in the middle, and another list box on the right. The name of each FKey on the list is preceded by its "resource number." If that number is between zero and nine, you can call up the FKey from the keyboard by pressing the Command, Shift, and appropriate number keys all at the same time. If the resource number is ten or higher, then the FKey is not normally available from the keyboard and you'll need to use special methods to open it (such as *FKey Manager*'s *Pop-Keys*, described on page 89, *QuicKeys*, described on page 99, or *Font/DA Juggler Plus* described on page 136).

To get started with FKeys:

1. Place a copy of the System file you intend to work on onto a floppy disk. Call it your work disk. If there is room, copy *FKey Manager*, the Finder, *Pop-Keys*, and *KeyPad* onto the disk as well, so you can use the disk as a startup disk and use *FKey Manager* to configure *Pop-Keys* and *KeyPad*. If there is not enough room, put them on a separate disk.



When you begin, make sure the System file on your work disk is displayed in the left list box.

2. Insert the work disk in a floppy disk drive and restart the Macintosh so you are running off the System file on the work disk. If *FKey Manager* is on a second disk, insert that disk in your other drive. Launch *FKey Manager* in the usual way.

3. If the text under *FKey Manager*'s left list box indicates that the file displayed is not the System file on the work disk, click on the Close button under the left list box. Click the button again when it says Open, and then locate and open the System file on the work disk.

If you have started with a fresh, unsullied System file which is configured for a Mac Plus, *FKey Manager*'s left list box will now contain only two FKeys, two of the four that are installed in every Mac Plus System file. (If you have a Mac SE or a Mac II, your list of FKeys may be slightly different.) Since this is the first time *FKey Manager* has had the chance to work with them, these FKeys should be unnamed (the space where the name should be is blank), but you can tell which is which by the number:

3, which saves the screen to disk

4, which prints the screen on the ImageWriter

The two FKeys which are not listed, by the way, are:

1, which ejects the internal floppy disk

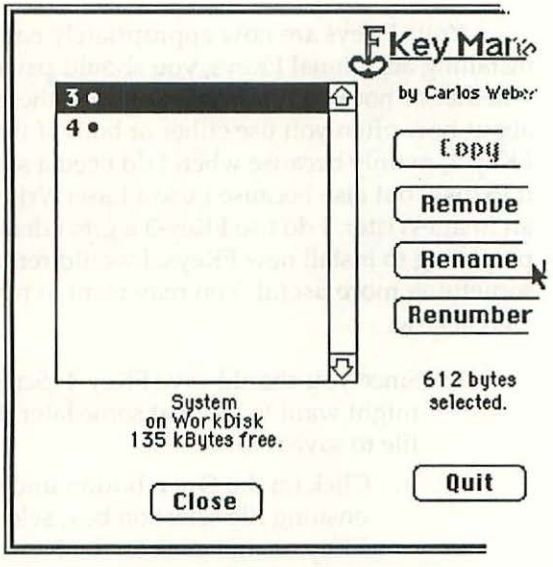
2, which ejects the external floppy disk

FKey-1 and FKey-2 are not listed in the FKey window because these FKeys cannot be removed or reassigned. As you will see, however, the other two standard FKeys can be renamed, reassigned, or removed, as you wish.

Let's work with FKey-3 and FKey-4 for a moment, both to see how some of *FKey Manager*'s commands work and to prepare for the new FKeys to come. First, let's give both FKeys a descriptive name. This is not just an idle exercise, for if you use the *Pop-Keys* menu, the names you assign in *FKey Manager* will be the ones displayed in the *Pop-Keys* menu. If the FKey has no name or an unhelpful name, it may be a bit difficult to figure out what the FKey on the *Pop-Keys* menu actually does. So the following instructions apply to any installed FKey whose name you want to change to something more clearly expressive of its function:

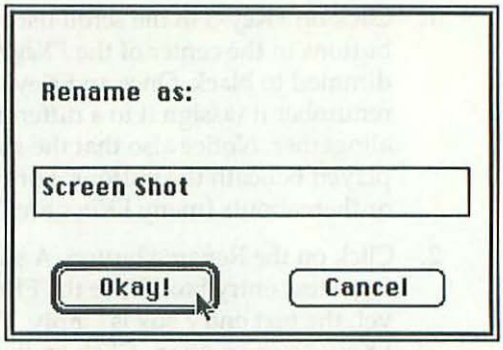
1. Click on FKey-3 in the scroll list box. Notice that three of the buttons in the center of the *FKey Manager* window change from dimmed to black. Once an FKey is selected, you can rename it, renumber it (assign it to a different FKey number), or remove it altogether. Notice also that the size of the FKey in bytes is displayed beneath the buttons. For FKey-3, it should read 612 bytes or thereabouts (many FKeys are, like this one, very small).

2. Click on the Rename button. A small window will appear containing a text entry box. Since the FKey in this instance has no name yet, the text entry box is empty. Type in the new name for the FKey: `Screen Shot`. Click on the **Okay!** button or press Return.



Click on FKey-3 to select it, then click on the Rename button.

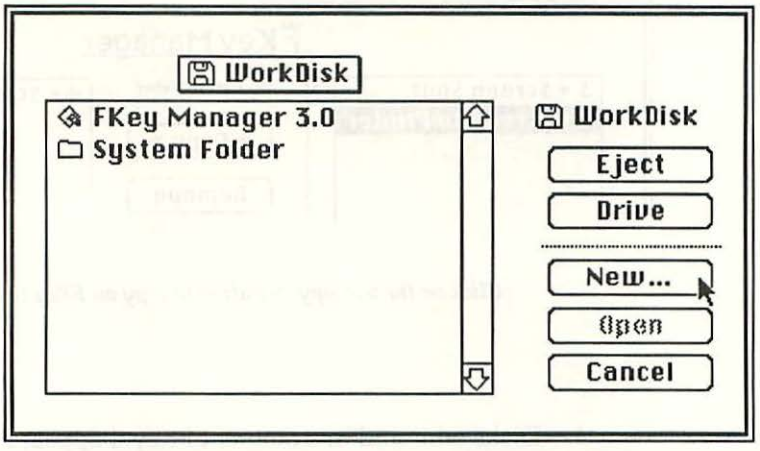
3. Now repeat the process with FKey-4: select the FKey in the list, click on Rename, enter the new name (`Screen to Printer`), and press Return.



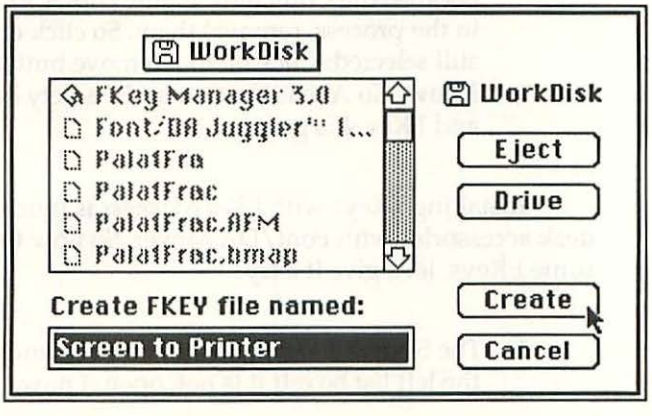
To rename the FKey, enter the new name and click on the Okay! button.

Your FKeys are now appropriately named, and before you start installing additional FKeys, you should pause for a moment to consider whether or not you really want or need these two standard FKeys. Think about how often you use either or both of these FKeys. I, for one, never use FKey-4, mainly because when I do need a screen shot I always want to save it to disk, but also because I use a LaserWriter, and FKey-4 only works with an ImageWriter. I do use FKey-3 a great deal, however. Thus if I were preparing to install new FKeys, I would remove FKey-4 to make room for something more useful. You may want to remove one or both FKeys as well. Here's how:

1. Since you should save FKey-4 (Screen to Printer), just in case you might want to use it at some later date, you need to create an FKey file to save it in.
   a. Click on the **Open** button under the right list box. In the ensuing file selection box, select the disk you want to save the FKey on and click on the **New...** button.
   b. Now you need to give your file a name. If you had selected the Screen to Printer FKey before clicking on the **New...** button, *FKey Manager* would suggest the name of the FKey as the name of the file. If no FKey was selected or if you want to give the file a different name, just type one in. Click on the **Create** button and you will be returned to the main window.
   c. The new, and so far empty, file will now be open in the right list box. (Check the file name under the list box to see that this is true.)

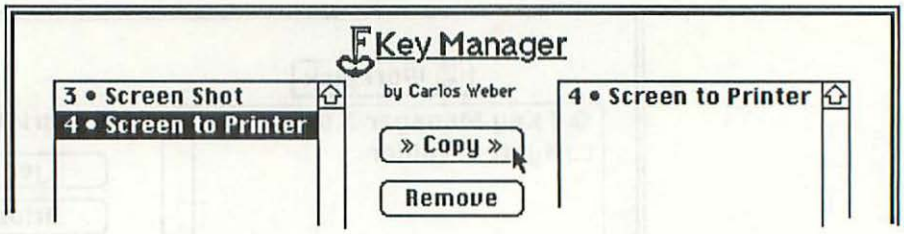


Click on the New... button to create a new FKey file.



Name the FKey file, if the default name needs to be changed, and click on the create button.

2. Click on FKey-4 if it is not already selected. Note that this time, since there is an open file in the other list box, all four buttons are black. Click on the >>Copy>> button.
3. A dialog box will appear asking you how to assign a number to this FKey. Since we are moving the FKey to an empty file, you can ignore this dialog box for now and simply press Return.



Click on the >>Copy>> button to copy an FKey to another file.

4. Disks whir and in a moment FKey-4 appears in the right as well as the left list box. Close the file on the right by clicking on the Close button. FKey-4 is now saved in its own FKey file.

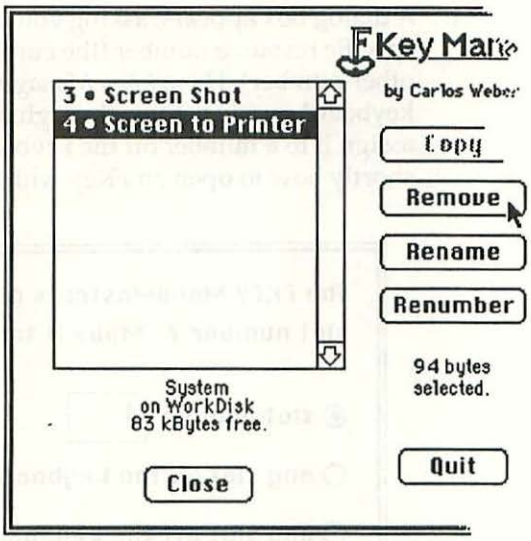
5. Finally, you need to remove FKey-4 from the System file. This is because the Copy button only copies FKeys, it does not move (and in the process, remove) them. So click on FKey-4 again if it is not still selected. Click on the Remove button and then click on the I Know, Go Ahead! button in the safety box. Wait a few moments and FKey-4 is gone.

Installing FKeys with *FKey Manager* is much like installing fonts or desk accessories with Font/DA Mover. So now that we are ready to install some FKeys, let's give it a try:

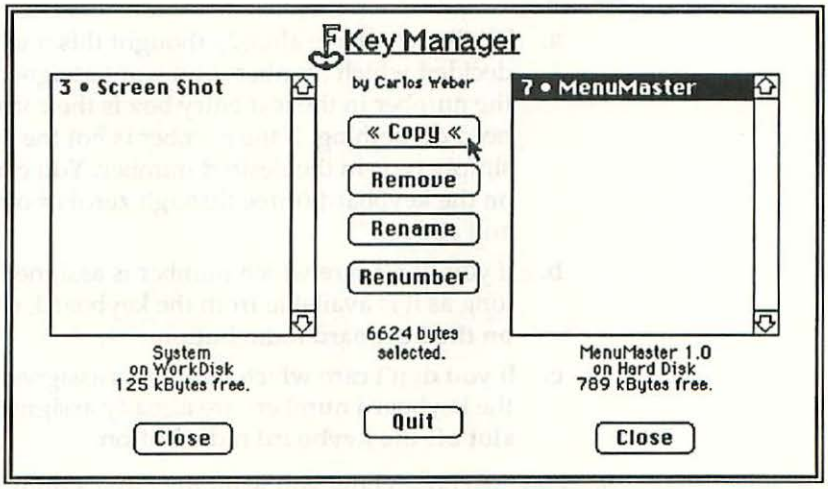
1. The System file should still be open and its contents displayed in the left list box. If it is not, open it now.

2. Click on the Open button under the right list box. Locate the file containing the FKey you want to install, clicking on the Drive button if you need to access a different disk. Select the desired FKey and click on the Open button. Let's say, for the sake of illustration, that you want to install *MenuMaster*. Locate the disk that *MenuMaster* is on and open the *MenuMaster* file.

3. Select the FKey on the right (*MenuMaster* in this example) so it is highlighted. Now click on the **<<Copy<<** button.

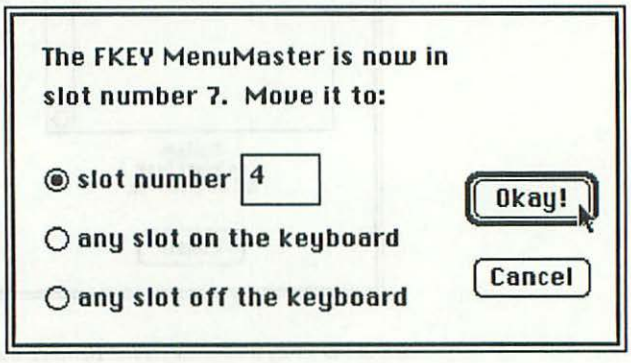


Select the FKey and click on the Remove button to delete it.



Use the <<Copy<< button to copy the FKey to your System file.

4. A dialog box appears, asking you if you want to assign the FKey a specific resource number (the currently assigned number or some other number). Have *FKey Manager* assign it to any available keyboard number (three through zero) or have *FKey Manager* assign it to a number off the keyboard (ten or higher). We'll see shortly how to open an FKey with a double-digit resource number.



When you install an FKey, you may want to give it a new number, or you can have FKey Manager *assign the new number for you.*

   a. Ideally, you have already thought this one through and decided which numbers you want assigned to which FKeys. If the number in the text entry box is the correct one, then you need do nothing. If the number is not the one you want, simply type in the desired number. You can enter a number on the keyboard (three through zero) or off the keyboard (ten and above).
   b. If you don't care which number is assigned to the FKey, as long as it is available from the keyboard, click on the **any slot on the keyboard** radio button.
   c. If you don't care which number is assigned to the FKey and all the keyboard numbers are already assigned, click on the **any slot off the keyboard** radio button.

5. Having chosen how you want the FKey's number assigned, click on the **Okay!** button or press Return. When *FKey Manager*'s main window next appears, the selected FKey will be listed in the left window.

You can continue in this way to add more FKeys to your System file. You can install as many FKeys as you want (within reason); you are not limited to eight, ten, or even twenty. Remember that only FKeys assigned single-digit numbers are available from the keyboard using the standard Command-Shift-number key combination. Thus, be sure to assign single-digit numbers to the FKeys you will want to use most often—the ones to which you will want quick and easy keyboard access. Try to number them in a way that makes it easy to remember which keyboard command goes with which FKey (that's not an easy task, I know, but at least you can try).

As you install your FKeys, also keep in mind that *FKey Manager*'s *Pop-Keys* menu (the subject of the following section), *FKey Sampler*, and *Font/DA Juggler Plus* (described at the end of this chapter) allow you to call up an FKey that is not installed but is in a standard FKey format file. Thus, you don't need to install every FKey you think you'll ever need. Indeed, the ones that will see infrequent use are best stored in an FKey folder or in a jumbo FKey file and called up when the occasion arises.

Setting Up a *Pop-Keys* Menu There are many who feel that FKeys, because they can only be called up from the keyboard using an arcane and somewhat complex key combination (and not from a menu), are regressive. With *Pop-Keys*, *Suitcase*, and *Font/DA Juggler Plus* however, you *can* select FKeys from a menu. You can select any of the ones which are already accessible from the keyboard, and, if you end up with FKeys assigned double-digit resource numbers in your System file, you can select all of them with keyboard commands as well.

To install *Pop-Keys*, you must place a copy of the *Pop-Keys* Init file in your System folder. Once installed, the *Pop-Keys* menu works in one of two ways. In one, the *Pop-Keys* menu opens anywhere on the screen whenever you click the mouse button while pressing a previously specified combination of Command, Shift, Option, and Caps Lock keys. In the other, you select a "hot spot" somewhere on the screen; if you click on (or simply put the pointer over) the hot spot, the *Pop-Keys* menu drops down. Once the menu is open, you either drag to or click on the desired FKey to launch it.

For example, my *Pop-Keys* menu uses the hot-spot, drop-down method. The hot spot on my Mac is a very small square to the left of the Apple on the menu bar, and the *Pop-Keys* menu drops down automatically whenever I put the pointer there. I click on an FKey name to select it or click anywhere else on the screen to close the menu. The menu itself lists all my FKeys and the resource numbers for those available from the keyboard. The numbers are there simply as a reminder of which FKey goes with which number on the keyboard. The last selection at the bottom of every *Pop-Keys*

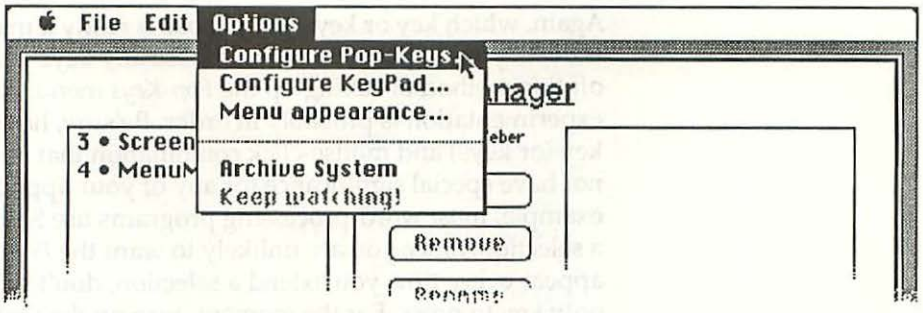
menu is the **Other...** command, which allows you to use temporarily any FKey which is not installed.

In order to set up your *Pop-Keys* menu the way you want it, you need to use *FKey Manager*'s **Configure Pop-Keys...** command. But first, you must make sure that all the required files are in their proper place, and that the System you want to configure is the startup System. Check the following points before attempting to set up your *Pop-Keys* menu:

- If at all possible, the System you configure should be the one on the work disk. This means that your work disk should contain the following files: System, Finder, *Pop-Keys* Init. If *FKey Manager* fits on there as well, great; if not, you can run it from a separate disk. Once everything is set up, restart your Mac with the work disk as the startup disk; this will ensure that the *Pop-Keys* Init is properly installed.
- If you have a hard disk and all the necessary files don't fit on a single floppy disk, then you'll have to configure the master System file on your hard disk. Altering an active System file is never the safest thing to do, especially on a hard disk. So caution and one or two backups of your System file are in order before you attempt this. (If you are careful, though, you shouldn't have any problems.) If you are working with a hard disk, don't forget to put a copy of the *Pop-Keys* Init file in the System folder. Then restart your Mac so that *Pop-Keys* is properly installed.

Now you're ready to configure your *Pop-Keys* menu. Here's what you need to do:

1. Launch *FKey Manager*. It will automatically open the startup System file, which should be the one you want to configure if you have everything set up correctly.
2. Select **Configure Pop-Keys...** from the Options menu. A dialog box, with an illustration above and a number of options listed below, appears.
3. On the right side, three options selected by radio buttons are prefaced with the rather cryptic heading "With mouse in rectangle." These three options have to do with the first of two ways in which the *Pop-Keys* menu can be made to appear on the screen. In this first method, the *Pop-Keys* menu only appears when the mouse pointer is over a hot spot (whose size and position you will



Select the Configure Pop-Keys... *command to set up your FKey menu.*

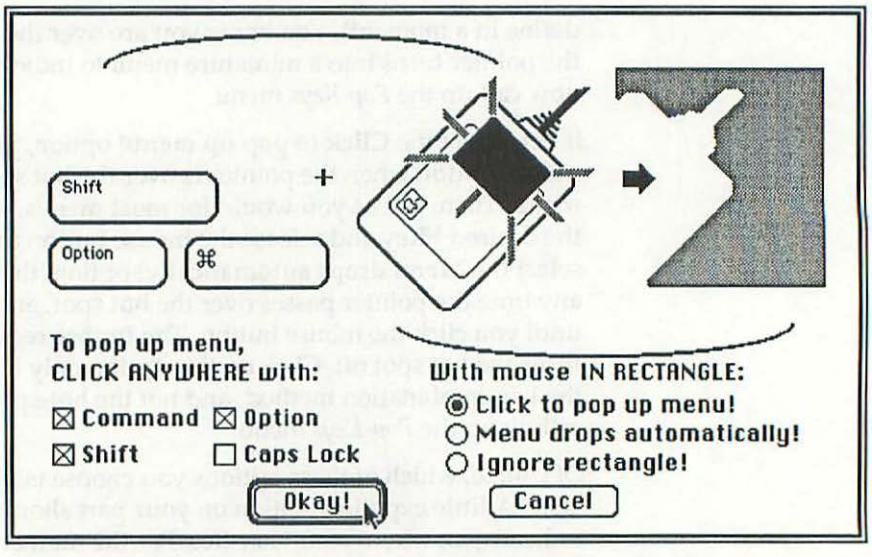
define in a moment). You know you are over the hot spot because the pointer turns into a miniature menu to indicate that you can now call up the *Pop-Keys* menu.

If you select the **Click to pop up menu!** option, you must press the mouse button when the pointer is over the hot spot to call up the menu. Then, just as you would for most menus, you must drag to the desired FKey and release the mouse button to select it. If you select the **Menu drops automatically!** option, the menu appears any time the pointer passes over the hot spot, and it stays down until you click the mouse button. The **Ignore rectangle!** option turns the hot spot off. Click on this option only if you want to use the key combination method, and not the hot-spot method of calling up the *Pop-Keys* menu.

Of course, which of these options you choose is purely a matter of taste. A little experimentation on your part should be enough to indicate just where your taste lies. For the moment, click on the **Click to pop up menu!** option.

4. On the left side are the four options for the second method by which the *Pop-Keys* menu can be called up. With this method, holding down one or several keys while you click the mouse button will make the *Pop-Keys* menu appear beneath the pointer, no matter where it is on the screen. The options on the left allow you to select just which key or keys must be held down: Option, Command, Shift, Caps Lock, or some combination of the four. Simply click on the check boxes of one, two, three, or (if you are really dexterous) all four keys to set the desired key combination.

Again, which key or keys you choose is really a matter of taste. Indeed, you may decide not to choose any keys at all and thus turn off this method of calling up the *Pop-Keys* menu. As before, a little experimentation is probably in order. Be sure, however, that the key (or keys) and mouse-click combination that you choose does not have special significance for any of your applications. For example, most word-processing programs use Shift-click to extend a selection. Since you are unlikely to want the *Pop-Keys* menu to appear every time you extend a selection, don't select Shift as the only key to press. For the moment, turn on the **Option, Command,** and **Shift** check boxes for an Option-Command-Shift-click combination.



The Configure Pop-Keys *window offers a number of different ways to call up your FKey menu.*

5. When you have chosen all your options, tentative though they may be, click on the **Okay!** button. A small notice now appears in the center of the screen, for there is one last thing to do: define the hot spot. The current hot spot is indicated by a black rectangle somewhere on the screen, probably in the upper-left corner, but you can put it anywhere you like. Simply drag the pointer (like the

marquee in your Paint program) to create a rectangle anywhere on the screen. Of course, you want the hot spot small and out of the way, or else your pointer will turn into the *Pop-Keys* pointer, or the *Pop-Keys* menu will drop down automatically (depending on which hot spot option you choose) at inconvenient moments.

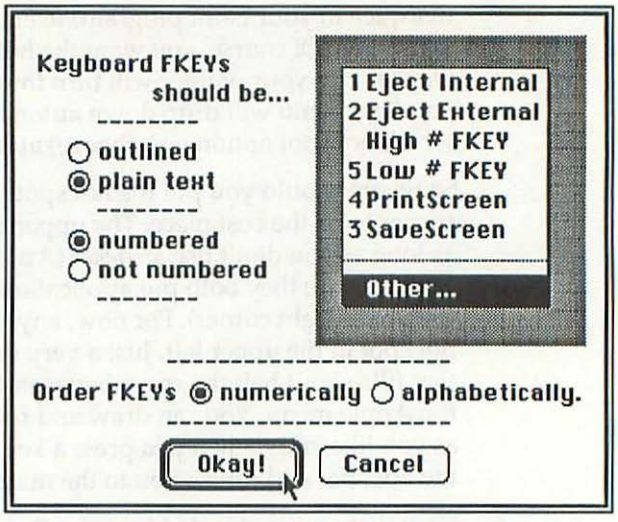
So where should you put the hot spot? I've found the upper-left corner to be the best place. The upper-right might work well too, as long as you don't use *Switcher* (Apple Computer) or Multi-Finder (since they both put application switching mechanisms in the upper right corner). For now, anyway, you might try creating a hot spot in the upper left. Just a very small rectangle will do, one that fills about half the space between the edge of the screen and the Apple menu. You can draw and redraw the hot spot as much as you like; only when you press a key does *FKey Manager* record the hot spot and return you to the main window.

6. Now that you've decided how the *Pop-Keys* menu should work, you can move on and customize the appearance of the *Pop-Keys* menu. To do so, select **Menu Appearance...** from the Options menu.

7. A new dialog box appears, containing a sample *Pop-Keys* menu and three pairs of options. Each pair determines which of two opposing display modes will be used for the menu. The sample menu on the right side of the dialog box illustrates the consequences of your choices as you select them:

   a. With the first pair of options, you decide whether the names of keyboard FKeys (those with single-digit numbers, which can thus be called up with a Command-Shift-number keyboard combination) will be displayed in outline text. To see how this will look, click on **outlined,** and the name of one of the FKeys in the menu will be displayed in outline text. Click on **plain text** and it will return to normal. I prefer a menu of more restrained appearance and thus select the **plain text** option. The choice, however, is yours.

   b. By clicking on **numbered** in the second pair of options, the menu will display the numbers assigned to the FKeys available from the keyboard (the single-digit numbers). Obviously, this does little to increase the utility of the menu itself. But if you, like me, tend to forget which number corresponds to which FKey, the numbers on the menu can serve as a re-



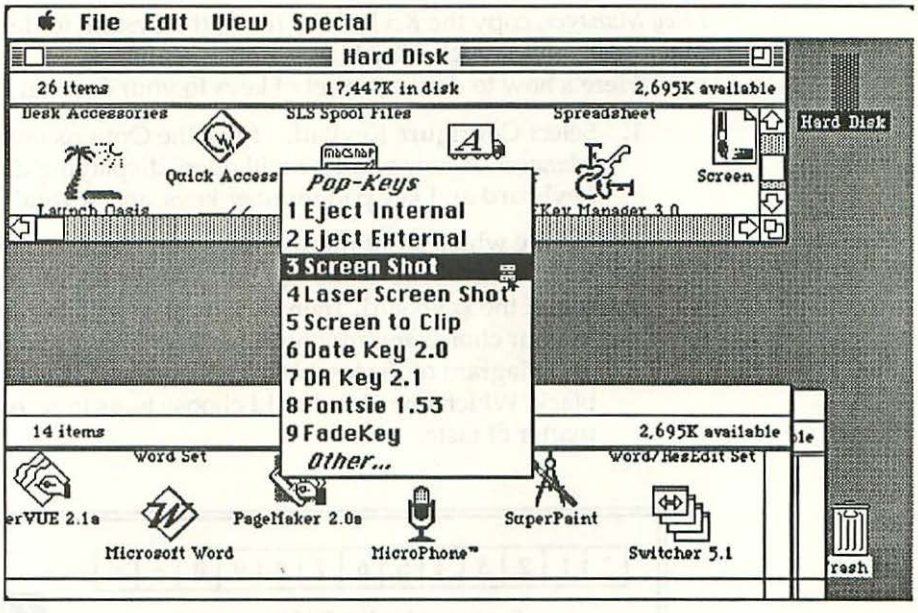
You can also customize the way your FKey menu looks.

minder. If you never use the keyboard commands, or you have a better memory for these things than I do, or you simply want a slightly less busy menu, you can choose not to have the numbers displayed by clicking on the **not numbered** option.

c. With the third pair of options, you can choose to have your FKeys listed in either numerical or alphabetical order. Numerical order makes the most sense if you have a small number of FKeys or if you have many FKeys but want the keyboard ones at the top of the menu. If you have a great many FKeys, however, alphabetical order probably makes more sense—unless you have numbered your FKeys so that they are ordered in some meaningful way (they are grouped by function, for example).

   This last option, by the way, is independent of the previous one. Thus, if you choose both the **alphabetically** and **numbered** options, the numbers will be there but not in numerical order.

8. When you have finished selecting all the options you want for your *Pop-Keys* menu, click on the **Okay!** button (or press the Return key) to save your choices and return to the *FKey Manager* window.



This Pop-Keys *menu was set to pop up in the center of the screen.*

Using *KeyPad* to Create Dedicated Function Keys on a Mac Plus

You can also use *FKey Manager* to customize the way the first ten FKeys are selected from the keyboard. Of course, the usual way is with Command-Shift-key combinations. But the *KeyPad* Init makes it possible, on a Mac Plus, to use single-key keyboard commands instead. You can, in other words, turn certain keys into dedicated function keys. The keys in question are the number keys across the top of the keyboard or the number keys in the key pad. Clearly, if you convert one or the other set of keys into dedicated function keys, you no longer have a choice of keys for entering numbers. If this sort of thing appeals to you, read on. If not, skip to the next section. (I, for one, had to eventually turn *KeyPad* off; I never could get used to the fact that it *did* matter which number key I pressed, and I was constantly frustrated by FKeys appearing when what I really wanted was a simple number.)

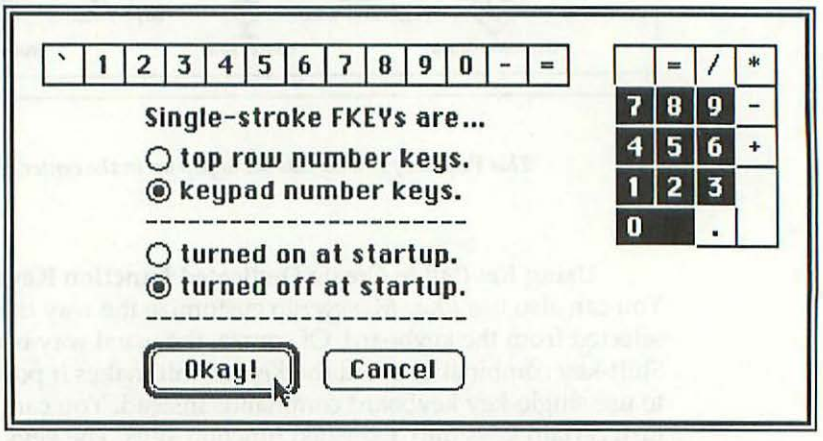
At this point, I should remind you that the *KeyPad* Init file must be in the System folder of the System you are working on in order to proceed. If it is not, the **Configure KeyPad...** command will be dimmed and you will be unable to select it. If your **Configure KeyPad...** command is dimmed, quit

FKey Manager, copy the *KeyPad* Init file to the System folder, restart your Macintosh, and launch *FKey Manager* once again.

Here's how to dedicate a set of keys to your first ten FKeys:

1. Select **Configure KeyPad...** from the Options menu in *FKey Manager*. A new window will open, displaying diagrams of the keyboard and key pad number keys, and several pairs of options.

2. Decide which set of number keys you want to convert into dedicated function keys: the key pad keys or the number keys at the top of the keyboard. Then click on the radio button corresponding to your choice: **top row number keys** or **keypad number keys.** The diagram of the key set you choose will invert to white on black. Which one you should choose is, as in so many things, a matter of taste.



From the Configure KeyPad *window, you can create a set of dedicated function keys on a Mac Plus keyboard.*

3. Since you can turn the *KeyPad* option off and on as you work, you now need to decide whether it should be on or off when you first start your Macintosh. Again, it's a matter of taste, although if you have any tendency to forget that the feature is on, I suggest that you start with it off, or you may have a rude surprise when you absent-mindedly press that first number key. Whatever you decide, click on the appropriate radio button: **turned off at startup** or **turned on at startup.**

4. To save your preferences and return to the *FKey Manager* main screen, click on the Okay! button. Since you are now finished with *FKey Manager*, select Quit from the File menu.

As mentioned, keyboard commands can be used to toggle the *KeyPad* feature on and off, so you don't permanently lose the numerical input capacity of the key pad or the top row number keys. To turn the *KeyPad* option on, press Command-Shift-Option-plus (that's the + on the key pad). To turn it off, press Command-Shift-Option-Clear (that's the Clear key on the key pad).

By the way, if you are uncomfortable with either of *KeyPad*'s options, that is, if you don't want your number keys to produce anything other than numbers, then the best thing to do is not put *KeyPad* in your System folder in the first place.

Some FKey Words of Wisdom

Finally, as you work with FKeys and *FKey Manager*, keep these points in mind:

- With *FKey Manager*, you can work with FKeys stored in FKey files, in System files *and* in applications; that's right, applications. As with fonts and desk accessories, you can store FKeys in applications. I for one, however, find a changing assortment of FKeys (as one changes from one application to another) too confusing. The fact that the available fonts and desk accessories might change with different applications does not bother me; but with FKeys, for some reason, it does. So my personal recommendation is to avoid putting FKeys in applications. However, if you do not have the same prejudices that I do, you may find that installing FKeys in applications is a useful technique.
- You can install as many FKeys as you like, but don't be greedy. *FKey Manager*'s author reports that he knows of some people who have as many as forty FKeys installed in their System file. Good grief! Whatever happened to moderation as a virtue?
- As you pick and choose among the many utility programs and techniques described in this book, always keep in mind that certain combinations can cause unexpected and undesirable problems. Utilities that alter the standard Apple keyboard layout (such as *MacKeymeleon*, which is discussed in Chapter 5) may cause conflicts with *Pop-Keys* and *KeyPad*. The click-anywhere

technique for calling up the *Pop-Keys* menu may simply stop functioning, for example. I've even had the standard keyboard commands for all my FKeys turned off by an unknown culprit. As you try this book's various suggestions on for size, always test before committing to any particular utility, technique, or combination thereof.

Other Ways to Work with FKeys

A number of utility programs can enhance your use of FKeys considerably. One, *FKey Maker* (public-exchange), is an easy-to-use program that creates a set of FKeys that call up the desk accessories in your System file. Another, *QuicKeys* (CE Software), provides—among many other things—a way to assign keyboard commands to your entire set of FKeys, rather than just the first ten. Thus, as an alternative to using the *Pop-Keys* menu and the mouse, *QuicKeys* makes it possible to call up all your FKeys without ever taking your fingers from the keyboard.

Homemade FKeys with *FKey Maker* *FKey Maker* is a relatively simple program in concept and operation. When launched, *FKey Maker* creates a file (called `Open DA FKeys`) full of FKeys. The function of each FKey in the file is to open one of the desk accessories currently installed in your System file. If you install one (or more) of these FKeys with *FKey Manager*, you will be able to call up the associated desk accessory with a Command-Shift-number keyboard command. (Of course, you must install it as one of the first ten FKeys in order to make it accessible with a Command-Shift-number command.) The price of this convenience, of course, is fewer keyboard FKeys devoted to "real" FKey functions.

To see how *FKey Maker* works and how it might be useful to you, consider the following illustration. The general idea of this example is to replace the standard Apple Command-Shift-3 FKey, which copies the current Macintosh screen to a disk file, with an FKey which calls up a desk accessory called *Camera*. (Consult the annotated desk accessory catalog later in this chapter to see why this might be desirable.)

1. First, make sure that a copy of your master System file, the one with the exact desk accessory collection with which you intend to use the new FKey, is currently in control. If not, restart your Macintosh with the correct System file.

2. Launch *FKey Maker*. A window will open which first introduces *FKey Maker* and then displays the names, one by one, of the desk accessories for which FKeys are being created. When the final information window appears, click the mouse button to quit the program.
3. A new file called `Open DA FKeys` should now be in the same folder as *FKey Maker*. This file contains a set of FKeys, each one of which opens a particular desk accessory in the System file.
4. Launch *FKey Manager*. The FKeys currently installed in the System file should be listed on the left. If Apple's standard FKey-3 (to which we gave the name `Screen Shot` a while back) is still on the list, select it and click on the Remove button (or copy it to a different file to save it and then remove it).
5. Click on the Open button on the right and open the `Open DA FKeys` file. You should now see a list of FKeys with names like `105•Control PanelFKEY`. Locate and select the FKey that has Camera in its name in the `Open DA FKeys` file; it should read something like `101•CameraFKEY` (the number of the FKey in your file will probably be different). Click on the copy button to copy it to the System file. In the resulting dialog box, enter `3` when asked what number to assign the FKey to, then click on the Okay! button.
6. Quit *FKey Manager*. Now test Command-Shift-3; the *Camera* dialog box should appear.

If you want to be able to select other desk accessories from the keyboard, simply install the appropriate FKeys from the `Open DA FKeys` file. Of course, it only makes sense to assign *FKey Maker* FKeys to the FKey numbers that have keyboard equivalents: FKey-3 through FKey-0. This means that you can install a maximum of eight *FKey Maker* FKeys, and if you install the maximum number, all your other FKeys will be available only from the *Pop-Keys* menu. To make all of your FKeys and all of your desk accessories available from the keyboard, you need a program such as *QuicKeys*.

FKeys and More from the Keyboard, Using *QuicKeys*

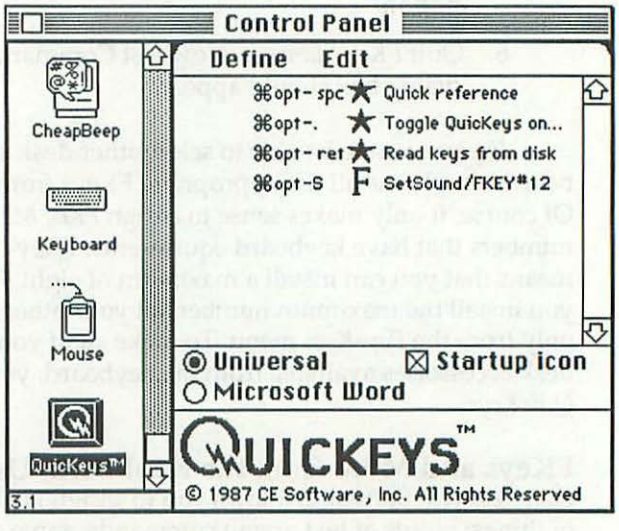
QuicKeys (CE Software) allows you to assign keyboard shortcuts to all sorts of things: pieces of text, menu commands, some other key or character, the current date or time, and quite a bit more. Although not as powerful as

Tempo II (which is described in more detail on page 410) and sometimes more awkward in operation, *QuicKeys* can do easily at least one thing that *Tempo II* would find more difficult: assign keyboard commands to all the FKeys in your System file.

With *QuicKeys*, you can forego the machinations of the last section and assign keyboard shortcuts to all your FKeys and desk accessories without resorting to *FKey Maker*. In order to use *QuicKeys*, you will have to buy a copy, however. Although I definitely think it is worth it, it has to be said that *QuicKeys* costs a lot more than *FKey Maker* (which is free).

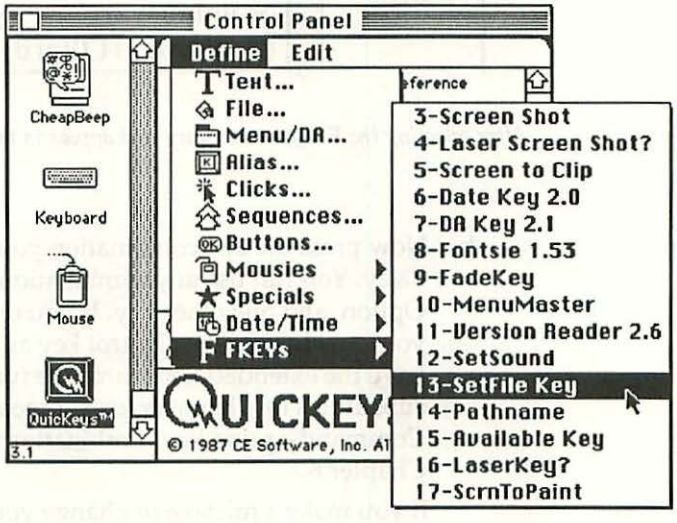
If you have *QuicKeys*, here's what you would do to assign a keyboard command to an FKey. First place the *QuicKeys* file in your System folder and restart your Macintosh. Then follow these steps:

1. Select the Control Panel from the Apple menu. Locate the QuicKeys icon in the scrolling list on the left side of the Control Panel window and click on it. This will open the QuicKeys configuration panel.
2. Make sure the **Universal** option, near the bottom of the configuration panel, is turned on. If not, click on the **Universal** radio button.



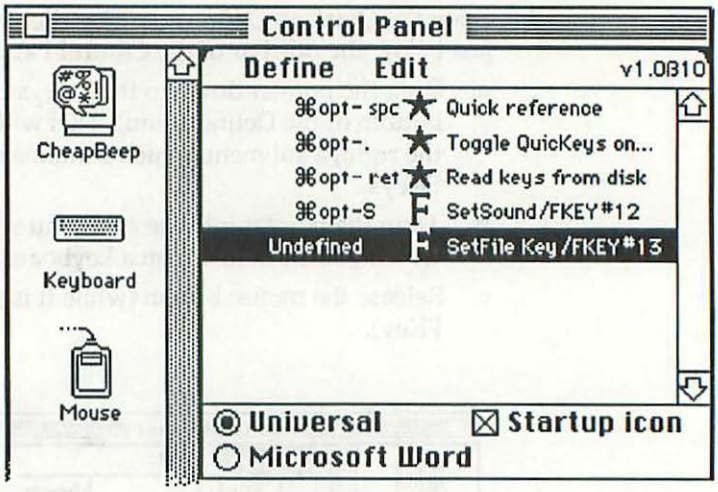
QuicKeys *shortcuts are created using the Control Panel.*

3. Open the *QuicKeys* Define menu by clicking on the word Define just below the title bar of the Control Panel window.
   a. Drag the pointer down to the **FKeys** command (at the very bottom of the Define menu). This will open a second menu to the right, a submenu which contains the names of all your FKeys.
   b. Drag the pointer into the submenu and select the FKey to which you want to assign a keyboard shortcut.
   c. Release the mouse button (while it is still over the desired FKey).



The QuicKeys *FKey submenu makes it easy to choose an FKey.*

4. Back in the main *QuicKeys* configuration panel, your FKey selection should now appear in the list of *QuicKeys* shortcuts. The listing should show (with a big F) that you have selected an FKey, display the FKey's name on the right, and on the left, indicate that the keyboard shortcut has yet to be defined. So let's define the key combination:
   a. Your new FKey listing should be highlighted. If it is not, click on the listing to select it.



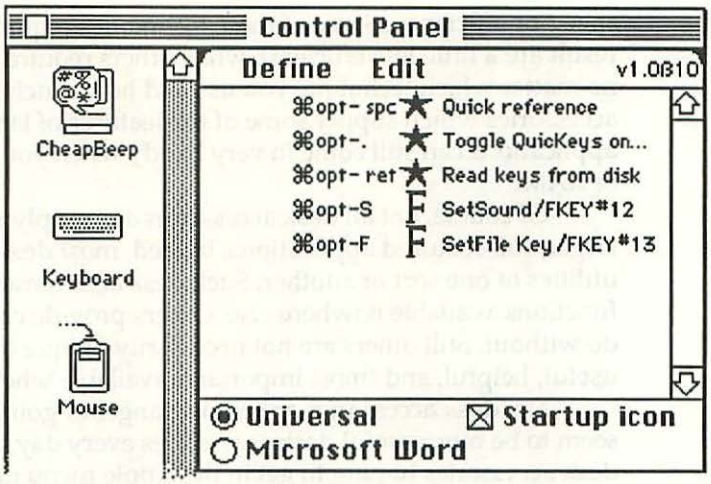
After selecting the FKey, a new entry will appear in the QuicKeys *shortcut list.*

b. Now press the key combination you want to use to call up the FKey. You can use any combination of Command, Shift, Option, and one other key. If you own a Mac SE or a Mac II, you can also use the Control key as a modifier, and, if you have the extended keyboard, the function keys can serve as substitutes for character keys. I recommend that you use Command-option key combinations, which is explained in Chapter 6.

   If you make a mistake or change your mind, you can keep pressing new key combinations until you get it right (and as long as the line remains selected). The last key combination you enter is listed on the left side of the shortcut description and is the one that *QuicKeys* remembers.

5. Close the Control Panel and test your new shortcut.

Continue in this way to assign new keyboard shortcuts to as many of your FKeys as you like. (The procedure for assigning keyboard shortcuts to desk accessories is much the same and is described in Chapter 5.) It's that easy. As long as *QuicKeys* is installed and the current System file is in control, the keyboard shortcuts you have assigned will invoke their corresponding FKeys.



The last step is to press the key combination you want to use to call up the FKey.

Desk Accessories: Why, How, and Which Ones

With MultiFinder, which makes it possible to keep more than one application open at a time, it would seem that the need for desk accessories is no longer as great as it once was. In some ways this is true. If you have RAM to burn, you could open *More* or *ThinkTank* (both by Living Videotext) instead of *Acta* (Symmetry Corporation) while you were working in some other program. You could launch *Chart* or *Excel* (both by Microsoft) just as easily as *MockChart* (MockPackage Plus, CE Software) or *MacPlan* (Borland). You could call up *MicroPhone* instead of a telecommunications desk accessory. You could launch your main word processor instead of the Notepad. Or you could use your graphics program rather than *Artisto* (public-exchange) or *DeskPaint* (Zedcor). You could do all these things—*if* you had enough RAM—and for many people, that's a big *if*. For most of us, RAM is a limiting factor, a brick wall we run up against frequently, especially when we try to pretend it's not there.

Whether the brick wall is up close or far away, we all run into it, and at such times, desk accessories that provide the basic features of a major application wrapped in a small package can come in very handy. In Chapter 6, we'll look at the many ways in which you can do what I described in the previous paragraph: open and close applications at will, without loss of

speed or efficiency. Some of these techniques require less RAM (and as a result are a little less efficient), while others require a great deal more. But no matter which technique you use and how much RAM you have, desk accessories which supply some of the features of larger, more sophisticated applications can still come in very handy when you are close to running out of RAM.

Of course, not all desk accessories are simply smaller versions of larger, full-featured applications. Indeed, most desk accessories are small utilities of one sort or another. Such desk accessories often supply essential functions available nowhere else. Others provide conveniences it is hard to do without. Still others are not necessarily unique or essential, but simply useful, helpful, and (most important) available when you need them.

No, desk accessories are not in danger of going extinct. Indeed, there seem to be more useful desk accessories every day, and my waiting list of desk accessories hoping to get in the Apple menu grows, rather than shrinks—despite MultiFinder. Thus the rest of this chapter is devoted to desk accessories. First, I look at the variety of possibilities in an attempt to help you choose the best desk accessories for your work. Then, I discuss a variety of desk accessory installation techniques, including several which sidestep the limit on the number of desk accessories imposed by the System file.

An Annotated Desk Accessory Catalog

In judging the value of a desk accessory, the main thing to consider is just how useful it is. This may sound obvious, but the concept can often be lost amid the hype of ads and the appeal of the cute or trendy. The most prominent commercial software companies do not necessarily sell the best desk accessories. In fact, no single company has produced a set of desk accessories I can't live without. Many of the most useful desk accessories aren't sold commercially at all but are shareware programs, or are simply free for the asking.

A quick perusal of the following list reveals that even if you are choosy about it, there are far too many desk accessories to fit in one System file. Actually, I've listed too many desk accessories on purpose, since no two people work in the same way, and my all-time favorites may not be yours. So, as you go through this list and read my brief comments about each desk accessory, think about the way you work and decide for yourself if the desk accessory makes sense for you. If it doesn't, cross it off the list and consider

yourself lucky to have reduced the size of your desk accessory waiting list. Nevertheless, between this list and the desk accessories you already use, you'll probably end up with a list too long to accommodate within the System file's self-imposed limit. Thus, in the sections following this one, I'll show you how to deal with the problem of too many desk accessories.

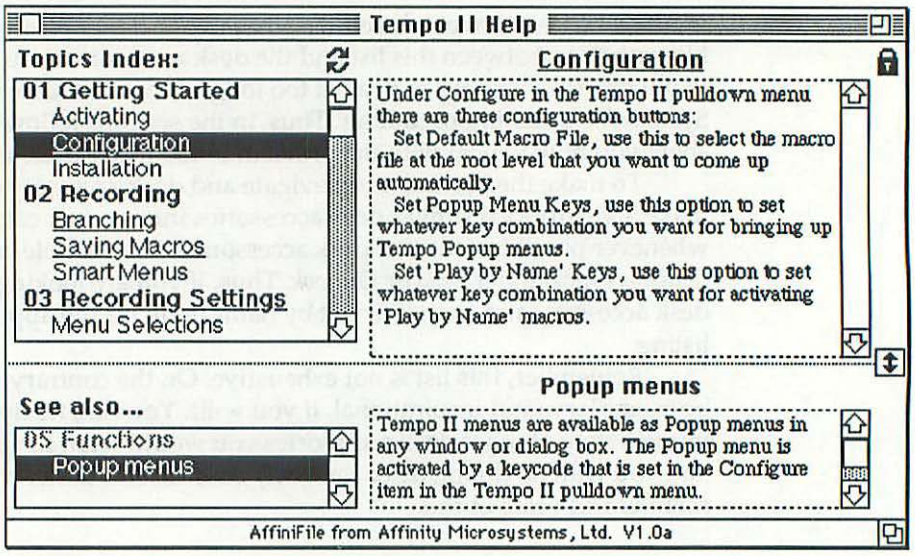
To make the list easier to navigate and desk accessory choices easier to make, I've grouped similar desk accessories into generic catalog entries whenever possible. Outliner desk accessories, for example, are listed under *Outliners*, not under *Acta* or *Outlook*. Thus, if you are looking for a particular desk accessory and cannot find it by name, look for the appropriate generic listing.

Remember, this list is not exhaustive. On the contrary, it is intended to be exemplary (and inspirational, if you will). You can, no doubt, think immediately of other desk accessories you would want in your System file, and you will, no doubt, discover many other useful desk accessories in the months and years ahead.

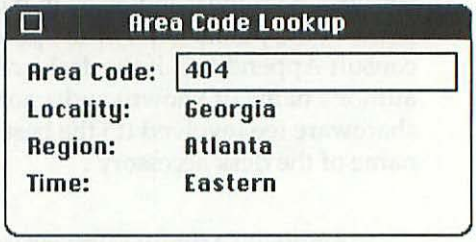
NOTE: Desk accessories marked with a § symbol are used in techniques and strategies described elsewhere in this book. You'll find more information about them in the discussions of these strategies later on. If the desk accessory is part of a commercial package sold under a different name, the name of the software package is indicated in parenthesis (along with the name of the publisher). For the publisher's name and address, consult Appendix C. If the desk accessory is public-exchange, the author's name (if known) and a note as to whether there is a shareware fee involved (to the best of my knowledge) follows the name of the desk accessory .

AffiniFile (Affinity MicroSystems) This desk accessory is described by its creators as a "dynamic filing system." That seems to mean that it can be used to store both text and graphics, that the different entries can be cross-referenced, and that a collection of entries (even one composed entirely of pictures) can be indexed for easy retrieval of all the entries.

Area Code Lookup (SideKick, Borland) This one does just what it says it does. Enter the area code and the desk accessory returns with the state, city, and time zone. Useful for people who call long distance a lot.



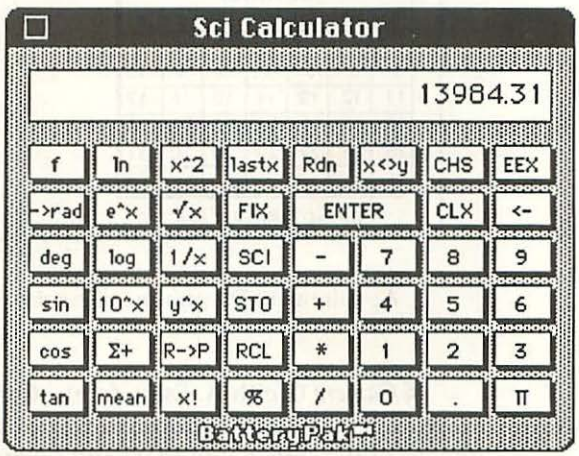
Entries in AffiniFile, *like the text entries in this* Tempo II *help file, can be indexed and cross-referenced for easy accessibility. Just click on a Topics Index or See Also... entry to view the new topic in one of the scrolling lists on the left.*



If you call out-of-the-way places, Area Code Lookup *can connect a geographic location with an area code.*

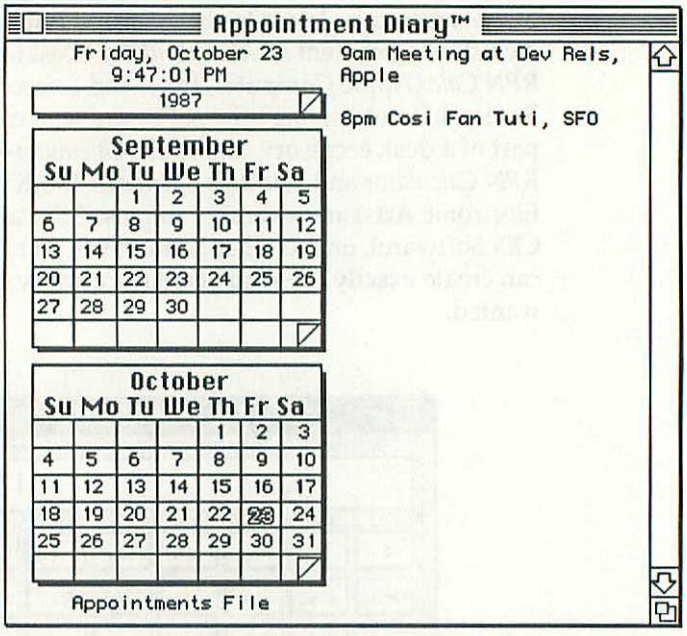
Calculators With my original Macintosh, I used to keep a real calculator on my desk because I disliked having to navigate the number keys at the top of the keyboard (or the mouse) to perform calculations with the calculator desk accessory. But now that all Macs have a key pad, real calculators are unnecessary (unless you have special needs that on-screen calculators cannot provide). There are lots

of on-screen calculators to choose from. Public-exchange calculators include the excellent *dCAD Calculator* (Desktop CAD, shareware fee), *RPN Calc* (Apple Computer, free), and *Science Calculator* (Rainmaker Research, free), among others. Commercial calculators are most often part of a desk accessory collection: *Calculator+* (SideKick, Borland) and *RPN Calculator* and *Scientific Calculator* (both in DiskTools Plus, Electronic Arts) are several examples. *Calculator Construction Set* (Dubl-Clik Software), on the other hand, gives you the tools with which you can create exactly the kind of desk accessory calculator you've always wanted.



DiskTools Plus' Scientific Calculator *does much more than add and subtract.*

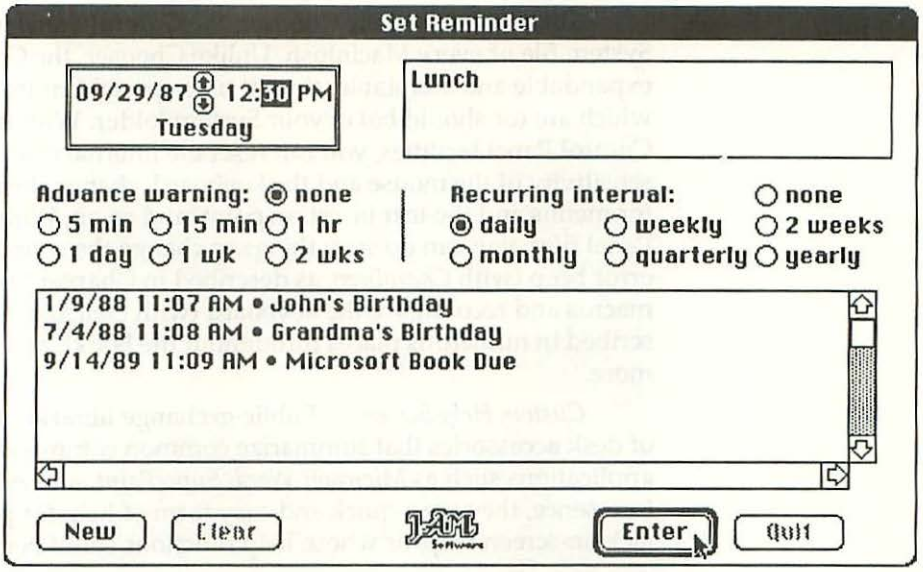
Calendars It's a great convenience having an on-screen calendar, in part because it is always at hand, and in part because you can look up any date of any year, not just the current year (or two), as is the case with print calendars. Most on-screen calendars also offer more space for notes, appointments, and the like. As with calculators, there are many on-screen calendars to choose from. Commercial offerings include *CalendarBook* and *Week-at-a-Peek* (SideKick, Borland), *Calendar* (DiskTools Plus, Electronic Arts), and *Appointment Diary* (Smart Alarms, Imagine Software). There are also a number of public-exchange calendars, one of the best of which is *Calendar* (Mike Schuster and David Oster, shareware fee).



Appointment Diary *is an endless electronic note calendar.*

§ *Camera* (Keith A. Esau, free) Camera is an alternative to the Apple screen shot FKey, Command-Shift-3. With it, you can convert images of the screen into MacPaint documents, even when a menu is open. It is an invaluable desk accessory for anyone who is documenting the way an application works (many of the illustrations in this book were taken with *Camera*, for example). It is also useful for making menu summaries (which are discussed in Chapter 5) and keyboard maps.

Clocks and Alarms Some people always like to have a clock nearby, and putting one in a window on your desktop is certainly one way of doing so. Simple clocks that only show the time or the time and the date are available from public-exchange sources. Apple's *Alarm Clock*, which is already installed in every straight-from-the-box System file, adds a simple, and single, alarm. *MacClock* (SideKick, Borland) is an analog clock (the kind with hands) for those who prefer the old way. *Smart Alarms* (Imagine Software) is the most sophisticated desk accessory of the lot. Indeed, to call it simply a clock or an alarm is



Smart Alarms *can remind you about anything, one-time occasions or recurring events, either at the correct time or in advance.*

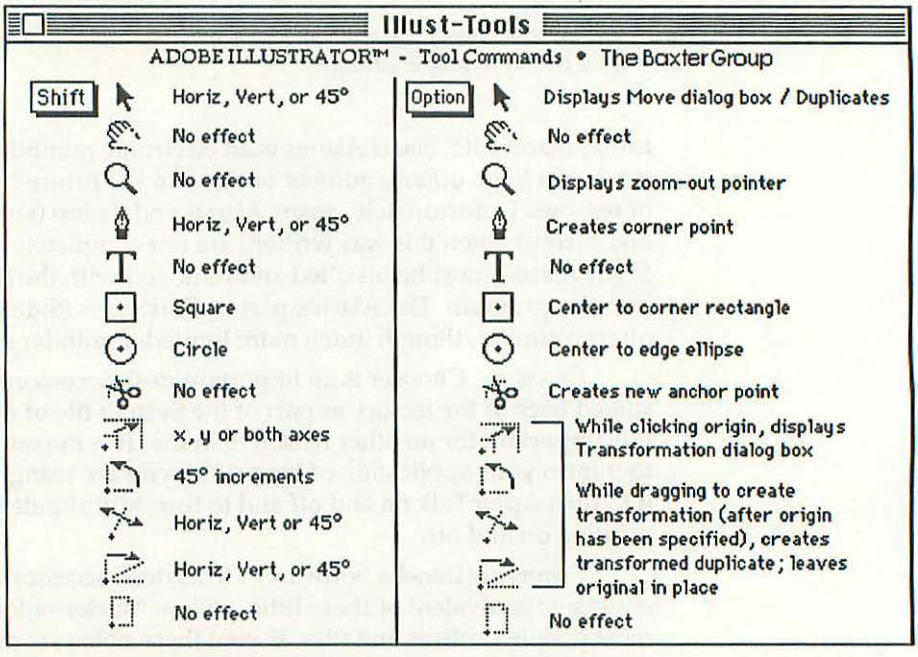
rather inaccurate. *Smart Alarms* is an electronic reminder system which can be set to go off any number of times in the future for any number of reasons. Unfortunately, *Smart Alarms* and *Tempo* (version 1.2, the one current when this was written) are not completely compatible. *Smart Alarms* must be installed and removed with the *Smart Alarms* installer program. The Alarms part of *QuikSheets* (SideKick, Borland) offers a similar, though much more limited, reminder system.

Chooser Chooser is an important desk accessory that is installed back at the factory as part of the System file of every Macintosh. It is important for no other reason than that it is the only way there is to inform your application of the printer you are using. You also need it to turn AppleTalk on and off and to turn MultiFinder's background printing on and off.

Comment (Deneba Software) This desk accessory provides the electronic equivalent of those little, yellow "sticky-notes" that festoon most people's offices and files. Except these notes are attached to your documents and pop up *in* the screen, not *on* the screen. If you are a red-pencil kind of person, *Comment* is just the thing for you.

Control Panel Like Chooser, the Control Panel is part of the System file of every Macintosh. Unlike Chooser, the Control Panel is expandable and adaptable, since it uses special Control Panel files which are (or should be) in your System folder. With the standard Control Panel facilities, you can reset the internal clock, change the sensitivity of the mouse and the keyboard, change the rate of flashing for menus and the text insertion point, and so on. With other Control Panel files, you can do such things as change the sound used for the error beep (with *CheapBeep*, as described in Chapter 5) and create macros and reconfigure the keyboard (with *QuicKeys*, which is described in numerous places throughout the book), and quite a bit more.

Custom Help Screens Public-exchange libraries offer a number of desk accessories that summarize common commands for popular applications such as *Microsoft Word*, *SuperPaint*, and *Adobe Illustrator*. In essence, they are a quick and easy form of help for programs which lack on-screen help, or whose help functions could benefit from some



Public-exchange help screens are available for a variety of programs. This one is part of a set by The Baxter Group for Adobe Illustrator.

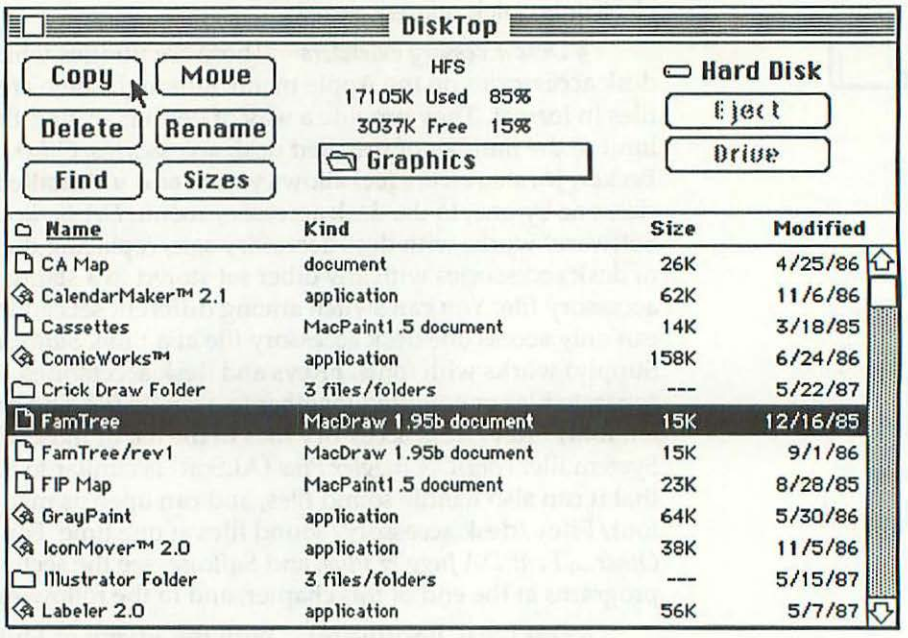
electronic quick reference cards.

§ *Desk accessory extenders* These are utilities which appear as desk accessories on the Apple menu, although some are actually Init files in format. They provide a way of getting around the System's limit to the number of installed desk accessories. *Other...* (Loftus E. Becker, Jr., shareware fee) allows you to add uninstalled desk accessories, one by one, to the desk accessory menu. *DA-Switcher* (Olduvai Software) works with desk accessory sets, replacing the System file set of desk accessories with any other set stored in a standard desk accessory file. You can switch among different sets ad infinitum, but can only access one desk accessory file at a time. *Suitcase* (Software Supply) works with fonts, FKeys and desk accessories, either in separate files or collected together in a single file, and can add up to ten font/FKey/desk accessory files to the list of those already in the System file. *Font/DA Juggler Plus* (ALSoft) is similar to *Suitcase*, except that it can also handle sound files, and can open as many as twelve font/FKey/desk accessory/sound files at one time. For more on *Other...*, *Font/DA Juggler Plus*, and *Suitcase*, see the sections on these programs at the end of this chapter, and in the following chapter.

§ *DiskTop* (CE Software) With the advent of MultiFinder, *DiskTop* may not be the sine qua non it once was. But if you work with MultiFinder turned off (with *Switcher*, for example, or with a Finder substitute such as *Oasis* or *SkipFinder*, all of which are discussed in Chapter 6), then *DiskTop* is just about essential. *DiskTop* provides nearly all of the Finder's functions, no matter where you are or what you are doing. With it you can delete, rename, copy, and move files, create new folders, find lost files, and get information about any file. Like a Finder substitute such as *SkipFinder*, you can use it to go directly to a different program (or program and document). If you are so inclined, you can change many "Finder attributes," such as file creator and type.

Expressionist (Allan Bonadio Associates) If your word processor lacks the ability to do mathematical typesetting (and most do), then the desk accessory form of *Expressionist*, a sophisticated equation editor, is an excellent way to make up for the deficiency.

File Utilities *DiskTop* may be the king of the so-called file utilities, but it certainly is not the only one. *DiskTools II* (DiskTools Plus, Electronic Arts) also provides many of the same functions (copying, renaming, deleting, finding and moving files, creating new folders, launching applications, and so on) and some prefer it to *DiskTop*. I used *DiskInfo* (David Dunham, shareware fee) all the time

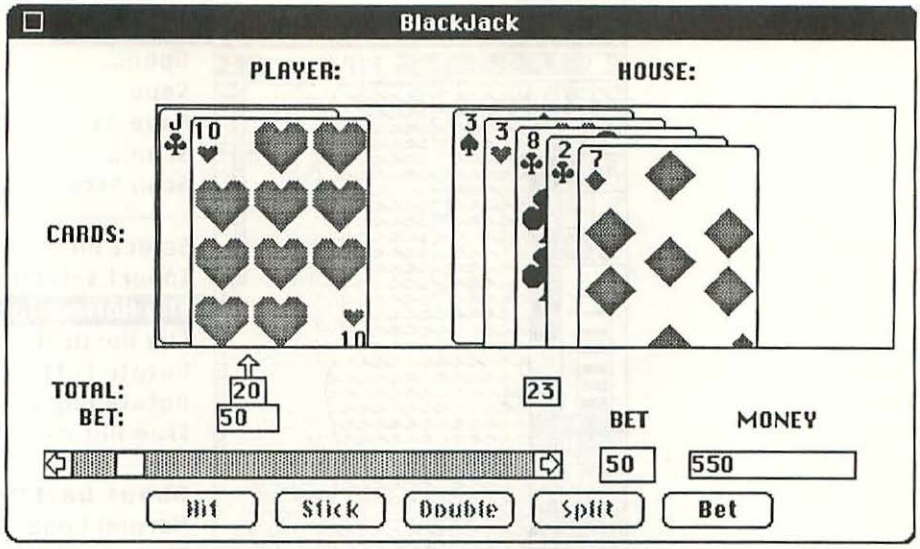


DiskTop *offers all the functions of the Finder, and more, from within any application.*

until *DiskTop* came along, and it is still a very worthwhile program. Although *HFS Locator Plus* (PBI Software) and Apple's *Find File* are more limited in scope, they work just fine for locating lost files.

Games Admittedly, games are not a high priority on many people's desk accessory lists. But if you sometimes hanker for a two-minute break and have long since tired of the old *Puzzle*, then one of the many public-exchange desk accessory games may be just the ticket. Among them are: *3D Tic-Tac-Toe* (Steve Christensen, free), *Bagels* (Steve Christensen, free), a version of the game Mastermind, *Rubik's Cube* (free), *Maxwell's Demon* (free), *BlackJack DA* (Brad Slutsky, shareware fee), and many others.

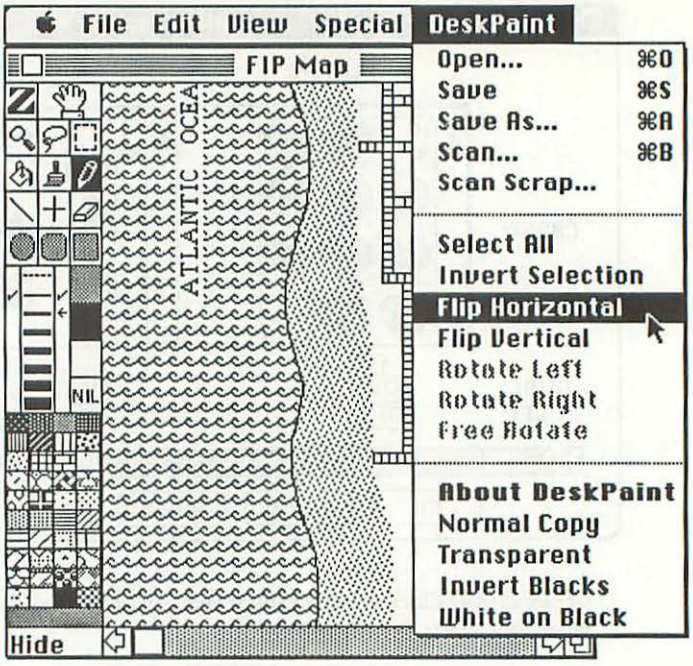
Graphics creators *HyperCard* has a wonderful built-in graphics capability (which is, I suppose, just what you'd expect from Bill Atkinson). But not all applications are similarly blessed. If you have sufficient memory and are running *PageMaker* under MultiFinder or *Switcher*, for example, you can switch to *SuperPaint* to whip up a special graphic without even thinking about *PageMaker*'s deficiencies in this regard. But if your Mac is a little short of memory or if you have



Games such as Blackjack *can enliven your Apple menu—and distract you from more productive efforts.*

already filled your otherwise more than adequate memory with other applications, you might find it convenient to use a desk accessory to create that graphic. One such desk accessory is *CheapPaint* (part of M.U.D., Spinnaker Software). It's simple, but it gets the job done. *DeskPaint* (Zedcor), on the other hand, offers quite a bit more, including free rotation, scaling, and flipping, as well as the usual Paint tools, in a desk accessory that's hardly more than 12K. *Canvas DA* (Deneba Software) is even more sophisticated, since it combines aspects of both Draw and Paint programs, but it requires quite a bit more disk space.

Graphics retrievers If you don't want to create new art (with the previous category of desk accessories) but do want some flexibility in locating, browsing, and pasting ready-made art , then you might want to look into one of the many graphics retrievers. *Art Grabber+* (part of M.U.D., Spinnaker Software; also packaged with *VideoWorks II* by MacroMind) and *Artisto* (Tom Taylor, shareware fee) are two such utilities. They both allow you to open a Paint document, browse through it, and select, with a marquee box, the image or part of an image you want to paste into your application. *QuickPaint* (EnterSet) is similar, except that it also has an eraser, with which you can erase (temporarily—your deletions are not preserved when you close the



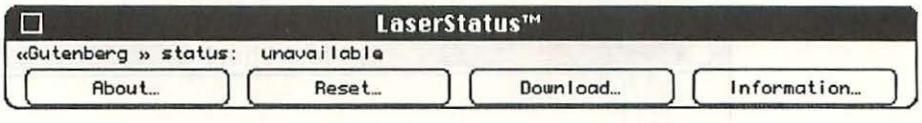
DeskPaint *is an amazingly powerful paint program in a desk accessory that takes very little disk space.*

document) any part of an image or surrounding images before you select what interests you. *PictureBase Retriever* (Symmetry) is the desk accessory part of the *PictureBase* graphics management system. To use it, your graphics must already be stored in one or more *PictureBase* libraries. With *PictureBase Retriever*, you can view the contents of a library file, search for a particular image, and copy all or a selected part of the image at standard or reduced size to the clipboard, and from there to your application. If you want to make changes to the library file itself, you must use the *PictureBase* application. *Graphidex* (BrainPower) is a graphics indexer similar to *PictureBase*. With this desk accessory, you can search for, retrieve, and perform limited graphics editing on images stored in standard Paint or Draw documents.

§ *Key Caps* Another desk accessory that comes already installed in every System file, *Key Caps* is indispensable. Without it, I

could never relocate such oddball characters as bullets, diamonds, or paragraph, trademark, and copyright symbols. For fonts with a large number of easily misplaced characters, *Key Caps* comes in very handy for making a set of character/keyboard maps. Unfortunately, when you use Command-Shift-3 to take a screen shot, the Command and Shift keys are intercepted by *Key Caps*, and as a result, a simple screen shot invariably produces a not very useful map of the Command-Shift keyboard layout. With *Camera* (as described in the menu map discussion in Chapter 5), however, you can easily obtain screen shots of the four basic keyboard layouts: no Shift or Option, Shift, Option, and Shift-Option.

LaserStatus (DiskTop, CE Software) Anyone using a LaserWriter will find *LaserStatus* very handy. With it, you can reset the LaserWriter without having to turn it off and on again, you can manually transmit downloadable fonts and PostScript files to the LaserWriter, you can monitor its status (useful if you share your LaserWriter with others on a network), and you can find out what fonts, including downloaded fonts, are currently in the LaserWriter's memory, and how much room is left for other downloadable fonts. *LaserStatus* is part of the *DiskTop* package.



LaserStatus *gives you more control over the LaserWriter.*

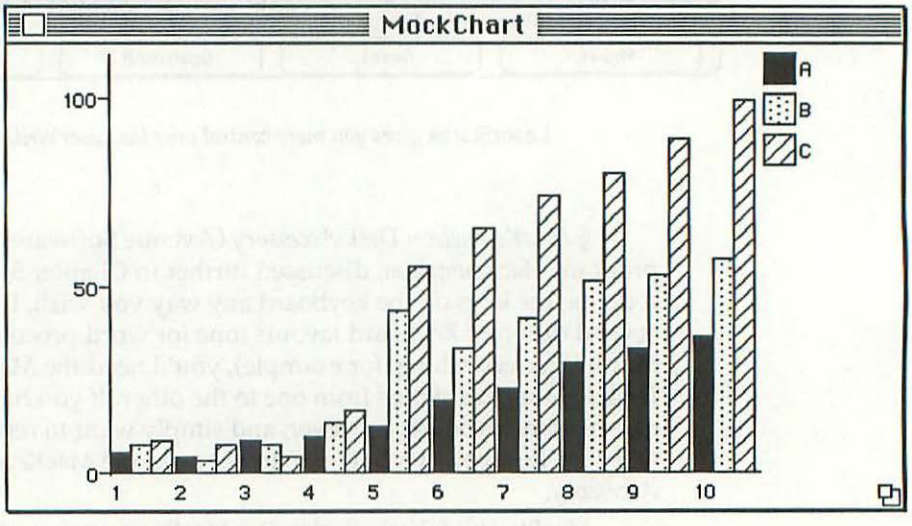
§ *MacKeymeleon Desk Accessory* (Avenue Software) The program *MacKeymeleon*, discussed further in Chapter 5, allows you to redefine the keys on the keyboard any way you wish. If you have several different keyboard layouts (one for word processing and one for number-crunching, for example), you'll need the *MacKeymeleon Desk Accessory* to switch from one to the other. If you have created only one custom keyboard, however, and simply want to replace the standard keyboard with it, you do not need the *MacKeymeleon Desk Accessory*.

MacPlan (SideKick, Borland) *MacPlan* is a spreadsheet in a desk accessory. Just as *DeskPaint* (described above) is not as powerful as

SuperPaint, *MacPlan* is not quite the same as *Excel*. But for a quick forecast on the side (especially when you are running short of memory), *MacPlan* is a useful solution. *MacPlan* can also chart the figures on your spreadsheet in several different ways.

§ *MenuKey* (TopDesk, Cortland Computer) *MenuKey* allows you to assign keyboard shortcuts to menu commands that don't already have shortcuts or change those that do. You can use the same repertoire of Command-key combinations used by most applications (in which the Shift key is ignored, so that Command-key and Command-Shift-key are interpreted as the same command), as well as Command-Option-key combinations. Your changes can be permanent or temporary (current session only), and they are shown on the menu just like any other keyboard shortcut. *MenuKey* must be installed and removed with the *MenuKey* installer program. (See Chapter 5 for more about *MenuKey*.)

MockChart (MockPackage Plus, CE Software) *MockChart* is a simple charting program in desk accessory form. It will create line, column, stacked column, pie, and hi-lo-close charts. You can import data or enter it directly into the rows and columns with which *MockChart* opens. When you have the chart you want, you can copy it to either Paint or Draw programs for fine tuning (or fudging, I sup-

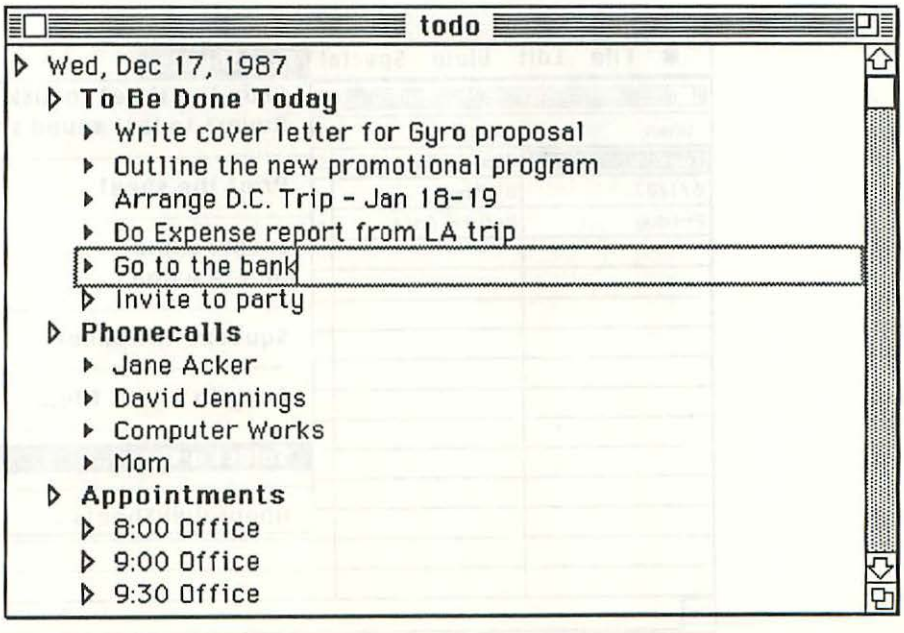


MockChart *can graph the decline and fall or a meteoric rise.*

pose). Obviously, *MockChart* will never replace *Microsoft Chart*, but for quickies or when memory is in short supply, it does just fine.

Notepads A notepad is a place to jot down notes and a method for storing them as well. The classic one is, of course, Apple's Notepad, which is part of every new System file. But as with the Scrapbook, there are alternatives that can do much more than the original. Among them is *Phone Pad* (DiskTools Plus, Electronic Arts), a combination notepad, phone book, and phone dialer all rolled into one. It contains 250 pages, which can be subdivided into sections for different purposes: phone numbers, phone notes, business notes, personal notes, and so on. *Phone Pad* has a Find function and a **Go To Page...** command, which help you navigate all those pages, and it can dial a selected phone number with any Hayes-compatible modem.

Outliners I like outliners, and I especially like desk accessory outliners. While it is true that application outliners have many more features (or at least some of them do), it's the easy accessibility of desk accessory outliners that I like. They're always there when I need them, even when very little free memory remains. There are currently two

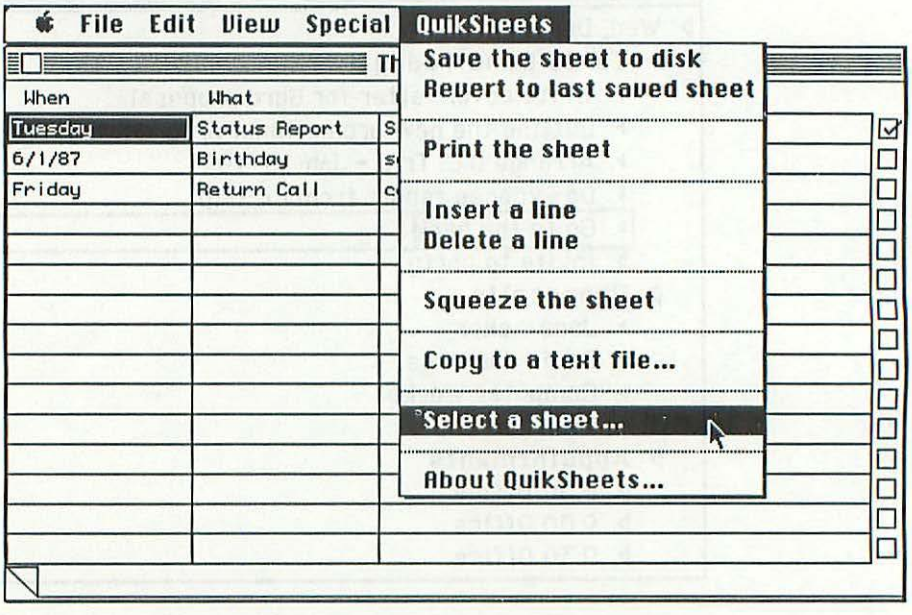


With a desk accessory such as Acta, *your outliner is available, no matter what you are doing.*

major players in the desk accessory outliner sweepstakes. *Acta* (Symmetry) is simple, straightforward, and allows you to put graphics or lots of text in a headline. *Outlook* (SideKick, Borland) uses separate windows for overflow text and graphics, and for some inexplicable reason, it takes up a whopping 113K of disk space.

§ *PowerDisk* (Software Power Company) *PowerDisk* is a RAM cache program. With a RAM cache, you can speed up your Mac's performance by minimizing the frequency with which it has to read information from a floppy or hard disk. *PowerDisk 2.0*, which should be available by the time you read this, comes in both Control Panel (cdev) and desk accessory format. For more information on RAM caches in general and *PowerDisk* in particular, see Chapter 7.

QuikSheets (SideKick, Borland) *QuikSheets* is a desk accessory database with diverse, though limited, capabilities. It comes with four preconfigured sheets (or forms), but you can create your own sheets with the accompanying *QuikEditor*, and have as many as twenty-one available at one time. The preconfigured sheets are an Alarms sheet, which can store and set off up to fifteen reminder alarms, a Things-To-Do checklist, and Credit Card and Expenses record sheets.



QuikSheets *is a multifaceted mini-database.*

Scrapbooks Scrapbooks are places to store multiple images or text; they are especially handy when the Clipboard's limit of one image or one piece of text is too confining. Apple's Scrapbook hasn't changed much over the years, and though it works well enough, there are now a number of more versatile alternatives. *Multi-Scrap* (Bob Luce, free) adds a new dimension to the original scrapbook format by permitting any number of scrapbook files, so you can organize your material by topic. *SmartScrap* (Solutions) and *New Scrapbook* (Tim Wasko, shareware fee) also allow you to create multiple scrapbook files. In addition, you can use them to retrieve selected lines of text or portions of images, and you can scroll to see text that is too long to fit in the window. *SmartScrap* also has a "table of contents" feature, which displays a scrollable list of thumbnail versions of images and text (although the text is essentially unreadable). Just click on a thumbnail to go directly to the desired image. *SmartScrap* also comes with another desk accessory called *The Clipper*, which allows you to scale or trim the image in the Clipboard before pasting it into the receiving application.

Screen Savers If you leave your Mac on all day long (as I do) and have a tendency to forget to turn the screen brightness down when you are away from your desk (which I also do), you run the risk of "burning in" an image on the screen. From that point on, the burned-in image will appear on the screen as a faint ghost. To avoid having a haunted Mac, you should use some sort of screen saver. Some, such as *AutoBlack* (Itty Bitty Computers, shareware fee) and *Pyro!* (Software Supply), both of which are described in Chapter 5, are Init files which install themselves at startup. Other screen savers are FKeys, and a few are desk accessories. I prefer the Init versions, but you may want to use a desk accessory. *Blank* (TopDesk, Cortland Computer) is one such desk accessory. *Blank* allows you to set the time interval of inactivity (in minutes and seconds) after which it darkens the screen. You can also set *Blank* to install itself automatically at startup. *Blank* must be installed and removed with the *Blank* installer program. *BCS*Sleep* (Ken Winograd, free), *Idle* (Larry Rosenstein, free), and *Sleep* (free) are public-exchange screen-saver desk accessories. All three require that you select them to turn off the screen; none are automatic like *AutoBlack* or *Blank*.

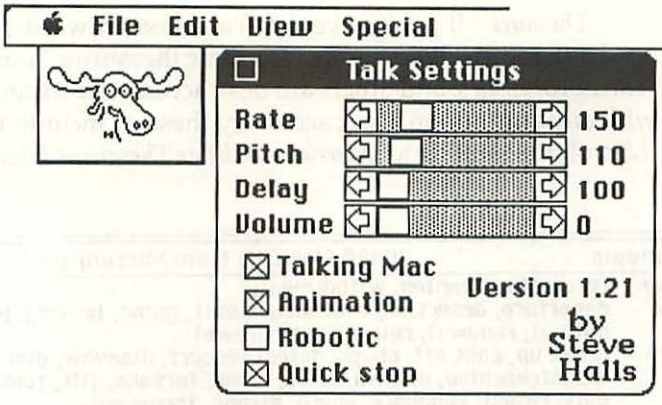
§ *SkipFinder* (Darin Adler, shareware fee) If you are not using MultiFinder, *SkipFinder* provides a way of jumping from one application to another without an intermediate stop at the Finder. Select *SkipFinder* once, and then every time you quit your application, you will be presented with *SkipFinder*'s file selection box instead of the

Finder. Select an application (and a document as well, if you like) and *SkipFinder* will launch it immediately. For more on program integration, on alternative methods for switching between applications, and on *SkipFinder* in particular, see Chapter 6.

Spelling Checkers Spelling checkers come in various forms: as stand-alone applications, as desk accessories, and integrated into high-powered word processors such as *Word*, *FullWrite* (Ann Arbor Softworks), and *WordPerfect* (WordPerfect Corp.). If your word processor (or any other word-based application) lacks a built-in spelling checker, and a separate checker application does not appeal to you, then you may want to add a spelling checker to your desk accessory menu. *Lookup* (Working Software) checks the spelling of individual words, and it operates only on request. (You have to use *Lookup*'s sister application, *Spellswell*, to check whole documents.) *Coach* (Deneba Software) will check a word or a selected portion of text, or it will check your spelling interactively (as you type). *Coach Professional* (Deneba Software), on the other hand, is more like a real dictionary; it provides the correct spelling and hyphenation, and defines the word in question. *Thunder!* (Electronic Arts) is another popular interactive desk accessory spelling checker; you can also use it to automatically expand shorthand abbreviations as you type.

Talking Moose (Steve Halls, shareware fee) This is clearly not a high-priority desk accessory, but we all have to have fun sometimes, right? *Talking Moose* installs a small animated moose wearing a bow tie (no fooling) in the corner of your screen, and this moose has some nerve. He pops up from time to time as you work, and he spouts off rude and impertinent comments about, among other things, your inability to get the job done any faster. *Talking Moose* requires a special phrase file and MacinTalk (the Apple System file for speech synthesis, available from most public-exchange sources), as well as the desk accessory itself. With *Moose Phrazer* (Jan Eugenides, free), you can edit and add phrases to the moose's collection of expressions.

Telecommunications Desk Accessories This category is really a potpourri. Some telecommunications desk accessories, such as *MockTerminal* (MockPackage Plus) and *MacTerm* (SideKick), are like scaled-down communications programs. For a quick check of your electronic mail when MultiFinder is turned off or memory is tight, these desk accessories can come in handy. *MacDialer* (SideKick, Borland) is an electronic telephone book/dialer that you can use to look up and dial any phone number in your phone list, no matter what you are doing at the moment. *MacDialer* uses SideKick's *PhoneBook* format files for

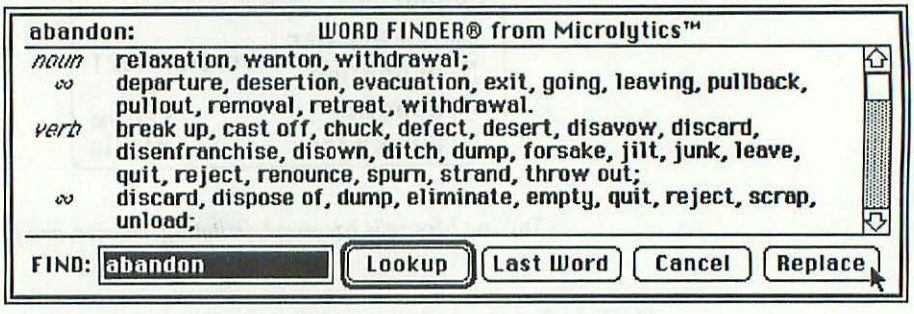


Talking Moose *is irreverent, irritating, abusive, and fun.*

storing phone numbers, and it also offers a phone log in which you can make notes on the calls you have made and store billing information. *BackDown* (Ron Risley, shareware fee) makes it possible to download files in the background (at speeds up to 9600 baud), even if you are not using MultiFinder. *QDial* (Leo Laporte, free) dials and redials your modem until it gets an answer from one of the five numbers you ask it to call. It is useful for reaching busy editors—oops—I mean bulletinboards.

Text Editors In the same way that I like to have an outliner readily available, other people like to have some sort of text editor (or small-scale word processor) close at hand so that they can, among other things, record those flashes of inspiration that hit at odd and usually inconvenient moments. Text editors differ from notepads in that you are not confined to the pages of the notepad. Text editors can open any text-only files from any disk, and they can save your writings as text-only files as well. There are many text-editor desk accessories to choose from. *MockWrite* (MockPackage Plus, CE Software) is the classic of its genre; it supports user-defined tab spacing, provides a method for easily combining several files into one, and contains a convenient Find function. *NotePad+* (SideKick, Borland) is pretty much the same program under a different name. *miniWRITER* (David Dunham, shareware fee) has many of the features that *MockWrite* left out: an Undo command; a choice of fonts, font sizes, styles, and line spacing; word and line counts; high-quality printing on the ImageWriter; and more.

Thesauri If you've ever been at a loss for words, then you may need a thesaurus. Perhaps an electronic thesaurus. Some are built into word processors, and others are desk accessories meant to be used with any application. Desk accessory thesauri include *Word Finder* (Microlytics) and *Coach Merriam-Webster Thesaurus* (Deneba Software).



Electronic thesauri such as Word Finder *can help if you're at a loss for words.*

Touch•n•Go (TopDesk, Cortland Computer) *Touch•n•Go* is a macro processor nowhere near as powerful as *Tempo II*. On the other hand, as part of the TopDesk set of desk accessories, it is nowhere near as expensive either. *Touch•n•Go* can replay pieces of text (in response to a keyboard command or by automatically replacing nonsense words such as abb), simple mouse movements (the screen locations of mouse-down and mouse-up), and a limited number of window actions (scroll and page in four directions, zoom, and close). To perform complex tasks, you must create a macro which combines a number of other, previously assigned, macros. *Touch•n•Go* must be installed or removed with the *Touch•n•Go* installer application.

Windows (DiskTools Plus, Electronic Arts) For programs that lack a windows menu (and that includes the Finder), the *Windows* desk accessory is a very handy addition. When invoked, a new menu item called Windows appears on the menu bar; in it are listed all the currently open windows. To bring any window to the front, you simply select it from the menu. This is a real time-saver if you have a large number of windows open at the same time. The names in the Windows menu are ordered according to their position on the desk-top; each time you bring a new window to the front, the names on the

menu are rearranged to reflect the new order. Of course, as new windows are opened and others closed, the menu is changed as well. A similar desk accessory, *Window DA* (Marsh Gosnell, shareware fee) and several FKeys which perform related functions are available from public-exchange software sources.

Now, before proceeding to the next and final section of this chapter, check through these listings and make a list of the desk accessories that interest you. They come from a wide variety of sources, so you'll have to do a bit of searching to find all the ones you want. When you have collected all the desk accessories on your list or as many as you can easily locate, then it's time to consider how to fit them all under the Apple menu, which is the topic of the next section.

Putting Your Desk Accessories under the Apple Menu

If you are one of those rare people who needs no more than a dozen or so desk accessories, then what follows is really not for you. As long as the number of desk accessories in your collection is less than fourteen or fifteen, then you can easily install them all in your master System file using Font/DA Mover and the methods outlined in the Macintosh manual.

But if you, like most Macintosh owners, have a list of desk accessories which is as long as your arm, or at least more than fifteen, then here is some help. In this section, we'll look at several techniques you can use to get around the System file's limit on desk accessories. One method entails a certain sleight-of-hand: installing desk accessories in applications. Another method uses font/desk accessory extenders, ranging from the low-key (*Other...*) to the sophisticated (*Suitcase*).

There is yet another method of increasing the number of available desk accessories which uses *ResEdit*, Apple's resource editor and programmer's tool. Although *ResEdit* is used elsewhere in this book for a variety of simple alterations, working over the System file with *ResEdit* so that it can accommodate a greater number of desk accessories is a complex task, not to be lightly undertaken. Thus I do not cover this method here, nor do I recommend it, lest you dig yourself a hole deeper than you can climb out of. If you are curious enough (and adventurous enough), and don't find the methods outlined here quite what you are looking for, check with other members of your local users group for more information on this method of stretching the desk accessory limit.

The Desk Accessories in Your System File Before we look at the desk accessories you don't have room for in your System file, however, consider briefly the fourteen or fifteen that you *can* install in your System file. Assuming you have more than fifteen you want to use, and assuming that you will install some of them (as many as you can) in the System file before resorting to other methods, then the question is which ones should you install in the System file?

The answer to that question is really a multipart answer, and it is an answer which depends to a great degree on the particular desk accessories you have in your collection. In other words, there is no simple answer. But there are a few guidelines you can follow:

- Almost any desk accessory which must be installed with its own installer program instead of Font/DA Mover should be installed in the System file. You may not, in fact, have any choice in the matter. I might note here that desk accessory installer programs are, for the most part, nothing more than a poorly disguised form of copy protection (see Chapter 8 for more on this topic). Like all copy-protection schemes, they can cause headaches and even more serious problems. Thus, to minimize the grief, here are a few simple rules you should follow with desk accessories that must be installed with their own installer program:

  - ◊ If at all possible, avoid using desk accessories which must be installed and removed with their own installer programs. It is too easy to forget which ones they are, and the results of an inadvertent attempt to manipulate the desk accessory with Font/DA Mover can be most unpleasant. In some instances, the best desk accessory for the job at hand is one with an installer program, in which case there's not much to do but put up with the inconvenience. Otherwise, I suggest you stick with standard desk accessories.
  - ◊ If you must use such a desk accessory, never attempt to copy it to another file with Font/DA Mover. Font/DA Mover may bomb in the attempt, or you may be sorry at some unexpected point later on. Either way, the end result is highly undesirable. (A notable exception here is most of the desk accessories in the *TopDesk* set of desk accessories from Cortland Computer. Although these desk accessories must first be installed with their individual installer programs, they can be manipulated with Font/DA Mover any time after the initial installation.)

◊ Never use Font/DA Mover to remove the desk accessory (unless the installer program is too dumb to know how to remove what it installed). Some of these desk accessories are accompanied by an entourage of resources (small program segments) which the installer program copies to your System file along with the desk accessory. If you use Font/DA Mover to remove the desk accessory, it may not remove all those other resources. The installer program, however, should know what needs to be removed, and is much less likely to leave some stray code behind.

- Desk accessories that require support files (usually data files of some sort) should be installed in the System file, and their support files placed in the System folder, whenever possible. This suggestion applies even to desk accessories which allow you to locate the file in question manually (as most spelling checkers do) or to those that search all of your folders automatically (as does *Appointment Diary*), for it makes finding the appropriate files faster and more convenient. But it is particularly important for desk accessories (such as those in *SideKick*) that need certain files and expect them to be in a certain place, usually the System folder. If such desk accessories are not installed in the System file but placed under the Apple menu using a desk accessory extender, for example, they may take some time finding their support files or fail to find them at all.
- Desk accessories that turn on background programs are best installed in the System file. These include RAM cache programs such as *PowerDisk* and automated screen-savers. Although these programs may work correctly if added to the Apple menu with a desk accessory extender, it's also possible that they may not. And a background program that fails, unbeknownst to you, is a dangerous thing. So the wisest move is to install desk accessories such as these in the System file so they are always available (and so you can keep your eye on them).
- Essential Apple desk accessories should be installed in the System file. Apple's Chooser and Control Panel are best placed where they were intended to be. I also keep Key Caps there, not because it is important to the operation of the Mac, but because I use it often to locate lost characters.
- If you are using *Other...* to extend your selection of desk accessories, you should be sure to install it in the System file. (*Other...* is described in more detail later in this chapter.)

- If there's any room left over, the desk accessories you use most often should be installed in the System file. This is an obvious suggestion, but one often forgotten. Unless you have no other choice, don't install desk accessories that you use in only one or a few applications in the System file. Save System file desk accessory slots for desk accessories that you use in a variety of different situations.

With these thoughts in mind, use Font/DA Mover to install the appropriate desk accessories in your master System file. Then, when you've installed all the desk accessories that you can in the System file, it's time to consider what to do with all the ones that are left over.

Installing Desk Accessories in Applications Many people are unaware of the fact that fonts and desk accessories can be installed in many kinds of files, not just the System file, using Font/DA Mover. As we will see in the next chapter, fonts can be installed in applications or documents. Similarly, desk accessories can be installed in applications. You may, in fact, have noticed that one or more of your applications already have "built-in" desk accessories. For example, *MicroPhone*'s MicroEditor, which appears under the Apple menu whenever you use *MicroPhone*, is simply a desk accessory installed in *MicroPhone* instead of your System file.

The advantages of installing desk accessories in applications are several:

- You can increase the number of desk accessories under the Apple menu by installing desk accessories in applications. Even if your System file is completely full of desk accessories, the Apple menu can accommodate any desk accessories it finds stored within the current application.
- By installing desk accessories that you use in only one or two applications, in those applications, the desk accessories become available only when the application is running. When you are not using the application in question, the desk accessories are gone. For example, if you install the *Adobe Illustrator* Help desk accessory (illustrated earlier) in *Illustrator*, then only when *Illustrator* is running will the desk accessory be listed under the Apple menu. In this way, specialty desk accessories will be there when you need them and not there (and thus not cluttering up the Apple menu) when you don't need them.

Installing desk accessories in an application is no more difficult than, and not much different from, installing them in a System file. The trick is the Option key. Here's what to do:

1. Launch Font/DA Mover and open the file containing the first desk accessory you want to install. Since you are working with desk accessories, there are two methods you can use to launch Font/DA Mover more efficiently:
   a. As you Launch Font/DA Mover, hold down the Option key so that Font/DA Mover opens to the desk accessory display instead of the font display. Then close the System file and open the desired desk accessory.
   b. Double-click on the Font/DA Mover file containing the desk accessory you want to install. This will launch Font/DA Mover, select the desk accessory display, and open the desired desk accessory file all at once.
2. Now open a copy of the application in which you want to install the desk accessory.
   a. To get applications to appear in the file selection list, hold down the Option key when you click on the Open button.
   b. Select an expendable copy of the desired application and double-click on it to open it.
3. Copy the desk accessory to the application just as you would to the System file or a desk accessory file. If you want to install several desk accessories in the same application, close the one you just installed, open another one and then copy it to the application.
4. Quit Font/DA Mover. When you next open this copy of your application, you will find the desk accessory (or accessories) you just installed listed in the Apple menu along with your other desk accessories, and in correct alphabetical order. The desk accessory (or accessories) will only be available when that application is running (unless you install it in other applications as well).

> *Remember: Install a desk accessory only in a copy of your application, never in the original.*

It is important that you make careful and considered decisions about what desk accessories to install in which applications. As a general rule, you

should install in applications only those desk accessories which you use with only a single application or a few applications at the most. Also, be sure that you do *not* install the same desk accessory in an application (or two) and the master System file. Having two copies of the same desk accessory available at the same time, one belonging to an application and one belonging to the System file, is not a good idea.

Installing desk accessories in applications has one significant hazard of which you should be aware, however. The problem centers around desk accessory ID numbers. Each desk accessory in the Apple menu must have a unique ID number, and to make sure that that is the case, Font/DA Mover reassigns ID numbers whenever there is a potential conflict. For example, when a desk accessory is installed in the System file, Font/DA Mover first takes stock of all the other desk accessories in the System file and compares their ID numbers to the ID number of the desk accessory it is preparing to install. If the new desk accessory's ID number is the same as any of the ones already installed, then Font/DA Mover gives the new desk accessory a new and unique ID number. Similarly, when Font/DA Mover installs a desk accessory in an application, it takes into account the ID numbers of the other desk accessories already installed in the application.

Font/DA Mover does not, however, check System file desk accessories when installing a desk accessory in an application. As a result, it is possible to have two different desk accessories (call them A and B) under the Apple menu, one desk accessory from the System file and one from an application, both of which happen to have, purely by chance, the same ID number. Although this situation sounds pretty serious, the problem is one of those "sleeping giant" sort of things. As long as you tiptoe, as long as you open only one desk accessory at a time, you may not experience any problems. Indeed, you may get away with any number of desk accessories open at the same time, as long as they all have different ID numbers. It's when you open desk accessory B when desk accessory A is already open that the code (often) hits the fan.

What to do? The only easy solution (other than not installing desk accessories in applications) is really just a computerish sleight-of-hand that tricks Font/DA Mover into giving all your desk accessories unique IDs. Although easy, it is not perfect, since there are occasions when it may not work (more on this in a moment), and since it can be used on no more than 52 desk accessories (the limit on the number of desk accessories you can put in a single file), but it's worth a try:

1. First, you need to create, with Font/DA Mover, one jumbo desk accessory file containing all the desk accessories you expect to use,

no matter where you intend to install them. In doing so, Font/DA Mover will renumber desk accessories as needed so that all of them have unique IDs. The following steps explain how to create such a jumbo desk accessory file:

a. Launch Font/DA Mover. When it opens, click on the Close button under the list box displaying the contents of the System file.

b. Click on either one of the Open buttons. In the resulting file selection box, click on the New button.

c. Give the new desk accessory file a name, such as *Jumbo DA File*, select a disk with enough space to store all your desk accessories, and click on the Save button.

d. Now open your individual desk accessory files, one by one, and copy each desk accessory to the jumbo file.

e. When all your desk accessories have been copied to the jumbo file, all of them should have unique ID numbers.

2. Next install your desk accessories using Font/DA Mover to copy the desk accessories from the jumbo file to the appropriate destination (the System file or an application). Be sure you copy the desk accessories from the jumbo file and not some other file. They are the ones that have unique IDs.

   Also, be sure that the destination file has been emptied of desk accessories *before* you install any from the jumbo file.

   As long as Font/DA Mover does not change any ID numbers again, all your desk accessories, no matter where they are installed, will have unique IDs. Unfortunately, there is no way to be absolutely certain that Font/DA Mover does not change the ID when it copies a desk accessory. That's why it is important to follow the instructions in paragraph 3.

3. Finally, and you've heard this before and will no doubt hear it again: test and test again before using the new setup with anything important. In this case, open the desk accessory (or accessories) in your application, then open the desk accessories in your System file, one by one. Use them, play with them, do whatever you can to trip them up. If all goes well, then you can be reasonably certain that it will continue to do so.

This technique works well for standard desk accessories that are installed with Font/DA Mover. It does not solve all your problems if you use desk accessories that must be installed in the System file with their own installer program. If you use such desk accessories, you have two choices. One is strenuous testing to make sure that there are no ID number conflicts. The other is to use a utility such as *Suitcase* or *Font/DA Juggler Plus* (both of which are described later in this chapter), instead of installing desk accessories in applications. Both of these programs resolve all desk accessory ID number conflicts automatically.

If you don't own a utility such as *Suitcase* or *Font/DA Juggler Plus* and prefer to play it safe, then the best bet is never to open desk accessories from two different sources at the same time. To help you differentiate application and System file desk accessories, you might want to place a bullet or some other mark in front of the names of desk accessories installed in applications (see Chapter 5 for details on how to do this). Thus application desk accessory names would rise to the top of the Apple menu (or sink to the bottom, depending on what prefix you use) and you would be able to recognize easily that they are not in the System file.

Using Desk Accessory Extenders Installing desk accessories in applications only makes sense for those desk accessories which are useful in one or two programs. If you have a long list of desk accessories that you want available all the time, then you'll have to turn to a desk accessory extender. Although they can differ significantly in the way in which they go about their business, all desk accessory extenders add desk accessories to the Apple menu temporarily (either during your work in the currently running application, or until you turn off the computer). Some add desk accessories one at a time, others can add many desk accessories at once. Some have a limit to the number of desk accessories they can add, others have virtually no limit at all. Some work automatically, others only on request. As we'll see in the next chapter, a few work with fonts as well.

Since desk accessory adders differ so greatly in the way they work, one may thus be more appropriate for you and your needs than another. What follows are general overviews of, and in some cases some tips on usage for, a few of the many desk accessory extenders which are available.

Other...* and *DA Key Two similar desk accessory extenders are the desk accessory *Other...* and the FKey *DA Key*. With either of these utilities, you can temporarily install desk accessories, one at a time, in the Apple menu.

Setup and operation of both *Other...* and *DA Key* is basically the same. For the sake of illustration, you would perform the following to use *Other...*.

If you want to work with *DA Key* instead, in most instances, a simple substitution of *DA Key* for *Other...* is all you need to do to use these instructions with *DA Key*.

1. Preparing your desk accessories so *Other...* can use them is important. Improper setup can lead to a system crash (in the worst possible case). A disorganized setup, on the other hand, can make *Other...* more difficult to use and thus much less efficient. The following are two simple steps which should prevent both problems:

   a. Collect copies (and be sure they are copies, not originals) of all the desk accessories you want to use with *Other...* in one place. Using Font/DA Mover, make sure that each desk accessory is in its own file (in other words, it does not share the file with other desk accessories). Now is the time, too, to make sure that the file names clearly indicate which desk accessory is inside each file.

   b. Create a new folder and place it somewhere convenient, such as directly on the desktop. Name it `Desk Accessories` (or something similar) and place all your desk accessories inside. You are now ready to use *Other...*.

2. Install *Other...* in your master System file with Font/DA Mover in the normal fashion. (If you are using *DA FKey*, you must install it with *FKey Manager*, of course.)

3. To call up a desk accessory stored in the Desk Accessory folder, select *Other...* from the Apple menu. A standard file selection box will appear. Locate the Desk Accessory folder and double-click on the desk accessory you want to use.

4. At this point, a dialog box may appear indicating that *Other...* needs to renumber the desk accessory in order to install it. This is nothing to be alarmed about, as long as all the desk accessories in your folder are copies, not originals. *Other...* has merely found an ID number conflict and needs to resolve it.

5. When *Other...* has completed the installation process, the desk accessory you selected will appear, ready for you to do whatever it was you had in mind. Close the desk accessory when you are done.

6. If you check the Apple menu, you will see that your desk accessory is now listed in the Apple menu (both *Other...* and *DA Key* prefix the desk accessory name with a diamond and place it at the bottom of the menu). As long as you are in the application from which you originally called it up, the desk accessory can be retrieved again and again at any time by selecting it from the Apple menu, just as you would any other desk accessory. If you quit the application, however, the desk accessory is automatically removed.

Desk accessories installed with Other... *are added to the bottom of the Apple menu and marked with a black diamond.*

In this way, you can call up any desk accessory in your Desk Accessory folder (or anywhere else for that matter) at any time, one at a time. The desk accessory will remain available in the Apple menu until you quit your application. Remember, you don't have to stop with one desk accessory. You can add desk accessories as you need them, within reason. Although there is no official limit to the number you can safely add, it is probably unwise to add more than five or so at any one time.

For many people, particularly those with modest desk accessory needs, this is a more than adequate way to get around the System file's limit on desk accessories. If you are looking for something with fewer limitations and more features, read on.

Suitcase *Suitcase* is an Init file which can greatly expand the number of desk accessories in your Apple menu. It can also be used to increase your supply of fonts and FKeys as well (see Chapter 4 for an explanation of how *Suitcase* can be used with fonts). Since it is an Init file, all you need to do to get it running is to copy it to your System folder and restart the computer. From that point on, *Suitcase* is always available.

Suitcase has two major ways of working, one automatic, the other manual. The automatic side of *Suitcase* goes into effect every time you start or restart your Macintosh. As part of the startup process, *Suitcase* adds to the Apple menu whatever desk accessories you have previously designated. It does so by searching your System folder for desk accessory files, or folders containing desk accessory files, which have been given one of four special names. All the desk accessories in these files or folders are then automatically added to those already in the Apple menu and arranged in correct alphabetical order.

Suitcase can open a maximum of ten FKey, font, or desk accessory files; this is not as much of a limitation as it sounds, however, since each desk accessory file can contain as many as 52 desk accessories and each font file can contain up to 200 fonts (where each individual font size counts as a single font). If there are so many desk accessories on the Apple menu that it runs off the bottom of the screen, the menu scrolls to provide access to those at the bottom of the list, just like an overly long font menu.

Suitcase's automatic mode offers a number of advantages over the usual Font/DA Mover to System file method, but two improvements stand out in particular. One is the fact that *Suitcase* expands the number of desk accessories you can have under the Apple menu into the hundreds. Hundreds of desk accessories may sound like more than you'll ever need (or want), but the important point is that by using *Suitcase*, there is no practical limit to the number of desk accessories you can have at your fingertips at all times. The other advantage is that since the desk accessories *Suitcase* works with are in desk accessory files, the Apple menu is expanded without increasing the size of the System file a single byte. Thus, your Apple menu can list 50 desk accessories (or more, if you like) and your System file can still fit easily on an 800K backup disk.

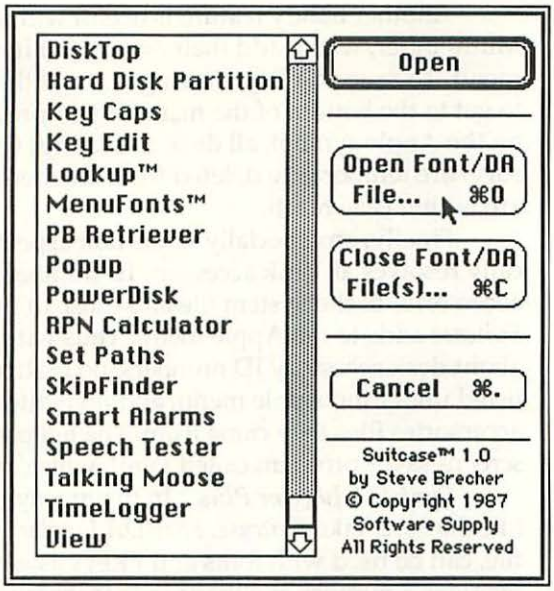
In its manual mode, *Suitcase* allows you to temporarily add any desk accessory or set of desk accessories, at any time, to the Apple menu. This feature of *Suitcase* is provided by the *Suitcase* desk accessory which appears

If Suitcase *adds more desk accessories than the Apple menu can display at one time, the Apple menu will scroll when the pointer hits the arrow at the bottom of the menu.*

automatically on the Apple menu (always at the very top), whenever the *Suitcase* is installed. To add a desk accessory that does not already appear in the Apple menu:

1. Select the *Suitcase* desk accessory (or press Command-K).
2. Click on the Open Font/DA File... button in the *Suitcase* window (or press Command-O).
3. Select the desired desk accessory file in the usual way.

The selected desk accessory (or desk accessories, if the file contains more than one) is added to the Apple menu, in correct alphabetical order, and remains available until you turn off or restart your Mac or until you remove it with the *Suitcase* window's Close Font/DA File... button. The desk accessory files opened in this way are included in the ten-file limit.



Click on the Open Font/DA File... button in the Suitcase *window to add more desk accessories to the Apple menu.*

Fortunately, *Suitcase* also sports a number of useful features which make handling large numbers of desk accessories more convenient. The *Suitcase* window, for example, displays all the desk accessories in the Apple menu in a scrollable list which acts just like a file selection box. This means that you can select a desk accessory without ever touching the mouse simply by:

1. Selecting the *Suitcase* desk accessory by pressing Command-K.
2. Selecting the desired desk accessory from the list in the *Suitcase* window by typing as much of its name as is needed to distinguish it from all the others.
3. Pressing the Return key to open the selected desk accessory.

Of course, you can also select any desk accessory in the scrolling list using the mouse in the usual way, but that's not much different from selecting the desk accessory from the Apple menu.

Another handy feature is useful with such programs as *Switcher* and MultiFinder, which add their own menu items to the bottom of the Apple menu. To save you from having to scroll through 50 or 100 desk accessories to get to the bottom of the menu, if you press the Option key when clicking on the Apple symbol, all desk accessories (except the *Suitcase* desk accessory) are temporarily deleted from the menu so that the special menu items are within easy reach.

Finally, an especially important aspect of *Suitcase* is that it automatically resolves all desk accessory ID number conflicts between the desk accessories in the System file and those in the desk accessory files which *Suitcase* adds to the Apple menu. Thus with *Suitcase*, you need never worry about desk accessory ID numbers, no matter how many desk accessories are listed under the Apple menu, and no matter how many different desk accessories files they came from. The *Suitcase* package also includes a screen- saver program called *Pyro!*, which is described in Chapter 5.

Font/DA Juggler Plus In many ways, *Font DA Juggler Plus* is much like *Suitcase*. Like *Suitcase*, *Font/DA Juggler Plus* is an easily installed Init file, can be used with fonts and FKeys as well as desk accessories, and includes a number of enhancements that make handling a large number of fonts easier. However, unlike *Suitcase*, *Font/DA Juggler Plus* works with sound files (which you can use to replace the beep sound), can add up to twelve font/FKey/sound/desk accessory files, and can automatically open any font/FKey/sound/desk accessory file, no matter what it is called or where it is stored on disk.

Font/DA Juggler Plus also has many features which make managing large numbers of desk accessories more practical. To reach the bottom of the Apple menu quickly, to select something from the MultiFinder application list, for example, hold down the Shift key when you click on the Apple symbol. As with *Suitcase*, the result is a special abbreviated menu in which all desk accessories (except the *Font/DA Juggler Plus* desk accessory) are temporarily removed. With a special "Hot Key" you can call up any installed desk accessory from the keyboard, either by typing its name or by using the arrow keys. And with *Font/DA Juggler Plus*'s *DA Utility* you can rename or renumber both desk accessories and fonts.

Summary

- Put together your master System file, using the information on FKeys and desk accessories in this chapter and the additional

material on fonts in the next chapter, before moving on to the other fully powered techniques and strategies described later in the book.

- FKeys are a convenient way of placing useful utilities at your fingertips. Two important points to remember about FKeys are:
  - ◊ If you don't have a desk accessory extender such as *Suitcase* and thus are limited in the number of desk accessories you can install, FKeys which duplicate desk accessory functions can eliminate the need for some desk accessories and thus free up space for others.
  - ◊ Unlike desk accessory windows, FKey windows cannot be kept open while you work; you must close them in order to return to your application. Thus, if faced with a choice of desk accessory or FKey, choose the desk accessory if you want to keep the window open in the background.
- With *FKey Manager*, you can install as many FKeys in your System file as you like. You can also use *FKey Manager* to rename, renumber, and remove FKeys. In addition:
  - ◊ The maximum number of FKeys available from the keyboard (using the Command-Shift-number key combinations) is normally ten. If you install more than that, those numbered ten and above must be called up from the *Pop-Keys* menu, which is configured with *FKey Manager*.
  - ◊ The *Pop-Keys* menu's **Other...** command provides a way to call up any FKey which is not installed in the System file.
  - ◊ If the *KeyPad* file is in your System folder, you can use *FKey Manager* to convert either the top row or key pad number keys into dedicated function keys which call up your FKeys at the press of a single key.
- With *FKey Maker*, you can create FKeys which call up your favorite desk accessories. This is one way to call up desk accessories from the keyboard (see Chapter 5 for another).
- If you want to call up all your FKeys from the keyboard (including those numbered ten and above), you can do so by assigning keyboard commands with *QuicKeys* to those FKeys which lack them.
- Even with MultiFinder, desk accessories are still useful in lots of ways. For example, they make good sense for small needs that

don't require large applications. They also come in handy when memory is too tight for a full-sized application and a desk accessory mini-application will get the job done.

- If you want to accommodate more than fifteen desk accessories, then first install the following types of desk accessories in the System file:
  - ◊ Desk accessories which must be installed with their own installer programs
  - ◊ Desk accessories which require support files (data files, dictionaries, and the like)
  - ◊ Desk accessories which turn background programs, such as screen savers, RAM caches, and print spoolers, on and off
  - ◊ Essential Apple desk accessories, such as Chooser and the Control Panel
  - ◊ *Other...*, if you are not using a desk accessory extender
  - ◊ If there's any room left over, the desk accessories you use most often
- If the number of desk accessories you want to use in addition to those in the System file is small, then use Font/DA Mover to install special-purpose desk accessories in the applications you use them with, and use *Other...* to call up additional desk accessories as required.
- If the number of desk accessories you want to use in addition to those in the System file is substantial, then use a font/desk accessory extender such as *Suitcase* or *Font/DA Juggler Plus* to expand the capacity of the Apple menu beyond most mortals' needs.
- Once you have decided on a combination of desk accessories and FKeys, be sure to test them, singly and in concert with all your major applications, before using them in everyday work.

4

The System File, Part 2: Fonts

Which Fonts for What Purposes *141*
- The Kinds of Fonts *142*
- ImageWriter Fonts *144*
- LaserWriter Fonts *148*

Fitting All Your Fonts in the Font Menu *161*
- Installing Only the Screen Sizes and Styles You Need *162*
- Installing Fonts in Applications and Documents *172*
- Adding More Fonts with a Font/Desk Accessory Extender *176*
- Space-Saving Strategies for Downloadable LaserWriter Fonts *182*

Customizing Fonts *185*
- Changing Screen Font Names—and Why *186*
- Adding Fractions to a PostScript Font *198*

Summary *215*

Most computers have one font for on-screen display and that's about it. On my Compaq, for example, even though I can print with any resident LaserWriter Plus font I want (if I'm using Microsoft *Word*), what I see on the screen never changes. With the Macintosh, on the other hand, there is nearly an unlimited variety of fonts to choose from, and with few exceptions, what I see on the screen is a reasonable approximation of what I'll get on paper. This wide variety of fonts accurately represented on the screen makes it easier to perform text-based tasks and to produce good-looking documents without a lot of trial and error printing. Screen fonts can even transform mundane word-processing into something approaching creative and enjoyable work.

The power to see different fonts on the screen and the multitude of fonts there are to choose from can create problems as well. The most basic problems, the ones that must be addressed by anyone putting together a master System file, are how to choose the right fonts and how to use them effectively. Unfortunately, the Macintosh manual fails to cover all but the simplest font facts, and since most people don't even read the manual, even what little is there is missed (or if read, forgotten). But a proper understanding of fonts, how they work, and how to put them to work, is essential if you want to produce first-class documents.

As I mentioned in the previous chapter, it is most important that the job of creating your master System file and folder and installing the fonts you want the System file to contain be completed before moving on to other fully powered strategies. Thus, the issues revolving around choosing and using fonts are the subject of this chapter. As you might expect, many Macintosh font problems are a result of their great abundance. With so many to choose from, it is easy to go on a "font binge" and end up with a System file bursting at the seams and a font list so long that you can't even remember what most of the fonts on the menu look like. Careful planning and organization are, therefore, just as important with fonts as with desk accessories, perhaps even more important.

In organizing the fonts for your master System file, it is essential that you choose the ones that are appropriate for you and your work. The information you need for the selection process is covered in the first third of this chapter. Since, after careful selection, you can still end up with a large number of fonts which occupy a considerable amount of disk space, the

middle third of the chapter discusses ways in which you can reduce the amount of disk space each font requires or get around space restrictions with a font/desk accessory extender. But even with the best and widest selection of fonts skillfully installed in your System file, there are still improvements you can make. Thus, the final third of the chapter discusses reasons for and ways to customize existing fonts or to create an entirely new custom font containing special symbols, logos, and other useful characters.

This chapter assumes, by the way, that you have a basic understanding of Macintosh fonts: how to recognize font files, how to install fonts in the System file, and so on. If any of these relatively elementary concepts are yet a mystery to you, you should go back to the Macintosh manual and the *Macintosh Utilities User's Guide* to review the material on fonts and Font/DA Mover.

Finally, a note on printers before we begin. The ideal fully powered Mac is attached to a LaserWriter of one sort or another, but for financial reasons many fully powered Macs will be connected to less than fully powered printers. Thus this chapter covers information for the two printer families used most often by Macintosh owners, LaserWriters and ImageWriters. Since LaserWriters and ImageWriters work with fonts in completely different ways, their needs and requirements are discussed in separate sections. For a well-rounded understanding of Macintosh fonts, you are encouraged to read both sections.

If you use a printer that is neither a LaserWriter nor an ImageWriter, what's written here will still be of value, although you may have to adapt my comments just a bit to fit the idiosyncrasies of your printer. Which section applies most directly to your printer depends more on whether or not it is a bitmap or PostScript printer, and less on whether it is a laser printer or a dot-matrix printer. If you have a PostScript printer, use the LaserWriter discussions as your guide; if not, read the ImageWriter discussions. Of course, not all the information is guaranteed to apply, but most of it should.

Which Fonts for What Purposes

Due to the abundance of fonts available for both the ImageWriter and the LaserWriter, the first step in setting up the fonts in your master System file

is to winnow out all the fonts which are unsuited to you and your printer. The first step in that process is to think about what you use your fonts for (manuscripts, fliers, newsletters, posters, correspondence, spreadsheets, whatever) and then to determine what kinds of fonts are best suited to those types of output. The next step is to separate out ImageWriter and LaserWriter fonts, and choose from among those which were designed with your printer in mind. This section, therefore, explains the differences among fonts in general and between LaserWriter and ImageWriter fonts in particular, and discusses how to choose the right kinds of fonts from among the many available.

The Kinds of Fonts

An essential step in choosing fonts for your font menu is analyzing how and why you use fonts. Do you, for example, draw up contracts with a lot of fine print? Do you create posters which require bold or eye-catching lettering? Do you write complex documents with headings of many levels or newsletters with headlines that need to be distinctly different in style from body text? Do you print a lot of correspondence that should look slightly informal, or do you self-publish books that should be as professional-looking as possible? What you do with words will determine to a great degree the fonts you should be using.

Although one can categorize fonts (or type families, to be typographically correct) in many different ways, I find dividing them into three broad categories—serif, sans serif, and everything else—most helpful when selecting fonts. Each one of the three groups has different characteristics and different uses. Because they have different uses, once you have determined just how you use text on the Mac, you can isolate immediately the font category or categories relevant to your work, and forget for the moment about all the others.

Serif fonts are characterized by thin, horizontal lines (serifs) embellishing the thicker, vertical strokes of most letters. The uppercase I, for example, has small horizontal flourishes at each end in a serif font, and the vertical stems of lowercase letters, such as m, n, h, r, and l, all have short horizontal bases to sit on and little flourishes up above as well. Serif fonts on the Macintosh include New York and Toronto for the ImageWriter and Times, Palatino, ITC (for International Typeface Corporation) Bookman, and New Century Schoolbook for the LaserWriter Plus. Because the serifs in serif

fonts help to define the text line and are, therefore, easier on the eyes, they are almost always used for lengthy pieces of text; that's why they are often referred to as text fonts. Look at almost any book (including this one), magazine, or newspaper and you will find the body of the text set in a serif font.

Sans serif fonts, in contrast, lack serifs (hence the name). All strokes are usually of equal weight, and the main stems usually lack embellishment of any kind. Sans serif fonts for the Macintosh include Geneva, Monaco, and Chicago for the ImageWriter; Helvetica for the LaserWriter, and ITC Avant Garde for the LaserWriter Plus. Sans serif fonts tend to look coldly modern, and in long pieces of text are tiring to read. Thus they are usually used as display type for headlines, section heads, chapter titles, and advertisements, where their bold appearance draws attention to what they have to say and the text is not overly long.

Sans serif fonts are noted for their lack of embellishment, and thin, horizontal flourishes identify a serif font.

I always prefer text in a serif font, but other people are apparently not so dogmatic on this point. For example, you might decide to print an entire brochure or short newsletter, both headlines and text, in a single sans serif font. In situations like these, in order to make the text easier on the eyes, you should choose a sans serif font with uneven stroke widths (thick verticals and thin horizontals) or a sans serif font containing a hint of a serif here and there on a few lowercase (a, j, i, r, and t) and uppercase letters (I and Z, for example). Such fonts are uncommon, but they do exist. Optima (a downloadable LaserWriter font by Adobe Systems) is an example of a sans serif font with variable stroke widths, and a very elegant one, too.

The special font category is where I put everything else. By and large, these are special fonts for special purposes: symbol fonts such as Symbol and ITC Zapf Dingbats for the LaserWriter and LaserWriter Plus; picture fonts such as Cairo for the ImageWriter; decorative fonts such as Athens and London for the ImageWriter and ITC Zapf Chancery for the LaserWriter; and foreign language fonts such as Greek and Cyrillic.

The uses to which symbol and picture fonts are meant to be put are obvious (if limited). If you need such a font, you'll know it. Decorative fonts are even more limited than picture fonts in their usefulness. But many people cave in to the seduction of decorative fonts, buy one or two or three, and end up using them in situations for which they are unsuited, such as long sections of text, or in ways that are simply unpleasant to look at. Decorative fonts sometimes work well on posters, in advertisements, or as mastheads or headlines. But they were never meant to be used in body text. Unless you want the medium to demand more attention than the message, stick to serif fonts for text. For display type, unless you really know what you're doing, you're better off using sans serif fonts.

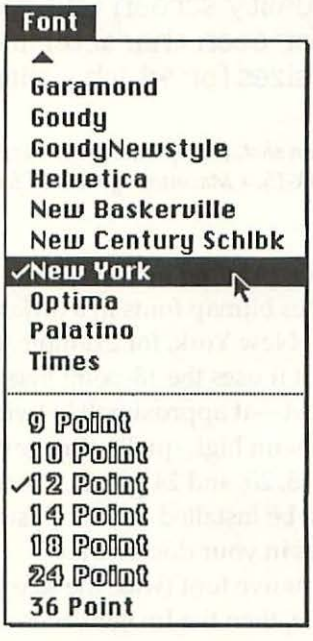
By now, you should have thought about how you use fonts and determined which of the three font categories are relevant to your work. You've probably figured out already that virtually everyone needs a basic set of serif fonts for text. The question is whether you need other types of fonts as well. If you write only letters and memos, for example, serif fonts are all you really need. If you create newsletters, manuscripts with chapter heads and subheads, that sort of thing, you should have a stock of both serif and sans serif fonts. If you design fliers, advertisements, and posters, for which strikingly different visual effects are the order of the day, then you may have to add one or more kinds of special fonts to your list as well.

Keeping in mind the types of fonts you need, let's turn now to the type of printer you have and consider which fonts fit both your needs and the requirements of your printer.

ImageWriter Fonts

ImageWriter fonts, often referred to as *bitmap fonts*, are the fonts that you install in the System file with Apple's Font/DA Mover. They are called bitmap fonts because each letter in each point size of each font is really just a predefined image composed of dots, or bits, at a resolution of 72 dots per inch (dpi). Because bitmap images degenerate when enlarged or reduced, a bitmap font usually contains five or six "native" sizes, each with its own set of bitmapped characters. ImageWriter fonts are also sometimes called *screen fonts* because the dots that make up the size and shape of the letters on an ImageWriter page also determine what you see on the screen. ImageWriter fonts thus serve double duty: They are used to create the letters on a printed page and the letters on the Macintosh screen.

On-screen versus on the Page The only twist to this simple arrangement involves the question of how different font sizes are used to create the text seen on-screen versus that on the page. As I mentioned, ImageWriter font files usually contain a number of different font sizes, each one with its own set of bitmap images for all the characters in the font. My New York font file, for example, contains 9-, 10-, 12-, 14-, 18-, and 24-point versions (these native font sizes appear in outline type on the font menu).



The native sizes of a font are listed in outline text in the font size menu.

This seemingly unnecessary duplication is actually quite important. First, consider screen text. Readable, high-quality screen text requires a tailor-made bitmap image for each character in each size that you are using. Text in sizes for which a bitmap is not available, such as 15-point New York, must be created by the Mac from some other size; the results are serviceable, but usually not very easy on the eyes. Thus, for high-quality, what-you-see-is-what-you-get screen text, you should have in your System file the same font sizes that you regularly use when writing (10, 12, and 18 points, for example).

of different font sizes, each one a set of bitmap images for all the characters in the font. My New York font file, for example, contains 9-, 10-, 12-, 14-, 18-, 20- and 24-point versions (these font sizes appear in outline type on the font menu).

This seemingly unnecessary duplication is actually quite important. First, consider screen text. Readable, high-quality screen text requires a tailor-made bitmap image for each character in each size that you are using. Text in sizes for which a bitmap is not available, such as

In this screen shot, the top lines are in New York-14, a native font, and the bottom lines are in New York-15, a Macintosh-generated font.

When printing in best-quality mode on the ImageWriter, however, the printer uses bitmap fonts in a different way. The Mac uses the 9-point version of New York, for example, to create 9-point New York text on-screen, but it uses the 18-point version (reduced 50%) to create 9-point printed text—at approximately twice the resolution of the Mac's screen. Thus to obtain high-quality ImageWriter text, the larger sizes of the fonts you use (18, 20, and 24 points, for example, for 9-, 10-, and 12-point printed text) must be installed in your System file, even if you never actually use those sizes in your documents.

If a native font twice the size you want to print is not installed in your System file, then the ImageWriter uses the same font size as that displayed on the screen, with some loss in print quality. If neither the double-size nor the actual size of the font to be printed is installed in your System file, then the Mac creates a bitmap for both screen (as above) and printer by resizing one that is installed. The results of this turn of events are, at best, ugly and, at worst, unreadable. Hence, you should avoid Macintosh-generated fonts at all costs. The easiest way to do this is to install all the sizes you need of all the fonts you regularly use.

We'll return to the inevitable proliferation of ImageWriter font sizes and how to deal with this unwanted abundance later in the chapter. For now, it is important to keep in mind that text intended for the ImageWriter should be formatted in native font sizes whenever possible, and that native fonts twice the size actually specified are required for best-quality printing.

Choosing ImageWriter Fonts Macintoshes come equipped with a number of ImageWriter fonts, including the serif font New York, sans serif fonts Geneva, Monaco, and Chicago, and special fonts Cairo, London, Venice, and Athens. For many, this may be enough. New York is one of the best-looking fonts on-screen, and it's almost as good-looking on paper. Geneva works well with New York, when used in headlines or section heads. As for Cairo, London, and Venice, my only advice is to use them sparingly, if at all.

As good as Apple's standard fonts are, however, few can resist the temptation to add more fonts to their System file. You can acquire a wide variety of ImageWriter fonts at reasonable prices from commercial software vendors such as CasadyWare, Century Software, and Dubl-Click Software. You can also add to your font collection by obtaining public-exchange fonts from electronic bulletin boards, users groups, and information utilities such as CompuServe. If free fonts such as these interest you, you might want to check out *Free (and almost free) Software for the Macintosh* (Crown Publishers) which illustrates over 200 free fonts and lists some of the places where each one can be found.

Serif Fonts If you are in the market for a new serif font, be sure it has the qualities every good text font must have: it should be easy to read on-screen and (this is, of course, especially important) on paper; and it should come in a wide variety of sizes, small for the screen and large for best quality printing on the ImageWriter.

Unfortunately, high-quality text fonts for the ImageWriter are not very common. Of the serif fonts available, one of the best is Boston II (public-exchange). It produces excellent-quality type on the ImageWriter, comes in an abundance of sizes (7, 8, 9, 10, 12, 16, 18, 20, and 24 points), and contains many useful extra characters, such as the complete Greek alphabet, fractions, and more. The smallest sizes are meant to be used for sub- and superscripts. Also, there is a 14-point size, which is really just a set of single-pixel-high blanks which you can use to add space between lines (*leading* in printers' jargon). There are two versions of the font, one for programs such as MacWrite, which cannot handle font sizes below 9 points, and one for programs such as Word, which can.

Other notable public-exchange serif fonts include Vancouver (by Jason Kanter), which is based on Compugraphic's Souvenir, and Willowdale. Commercial font-makers offer very few useful text fonts. The best of these, to my knowledge, is Fluent Fonts' (CasadyWare) ImageWriter version of Bodoni in sizes from 9 to 36 points.

The fact that I can recommend only four ImageWriter serif fonts is not especially good news for those who are looking for variety in text fonts. Understandably, frustration caused by this lack of quality text fonts may tempt ImageWriter owners into using the screen versions of LaserWriter text fonts, which are fairly abundant (see below for more about LaserWriter fonts and an explanation of their use of screen fonts). In most cases, however, this is a temptation worth avoiding. LaserWriter screen fonts are primarily designed to duplicate on-screen the character spacing that will be printed on paper. They are designed only secondarily to look good on the screen and were never intended for printing by an ImageWriter. While it is true, for example, that Palatino printed by the LaserWriter looks terrific, Palatino's screen font printed on the ImageWriter is disappointing. Thus if you are looking for good-quality ImageWriter output, take my advice and be sure to use fonts designed specifically for the ImageWriter.

Sans Serif and Special Fonts Sans serif fonts are relatively abundant, both commercially and in the public-exchange arena, although those with plenty of different sizes are in the minority. Commercial sans serif fonts include: Micro from Fluent Fonts (CasadyWare); Epsilon (an ImageWriter version of Optima), Omicron Bold (an ImageWriter version of Futura Bold), and Sigma (an ImageWriter version of Helvetica) from Professional Type Fonts (Kensington Microware); and Copenhägen, Montreal, Paris, and Stuttgart from World Class Fonts (Dubl-Click Software). Good quality public-exchange sans serif fonts include Long Beach, Prague, and Wall Street.

The vast majority of both public-exchange and commercial fonts are decorative fonts, wonderful to look at but difficult to use artfully, and other specialty fonts. Although abundant, these fonts are primarily of use to graphic designers who need bold and eye-catching type or special symbols. As a rule, they are available in only a few and usually large sizes. Decorative and other special-purpose fonts of all descriptions can be obtained from public-exchange sources and offer something for every taste. Fluent Fonts, World Class Fonts, and other commercial font producers also offer imaginative decorative fonts, as well as picture and symbol fonts.

LaserWriter Fonts

LaserWriter fonts are a more complex beast than ImageWriter fonts. Unlike ImageWriter fonts, LaserWriter fonts are comprised of two parts: one part

draws the letters you see on the screen and the other part creates the letters printed on paper. The part that creates the printed text is written in PostScript, Adobe's page description language, and for that reason LaserWriter fonts are often called PostScript fonts. PostScript font is also a more accurate term, since these fonts can be used on any PostScript printers, not just the LaserWriter. However, since only a handful of people have a Linotronic in the living room but many more own a LaserWriter, the term *LaserWriter font* is used here without apology.

Unfortunately, working with LaserWriter fonts is often more difficult and sometimes more frustrating than working with ImageWriter fonts. It is also more expensive. You can, of course, use the cheaper and simpler ImageWriter fonts on the LaserWriter. Some of them even look quite respectable if you turn on the Smoothing option in the Print dialog box; but you're really wasting your time and the money invested in your LaserWriter if you do. The quality and versatility of LaserWriter fonts far outweigh whatever drawbacks they may have. If the printed page is capable of giving you pleasure, then a page printed with a LaserWriter font cannot fail to please you more than the same page printed with an ImageWriter font.

On-screen versus on the Page Each LaserWriter font consists of a screen font file and a PostScript printer font file (and sometimes an AFM—for Adobe Font Metrics—file, which Macintosh owners can safely ignore). The screen fonts are bitmap fonts (with a resolution of 72 dpi); they usually come in a number of different sizes and thus are not much different from ImageWriter fonts. Like any ImageWriter font, LaserWriter screen fonts are installed in the System file and chosen from the font menu, but the similarities end about there. This is because the primary purpose of a LaserWriter screen font is to portray on the screen the way the font's PostScript-produced text will look on paper. Especially important in this regard are individual character widths (as opposed to their shapes), and their effect on line lengths, tables, columns of figures, and so on. LaserWriter screen font designers must often sacrifice the on-screen appearance of a character to preserve its proper width, and as a result, LaserWriter screen fonts are not always easy to read or even particularly good looking.

The differences between the two types of fonts are not restricted to matters of screen font design. When you print with an ImageWriter font, bitmap representations of every character, obtained from the font installed in the System file, are sent to the ImageWriter for printing. When you print

with a LaserWriter font, however, quite a different process occurs. Since the screen font is just for viewing on the screen, a different technique is used to create the printed page. First, the Macintosh locates the PostScript version of the font in the printer's memory, or puts it there, if necessary. (See the next section for more information on downloadable LaserWriter fonts.) It then sends the text to the printer, not as bitmapped images, but as information about the letters, numbers, and symbols which are to be printed in the selected font, and their formatting.

For example, the Mac would tell an ImageWriter to print a vertical bar of eight dots followed by a horizontal row of three dots along the baseline, but it would tell a LaserWriter to print an uppercase L of a certain font, size, style, and position on the page. The LaserWriter would then take the information about the size of the letter, use the PostScript font file in memory to produce a bitmap (at LaserWriter resolution) of the specified size. Finally, it would place the L in the correct location on the image of the page being assembled in the printer's memory, and then when the page is complete, it would print the page.

The quality of ImageWriter fonts is limited by the resolution of the bitmap image used to define the characters in the screen font. On the ImageWriter, when you are printing in best-quality mode, the resolution of printed text is approximately 140 dpi. But best-quality printing is only possible when a font twice the size of the one you specified is available (when there is a 24-point size in the System file and the text to be printed is 12 points, for example). Otherwise, if a double-size font cannot be found, the ImageWriter will print at only half that resolution. Thus ImageWriter fonts can produce high-quality text only in a limited number of sizes.

LaserWriter fonts, on the other hand, can print high-quality text of virtually any size. The only limits are the font sizes your application can specify and the size of the paper you're printing on. This is because PostScript fonts are descriptions of detailed shapes (created with mathematically precise curves and lines) and not collections of dots. These detailed shapes are scaled up or down by the LaserWriter to any size you specify and only then converted into bitmaps. Since this conversion process is performed by the printer and takes into account the printer's resolution, the quality of PostScript fonts is limited only by the resolution of the printer, which for the LaserWriter is 300 dpi. Thus LaserWriter fonts print at 300 dpi, no matter what size text you choose, no matter what size screen fonts are installed in your System file, and no matter what the resolution of the screen font. With

Linotronic printers, PostScript fonts print at resolutions many times higher than that. This combination of detailed character shapes and the highest possible resolution is what makes LaserWriter fonts look so good.

There is also another reason PostScript fonts are of such high quality. In many PostScript fonts, italic, bold, and bold-italic faces are designed and stored separately from the plain (or roman) style. Thus instead of taking the roman style and altering it according to some formula to make bold or italic (which is what usually happens in ImageWriter fonts), the LaserWriter uses a specially designed bold or italic PostScript font when these styles are called for. Such carefully designed bold and italic fonts are inevitably better looking than the same styles derived mechanically from the roman font.

Resident versus Downloadable Fonts There are two major categories of LaserWriter fonts, and they are easily distinguished. There are the fonts that come with the printer (and are stored in its ROM), and there are those that do not (they come on a disk and are sent to the printer, to be stored in its RAM only when needed). Fonts in the first group are called *resident* or *built-in* fonts, and those in the second group are referred to as *downloadable* fonts. Resident LaserWriter fonts are about as easy to use as ImageWriter fonts; it's usually with downloadable fonts that people run into difficulty.

Resident Fonts The LaserWriter comes with four fonts in ROM: Times, Helvetica, Courier, and Symbol. The LaserWriter Plus ROM contains, in addition to the LaserWriter's four, ITC Avant Garde, ITC Bookman, Helvetica Narrow, New Century Schoolbook, Palatino, ITC Zapf Chancery, and ITC Zapf Dingbats. Because the PostScript descriptions of these fonts are in the printer's ROM, you really don't have to deal with them very much. As long as the corresponding screen fonts are installed in your System file (or are made available by a font/desk accessory extender such as *Suitcase*), you can select resident fonts at any time, in any style, and in any combination.

By the way, the screen fonts for almost all LaserWriter and LaserWriter Plus resident fonts come in four different styles (roman, bold, italic, and bold-italic). For reasons unknown, however, Apple provides LaserWriter owners with the roman screen fonts only. But as you will see later in this chapter, failure to use the italic and bold screen fonts can result in screen text that is difficult to read and a less-than-accurate representation of the printed page, and printed text that is poorly spaced.

If you use italic or bold faces in any of the LaserWriter's text or display fonts, you may find it worthwhile to obtain the complete set of LaserWriter and LaserWriter Plus screen fonts from Adobe Systems. Or if you use and need only a few screen fonts and are a member of CompuServe, you can download just the ones you need from CompuServe's Adobe forum at no charge other than your usual connect-time fees. To do so, simply type `Go Adobe` at any CompuServe prompt (an exclamation point). Once you have entered the forum, go to Data Library 6, where the font files are stored. With the Browse command, search for the screen font you need by entering the name of the desired font, such as Palatino, or the word `screen` (for screen font) as the key word. Once you locate the screen font you need, download it in the usual way.

Downloadable Fonts The small number of fonts built into the LaserWriter and LaserWriter Plus are but a drop in the bucket compared to the choices available to traditional typesetters. Fortunately, a number of firms are busy converting many traditional typefaces into PostScript fonts that can be used by the LaserWriter and other PostScript printers. The fact that these fonts are not built into your printer means that they are not quite as easy to use as those that are. This should not deter you, however, from taking advantage of the variety of high-quality typefaces available in downloadable form.

Like any PostScript font, downloadable fonts come in two parts (at least): screen fonts and PostScript font file. As with any other LaserWriter font, the screen fonts must be installed in your System file (or are made available by a font/desk accessory extender such as *Suitcase*) so you can select the desired font from the font menu. Unlike resident LaserWriter fonts, however, the PostScript font file is not in the LaserWriter's ROM, but on a disk. Most PostScript font files can be identified by their icon, a small LaserWriter with a piece of paper displaying the letter T sticking out of the top. (Some companies, such as Adobe, use special PostScript font icons.)

There are two ways in which a downloadable font can be transferred from the disk to the LaserWriter: manually or automatically. As you might guess, the easiest one to deal with is automatic downloading. With this hands-off method, when the LaserWriter needs a PostScript font file not already in memory (ROM or RAM), the appropriate PostScript font file is located (it should be in the System folder) and copied to the LaserWriter's RAM automatically. Once in RAM, the font file is used just like a resident font. Finally, when the print job which required the downloadable font is

finished (or with *PageMaker*, when the text segment that required the font is completed), the font file is flushed from the LaserWriter's RAM to make room for future downloads.

To manually download a PostScript font file, you need a special download program. Commercial LaserWriter fonts characteristically come with such programs, which are easy to operate. With this method, you would launch the download program, select the desired PostScript file (from any folder or disk) and then invoke the command that sends it to the LaserWriter. Unlike automatically downloaded fonts, manually downloaded fonts remain in the LaserWriter's memory until you reset the printer or turn it off.

Once it is sent to the printer, a manually downloaded font is immediately available to every application, and no time is wasted downloading it every time a document needs it. On the other hand, with the manual method, you must always remember to download the fonts you need *before* you print, perhaps even before you launch your word processor. If you forget, the LaserWriter may print your document in some other LaserWriter font (such as Times or Courier) or even worse, copy the screen font bit-for-bit. Because manually downloaded fonts are not flushed from RAM after they are used, they can take up valuable memory when they are not needed, and in extreme cases, can even prevent other fonts from being downloaded (manually or automatically) if the remaining free memory is insufficient. (For more information on the LaserWriter's memory limitations and how to keep on top of it, consult the section on space-saving strategies beginning on page 182.)

Downloadable fonts can be used either manually or automatically; the choice is up to you. Because automatic downloading almost always works (although more slowly, at times) and is a basically mindless enterprise, it is thus the best alternative if you're looking for the easy way out. It is the best way to go if you use many different downloadable fonts during the course of a day and need to have previously used fonts flushed from memory to make room for the next ones. Also, you should use automatic downloading if you share a LaserWriter with other Macs; that way, the printer will take care of memory management for everyone, and automatically clear its memory for the next person. If you are using *PageMaker*, only automatic downloading can take advantage of *PageMaker*'s special form of memory management, which makes it possible to have an almost unlimited number of fonts within a single *PageMaker* document.

When does it make sense to take the trouble to download fonts manually? If you (or all the people connected to your LaserWriter) use one or two downloadable fonts exclusively, your printing will go much faster if you manually download the font or fonts at the beginning of the day. This way, your applications won't have to search for the font files on disk or download them for each print job. Also, not all LaserWriter print spoolers are able to cope with automatic downloading. Thus, to make up for the deficiencies of such spoolers, you may need to manually download the fonts you want to use before you set the spooler in motion. (For more on spoolers, see Chapter 7.) Finally, if you don't have a hard disk and simply don't have room to store the needed PostScript font files on your System disk and in your System folder, then you'll have to download the font files manually from another disk before you begin working.

Choosing LaserWriter Fonts Putting together even a small collection of LaserWriter fonts can be an expensive proposition. In addition, the choices are many and the correct choice is not always obvious. Indeed, the number of fonts available is sufficiently daunting that many simply forgo the pleasures of downloadable fonts and stick to using resident fonts. Thus the following three sections offer some help and advice on choosing a basic library of downloadable LaserWriter fonts. If you already have a more-than-adequate collection, then you may wish to skip to the next section. If, however, you are still working on your font library or have yet to begin one, this section is for you.

Serif Fonts The LaserWriter comes with two built-in serif fonts, Times and Courier. Unless you want your expensive LaserWriter to print like a cheap typewriter, you might as well forget about Courier completely. To save space, I don't even bother to install the screen fonts in my System file; if I had wanted a typewriter, I would have bought a typewriter, not a LaserWriter. Times, on the other hand, is an industrial-strength text font. A little dull, perhaps, but it's used in all sorts of publications. As a result, it should be near the top of your list of text fonts.

The LaserWriter Plus contains three additional serif fonts: ITC Bookman, New Century Schoolbook, and Palatino. All three are excellent text fonts. ITC Bookman and New Century Schoolbook are more traditional in appearance, and Palatino is nothing if not elegant. These three fonts, plus Times, comprise a good, basic collection of text fonts. At least one of them should work just fine for you, no matter what sort of text-intensive job you

have in mind. With four fonts to choose from, you can easily vary the appearance of your output from one job to the next. Unless a certain project requires a particular font, or unless you want more choices, these four fonts may be more than enough.

If you are in the market for a greater variety of text fonts, however, there are many excellent choices. For example, Adobe Systems, which created the fonts in the LaserWriter and LaserWriter Plus, publishes an extensive library of fonts, many of which are based on well-known Mergenthaler and International Typeface Corporation faces. Adobe's fonts are among the best available and also among the most expensive. Some of the ones you might want to consider for text purposes are ITC Souvenir, ITC Garamond, ITC Benguiat, ITC Galliard, ITC New Baskerville, Goudy (rhymes with "howdy") Old Style, Century Old Style, Bodoni, Melior, Trump Mediaeval, and ITC Cheltenham.

Although Adobe publishes the lion's share of traditional, text-quality serif fonts, there are a few alternatives, and they usually cost less. CasadyWare publishes its own version of Bodoni, for example. Goudy Newstyle, a reconstruction of Goudy's elegant original, Goudy Old Style, and Cooper are available from Altsys, and Century Software has ITC Garamond. As of the time this was written, there were no public-exchange LaserWriter serif fonts suitable for serious work.

Sans Serif Fonts For a sans serif font, the LaserWriter has that old standard, Helvetica, in ROM. The LaserWriter Plus also has Helvetica Narrow and ITC Avant Garde. (Helvetica Narrow is not actually a completely separate font, but is produced by mathematically altering Helvetica. Adobe's downloadable Helvetica Condensed is the "real," hand-crafted version of Helvetica Narrow.) Helvetica works well for headlines and such in combination with Times, and Avant Garde and Palatino make a nice pair. If you print only an occasional newsletter or flier, these four fonts are good choices and may well provide enough variety for your needs.

On the other hand, you may wish to have more choices than Helvetica, Helvetica-Narrow, and Avant Garde provide, and there are quite a few sans serif fonts available. Adobe has Helvetica Light, Helvetica Black, Helvetica Condensed, and Machine, for example. CasadyWare has Sans Serif (which includes roman, book, and demi-bold), Sans Serif Bold, Sans Serif Extra Bold, Micro, and Micro Extended. Century Software has its version of ITC Avant Garde, plus ITC Kabel, ITC Eras, Gothica (which is based on Futura), Micron (based on Eurostyle), Manistee, Neosho, Potomac,

and Willamette. Altsys has Black Sans Serif, and more keep appearing every month.

The great majority of downloadable sans serif fonts have uniform stroke widths and no serifs at all, but a few do not. Optima is perhaps the best known of the nonconformers, and Adobe sells an excellent PostScript version. Although Adobe's Optima looks good on the LaserWriter, it really shines on higher resolution printers such as the Linotronic. This is because the PostScript file contains two descriptions for many characters, one for the medium resolution of the LaserWriter and one for higher resolution printers. Century Software's Option is an Optima mimic, but it doesn't have Adobe's double character descriptions (and as a result, its PostScript file takes up less space on disk). Other nonconformist sans serif fonts include CasadyWare's Monterey, Century Software's Terra (which is patterned after Globe Gothic), and the public-exchange Princeton (by D.E. Dougherty and S.H. Lam).

Special fonts Decorative fonts are attention grabbers for those occasions when style triumphs over substance. Of the two LaserWriters, only the LaserWriter Plus has a decorative font. It is ITC Zapf Chancery, a script font that says wedding invitation even if the text does not. That's as flamboyant as the LaserWriter's fonts get.

You can get far more flamboyant than ITC Zapf Chancery if you want to. There are many downloadable LaserWriter fonts suitable for broadsides and similar projects and little else. It is, in fact, dangerously easy to fall in love with a gorgeous font, only to discover (too late, after the money has changed hands) that its beauty is only skin-deep. Art Deco, Art Nouveau, ultramodern, script, calligraphic, Germanic, and other impressive styles are just too overpowering for most uses. If you need attention grabbers, then by all means spend money on them. If you don't need a font that makes a statement even before the words are read, then resist the temptation to buy fonts just because they look spectacular.

Should you need a florid font or two, however, there are plenty to choose from: CasadyWare's Calligraphy, for example, or their Prelude Script, Regency Script, Right Bank, or Ritz; Park Avenue by Adobe; Nouveau from the Complementary Type series; or from Century Software, there's Columbia, Congo (which looks like a stencil), Cumberland and Yukon (overweight serif fonts), Missive (which is based on Mistral), Devoll, Rhine and Trent (Germanic style fonts), Thames (a less flowery old English), Seine, and Styx. Century also has Shadow Effects and Shadow Effects 2,

which transform the LaserWriter's Times and Helvetica into special-effects fonts: light gray, dark gray, shadow (like the kind made by the sun, with or without the letters creating the shadow), different types of outline, shaded characters, candy-stripe characters, and more.

Finally, there are special fonts that include characters that are not in the Roman alphabet. The LaserWriter has one special, built-in font, Symbol. It contains the letters of the Greek alphabet in both uppercase and lowercase, as well as a random collection of symbols, arrows, and straight and crooked lines, which few LaserWriter owners ever use. The LaserWriter Plus has, in addition to Symbol, a font called Zapf Dingbats. This font has all kinds of widgets you can use for bullets, plus arrows of all descriptions, and a few tiny pictures of this and that.

Although I don't find the characters in Symbol and Zapf Dingbats all that useful, there are other fonts which do have special symbols that appeal to me. Century has the best collection of picture fonts, including PostScript versions of Cairo and Taliesin (sometimes called Mobile); Doodads, which includes animals, plants, and other objects; Gimcracks, which covers modes of transportation, arrows, and such; and Thingamajigs, which has people, active and at rest, and a variety of Macintosh icons.

Another type of font is the border font. The best are from Altsys. Their four border fonts (which can be edited with Fontographer, if you like) are loaded with classic and elegant borders. Many of these borders are designed by Richard Mitchell and regularly appear in his broadside, *The Underground Grammarian*. Far more esoteric is Century Software's Weather&Nautical font, which contains all the standard National Weather Service and nautical chart symbols. Keep the Key Caps desk accessory handy when using these fonts; otherwise, you'll have little or no idea what character is produced by which key or key combination.

I could write a whole book on fonts (as they say), but that is a different book, not this one. Indeed, as far as LaserWriter fonts are concerned, the book you need has already been written. For complete samples of all current LaserWriter fonts plus helpful information, such as the amount of space the screen and printer fonts require, font ID numbers, keyboard arrangements, and special characters, your best bet is the two-volume reference work, *LaserSampler II* (MacTography).

With so many fonts to choose from, it is easy to drown in a quagmire of typographical aesthetics (or the lack thereof). Although the aesthetics issue is not really within the scope of this book, I can briefly describe the

cures most often recommended: use no more than two or three different fonts per page and, in most instances, only the basic styles (roman, bold, and italic) for each font. The best-looking pages are those composed with only one or perhaps two well-matched fonts. If you are using several fonts, mix them judiciously and systematically, not haphazardly. For example, use a serif font such as Times for body text and a sans serif font like Helvetica for headlines. For emphasis, use italic and bold styles, rather than completely different fonts. Avoid overdosing the reader with outline, shadow, or other exotic forms, and try to avoid underlining. For one thing, it makes text hard to read and for another, it is a throwback to typewriter days when italics were something only a typesetter had and underlining was used as a substitute. If you want italics, use italics.

Troubleshooting: When Bad Things Happen to Good Fonts

LaserWriter fonts are complex and hence sometimes difficult beasts. Although LaserWriter fonts are normally well-behaved, they can occasionally surprise you, unpleasantly. Fortunately, most LaserWriter font problems have readily identifiable and rectifiable causes.

The following are some of the more common problems and their solutions:

- Sometimes the LaserWriter will print (very slowly) using the screen font instead of the PostScript font, creating output that looks more like a product of the ImageWriter. This is usually a problem with downloadable fonts and is caused by a failure, during the automatic downloading process, to locate the relevant PostScript file. During automatic downloading, printer drivers expect to find PostScript files either directly on the desktop (not in any folder) or in the System folder, and they rarely look elsewhere.

  To avoid problems, be sure to put your PostScript files where they can be found; the safest and surest place is the System folder. If you have a lot of PostScript font files and don't want to overwhelm your System folder with all those font files in icon view, simply put all the font files in a stack, one on top of the other (or in several stacks divided by manufacturer, for example). You'll only be able to see the font file on the top of the stack, but your System folder will look much neater. To move the pile, by the way,

surround it with the marquee to select all the files in the stack, then drag as if it were a single file.

- Strange things can also happen if a font's ID number is changed behind your back. First some background: Every font must have a unique ID number in the System file. On rare occasions, Font/DA Mover will change the ID of a font being installed if its ID is already in use by another font (usually one from a different font manufacturer). This is not a problem if you always print using the same System file. But it can cause big headaches if you switch System files (such as when you take your documents to a commercial printing service), for under a different System file, your document may print with a different font. The reason, of course, is that even though the other System file contains your font, the font's ID number may be different from the one in your System file.

  For example, if Glypha's ID number is 134, but it was changed to 165 when you installed it in your System file, your document probably won't be printed by a commercial service in Glypha (which may still have an ID of 134 in the service's System file, or it may have been reassigned to yet another number). Instead, the service will print it in whatever font has the ID of 165 in *its* System file, and there's no telling what that might be. If no font has the ID of 165 in the service's System file, then your document will probably be printed in Courier (or possibly in bit-for-bit copy of the screen font). The most insidious part of this problem is that the screen display always looks just fine, since screen fonts are assigned by name, not number. Thus, the fact that your document appears in Glypha on the screen of the commercial service's Mac is no indication that it will print out that way.

  To avoid unexpected font substitutions, always provide a commercial printing service with a copy of your System file along with your document, and always remember to ask them to print your document using the supplied System file. (This, you may notice, is yet another reason to keep your System file small enough to fit within the confines of a single disk.)

- Changing font ID numbers is a sneaky trick, to be sure. But the most unkind thing that Font/DA Mover can do is refuse to install

any more fonts in your System file. Although few people have yet to reach it, there is a limit to the number of fonts the System file can hold. That limit is 200 fonts (where one font size is counted as a single font), which should be plenty for most people. If it's not enough for you, you'll need a font/desk accessory extender such as *Suitcase* (which is described, in font terms, later in this chapter).

- Hardware can cause a few surprises as well. The most common hardware complaint is that the same font printed on different printers produces slightly different results. Unfortunately, the differences are frequently real and often unavoidable. Due to differences in printer resolution, the details of some fonts are more finely wrought by the Linotronic than the laserWriter, for example, resulting in a somewhat lighter, cleaner, and often more handsome face than the one produced by the LaserWriter. At very small sizes, differences in printer resolution may also cause slight differences in character spacing (not to mention a more readable face on a higher resolution printer). As mentioned, Adobe's Optima is a special case. The fact that its PostScript file contains two versions of most characters, one for the LaserWriter and one for higher resolution printers, means that differences between LaserWriter and Linotronic output are often pronounced and intentional.
- PostScript fonts are known to have a few built-in eccentricities and limitations, as well. The most serious limitation occurs on the LaserWriter, where the quality and legibility of many PostScript fonts deteriorates at sizes below 12 points: curves acquire measles-like bumps, and variations arise in stem widths which should be identical. Some fonts, most notably those of Adobe's Type Library and Century's Designer Series, minimize the problem better than most. So if you must include the fine print on the LaserWriter, then be sure to use the best quality fonts you can find.
- Another point to keep in mind is that fonts from different manufacturers which are based on the same typeface model are never exactly the same. Sometimes these fonts are referred to by same name (such as Goudy Old Style, from both Adobe and Altsys), sometimes by different names (Adobe's Optima and Century's Option). Whatever their names, however, though they may be

painstaking copies of the same original, differences in artistic techniques and PostScript coding abilities conspire to make each company's rendition unique in one way or another. Thus, if the (usually) small differences that occur in fonts based on the same typeface concern you, be sure you know not just whether a commercial printing service has the font you want, but who the manufacturer of that font is as well. Better yet, if possible, have the commercial service use both your System file and your downloadable font files.

- But even the same fonts from the same company sometimes differ in odd little ways. Take the built-in fonts (such as Palatino and Avant Garde) found in the LaserWriter Plus, for example, and the disk-based, downloadable versions of the same fonts used by owners of LaserWriters and Linotronic printers. In some instances, there are subtle differences in character shape and spacing between the built-in fonts of the earliest LaserWriter Pluses and the more recent revisions of the disk-based versions. Oddly enough, different versions of the screen font can affect the leading (the space between lines) produced by identical PostScript font files. The only solution to the former problem is not to change printers. The solution to the latter difficulty is, once again, to always print using your own System file, no matter what or whose printer is doing the printing.

Despite their foibles and imperfections, PostScript fonts are the only game in town for LaserWriter printing. In any case, don't be put off by the trouble-shooting list; you are unlikely to encounter many, or even any, of the problems enumerated. But if you do, forewarned is forearmed.

Fitting All Your Fonts in the Font Menu

After having narrowed down the field, you may still be faced with a list of fonts which would, were you to install all the corresponding screen fonts, inflate your System file well beyond the bounds of common decency. This is an especially difficult problem for anyone who does not have a font/desk accessory extender and wants to keep his or her System file within the one-

backup-disk size limit. If you have a hard disk, the following pages describe many things you can do to prevent a bloated System file. If you have a floppy-disk-based system and intend to use many downloadable fonts, there is advice for you as well in what follows. Be forewarned, however, that your task will be quite a bit more challenging.

If you use a font/desk accessory extender such as *Suitcase* or *Font/DA Juggler Plus*, you need install only a few screen fonts in the System file and let your font extender take care of the rest. Thus the size of the System file is not the problem. Nevertheless, the space-saving schemes in the next few sections are relevant to your situation as well. By following the strategies that are outlined, you can shorten the length of overly long font menus and, more important, recover wasted disk space occupied by unneeded font sizes and styles.

Whatever your situation, it probably makes sense to go through your list one last time to see if there are one or two fonts you could live without. But don't despair if, even then, your font list is a long one. There are a variety of ways in which you can save space and cut corners as you install your fonts, and still make available all the fonts on your list.

Installing Only the Screen Sizes and Styles You Need

It may sound rather obvious to you, but many people ignore the fact that you need to install only those screen font sizes and styles that you actually use. Many ImageWriter and LaserWriter screen fonts come in a multitude of sizes, and many LaserWriter fonts are represented by up to four different screen font styles (roman, bold, italic, bold-italic). In some instances it makes sense to install them all, but often it makes more sense to install only a few of the available sizes and styles, and in some instances, only one. How you determine which sizes and styles to install and which to leave behind depends, first and foremost, on the printer you use.

ImageWriter Fonts: Which Sizes to Install This section is a step-by-step guide to winnowing out unnecessary font sizes and styles and to reduce wasted disk space to a minimum. If you have a very small collection of fonts, you may be able to install all of their screen fonts and ignore this section, wasted disk space or not. To find out if you are in that enviable position, put all the fonts you want to install in a folder and use the Finder's

Get Info command to determine the amount of disk space the entire collection requires. Add that to the size of your System file (the one containing the FKeys and desk accessories of your choice; the one you ended up with at the end of the last chapter). If the total is within the limits you have set for yourself, then you can install them without further ado.

If you have a moderate collection of fonts or use a font/desk accessory extender, you may need only work through the first few steps of the winnowing process. But if you prefer to have a large number of fonts always at hand and do not have a font/desk accessory extender, then you may need to work through all the steps in the following process.

When dealing with ImageWriter fonts, keep in mind two things: which font sizes you normally work with on the screen, and which font sizes are actually used when you print your documents. Say, for example, that you normally work with New York-10 and -12, both on-screen and on paper. As explained earlier, the Mac uses New York-10 and -12 to generate the characters you see on the screen, and New York-20 and -24 to generate the characters on the printed page. The important thing to note here is that although you are seeing only two font sizes, the Mac is actually working with four, and it's what the Mac works with that is important.

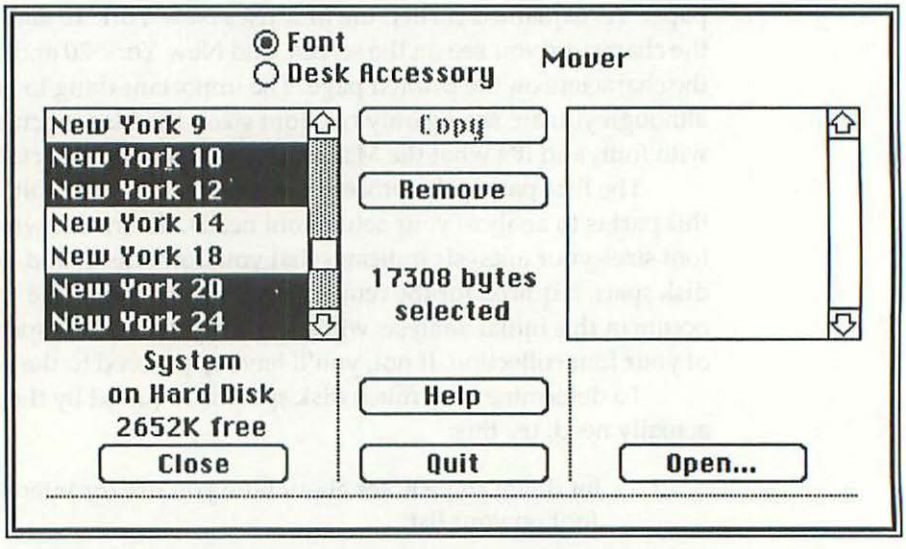
The first part of the process is not really very difficult. The object of this part is to analyze your actual font needs, determine which fonts and font sizes your analysis indicates that you don't need, and then tally up the disk space required for the remainder. If you're lucky, the winnowing that occurs in this initial analysis will adequately reduce the space requirements of your font collection. If not, you'll have to proceed to the next part.

To determine how much disk space is required by the fonts you actually need, try this:

1. Jot down some notes about how you use (or intend to use) each font on your list.

2. Go back over the list and take note of which font sizes the Mac would need, both for the screen display and for printing, to accomplish the desired tasks. For example, you need only install the largest sizes of a font you use for display type, but a text font will need small sizes for the screen and large sizes for printing. Be as precise as you can in this list-making process. If you always use 10- and 12-point type for text, then be sure to note that 9-point type (and the corresponding 18-point size for printing, unless you use 18-point type for headlines) can be safely left behind.

3. Now use Font/DA Mover to find out how much space the sizes you have listed for each font will require. Say you decide you need New York-10, -12, -20, and -24 (so you can print and view on-screen the 10- and 12-point sizes), but not New York-9, -14, or -18:
   a. Open the font file containing New York.
   b. Shift-click on the 10-, 12-, 20- and 24-point sizes in order to select them all as a group.
   c. Make note of the total number of bytes displayed in the center of the Font/DA Mover window, 17,308 in this case, and divide by 1024 (or 1000 if precision is not your style) to get the total in kilobytes.
4. Do this for every font you want to install and add up the results.



You can use Font/DA Mover to add up the disk space required for selected sizes of each font.

Here's an illustration using this procedure. Let's assume that you have a hard disk and 800K disk drives, and that you therefore want to be able to back up your System file on a single 800K disk. Assume further that after installing the desired FKeys and desk accessories, there is 231K left on your System file's current back-up disk. You have placed all the fonts you want

to install in a folder and discovered, to your horror, that their combined size is 456K, nearly double the amount of room available.

So you make your list of fonts, think about how you intend to use them, and make a list of the sizes you think you'll actually need. You open Font/DA Mover and start calculating how much space your new list of fonts and sizes will require: New York, approximately 17K; Boston II, 24K; Vancouver, 16K; Bodoni, 14K; Micro, 13K; Epsilon, 10K; Omicron, 27K; and so on. The total of all these turns out to be 268K, which is a significant reduction from 456K (although still not small enough to fit within the remaining 231K).

With luck, and unlike the example above, the amount of disk space required by your revised list of font sizes is within the limits set by the free space on your backup disk. If so, simply install the fonts and sizes on your list in your System file.

If you use a font/desk accessory extender, you'll probably want to stop at this point regardless of how much disk space your font list requires. Use Font/DA Mover to remove the sizes not on your list (to avoid wasting space unnecessarily) and install a few of your most widely used fonts in the System file. Then put the rest of your fonts in one or more font files for the font/desk accessory extender to work with.

If you don't have a font/desk accessory extender and your font list still requires more disk space than you have to offer or are willing to part with, then it's time to go to the next step. Unfortunately, this next step involves some sacrifice; it's more than just a simple trimming of the fat, so get ready to tighten your belt.

Since the ImageWriter uses fonts twice the size you specify, in most instances the smaller font sizes are used only for viewing on the screen. You can thus eliminate most or all of the smaller font sizes, with no loss of print quality. There will be a loss in screen quality (and accuracy), however, which could cause some eyestrain. The trick is to minimize the eyestrain while maximizing the number of font sizes you can live without. There are at least two ways of accomplishing this.

The first alternative and the most obvious method for trimming font sizes is to install for each font only the larger sizes on your list (which on the ImageWriter are used as is for large type and in reduced form for small type), plus, at most, one small size (whichever one you use most often) for screen work. With a setup such as this, you would do all your initial on-screen work in the one installed small size (12 points, for example), except

for those instances where the larger installed sizes are also needed. After final editing, you would then format the portions of text you want in other small sizes (10 or 9 points, for example, when 12 points is the only small native size), a process that will probably make them unreadable on the screen. The last step, of course, is printing; since the ImageWriter uses the large (installed) sizes to print, the small point sizes will look fine on paper, even if they look terrible on the screen.

The second alternative and a more drastic version of the first scenario eliminates all small fonts completely. In some cases, a 12-point font created from an installed 24-point version, for example, may be good enough to work with without causing severe eyestrain (no worse, for example, than the lower example in the illustration on page 146). In other cases, the resized font may well be indecipherable, or nearly so. If you can stand it, if the reduced version is more or less readable, you may be able to work with the font despite the fact that none of the smaller sizes are installed. If you can't stand it, you can do all your initial work in one font, such as New York, which has all of its sizes (or at least the most important ones) installed. Then, just before printing, convert to the font you actually want to use. That may turn the screen into a tangle of nonsense shapes, but it, again, it should print just fine.

Last minute switcheroos such as these, from one size to another or from one font to another, make it possible to eliminate quite a few font sizes from your font list; virtually all of the smaller sizes, in fact. But such switches may also introduce last minute changes in your documents. Changing sizes or fonts will probably affect your line counts, line lengths, and page breaks, so you have to be careful to check for formatting problems after you make the changes and before you print. Also, resized fonts portray actual character sizes, character spacing, and line lengths less accurately than native sizes do, so what you see on the screen will not only be less readable, but less faithful to the final appearance of the printed version as well.

In the end, the advantages of a trim System file must be weighed against the disadvantages brought about by the lack of a full range of font sizes. Clearly, the most drastic font-size-trimming should be reserved for the fonts you use least often. Those fonts that you use all the time and those you use for projects (such as newsletters) which require highly accurate letter spacing should be trimmed the least, if at all. Fonts that get an intermediate amount of use should be trimmed at an intermediate level.

If all of these calculations and compromises seem like too much to contemplate, then you might try achieving the same result by running the trimming process in reverse. By this I mean loading up your System file with all the fonts on your list (if you have the disk space to spare). Then, little by little, with the help of experience, slowly eliminate the font sizes you never use. Then, when the obvious eliminations have been made, experiment with drastic reductions in the installed sizes of the fonts you rarely use. (If you are working with a floppy-disk-based System, install as many fonts as you can at first, and then install the remaining fonts on your list, one by one, as you eliminate unneeded font sizes.) The problem with this method is that it tends to "punch holes in" or fragment your System file. To reconsolidate your System file when you are finished deleting bits and pieces of fonts, copy it to a floppy disk, delete the original, and then copy it back to your System disk or System folder. (See Chapter 8 for more information on file fragmentation and what to do about it.)

LaserWriter Fonts: Which Sizes and Styles to Install The process of weeding out unnecessary LaserWriter font sizes is similar to that described above for ImageWriter fonts, with the added complexity that many LaserWriter screen fonts come in as many as four different styles: roman, italic, bold, and bold-italic. But added complexity is not all LaserWriter owners must face. There are also space problems far worse than what ImageWriter users deal with. For example, Adobe's collection (all four styles) of screen fonts for the LaserWriter Plus' resident fonts occupies three 400K disks. Obviously, a System file with no more than a complete set of screen fonts for the LaserWriter Plus wouldn't have a prayer of fitting onto an 800K backup disk. This section should help you make the decisions that must be made so that your System file contains the necessary screen fonts and still fits on an 800K backup disk.

As I mentioned earlier, if you have a hard disk and a commercial font/desk accessory extender such as *Suitcase*, you should have little difficulty loading up on as many fonts as you want, including the bold and italic screen fonts which many font manufacturers provide. But to save valuable disk space, it makes sense to eliminate those font sizes and styles that you don't use or don't really need. Thus, even if you use a program such as *Suitcase*, you'll find the information in this section useful as you contemplate a judicious trimming of font sizes and styles.

Since LaserWriter and ImageWriter screen fonts serve different purposes, the criteria used in the font size and style elimination process are

different. One of the most important differences between ImageWriter and LaserWriter fonts, the fact that the screen fonts installed in the System file are not used to create printed output, has important consequences. For example, with LaserWriter fonts, you need only install those font sizes that are critical for on-screen work. In fact, it is possible to get away with a single font size and style per font, whichever one you use most frequently. As you might expect, however, you pay a price for such drastic cuts in your screen fonts.

The obvious price is in readability: only the installed size and style will look very good on-screen; all the other sizes and styles will be difficult, if not impossible, to decipher. This is because, for every size and style not installed in the System file, the Macintosh must generate characters based on whatever font sizes and styles are installed. These generated characters are always more difficult to read on-screen than the native, or installed, sizes and styles.

Screen legibility varies widely in Macintosh-generated characters, but the worst offenders are characters styled in italics and those of any size that is not a simple multiple of an installed size (15 or 17 points, for example, instead of 18 points, when 9 points is the installed size). Thus, if screen legibility matters to you and, for example, you print a lot of text in italics or in a variety of sizes, then you will want to install the italic screen font or a wider range of sizes. An alternative is to do all your text entry and editing in installed sizes and styles for better legibility, and convert to the desired sizes and styles only at the last minute, when you are ready to do final formatting and printing.

There are other, less obvious problems than screen legibility that result from cutting your screen fonts to the bone, however. What are they, and are they of any consequence to you? The answer to this question is rather complex and depends on the specific fonts you are installing, the applications you use, and how precisely you want your printed output portrayed on the screen.

First, as mentioned earlier in the chapter, many PostScript fonts have separate PostScript files for roman, italic, bold, and bold-italic styles, and separate screen fonts to match. Fortunately, most fonts, such as those from Adobe, CasadyWare, and Century, are smart enough to know that if the roman screen font is styled with the Italic Style command, the special italic screen font should be displayed on the screen (if it is available). This is, in fact, the preferred way of selecting font styles, since only if you use style

commands will your font style formatting be preserved in a global font change, from Palatino to Times, for example. (Had you selected the styled fonts directly from the Font menu, all your style selections would be lost in a global change of this sort; for example, Palatino roman, Palatino bold, and Palatino italic would all end up as Times roman in the previous illustration.)

These same fonts are also smart enough to know that they should print with the corresponding italic PostScript printer file when the roman screen font is italicized with the Italic style command. With smart fonts like these, you can either make the italic, bold, and bold-italic screen fonts invisible to the font menu (using the technique described at the end of the chapter) or eliminate them entirely (assuming you can deal with the legibility problem) and still have full access to the PostScript font files via the style commands.

A few fonts, however, such as those created with *Fontographer* (Altsys), are not so smart. For these fonts, you *must* install screen fonts for each style in order to utilize the bold and italic PostScript printer files, or you have to be content with derived faces, which are created, using a stock formula, from the PostScript file of the roman face. (Altsys has recently come out with a separate program called *Family Builder*, which corrects this problem. But the utility requires a certain amount of technical expertise and is not intended for casual use.) If your font manual is unclear as to whether the styled screen fonts are required in order to print with the bold and italic printer files, try some experimental output, or call the manufacturer to find out. If they are essential, then be sure to install screen fonts for each style you intend to use, and to install *exactly* the same font sizes for each style.

The accuracy with which Macintosh-generated characters portray printed output is another issue of which you should be aware. Oddly enough, accuracy depends on the application you are using. Some applications, such as Microsoft *Word* and *MacDraw*, force the printed page to match the screen view exactly. But Macintosh-generated styles (such as bold and outline) and large sizes are often more spread out on the screen than they normally would be on paper. Thus with programs such as *Word* and *MacDraw*, the lack of a full complement of screen fonts can result in bold and italic-styled text which contains, both on-screen and on paper, a little more "air" than it should. In some instances, this additional spacing can be quite noticeable and rather unsightly.

Other programs, such as *PageMaker* and *MacWrite*, use spacing rules that are independent of the screen view. As a result, for generated charac-

ters the line lengths shown on the screen do not always precisely match those on the printed page. The printed text is now correctly spaced, but the lines on the screen may be slightly longer (which is usually the case) or shorter than they are when printed (which is a real headache when trying to place headlines in *PageMaker*, by the way). Line breaks, fortunately, are the same on-screen and on paper. Line breaks within a single document may change, however, if you switch from using a System file, which must use a Macintosh-generated font, to one that can display your text in an installed font.

To minimize the accuracy problem, yet still save disk space, the obvious solution is to install the sizes and styles with which you print most frequently. If you use a great many sizes, install the largest and the smallest ones that you normally use, so that the Macintosh always has two sizes from which it can interpolate its generated characters. Again, to properly take advantage of them, if you are installing italic, bold, or bold-italic screen fonts in addition to the roman font, be sure to install *exactly* the same font sizes in all the different styles. In other words, if you have installed 10-,12-, and 24-point Goudy, and you want to install the bold and italic screen fonts as well, you should install the 10-, 12-, and 24-point sizes of Goudy Bold and Goudy Italic, not just the 10- and 12-point sizes, or some other combination of sizes.

There is one other point to consider as you think about the LaserWriter screen fonts you have chosen and which sizes you should install in your System file. As has been mentioned previously, LaserWriter screen fonts are not designed for appearance as much as for the faithfulness with which they convey line, word, and character spacing on the final, printed page. Thus, many LaserWriter screen fonts are not as easy on your eyes as New York or Geneva, for example, which were designed for on-screen viewing (and ImageWriter printing, of course). Thus if you do a lot of writing, you may want to do all your drafts and editing in New York, and only switch to a LaserWriter font when all the text-intensive work is done. If so, then it may make sense for you to install a complete set of all the sizes of New York and only one or two sizes for each LaserWriter font.

To summarize, although your decisions will have some effect on the appearance of the printed page, which LaserWriter screen font sizes you install will have a great effect on the convenience, readability, and accuracy of the on-screen view. Your decision will be a compromise between readability and accuracy on the one hand, and the size of your System file and/or

the disk space occupied by your font files on the other. How you negotiate this compromise will depend on how you work with fonts. Thus it is impossible for me to give you explicit advice on which fonts and which sizes to install, since I can't possibly cover all conceivable combinations of applications, fonts, and work habits.

I can, however, give you an example of the sort of thinking that takes place and the compromises that are sometimes needed when space is at a premium. The example is my own System file and the way I configured it in my pre-*Suitcase* days. First, I spend the majority of my work time at the computer writing and reading text. I do all my preliminary text input and formatting in New York, a font I find easier to read on the Mac screen than almost any other. Only when I am finished with input, spell-checking, and character formatting (selecting different styles and point sizes) do I convert the document to the LaserWriter font or fonts I intend to print with. I use Microsoft *Word* for long, text-intensive documents such as this book and *PageMaker* for a 16-page newsletter I put out every three months.

Since I spend most of my time working and reworking text formatted in New York, my pre-*Suitcase* System file had a complete set of New York sizes, from 9 point to 24 point. For LaserWriter text fonts, on the other hand, I had only one, two, or rarely three small sizes installed (the ones I use most frequently: 9-, 10-, or 12-point) and for LaserWriter display fonts, I had only one or two large sizes installed (again, the ones I use most frequently; usually 18- and 24- point). Most of the time, the final converted document used these installed fonts and thus was still readable, accurate, and relatively easy to go over for that final last-minute edit. Less frequently, the resulting document used resized fonts and was not as readable or as accurate as it might be. In these circumstances, a little extra tweaking and a few test prints were sometimes required, but the extra minutes this work necessitated was worth it for the space in the System file I was able to recover.

The only exceptions to this setup were the LaserWriter fonts that I used all the time in my newsletter. Since the newsletter is in three-column format, even minor inaccuracies in character spacing can throw off alignment, cause inappropriate hyphenation, and create other problems. Thus for Palatino and Avant Garde, which are the fonts I use in the newsletter, I installed all the screen font sizes I thought I'd ever need. I did not install the screen fonts in styles other than roman, however. As a result, there were times when I couldn't quite decipher a piece of italicized, 9-point

text and other times when italic text was too widely spaced. But I guess you have to draw the line somewhere, and that's where I drew mine.

So, my System file had a font list something like the following:

New York—all sizes; for on-screen writing and editing

Palatino and Avant Garde—almost all sizes; roman only, for desktop publishing precision

Other LaserWriter text fonts—standard text sizes only (9 to 12 point)

LaserWriter display fonts—large sizes only (18 point and above)

Even if you work the way I do (which I doubt), your font list will probably look somewhat different in its details because of different needs and different tastes in fonts. You'll notice that I use resident LaserWriter fonts for newsletter work; this is because they are faster. But you may want to add a little more spice to your desktop publishing projects, for example, and use downloadable fonts instead. Thus you could get away with only a few installed sizes for Palatino and Avant Garde, but you'd want a more complete selection of the downloadable font or fonts you use in *PageMaker* projects.

On the other hand, if you prefer to use the font or fonts you will ultimately print with from the very start, your font selection will look quite different from mine. If you use the same fonts from start to finish, there is no need to have New York or any other ImageWriter font (except for the fonts that Font/DA Mover won't allow you to remove) in your System file. Instead, you would want to install a full range of sizes (and perhaps styles) for the two or three LaserWriter fonts you use most frequently. To avoid wasting disk space, however, it would still make sense to install only a few sizes of the fonts you use less frequently: smaller sizes for text fonts and larger sizes for display fonts.

By the way, now that I use *Suitcase* and have bold and italic screen fonts installed as well, font legibility and accuracy are as good as they can be, and my System file is no larger than it was before. For more about the joys of font/desk accessory extenders such as *Suitcase*, read on.

Installing Fonts in Applications and Documents

Even after you've pared down your screen font list according to the strategies on the previous pages, you may still find yourself unable to install

every font on your list. If your software collection includes a font/desk accessory extender, then you already own the best solution to the problem. If the number of fonts you want to add to your Font menu is large and you don't own a commercial font/desk accessory extender, then you should seriously consider purchasing one. Those who already own, or should own, a font/desk accessory extender may wish to skip the next few pages and go directly to page 180.

If the number of fonts you are dealing with is small and you don't yet own a commercial font/desk accessory extender, then consider an alternative, double-pronged attack on the problem. One prong involves *Fontsie*, a public-exchange font extender (of somewhat more limited means than the commercial ones) described in the next section. The other prong is to install less frequently used fonts in applications or even in individual documents, instead of installing them in your System file. Fonts contained in an application are added to the font menu only when that application is running, and fonts installed in a document are similarly available only when that document is open. When the application or document is closed, the font disappears, and fonts installed in applications or documents have no effect on the size of your System file.

To find out if this alternative will work for you, you need to analyze, once again, when and how you use fonts. Do you use some fonts frequently and in all kinds of applications, and others in only one, or at most two, applications? Do you use some fonts very infrequently, so rarely, in fact, that a little inconvenience in using them would not be a serious problem? If you can see a pattern in the way you use fonts and can answer either of these questions in the affirmative, then it may make sense to install some of your fonts in applications or documents, instead of the System file.

Although it doesn't make much sense to install widely used fonts in specific applications, it makes a lot of sense if you use certain fonts with only one or two programs. For example, if you use New Baskerville only with *PageMaker*, Doodads (a picture font) only in *SuperPaint*, or American Typewriter only with *Word*, you could install New Baskerville in *PageMaker*, Doodads in *SuperPaint*, and American Typewriter in *Word*. They would always be available when you need them but would not clutter up the font menu when they were not needed, and they would not take up space in your System file.

Similarly, you can install rarely used fonts only in the documents that need them. Not only does this trick save space in your System file, it also

makes it easier to pass around documents which require unusual fonts; whoever gets a copy of the document also gets a copy of the (screen) font automatically.

The process of installing fonts in applications and documents is no different than that of installing a desk accessory in an application. To do so, simply hold down the Option key when clicking on the Open button in Font/DA Mover. A list of all available files, not just System files, will then appear in the file selection box. Select the desired application or document in the list, click again on the Open button (or double-click when you select the application or document), and proceed as you would when installing a font in a System file.

While the process of installing fonts in applications or documents is a familiar sounding one, readers who are new to this concept are likely to feel as though they have entered unfamiliar territory. To help you feel more comfortable during your first few forays, here are some points to keep in mind as you work with fonts in files other than the System file:

- You may well find that some of your applications already have fonts installed in them. *MacPaint*, for example, has a font which contains the tool palette's icons. As with the System file, application files can have more than one font, so there's no need to worry if you see one or two already installed. Be careful, however, not to remove any of these previously installed fonts. If you do, you may well ruin your application. *Thus it is imperative that you use Font/DA Mover only on work copies of your applications and files, never on your originals.*
- If you install a font in an application (or several applications), be sure you remove that font from the System file—if it was there to begin with. Failure to do so may cause the font to appear twice on the font menu, which may (or may not, if you're lucky) cause problems.
- The convenience of fonts installed in documents can be greatly enhanced by keeping on hand a supply of blank documents in which the fonts are already installed. To do this, simply create a few blank documents (by opening new documents and immediately saving them) with the relevant application, and give them names that indicate the font they will contain, such as `Garamond Blank`. Then use Font/DA Mover to install the appropriate font in each document. The next time you need a document with

Garamond installed, for example, simply select the file `Garamond Blank` and go to work. Be sure to use the **Save As** command and a different name when you first save your work, to preserve the blank file for future use. (Or, if you are forgetful, after installing the font, you can lock the blank file with the **Locked** check-box in the Get Info window.)

- Although installing fonts in applications (and documents) is a simple and useful technique when circumstances allow it, it contains within it a potential problem that you should be aware of. The problem has its roots in the font ID number discussed earlier in this chapter. As I mentioned, because there are so many fonts now available, some fonts (normally those from different manufacturers) carry the same ID number. Font/DA Mover takes care of this problem within a single file by changing ID numbers when potential conflicts arise. But Font/DA Mover makes no attempt to resolve ID number conflicts between the fonts installed in two different files. Also, in changing a font ID number to avoid a conflict in one file, Font/DA Mover may create a new conflict with a font in another file.

  You could, for example, end up with two different fonts with the same ID number, one font in your System file and the other font in an application, and not know it. Although such occurrences are rare, the results can be frustrating, or worse. To avoid a problem like this, have Font/DA Mover first check and change font IDs for you by loading every font you intend to use into a single, jumbo font file. Use the technique described on page 128–129 for renumbering desk accessories when you install them in applications. Then copy the applicable fonts *from the jumbo file (not the original font file or files)* to the appropriate applications or documents after deleting all the fonts which were already installed in each file. As long as Font/DA Mover does not renumber your fonts yet again (unfortunately, there is no guarantee that it won't), you should have no font ID conflicts between files.

  If you own *Font/DA Juggler Plus*, you can use its *Font/DA Utility* to check, and if necessary, change the ID numbers of problem fonts installed in applications. Then again, if you own *Font/DA Juggler Plus*, you probably won't feel much need to install fonts in applications.

Adding More Fonts with a Font/Desk Accessory Extender

Font/desk accessory extenders add fonts that are still in their suitcase (Font/DA Mover format) files to a Font menu by fooling the System file into thinking that the additional fonts are installed, when in fact they are not. The result is that they make fonts available which would not normally be listed in the Font menu, without adding to the size of the System file. The FKey version of *Fontsie* (public-exchange), for example, allows you to add fonts to your font menu, one by one, while you work. Commercial programs such as *Suitcase*, on the other extreme, make it possible to add literally hundreds of fonts to the font menu, as well as FKeys and desk accessories.

Thus, if you are working with a small number of fonts, or if you don't own a commercial font/desk accessory extender, then the combination of installing fonts in applications and documents (as described in the previous section) and using *Fontsie* is a sensible way to put all your fonts in the Font menu. If you are working with a large number of fonts, however, you would be well advised to purchase a font/desk accessory extender, if you don't already own one, and to skip to page 180; where such programs are described in more detail.

Using *Fontsie* *Fontsie* will add any font in a standard Font/DA Mover format file to your font menu. *Fontsie* doesn't work with all applications; it can cause problems if you ask too much of it, and on rare occasions, it can blow up in your face. *Fontsie* is thus not for everyone, but it is an alternative to the commercial font/desk accessory extenders described in the next section, and it is one possible solution to the dilemma of what to do with fonts you use only once in a while, especially if the number of fonts you ask *Fontsie* to work with is small.

Fontsie is available in several versions, as both an FKey and an Init resource. The FKey and Init versions, though similar in purpose, are different in function. Here are some other points to keep in mind as you use one or the other version of *Fontsie*:

- *Fontsie* works best with programs that have a standard Font menu (one called Fonts or Font). *Fontsie* may or may not work with applications that do not have a font menu, depending upon the

application. You should test using *Fontsie* with any applications not containing a standard font menu (applications that have you select fonts from a dialog box, for example, or whose font menu contains both fonts and font sizes) before entrusting the combination to serious work.

- *Fontsie* the Init resource does not check for font ID number conflicts. To avoid having two fonts with the same ID number on the menu, you may wish to install *all* the fonts you intend to use in the System file first, including those you intend to use with *Fontsie*. Then move the ones destined for *Fontsie* back into appropriately identified font files. Even though *Fontsie* the FKey does check for conflicts, you may want to follow the above procedure anyway and thus eliminate the need for *Fontsie* to change IDs.
- Do not attempt to install a font that already appears in the Font menu. The results of doing so are unpredictable and probably undesirable.
- Some applications, such as *MacWrite*, do not show a check mark on the font menu when a *Fontsie*-installed font is selected, and do not show the native (installed) sizes in outlined type. This inconvenience is merely cosmetic; all other font functions should work as expected.
- In a sense, *Fontsie* tricks the application into thinking the uninstalled font or fonts are, in fact, installed. As with any trick, the illusion sometimes fails to work as planned. *Fontsie* has evolved for several years, and as it has, these failures have become extremely rare. But they do still occasionally occur, usually with programs whose willingness to be deceived is limited. Thus it is a good idea to save your work just before invoking *Fontsie*.
- Documents created while *Fontsie* is in effect retain information about the fonts you used. They do not, however, contain the font itself. Thus if you open the document in a subsequent work session but neglect to reload the font with *Fontsie*, your text will appear in a "generic font" such as Geneva, not in the originally selected font. The remedy, of course, is to be sure to reload the font in question with *Fontsie*.

The fact that *Fontsie* the FKey and *Fontsie* the Init work differently means that one will probably be more suited to you and your style of working than the other. The FKey version, *Fontsie* 1.53, comes only in "manual transmission," installs fonts one by one, and can be used at any time during the course of your work. The Init version, *Fontsie* 3.08, is more or less automatic, installs preselected fonts as a group, and must be used at the time you launch an application. The FKey version makes sense if you prefer not to have to decide whether to invoke *Fontsie* before you start your application. On the other hand, the Init version is probably a better choice if you have only a handful of rarely used fonts, know in advance that you will need them, and don't mind adding them all to the font menu just to use a particular one. The following two sections are brief run-downs on the two different versions of *Fontsie.*

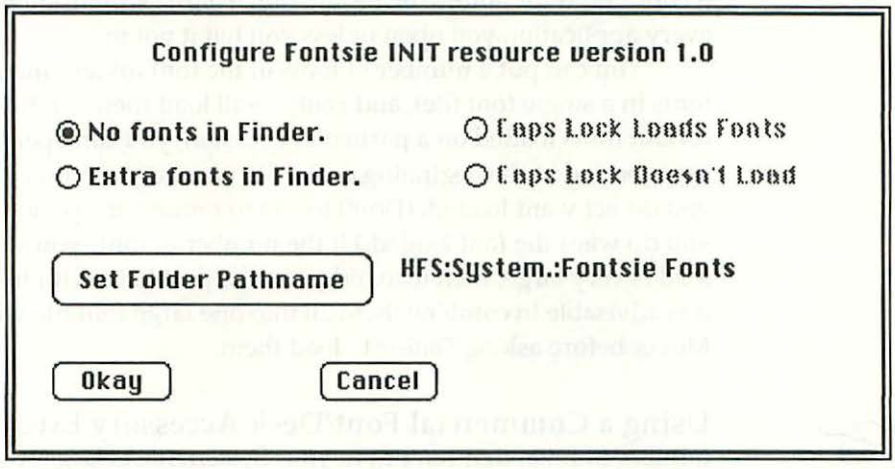
***Fontsie* 1.53: The FKey** *Fontsie* 1.53 is installed like any other FKey (see Chapter 3). The fonts you intend to use it with should be stored one font (any or all sizes) per font file. Just to make sure that all is as it should be, before using *Fontsie* the first time, you should check all the font files you intend to use it with, deleting unwanted sizes and separating different fonts into different files as required. For convenience's sake, your font files should be collected in a single folder, and the folder should be in a location that can be easily remembered and quickly found. Your fonts folder does not need to be in the System folder or even on the System disk—just somewhere handy.

To use this version of *Fontsie,* simply select the *Fontsie* FKey, either through the key combination you have assigned to it or from the *PopKeys* menu, from within any application. *Fontsie* will ask you to find your fonts folder and select a font file. Click on Open and *Fontsie* will install the selected font in the application's font menu for you to use just like any other font until you quit the application. If you want to install two or three or more fonts, simply repeat the procedure as often as necessary. I don't recommend using *Fontsie* to install a large number of fonts at once, however. If you do need to install more than three fonts during a single work session, I suggest you thoroughly test the anticipated combination of application and fonts before using it in your work.

***Fontsie* 3.08: The Init Resource** *Fontsie* 3.08, the Init version, functions more like a commercial font/desk accessory extender (for fonts only). It should be configured before you use it, using the configuration program that accompanies *Fontsie,* and before you start the configuration program, you need to decide where to store the font files *Fontsie* will be using. If you have room to store them in the System folder (if you have a hard disk, in

other words), you need do nothing now. If you have no room in your System folder, you need to create a folder for your font files, give it an appropriate name (such as `Fonts`), and place it where you want it. Also, for now, place at least one font file inside the folder.

Now start the configuration program. In the file selection window that appears, first locate *Fontsie* itself, click on it, and click on the Open button. In the following window, you need to click on one each of two pairs of buttons. One pair determines whether or not you want *Fontsie* to install fonts in the Finder. Turn on **Extra Fonts In Finder** only if you think you'll want to use uninstalled fonts in one or more of your desk accessories while in the Finder. Otherwise, select **No Fonts In Finder.**



Use the Configure Fontsie *program to choose among the options which apply to the Init version of* Fontsie.

The other pair of buttons determines whether you want *Fontsie* to load fonts automatically unless you tell it not to, or not to load fonts until you tell it to. The former option makes sense if you want to use *Fontsie* as a way of adding to your font list on a daily basis. The latter option—*Fontsie* loads fonts only when you tell it to—is a way to keep rarely used fonts off the Font menu, except for those occasions when you need them.

If you plan to put your fonts in a `Fonts` folder inside your System folder, then you need do nothing more. If, however, you want to put your

fonts elsewhere or in a folder with a different name, then click on the **Set Folder Pathname** button and, in the file selection box which follows, locate the font folder you created a minute ago, select the font file within it, and click on the **Open** button. Finally, to save your choices and quit the configuration program, click on the **Okay** button. Copy the Init resource to your System file and restart your Macintosh.

Once configured and installed, the Init version of *Fontsie* is easy to use. You simply place the fonts you want *Fontsie* to install in the designated fonts folder. Then, if you have configured *Fontsie* to install fonts only when you tell it to, when you want *Fontsie* to install your extra fonts, you press the Caps Lock key before launching the application in which you want the fonts to appear, and keep it down until the program is fully launched. If you have it configured for automatic installation, *Fontsie* will install your fonts in every application you open unless you tell it not to.

You can put a number of fonts in the font folder (and a number of fonts in a single font file), and *Fontsie* will load them all. If you only want certain fonts loaded on a particular occasion, you can open the font file and put a period at the beginning of the file name of each font (as in `.Optima`) you do *not* want loaded. (Don't forget to remove the period the next time you do want the font loaded.) If the number of fonts you want *Fontsie* to load is very large, then to avoid potential problems with fussy applications, it is advisable to combine them all into one large font file with Font/DA Mover before asking *Fontsie* to load them.

Using a Commercial Font/Desk Accessory Extender If the number of fonts that can't fit in your System file is larger than the number of files that can, then you should use a font/desk accessory extender such as *Suitcase* or *Font/DA Juggler Plus* to manage your fonts (as well as your desk accessories and FKeys). As we saw in the previous chapter, these two programs are simple yet powerful utilities which can vastly increase the number of desk accessories in the Apple menu with little or no increase in System file size. Almost everything that I said earlier about these two programs in regard to desk accessories applies equally well to their use with fonts.

Suitcase, for example, can automatically add at startup the fonts in appropriately named files or folders to the fonts already installed in your System file. You can also use the *Suitcase* desk accessory to add even more fonts after startup. Although the maximum number of font and desk

accessory files which can be open at one time is ten, a judicious use of Font/DA Mover to collect a large number of fonts into a small number of files can result in an almost unlimited number of immediately accessible fonts. *Font/DA Juggler Plus* also contains a compression utility that can shrink a font file by as much as forty percent (according to the manufacturer). Thus for those who have little disk space to spare (which probably includes most who have yet to spring for a hard disk), *Font/DA Juggler Plus* can load more fonts at startup than *Suitcase* can because its compression feature provides a way of packing more fonts in a limited amount of disk space. Compressed font files are automatically decompressed by *Font/DA Juggler Plus* whenever they are called into use. Keep in mind, however, that font files that have been compressed by *Font/DA Juggler Plus* are not usable if you are operating from a System file in which *Font/DA Juggler Plus* is not installed.

Should you decide to use one of these programs to expand the number of fonts in your font menu, keep the following points in mind:

- Unlike the Apple menu, font lists and menus in most applications do not update themselves automatically if new fonts are added in the middle of a work session. Thus with both *Suitcase* and *Font/DA Juggler Plus*, you should be sure to complete all your font additions at startup, from the Desktop (when you are using the Finder) or if you are using MultiFinder, when all your applications are closed. Otherwise, your font menus will not accurately reflect what fonts are actually available.
- Neither *Suitcase* nor *Font/DA Juggler Plus* attempts to resolve font ID number conflicts (in contrast to the fact that they both resolve potential desk accessory ID number conflicts). Although duplicate font ID numbers rarely cause system errors (crashes), they can create annoying printing glitches: Your document might print in Glypha, for example, even though you selected Garamond. One way to avoid such problems is to employ the jumbo file technique described in the previous chapter and briefly summarized earlier in this chapter. Or, if you own *Font/DA Juggler Plus*, you can use its *Font/DA Utility* to resolve font ID number conflicts.
- When using a font/desk accessory extender, be sure to put all the screen fonts—all sizes and all styles—of a single font family in the same file, either a single font file or the System file. Do not, for example, put the roman screen fonts in one file and the italic and

bold screen fonts in another. If you do split the screen fonts between several files, only the sizes and styles in the file which was loaded last will be available to you.

- Remember that with both *Suitcase* and *Font/DA Juggler Plus* you can combine your desk accessory and font strategies. Both fonts and desk accessories can be installed in a single font or desk accessory file. To open a font file while in desk accessory mode in Font/DA Mover (or vice versa), hold down the Option key when clicking on the Open button, then add desk accessories as usual.

Space-Saving Strategies for Downloadable LaserWriter Fonts

If you've worked at all with downloadable fonts, you've probably noticed that the large number of files contributed by just a few font families can present some difficult space problems. They quickly consume, for example, immoderate amounts of disk space. On a floppy-disk-based system, you can run out of room for PostScript font files on your System disk after installing only one or two font families. As for the LaserWriter, it has an unreasonably low limit to the number of downloadable fonts it can accommodate at any one time. If you attempt to load more than the legal limit, the LaserWriter may respond with a firm "no," or worse, reset its memory, purging all the currently loaded fonts, prep files, and partially printed documents.

None of these problems has an entirely satisfactory solution, but there are ways to work around them. The following are a few suggestions for those of you who own more than one or two downloadable fonts, or someday hope to.

On Disk If your font menu is loaded with downloadable fonts, you undoubtedly have quite a stack of PostScript font files to go with them. This is partly because, as explained above, each font on the menu may have as many as four corresponding PostScript files: one each for roman, italic, bold, and bold-italic faces. Another factor is that you can trim the excess fat off the screen fonts by deleting unneeded styles and sizes, but there's no way you can trim a PostScript font file, short of dropping it altogether. Thus a single screen font, trimmed down to only two sizes and, for example, 7.5K, might have associated with it 120K or more in PostScript files. It doesn't take much imagination to see how fast numbers like these can add up.

Those of you with hard disks may take a brief moment to feel smug about your hardware's ability to absorb large numbers of downloadable font files. A well-stocked, hard-disk System folder can exceed one megabyte (1024K) relatively easily, even before considering any PostScript font files. Add half a dozen or so downloadable fonts and all their associated files and your System folder can come close to doubling in size. A baker's dozen might well produce a three-megabyte System folder. Clearly, the total space required by System files, screen fonts, and PostScript files can easily amount to more than an 800K disk, or even a 1600K disk, can handle. So if you must have a variety of PostScript fonts readily available, a hard disk is a virtual necessity.

But owners of floppy-disk-based systems need not despair altogether. There are a few strategies you can use to increase the number of downloadable fonts available to you. None are likely to win any awards, but you may find that one or another is an acceptable solution to a difficult problem.

- You could create not one but several System disks, each with a different set of downloadable fonts. The fonts on each disk could be organized by application, by project, or by a combination of both, depending on how you use your downloadable fonts. The drawback to this technique is that you have to change System files every time you change applications or projects. If you use a RAM disk, RAM cache, or print spooler (more on these topics in Chapter 7), changing System files can be a hassle. Only you can decide if such inconveniences are an acceptable price to pay for the added number of fonts this scheme puts at your disposal.
- If you find that you never use the bold-italic style combination (I almost never do), you can remove the bold-italic PostScript file from your System disk and the bold-italic screen fonts from your System. Far more drastic is the elimination of either the bold or italic, or most drastic of all, both bold and italic files as well. Perhaps you never use one or the other style. Or you may use one (or both) so infrequently that only the most careful observer would notice that that rare piece of italic text is a slanted version of the roman font and not true italic.

  It's your printed page, so only you can decide if you are willing to compromise in this way. I know it sounds odd to recommend something which decreases (even if only slightly) the quality of a LaserWriter font, since print quality is the primary reason for

using a LaserWriter. But if you need the disk space for other fonts, it is worth your time to experiment and see what you think of the results.

- The most reliable place to put your PostScript font files is in the System folder. But placing them in the folder which contains the currently running application sometimes also works. I say sometimes because, well, sometimes it works and sometimes it doesn't. The reason for this is unclear to me (although it is probably clear to someone). So your best bet is simply to experiment and see if it will work for you and the combination of application and fonts you are working with. As you experiment, there's no need to actually print a sample on paper. Just start the printing process and keep your eye on the LaserWriter information box at the top of the screen. If the Mac reports that the PostScript file for the font in question cannot be located, you know that the experiment has failed. Once you know the outcome of the experiment, press Command-period to halt the printing process.

In the LaserWriter Another concern created by a proliferation of PostScript font files and their frequently large sizes involves the LaserWriter. Both LaserWriters devote only a small fraction of their RAM to downloadable font storage. You may have as much as 130K of RAM at your disposal (if you have a LaserWriter Plus and are printing with "standard" applications that use the Laser Prep file), or as little as 40K (if you have a LaserWriter and are printing with both "standard" programs and *PageMaker* during the same session). Since PostScript fonts can range from 25K to over 60K per face (they average about 40K), the LaserWriter can accommodate three average-sized faces at best, and none at all at worst.

What to do? Well, don't do anything if your LaserWriter has never complained of an overloaded memory. If it has complained, the following are some strategies you can use to get around the limitations imposed by a shortage of RAM in the LaserWriter:

- If you have a LaserWriter, you might consider upgrading it to a LaserWriter Plus. The Plus is faster for one thing, but more to the point, it is more efficient in the way it utilizes RAM. The difference can be significant: by upgrading, you can, under certain circumstances, add as much as 40K to the RAM available for storing downloadable fonts.

- Most programs copy the Laser Prep file to the LaserWriter's RAM before they print. As noted earlier, *PageMaker* has its own prep file, Aldus Prep, which it copies to the LaserWriter instead. If you are printing with *PageMaker* and other programs as well, both prep files will be copied to the LaserWriter's RAM and stored there until you turn it off or reset it. Since only one prep file is used by a given application, the other prep file sits idle and occupies space that downloadable fonts might otherwise have used.

  There are two ways to get around this problem. One is to confine your use of *PageMaker* to sessions in which you use only *PageMaker* to print. The other is to reset the LaserWriter each time you switch print jobs from a "standard" application to *PageMaker*, or from *PageMaker* to a "standard" application. You can reset the LaserWriter by turning it off and then on again, or with a desk accessory such as *LaserStatus*, which has a LaserWriter reset command which works entirely through software.

- Be aware of your use of boldface and italics. Remember that although you may have selected only one font in the font menu, you may actually be using two or three or even four different PostScript files, one for each style. If you carefully observe the reason why your LaserWriter RAM is overloading and determine that it is, in fact, because of your use of styled text, there are several possible remedies to your problem. One, of course, is to stop using italics or bold text. Another is to delete the bold and italic PostScript files from your System folder and force the LaserWriter to create them from the roman style (as in the disk space suggestion made in the previous section). Neither remedy is entirely palatable, but they are better than not being able to print at all.

Customizing Fonts

Font manufacturers are pretty good about responding to the needs of their customers, even when those needs are highly specialized. There are, for example, some rather peculiar fonts available; fonts which only a very few could find useful. Nonetheless, you may one day discover a need that

cannot be fulfilled with existing fonts. Or, after working with a font for a while, you might want to fine-tune it to your own particular needs, which may be one-of-a-kind.

To attempt to cover all the ways in which one could improve upon existing fonts is beyond the scope of this book. Indeed, one could fill a whole volume with advice on creating new fonts and customizing old ones. But from among the many things such a book would cover, I close this chapter with two that apply particularly well to the fully powered Macintosh: One is the altering of screen font names to reorder the font menu and to make the bold, italic, and bold-italic screen fonts disappear from the font menu. The other is the creation of fractions for those PostScript fonts which lack them.

Changing Screen Font Names—and Why

Have you ever wished that the fonts you used most often were the ones at the top of the font menu? Or that you could group your fonts by type—first text fonts, then display fonts, and then symbol and picture fonts? Or have you ever noticed that, for some reason (known only to font manufacturers, apparently), the different style screen fonts of a PostScript typeface are named in such a way that they never end up together on the Font menu (Garamond roman, bold, italic, and bold-italic are not listed next to each other, for example). Instead, all the bold versions of a collection of fonts are in one clump (because the names all begin with B), all the italic in another (because their names all begin with I), and so on. Wouldn't it be nice if you could put all the faces of the Garamond family together in the Font menu?

The fact is, you can reorder the fonts on the Font menu, and it's not as hard as you might think. All you have to do is change the name of the font (the screen font only) so that the sort which orders the font menu arranges your fonts the way you want them—in much the same way that you altered file and folder names in Chapter 2 to change their order in file selection boxes.

There is another use to which you can put font renaming as well. This is to mark all styled screen fonts (such as Garamond Italic and Garamond Bold) in such a way that they disappear, and only the roman font (plain Garamond) appears on the Font menu. This is easily done, simply by

adding a period or a percent sign as a prefix to the font's name. As I noted earlier, if the font is a "smart" one (and most are nowadays), the italic screen font will be displayed on the screen (and the PostScript version printed as well, of course) if you use the Italic Style command on the roman screen font, even though the italic screen font has been altered so that it does not appear in the Font menu. Having the styled screen fonts listed on the Font menu is thus unnecessary. Indeed, because long lists of styled screen fonts can make the roman font difficult to find, listing the styled screen fonts in the Font menu can be a distinct disadvantage.

For example, assume that Garamond Italic was renamed .Garamond Italic and then installed in the System file. If you apply the Italic style command to Garamond, the Garamond screen font changes to the Garamond italic screen font, even though the Garamond italic screen font is not listed as a separate font in the Font menu. Of course, making a font name invisible in this way does nothing to decrease the disk space required by your font files. But making styled fonts invisible is a simple and extremely effective way of cleaning up and shortening cluttered and overly long Font menus.

Unfortunately, there is one minor glitch in the font disappearance act. This is the fact that a few programs, such as *PageMaker*, for example, deal with fonts in nonstandard ways and continue to display fonts which are marked as invisible. Suffering through the Font menus or list boxes of these few programs, however, is a small price to pay for shorter and neater Font menus in your other applications.

In the pages that follow, we look at two different programs with which you can change the names of your screen fonts. One uses the commercial bitmap font editor *FONTastic Plus*; the other uses the Apple programming tool called *ResEdit*. I use the first program to illustrate how to rename a font so that it rises to the top of the Font menu. I use the second program to illustrate how to rename a styled screen font so that it disappears. The type of illustration used in each case is of no particular consequence; both programs can be used for both purposes.

But first, no matter which program you use or what reason you have for renaming a screen font, it is important to keep the following points in mind:

- Since you will be tinkering with something of value, and it is possible that you will make a mistake, *do all your tinkering on a*

copy, never on an original, and put the copy you are tinkering with on a floppy disk, even if you have a hard disk. Failure to follow these two caveats may result in the irredeemable destruction of your font—or worse.

- You can work on your fonts either before or after they are installed in your System file. If you choose to work on the fonts after they are installed in the System file, do not work on the currently active System file. Make a copy and work on the copy. When you are finished, replace the old System file with the new one. (To do so, restart your Mac and run it from the altered System file or from a third System file. Replace the old System file with the new one, and then restart the Mac again to activate the newly altered System file.)
- An old font with a new name will not be recognized by your old documents. If you rename New York as !New York, for example, all documents formatted in New York will fail to appear in !New York. If you want to see them again in New York (or I should say, !New York), you will have to reformat them under the new font name. (The only exceptions to this rule are fonts which you prefix with a period or percent sign in order to make them disappear.)

Note that if your renamed fonts are the screen fonts of a PostScript typeface, they should print correctly on the LaserWriter even if the correct font does not appear on the screen. This is because font identification for printing is accomplished not by the font's name, but by its ID number, which remains unchanged throughout the procedures described in the following sections.

By the way, if you own a copy of *Font/DA Juggler Plus*, you can skip the procedures described below and use that program's *Font/DA Utility* to rename your fonts safely and easily.

Reordering Fonts With *FONTastic Plus* Before you attempt to reorder your fonts by renaming them, you should decide just how you want them arranged on the font menu and determine what name changes will be required to get them in that order. Try to accomplish your rearrangement in as few name changes as possible; you might think of this exercise as something similar to those puzzles where you have to get from A to B by passing through the smallest number of intermediate points.

For example, if you want to put the fonts you use most frequently at the top of the list, then you could start the names of these fonts with an exclamation point, a dollar sign, a comma, a space, or an asterisk. If you want to send certain fonts to the bottom of the list, you could start their names with a tilde, a bullet, or a diamond. If you want to put your fonts into two or three subgroups (such as text, display, and special fonts), then you could prefix the first group with an exclamation point, leave the second group alone, and prefix the third group with a diamond.

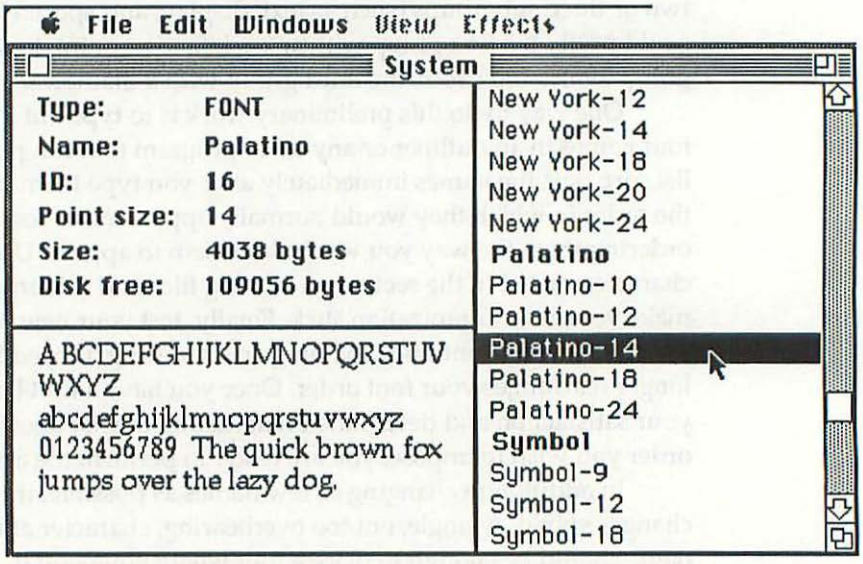
One way to do this preliminary work is to type out a list of all your font names in an outliner or any other program that can perform a simple list sort. Sort the names immediately after you type them in to put them in the order in which they would normally appear. Now rearrange them, ordering them the way you would like them to appear. Use the special characters listed in the section on naming files and folders in Chapter 2 to make your new organization stick. Finally, test your new names by sorting them again; and continue renaming and resorting, if need be, until a sort no longer rearranges your font order. Once you have a font list organized to your satisfaction and determined that sorting the list does not undo the order you wish to impose, you are ready to perform the actual changes.

In addition to changing as few names as possible, try to keep your changes subtle. A single, not too overbearing, character at the beginning of a name should be enough to place a font where you want it. If you have a choice, use characters which don't draw attention to themselves, such as a tilde, a space, or a comma. Avoid changing names altogether to get them to change position. Remember, being able to recognize your fonts quickly and easily is still a primary concern.

In the following example, we use *FONTastic Plus*, a bitmap font editor from Altsys. With it you can change existing bitmap or screen fonts in almost any fashion or create new ones. For our purposes, however, all we want to do is change a font name. The procedure for changing a font's name with *FONTastic Plus* is:

1. Launch *FONTastic Plus*. The program will open to a file selection box. Locate the file containing the font you want to rename. Double-click on the file name or click once and then click on the **Open** button. If the program responds by asking if you want to update the file to the *FONTastic Plus* version, click on the **Yes** button.

2. The File window that now appears contains a list on the right of all the fonts within the file just opened. Each font name will probably be listed a number of times, once in boldface, and one or more times in plain text.

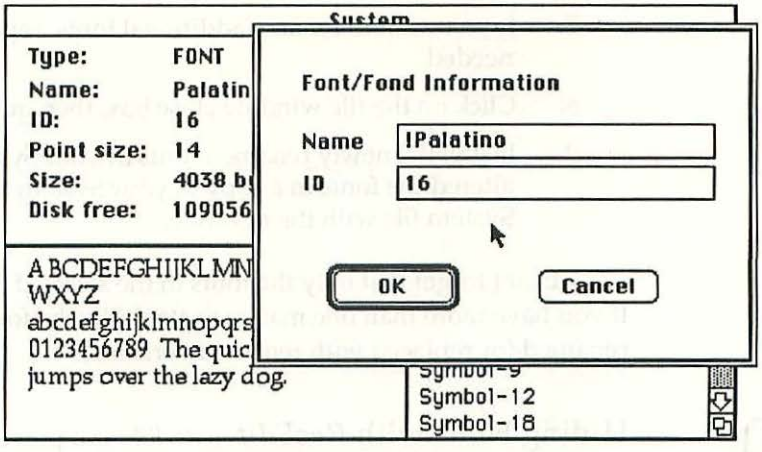


FONTastic Plus' *file window (in this case, the one for the System file on the work disk) displays a list of all the fonts in the file.*

3. Select (click on) the name of the font you want to revise. If the name appears more than once, click on any one of the names displayed in plain text and followed by a number; which one you choose does not matter.

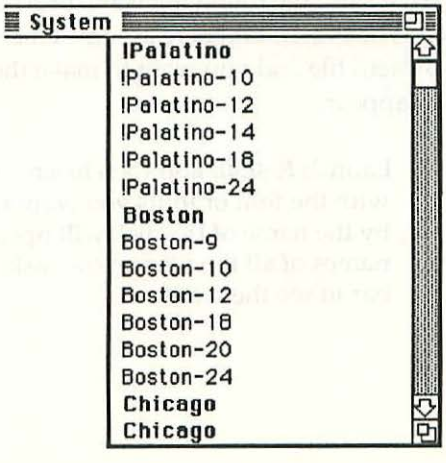
4. Select the Font Attributes command from the File menu or press Command-I.

5. In the resulting dialog box, insert the desired prefix in front of the font name. For example, put an exclamation point in front of *Palatino* to send the font to the top of the Font menu. *Be careful not to change the font's ID number.* Click on the OK button to close the dialog box. Note that all plain-text listings of the selected font (if there are more than one) display the new name.



In the Font Attributes dialog box, insert the desired prefix in front of the font name.

6. If there is also a boldface listing of the font, perform the procedure in the previous paragraphs on this entry as well: scroll to locate the boldface listing if necessary, select it, invoke the Font Attributes command, and change the font name *exactly as before*. If you placed an exclamation point in front of the name in step 5, do so again here. Now click on the OK button to close the dialog box.



Your renamed font moves immediately to the top of the file window font list.

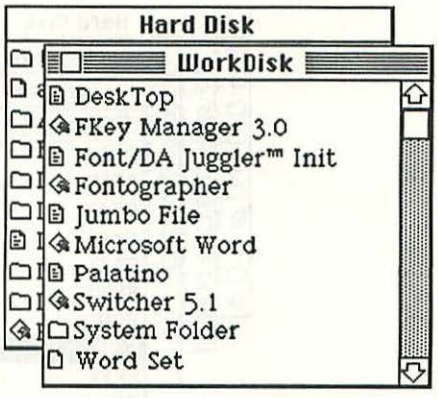
7. If you want to rename additional fonts, repeat steps 3 through 6 as needed.
8. Click on the file window close box, then quit *FONTastic Plus*.
9. Install the newly renamed fonts in your System file or, if you altered the fonts in a copy of your System file, replace the original System file with the new one.

Don't forget that only the fonts in the selected file have been renamed. If you have more than one master System file, the fonts in each file must be renamed (or replaced with renamed versions).

Hiding Fonts with *ResEdit* *ResEdit* is a powerful programmer's tool created by Apple. Since *ResEdit* is discussed more extensively in the next chapter, that is where the section containing comments about and advice concerning the program has been placed. Thus if you want to use *ResEdit* instead of *FONTastic Plus* to change font names, I strongly suggest that you turn to the introduction to *ResEdit* in Chapter 5 before proceeding. For those who like to take chances, suffice it to say here that *ResEdit* is a useful program which permits easy access to the internal workings of most any program. With *ResEdit*, you have the power to alter a program in ways more to your liking and the power to make a mess of everything. So be careful, and be sure to *work only on copies of programs, never on the original.*

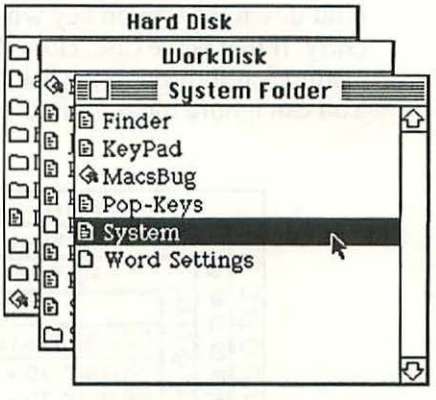
That said, the following is the procedure for changing font names with *ResEdit*. The illustration here assumes that you are working on (a copy of) your System file and you want to make the italic screen font of New Baskerville disappear.

1. Launch *ResEdit* and then insert the floppy disk containing the file with the font or fonts you want to rename. A window identified by the name of the disk will open; listed in the window will be the names of all the files on the disk. You may need to use the scroll bar to see them all.



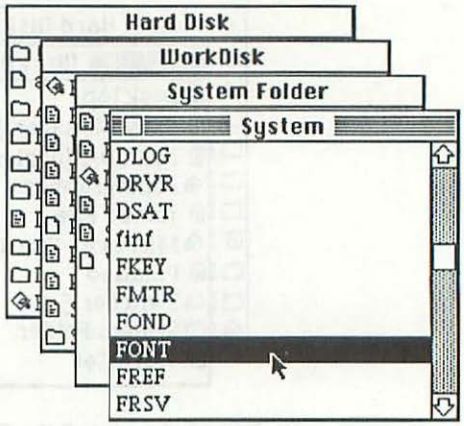
ResEdit *windows list all the files on all mounted disks.*

2. Double-click on the folders needed to locate the file entry in question, and then double-click on the file entry to open that. In this example, you double-click on the System file entry.



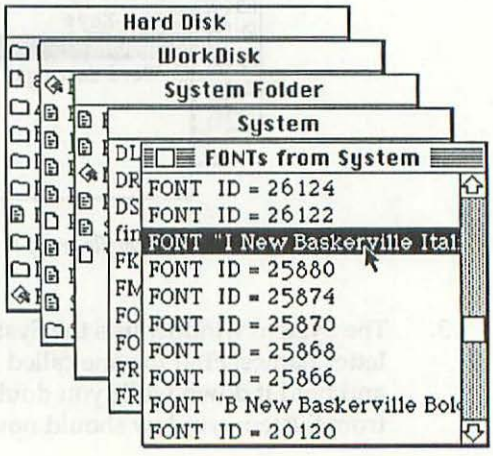
Double-click on the System file entry to open the System file window.

3. The System window lists the System's resources by their four-letter names. Find the one called FONT. Now press the Option key and hold it down while you double-click on FONT. The FONTs from System window should now appear.



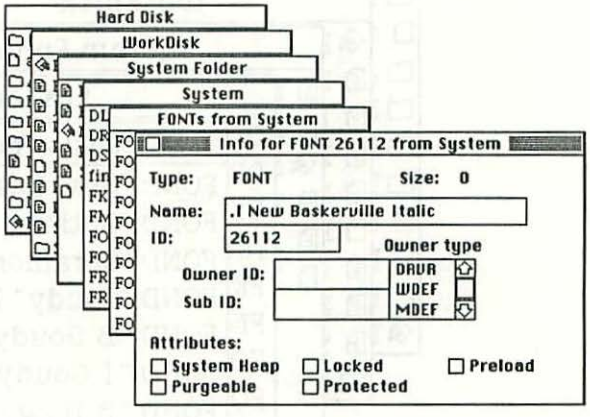
In the System file window, locate the FONT resource.

4. In the FONTs from System window, you should see a list of resources, only some of which have names as well as ID numbers. (If every line contains the name of a font, you probably forgot to hold down the Option key when you double-clicked on the file entry. If that is the case, close the window, return to step 3 and try again.) Scroll through the list to locate the font that is of interest. You can ignore the resources that lack names.



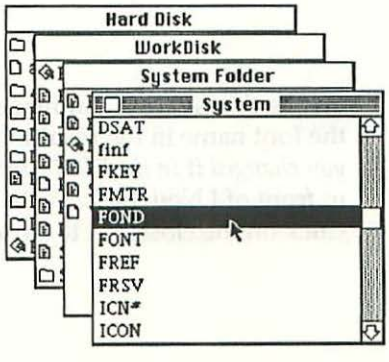
In the FONT window, find the font you want to hide and select it.

5. Select the resource with the font name you want to change by clicking on it once. Then choose the Get Info command from the File menu (or press Command-I). In the resulting FONT Info box, insert a period in front of the font name in the Name text entry box. In this example, put a period in front of I New Baskerville Ital. *Do not change anything else.* Close both the FONT Info window and the FONTs from System window.



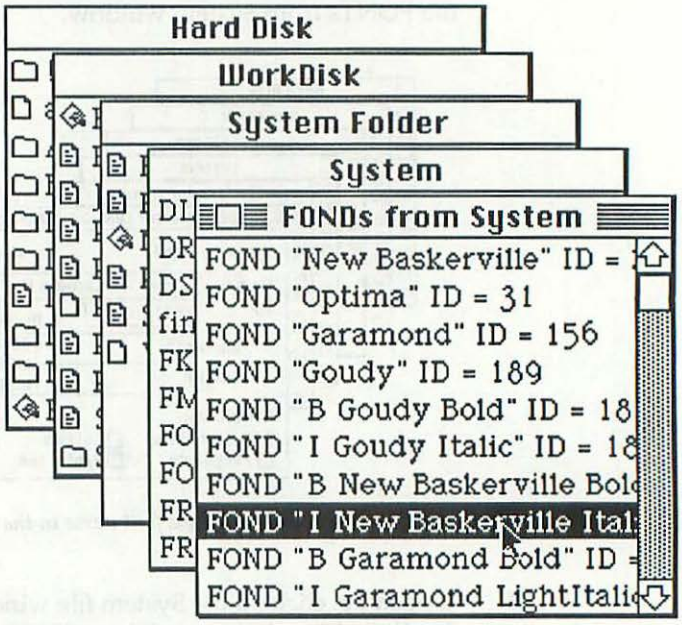
Insert a period in front of the font name in the FONT Info window.

6. If need be, click on the System file window to bring it to the front. In all likelihood, there will be a FOND resource just above the FONT resource selected previously.



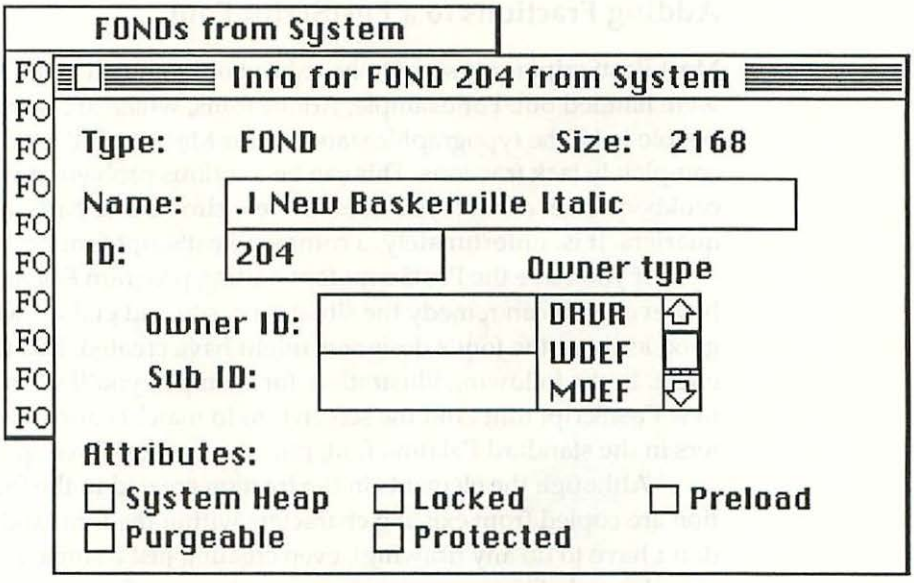
Go back to the System file window and find the FOND resource.

7. Double-click on FOND to open the FOND window (do not press the Option key this time) and select the name of the font you just changed in step 5.



In the FOND window, locate the font name you just altered in the FONT window.

8. Invoke the Get Info command again (press Command-I). Change the font name in the resulting FOND Info box, *exactly the same way you changed it in the FONT Info box.* In our example, insert a period in front of I New Baskerville Ital. Do not change anything else. Click on the close box to close the FOND Info window.



Change the name in the FOND Info window to exactly match the newly changed name in the FONT Info window.

9. If you want to change more than one font name in this file, repeat this process as needed, starting with step 3.

10. Recheck both FOND and FONT windows to make sure that the names you have altered are precisely the same in both windows. Now close the file window—in this case, the System window. You will be asked if you want to save your changes. Click on Yes or press Return.

Replace your old System file with this newly altered one, or install your renamed fonts with Font/DA Mover if you worked on a font file. That's all there is to it.

Adding Fractions to a PostScript Font

Most PostScript fonts seem to have been out to lunch when the fractions were handed out. For example, Adobe fonts, which are held by many people to be the typographic standard for Macintosh PostScript fonts, completely lack fractions. This can be a serious problem if you are printing a cookbook or even if all you need is a few simple one-halfs and three-quarters. It is, unfortunately, a common PostScript font problem.

If you have the PostScript font editing program *Fontographer* (Altsys), however, you can remedy the situation easily and end up with fractions as good as those the font's designers might have created, had they made the effort. In the following illustration, for example, you'll see how to create a new PostScript font (and the screen font to match) containing all the characters in the standard Palatino font, plus the fraction three-quarters.

Although the elements in the fraction created in the following illustration are copied from existing characters within the font (and as a result, we don't have to do any drawing), even creating just a single fraction is a rather complex task. Of course, creating more than one fraction is even more time consuming. Clearly, efforts like these are serious business, and should only be undertaken when the rewards justify the amount of work involved; in other words, when you really need fractions or whatever it is you are adding to your font repertoire. On the other hand, you don't have to limit yourself to fractions in endeavors like these; you can add any kind of special characters, symbols, or graphics, the company logo, or even your initials or facsimile signature.

Readers who have no need of fractions, who do not own *Fontographer*, or who have no interest in adding special characters of their own design to existing PostScript fonts, are hereby warned that the following illustration is rather lengthy and detailed (sorry but there's no other way to get the job done). Such readers may want to skim or skip the remainder of this chapter and head directly for the first page of Chapter 5.

Readers who wish to follow along in this final exercise will need a copy of *Fontographer* and the roman screen font for Palatino (or you can improvise with a different font). The following are the steps for creating a new version of Palatino roman that contains one additional character, the fraction three-quarters:

1. Prepare a work disk containing a skeleton System folder (bare-bones System file, Finder, LaserWriter, and LaserWriter prep files),

Fontographer, Font/DA Mover, and the Palatino screen font file. If you have a copy of the Palatino roman AFM (Adobe Font Metrics) file, copy that to the work disk as well; it will save you some time, but it is not essential.

a. Restart your Macintosh so that the work disk is the startup disk.

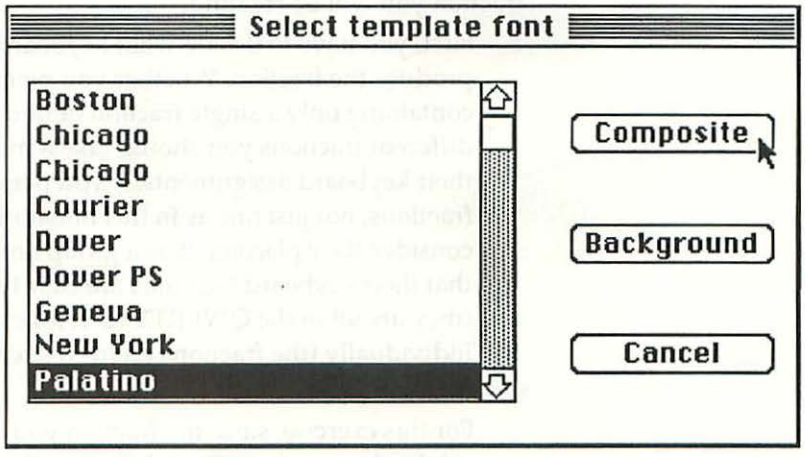
b. Install the entire set of screen fonts for Palatino roman in the work disk's System file using Font/DA Mover and the usual font installation procedure.

c. Put Font/DA Mover in the trash if you are short of disk space.

2. Launch *Fontographer*. When the opening file selection box appears, click on the Cancel button to make it go away.

3. Select **Open Template...** from the File menu.

a. In the file selection box which results, select Palatino and then click on the Composite button.



Select the font you want to work with and click on the Composite button.

b. Do not change any of the figures in the Font Attributes dialog box, which now appears. You might note, however, that Fontographer has already chosen an ID number for the

composite font that is different from the ID number of the original Palatino file (the new font ID number appears at the lower left).

c. Click on the OK button in the Font Attributes dialog box.

d. If *Fontographer* states that it cannot locate an AFM file corresponding to the selected font, click on the OK button to proceed.

4. The *Fontographer* font window now opens, displaying the standard ASCII character assignments.

   a. To save Fontographer's efforts so far, select either **Save** or **Save As...** from the File menu.

   b. Enter a name for your new font. If you aren't familiar with PostScript font naming rules (which are a bit fussy), or you can't think of anything better, name the font `PalatFrac`.

   c. Click on the Save button or press Return.

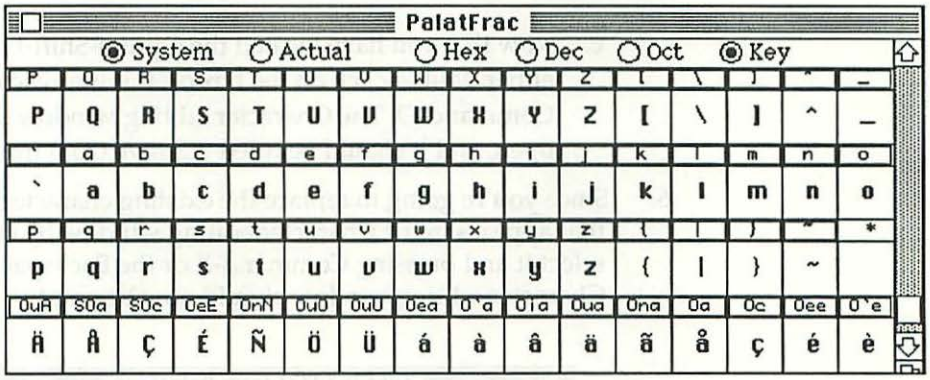
5. Now you need to open the Character editing window for the fraction you will be creating.

   a. First, you have to decide what keyboard combination will produce the fraction. Whether you plan to create a font containing only a single fraction or one containing many different fractions you should take a moment or two to check their keyboard assignments. If you plan to add a number of fractions, not just one as in this illustration, you should consider their placement as a group and try to assign them so that their keyboard locations are easy to remember collectively (they are all in the QWERTY row, for example) as well as individually (the fractions go from small to large, left to right, for example).

      For this exercise, since the fraction you are creating is 3/4, you might choose a key about three fourths of the way across the QWERTY row, such as the I key, thus leaving room for other, smaller or larger fractions to be created at some later date. Add Option-Shift to the combination to avoid eliminating an important character, and thus assign your fraction to Option-Shift-I.

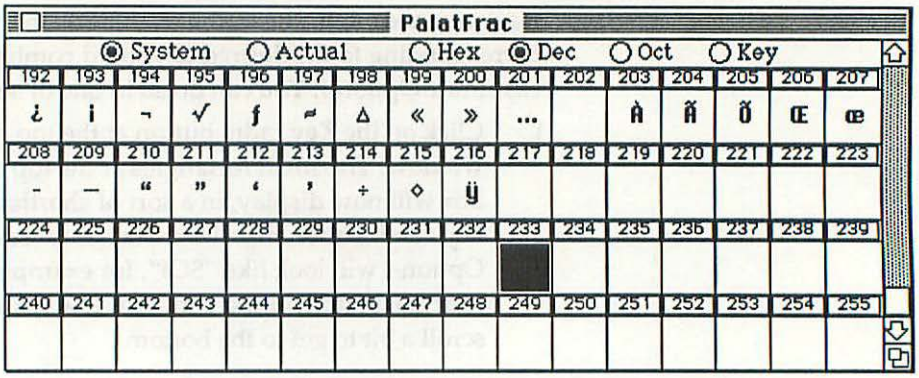
b. Next you need to locate the character box in the Font window corresponding to the desired keyboard combination, in this case Shift-Option-I. You can do so in one of several ways:

   i. Click on the Key radio button at the top of the Font window. The small rectangles at the top of each character box will now display, in a sort of shorthand, the key or key combination that the character is assigned to. Shift-Option-I will look like "SOi", for example. You'll find it near the bottom of the Font window (you may have to scroll a bit to get to the bottom).



Click the Key radio button at the top of the Font window to label each character with a code indicating the corresponding key combination.

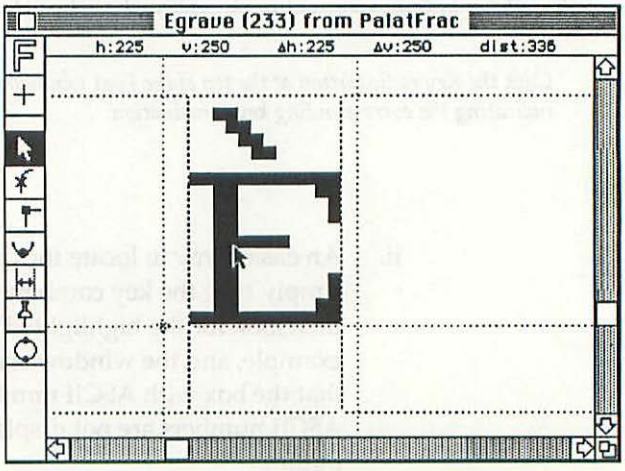
   ii. An easier way to locate the correct character box is to simply type the key combination you have in mind and then look for the highlighted box. Type Shift-Option-I, for example, and the window scrolls (if need be) to reveal that the box with ASCII number 233 is highlighted. (If ASCII numbers are not displayed, click on the Dec radio button.)



Typing the desired key combination will highlight the corresponding character box.

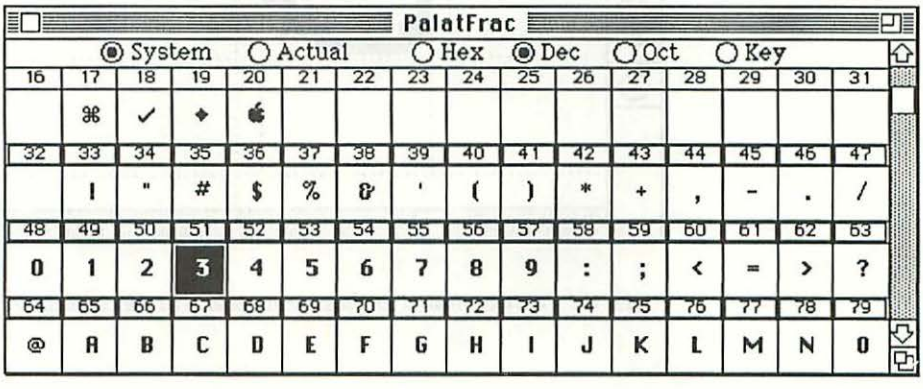
c. Now that you have located the Option-Shift-I character box, either double-click on the box or, if it is already selected, press Command-O. The Character editing window should now open, and it should be titled "Egrave (233) from PalatFrac".

6. Since you're going to replace the existing character, erase the one that appears in the Character editing window by clicking on it to select it and pressing Command-X or the Backspace key. The Character editing window should now be empty.



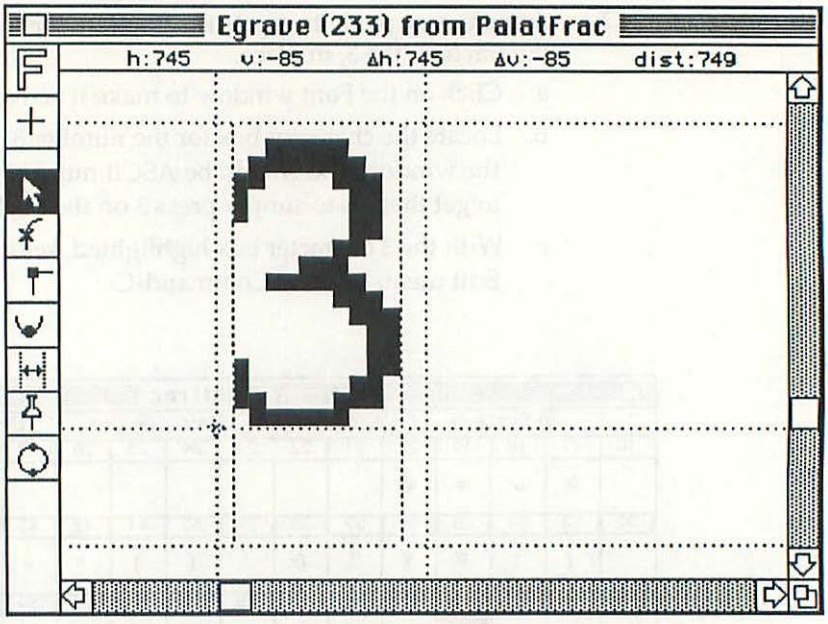
Select the character—the two shorter vertical lines indicate that it is selected—and press Command-X to delete it.

7. You are now ready to create the fraction. First, put the top part of the fraction, the 3, in place.
   a. Click on the Font window to make it active again.
   b. Locate the character box for the number 3. It's near the top of the window and should be ASCII number 51. The easiest way to get there is to simply press 3 on the keyboard.
   c. With the 3 character box highlighted, select **Copy** from the Edit menu or press Command-C.



With the number 3 character box selected, invoke the ***Copy*** command.

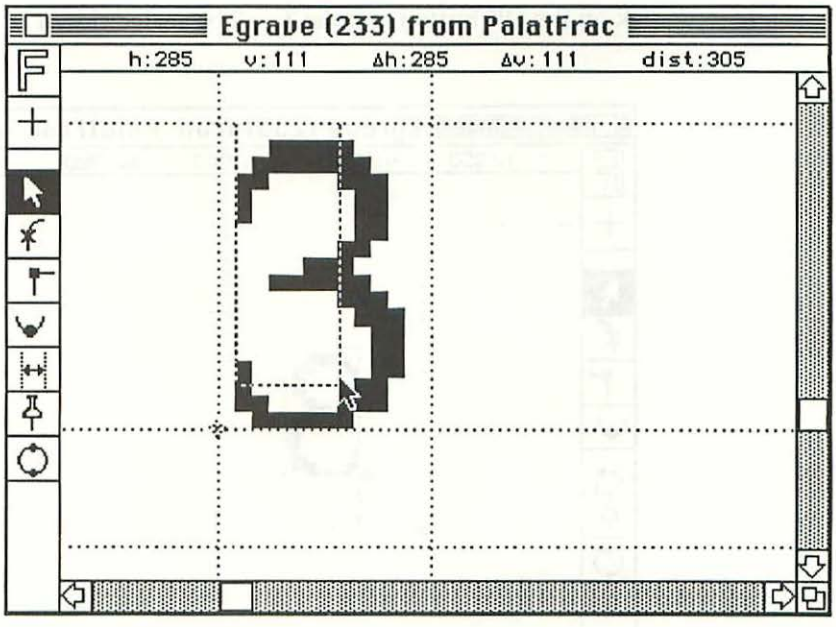
   d. Click on the Character editing window to make it active again.
   e. Select **Paste** from the Edit menu or press Command-V. Although it is clearly too large for a fraction, the number 3 should now be in the Character editing window. What you see on the screen, by the way, is just a crude bitmap representation of the selected character. When you print, the PostScript description will be used to create the expected, high-quality character. (See graphic next page.)



*Use the **Paste** command to place the number 3 in the character editing window.*

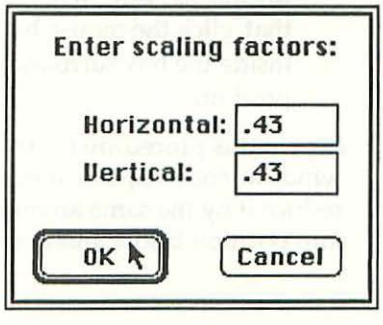
f. Since the 3 is too big, you need to reduce it to a more appropriate size. There are two ways you can do this:

   i. With the 3 still selected (it will remain selected after it is pasted in, unless you click the mouse while the pointer is elsewhere in the window), position the pointer over the lower right corner of the box surrounding the 3, click and hold the mouse button down, press the Shift key and then drag the pointer up and to the left. (The order in which these actions are done *is* important). Release the mouse button to see just how small the 3 has become. Repeat the procedure until the 3 appears to be the right size (between one-half and one-third of its original size).



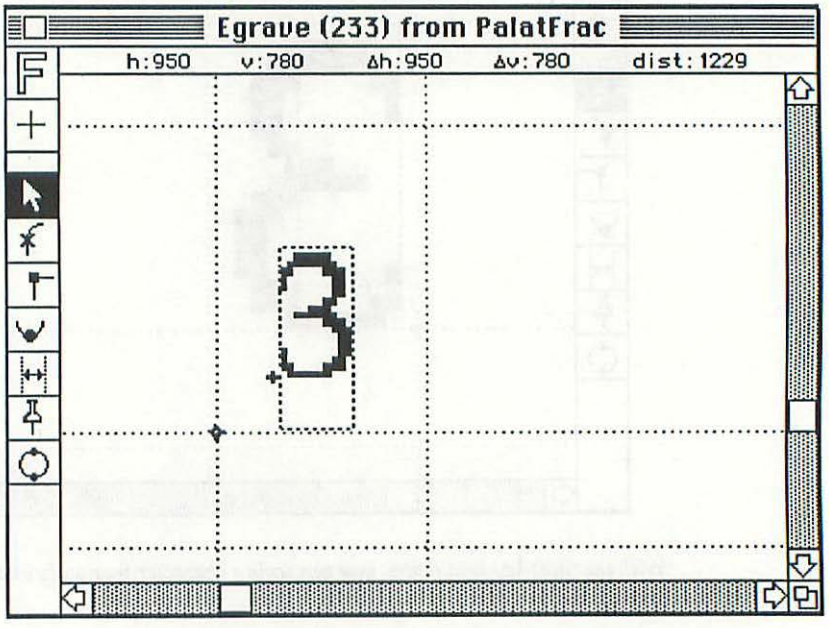
With the Shift key held down, you can scale a character manually and maintain its original proportions.

ii. A more precise method is to select **Scale...** from the Effects menu. In the dialog box which results, enter the same factors (something between .50 and .33) for both horizontal and vertical scaling.



Select the Scale... command and enter the desired scaling factors.

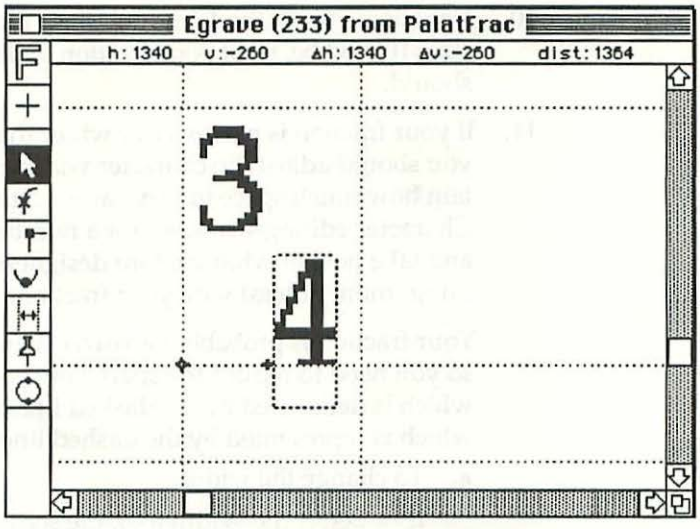
Click on the OK button and the 3 will be scaled down as requested.



Click on the OK button and Fontographer *will reduce the character as requested.*

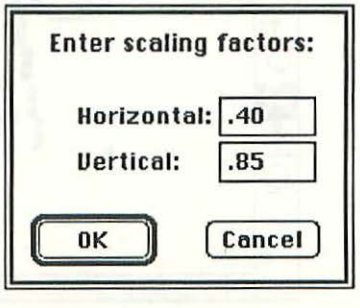
g. Move the 3 to the upper left corner of the square formed by the dashed lines within the Character editing window. To do that, click the mouse button while the pointer is anywhere inside the box surrounding the 3 and drag the 3 to the desired position.

8. Repeat this procedure for the number 4: select it in the Font window, copy it, paste it into the Character editing window, and reduce it by the same amount that you reduced the 3. Drag the 4 into position below and to the right of the 3.



With the 3 and the 4 in place, it's time to add the slash between them.

9. Once again, repeat this procedure for the slash (ASCII character 47) which will separate the 3 and the 4. This time, however, you may not want to scale the slash uniformly. Instead, you might want to make it thinner, but keep it as long or almost as long as the original. To accomplish such asymmetrical scaling, either do not press the Shift key when you drag the mouse, or enter two different numbers in the Scale dialog box (depending on which scaling method you use). Experiment a little to get the slash the way you want it. Finally, drag the slash into position between the 3 and the 4.



You can scale the slash asymmetrically by entering different scaling factors in the Scale dialog box.

10. Now jiggle the 3, the 4, and the slash around a bit, and adjust their sizes if need be, until your fraction looks the way a fraction should.

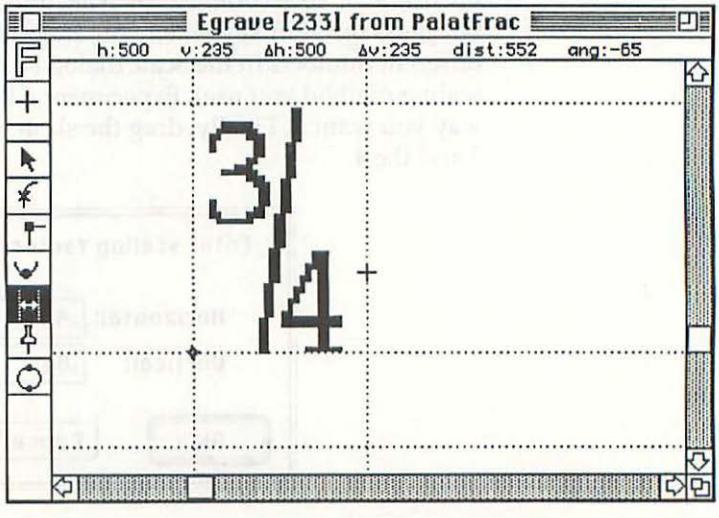
11. If your fraction is narrower or wider than the character it replaced, you should adjust the character width and offset. If you're uncertain how much space to leave around the fraction, open the Character editing windows for a number of different characters and take note of what the font designers did. Then do the same thing (more or less) with your fraction.

    Your fraction is probably narrower than the character it replaced, so you need to reduce the space between the character width, which is delineated by the dashed line to the right, and the offset, which is represented by the dashed line on the left.

    a. To change the width:

        i. Select the Width tool, the second-to-last tool in the tool palette, the one that looks like a double arrow between two dashed lines.

        ii. Using the Width tool, drag the dashed width line (the one on the *right* side of the character) closer, or simply click where you would like the width line to be.



With the Width tool, you can adjust the width line on the right side of the fraction.

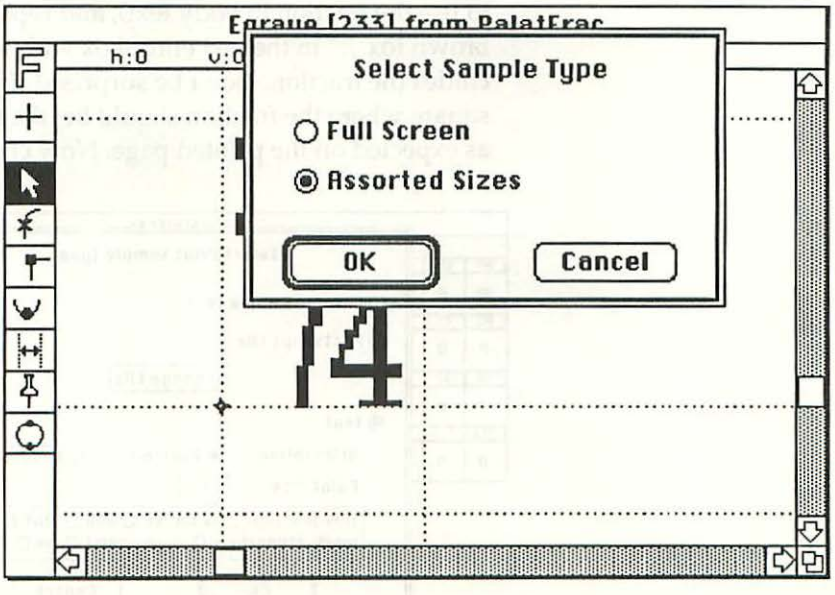
b. If the offset line is too close to the left edge of the fraction or too far away, you'll have to drag the fraction to get the line where you want it in relation to the new character; the offset line does not move. To move the entire fraction at once:

   i. Shift-click on each individual part (or choose the **Select All** command) to select them all.

   ii. Click on any piece of the whole and drag the entire fraction to the correct position in relation to the offset line.

12. Now comes the day of reckoning—the test printout.

   a. Make sure your LaserWriter is turned on, that the Character editing window is active, and then select **Print Sample...** from the File menu. In the Print Sample dialog box, click on the **Assorted Sizes** radio button. Click on the OK button or press Return.



*When the Character Editing window is active, the **Print Sample...** command will display this dialog box. Click on the **Assorted Sizes** radio button, then on the OK button.*

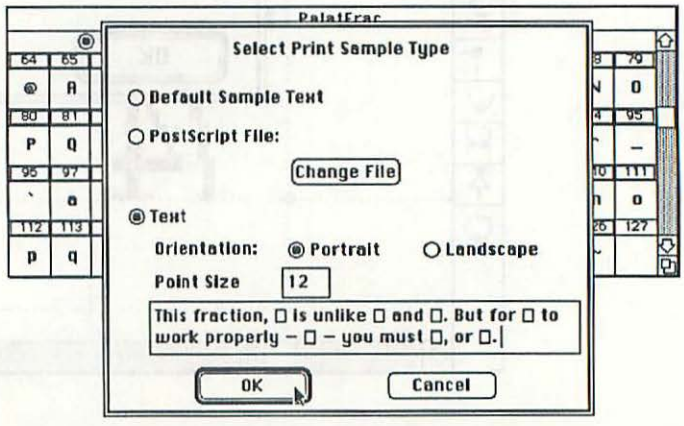
b. The LaserWriter will print about 18 different sample sizes of the fraction, ranging from the exceedingly small to the implausibly large. Look at them carefully to see if the three parts of the fraction need any adjustment in spacing or relative sizes.

c. Since the bitmap image on the screen is rather crude compared to the actual LaserWriter output, you will probably need to make some adjustments, run another Print Sample, then make a few more adjustments, and so on, until the fraction looks the way you think it should.

13. The real test, of course, comes when you actually use the fraction in a line or two of text. Does the fraction seem too big or too small? Is the spacing correct? Here's how to find out:

a. Select **Print Sample...** again, but this time make sure the Font window is active.

b. In the Select Print Sample Type dialog box, click on the **Text** radio button, enter a font size such as 12 points (if you expect to use the fraction in body text), and replace "The quick brown fox..." in the text entry box with some text that includes the fraction. Don't be surprised if you get an empty square where the fraction should be; the fraction will appear as expected on the printed page. Now click on the **OK** button.



When the Font window is active, ***Print Sample...*** *presents this dialog box. Select the* ***Text*** *option, enter an appropriate size, and type some text containing the new fraction (which will probably appear on the screen as an empty box).*

c. Again, because the bitmap image on the screen is much less refined than final LaserWriter output, you may need to go back and fine-tune the size or placement of the fraction, or adjust the character width, so that it fits in properly with the other characters of the font. You may also want to test the fraction with text of different sizes to make sure that your work translates properly both up and down the font size scale.

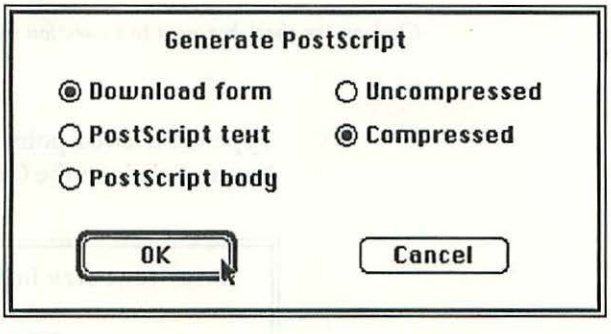
14. Once you have the fraction the way you want it (or fractions, if you have decided to be more ambitious), there are a few final steps you must perform to create a functioning PostScript font:

    a. First, you need to create the downloadable PostScript font file.

        i. With the Font window still the active window, select **Generate PostScript** from the File menu.

        ii. In the Generate PostScript dialog box, click on the **Compressed** and **Download Form** radio buttons, and click on the OK button.

        iii. Wait for *Fontographer* to create the PostScript font file.



When you ask Fontographer *to create the PostScript file, specify the* ***Download Form*** *and* ***Compressed*** *options.*

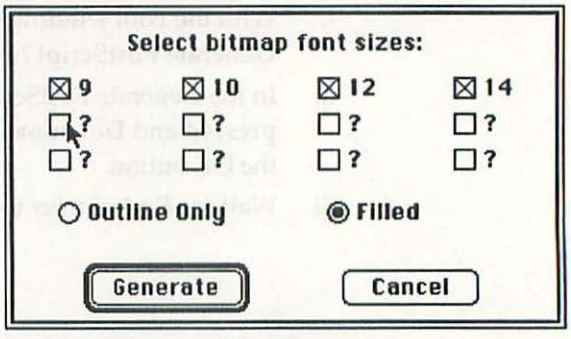
b. Next, create the bitmap screen fonts.

i. With the Font window active, choose **Select All** from the Edit menu or press Command-A.

ii. Select **Generate Bitmaps...** from the File menu.

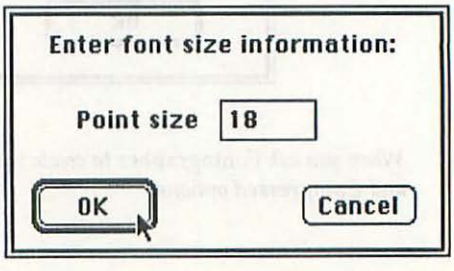
iii. In the Select Bitmap Font Sizes dialog box, replace the question marks with the same font sizes that the original Palatino bitmap font file contained. For example, to replace the first question mark:

◊ Click on the check box next to the first question mark.



Click on the check box next to a question mark to add a new bitmap font size.

◊ Type the desired point size in the resulting dialog box and click on the OK button.

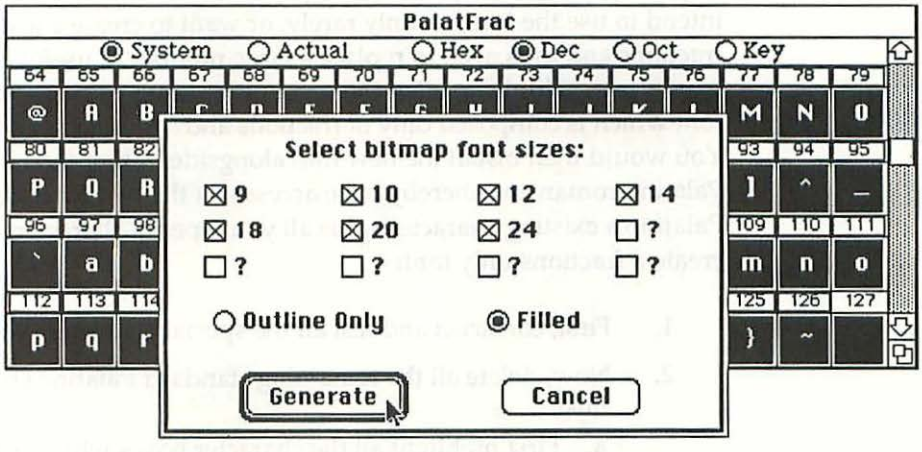


Enter the new point size and click on the OK button.

◊ Don't change anything in the next dialog box, the Font Size Information dialog box; just click on the OK button.

iv. Repeat as needed to enter and select all the font sizes in the original Palatino bitmap file.

v. Make sure that the **Filled** option is selected. Then click on the Generate button and wait for *Fontographer* to create the new bitmap file.



Once the proper bitmap sizes are entered and checked, click on the Generate button.

vi. If desired, use *FONTastic Plus* to clean up the bitmap image of the fraction. (This is more a cosmetic decision than anything else. If you don't care how "amateurish" your fraction looks on the screen, don't bother.) All the other characters should have been copied directly from the original Palatino bitmap file and thus should not need any additional work.

c. Finally, install the new font as you would any downloadable PostScript font:

i. Place a copy of the PostScript font file in your System folder.

ii. Since the new file is a duplicate of standard Palatino (except for the one deleted character and the added fraction), use Font/DA Mover to delete the original Palatino screen font from your System file.

iii. Copy the new screen font to your System file with Font/DA Mover.

Your new version of Palatino is now ready to use.

In this illustration we created a font which contains all of Palatino's usual characters (except the one replaced with the fraction). But, if you intend to use the fraction only rarely, or want to create a wide variety of fractions and thus need to replace a large number of useful characters, you may want to eliminate all of the remaining standard characters and create a font which is composed only of fractions and/or other special characters. You would then install the new font alongside, rather than instead of, Palatino roman and thereby have access, via the two fonts, to all of Palatino's existing characters plus all your special characters as well. To create a fractions-only font:

1. First, construct and test all the special characters you have in mind.
2. Now, delete all the remaining standard Palatino characters. To do this:
   a. First highlight all the character boxes which still contain a standard Palatino character. Select them one by one while holding down the Shift key.
   b. Then select **Clear** from the Edit menu or press Command-X (Cut).
3. Finally, create the bitmap and PostScript font files as in the previous example.

One final note about composite *Fontographer* fonts such as this one. In order for your new font to print properly, the PostScript file of the font on which the composite is based must be in the LaserWriter's memory (either ROM or RAM) before the new font is called upon. Otherwise, the new font will not be able to locate the individual parts of the composite characters and will not print correctly. In other words, composite characters do not initiate automatic downloading of the PostScript font files on which they are based.

This requirement poses no problem if the base font is a resident LaserWriter font: if, for example, your composite font is based on Palatino, and you are printing on a LaserWriter Plus (in which Palatino is a resident font). It can pose a problem, however, if the composite font is based on a downloadable font. In that case, you want to be sure that the base font is downloaded to the printer before the composite font is used.

There are two ways to ensure that the base font is in the LaserWriter's memory when it is needed. One is to manually download the base font before you print a document using the composite font. The other way is to use the base font somewhere in the document *before* you use the composite font. If you don't want to format any visible characters in the base font, you could format a space or a blank line (if one is handy) instead. The space or line, of course, will look just like any other space or line. And, even though no visible base font characters are actually printed, the composite font characters will now print as they are supposed to.

Summary

- Different fonts are designed for different purposes. General rules of thumb for choosing among them include:
  - ◊ Use serif fonts for text.
  - ◊ Use sans serif fonts for display type (headlines, section heads, titles, and so on).
  - ◊ Use decorative typefaces sparingly.
- For the best quality bold and italic text, LaserWriter owners should obtain the complete set of LaserWriter and LaserWriter Plus screen fonts from Adobe.
- It may sound obvious, but it's true: to save disk space, only install the screen font sizes and styles you actually need and use. To conserve even more disk space, you can sacrifice screen readability (and to a lesser degree, screen accuracy and print quality) by eliminating all but a few screen fonts. If your font sizes and styles are cut to the bone in order to accommodate many different fonts, you may want to install a complete set of New York screen fonts, do all your major document work in that font, and then reformat in the correct font just before printing.

- If, after a reasonable attempt at cutting out unneeded screen font sizes and styles, you still can't fit all your fonts in the System file, then try the following:
  - ◊ If the fonts left over are few in number, install specialty fonts in applications or documents, and add the others to your Font menu with *Fontsie*, on those occasions when you need them.
  - ◊ If the fonts left over are many in number, you should consider purchasing a commercial font/desk accessory extender, such as *Suitcase* or *Font/DA Juggler Plus.*
- If you use downloadable PostScript fonts, you should manually download them if you use a few downloadable fonts all the time, if you use a print spooler which cannot automatically download fonts, or if you don't have room to store all the printer files in your System folder. Otherwise, you're probably better off letting the Macintosh download your fonts for you automatically.
- Downloadable PostScript fonts can cause serious space problems. To save space on disk (especially floppy disks):
  - ◊ Create several floppy-disk System disks, each with its own set of downloadable fonts. At the beginning of a work session, restart your Macintosh with the disk containing the fonts you intend to use.
  - ◊ Remove the bold-italic printer file from your System folder. If you never use bold or italic text, or in those circumstances where Macintosh-generated bold face and italic will suffice, delete the bold and italic printer files as well.
  - ◊ Experiment and see if you can put some of your printer files on your application disk and/or in your application folder.

  To keep from overloading the LaserWriter's memory:
  - ◊ Avoid printing from *PageMaker* and programs that use Laser Prep (which is most programs) during the same printing session.
  - ◊ Eliminate bold-italic, bold, or italic text in your documents, since each one (usually) requires a different PostScript printer file.
  - ◊ If you have a LaserWriter, upgrade to a LaserWriter Plus.

- With *FONTastic Plus* or *ResEdit* (or *Font/DA Juggler Plus*'s *Font/DA Utility*), you can add prefixes to the names of your screen fonts to reorder your font menu in a more efficient way, such as all text fonts at the top and all decorative fonts at the bottom. If you prefix styled screen fonts with a period or percent sign, they will disappear from most Font menus and lists, making your roman fonts much easier to find. The bold, italic, and bold-italic screen fonts will still be available, however, via the Bold and Italic style commands.
- You can use *Fontographer* to add fractions (easy) or a logo (more difficult) or a facsimile of your signature (even more difficult) to any existing PostScript font.
- If your PostScript fonts (both resident and downloadable) are not printing as you think they should be, check the following:
  - ◊ Make sure that your downloadable PostScript printer files are stored in your System folder.
  - ◊ Print your documents, even if you are at a commercial printing shop, using the same System file that was in charge when the documents were originally created.
  - ◊ Try to avoid printing on the LaserWriter in sizes smaller than 12 points. If you must include the small print, be sure to use the LaserWriter's resident fonts or the best quality PostScript fonts you can find.
  - ◊ Expect (usually) slight changes in your documents when you change from a low-resolution printer (like the LaserWriter) to a high-resolution one (such as a Linotronic), or when changing from one manufacturer's Bodoni (or whatever) to another's.

5

Customizing the Desktop (and Other Things) to Fit the Way You Work

Custom Windows *221*
- Icon Views *223*
- List Views *226*
- Font and Font Size *227*
- Other Window Options *229*
- The Default Window *230*

Custom Menus and Menu Command Shortcuts *232*
- Customizing the Desktop Menu Bar *232*
- New Menu Command Shortcuts: Getting Ready *236*
- Assigning New Shortcuts to Existing Menu Commands *245*
- Creating New Commands and Their Shortcuts *255*

Custom Keyboards *264*
- Using *MacKeymeleon* *264*
- Using *QuicKeys* *270*

Making a Custom Keyboard Map *274*
Custom Dialog Boxes *276*
Custom Icons *278*
Other Customizations You Can Do *286*
- Custom Command and Menu Names *286*
- Custom Desktop Patterns *295*
- Custom Scroll Bars *297*
- Custom Startups *299*

Summary *310*

For most of us, the Desktop is an integral part of our work environment. True, some people avoid the Desktop like the plague. Indeed, the following chapter contains much information concerning ways to bypass the Desktop, either when the spirit moves you or all the time. But for most of us, even those of us who like to detour around the Desktop on occasion, the Desktop is a home base, an organizational focus, the place to which we always return to put our house in order before setting off on some other task.

The Desktop is created by the Finder, and because the Finder is the application to which we turn more often than any other, it makes sense that it should be fine-tuned as closely as possible to our personal preferences. In the sections on viewing and naming files and folders in Chapter 2, we began the process of customizing the Desktop to fit the way we work. But there is much more that can be done. Some of these fine-tunings are easy to perform, some are more difficult. Some are real time-savers, some make organization easier, and some are just for fun. Taken together, they can make the Macintosh Desktop reflect your work habits and personality, as much or more than your real desk top does.

This chapter contains instructions for altering the Desktop in a number of different ways. Actually, some changes will apply across the board, not just to the Finder. Nonetheless, they all will alter the way the Desktop looks and feels (as the saying goes). In some cases, small utility programs make the process easy, painless, and safe. In others, you'll need a program such as *Tempo II* (Affinity Microsystems) or *QuicKeys*. In still other cases, you must (or may choose to) use *ResEdit*, which was introduced at the end of the last chapter. Those who get nervous at the idea of tweaking the innards of a real live program may wish to restrict themselves to working only with the utilities. But as long as you are careful to follow the precautions I enumerate a little later on concerning the use of *ResEdit*, even the squeamish should have little to fear.

Rather than suggest that you combine all of these alterations in a single System file, I recommend that you read through the following sections and decide on a subset of customizations which interest you most. Then gather the necessary software tools and experiment on a copy (scaled down, if need be) of your master System file. Experimentation is important, because not all of these customizations work happily in concert. *Pop-Keys*,

for example, was incompatible with *Tempo II* at the time this was written. (*Switcher* was not happy with *Tempo II* either, but that is a matter best left to the next chapter.) Only after you have determined to your satisfaction that the combination of changes you have made is both productive and safe should you use it during your normal work routine.

Need I also add that once you have configured a custom desktop and become comfortable with it, there is a temptation to add new and interesting custom features to the existing ones? (These could be other changes described in this book or new alterations that you have come across elsewhere.) It is most important, however, that you not revise your previously tested custom desktop without testing it thoroughly once again. New changes can create new incompatibilities, which can wait to make themselves known until the least desirable moment. To avoid such unpleasantness, always experiment on copies of your System folder, and test new customizations thoroughly before placing your trust in them.

These warnings are not meant to scare you off, by the way. Incompatibilities among popular programs are certainly nothing new. Macintosh software is highly complex, and it is difficult for programmers to predict with assurance how all possible combinations of applications, System files and Finders, and the changes we are about to make to them, will interact. So while I encourage you to experiment until you find a Desktop that fits you like a glove, I also encourage you to be your own beta test site before you add that glove to your daily wardrobe.

Thus I invite you to try those customizations in this chapter that interest you. You will, it is hoped, end up with a distinctly more comfortable and efficient Desktop, and learn a little something about Macintosh programs, especially the Finder, in the process.

Custom Windows

We are all so familiar with the arrangement of icons and lists in Macintosh windows that we hardly think about it. Yet the way they come, straight out of the box, is not necessarily the optimal arrangement for everyone. If your files tend to have long names, for example, you may want to stagger the icons in icon view so that the names are less likely to overlap (as in the

offset rows illustrated in Chapter 2); or you may want more room for longer names in the list view; or you may prefer dates in short form, rather than spelled out in detail; or perhaps you'd like your disk and file names spelled out in a different font or a larger font size.

All of these things, and more, can be changed with some effort using *ResEdit*. Fortunately, a utility program called *Layout* (public-exchange) can take the pain out of the process. With *Layout*, you can easily change the arrangement of icons (both large and small), shift the space allocated to the columns in list windows, and even delineate the size and location of the first window to open from a newly created folder or initialized disk. You can do all this by pointing with the mouse and demonstrating what you want things to look like, instead of fiddling with arcane numbers. (If you want to see what kind of arcane numbers you'd have to fiddle with, use *ResEdit* to open the Finder resource called LAYO. Open the single resource within the LAYO window, and then scroll through the resulting window and its innumerable text entry boxes.)

To begin this exploration of custom windows:

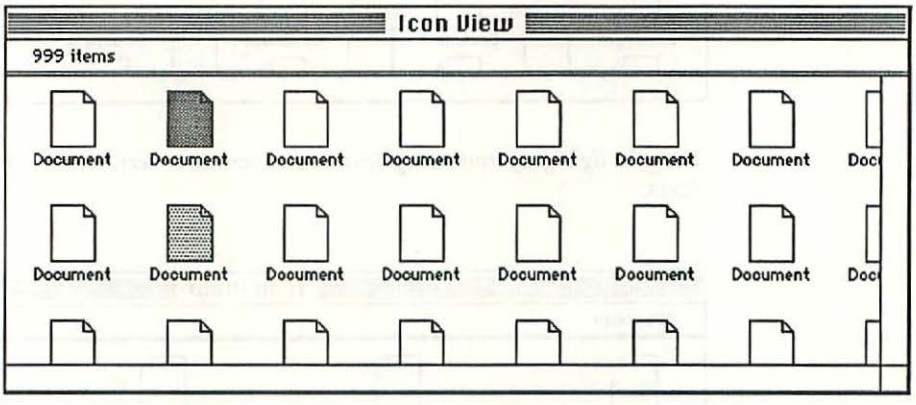
1. Copy the Finder to a floppy disk.
2. Launch *Layout* with your standard System disk in control.
3. Since *Layout* automatically opens the Finder on the Startup disk, select **Open...** from the File menu and select the backup copy of the Finder you just made.

Now you are ready to customize a copy of the Finder. When you have completed the customizations described in the following sections, replace the Finder in your System folder with this newly altered one.

Before we begin, however, it should be noted that the icon arrangements you set up for both small and standard icon views will not immediately appear when you start using your newly customized Finder. Your windows will look exactly as they did before—until you use the **Clean Up** command, or if *Layout*'s **Always Grid Drags** option is turned on, until you start dragging icons from place to place. If your windows are in one or the other icon view and you want your Desktop to reflect the new icon arrangements immediately, open all your windows one by one and select **Clean Up**. Otherwise, simply select **Clean Up** each time you encounter a window in the old arrangement during your regular work routine.

Icon Views

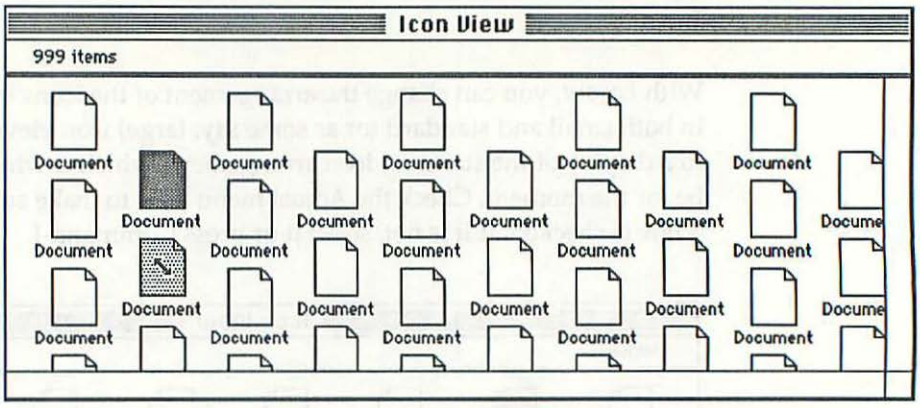
With *Layout*, you can change the arrangement of the icons in your windows in both small and standard (or as some say, large) icon views. *Layout* opens to a display of the standard icon arrangement, which is where we want to be for the moment. Check the Adjust menu now to make sure that **Icon View** is checked. If it is not, select it or press Command-I.



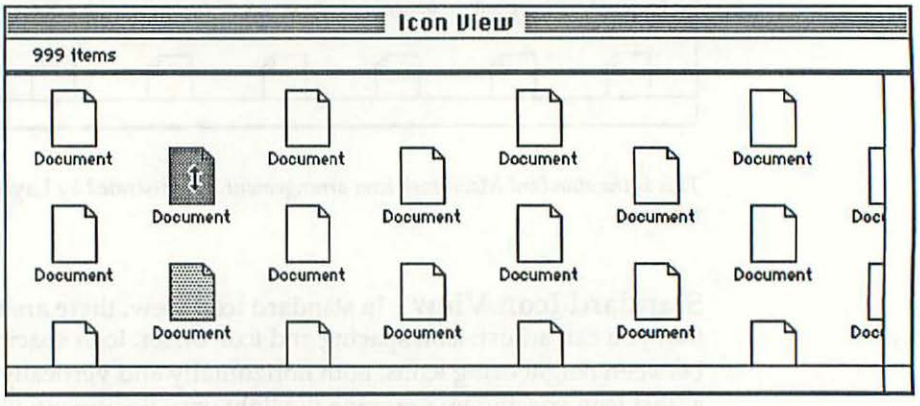
This is the standard Macintosh icon arrangement, as illustrated by Layout*'s Icon View edit window.*

Standard Icon View In standard icon view, there are two parameters that you can adjust: icon spacing and icon offset. Icon spacing is the space between neighboring icons, both horizontally and vertically. In *Layout*, you adjust icon spacing by dragging the light gray document: up or down to change the vertical spacing, left or right to change the horizontal spacing, or diagonally to change both horizontal and vertical spacing at the same time. In this way you can squeeze out a lot of the air between your documents or let some in.

Offset moves alternate vertical columns of icons up or down, so that a single row of icons seems to jog up and down as it advances across the screen. This odd-sounding concept actually makes a lot of sense, especially if you want to shoehorn a lot of icons with long labels into a small space. With offset icons, long document names do not overlap the names of



Drag the light gray icon in any direction to increase or decrease the space between adjacent icons.



Drag the dark gray icon up or down to create offset rows of icons.

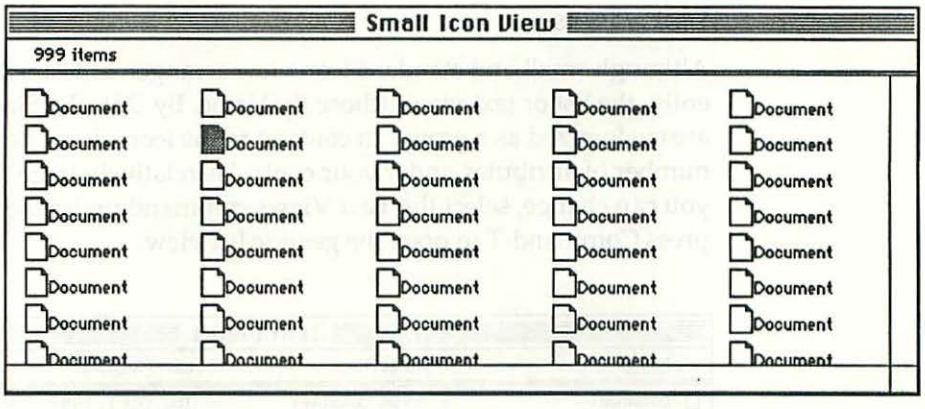
neighboring icons, even when icons are relatively closely spaced. (See Chapter 2 for illustrations of offset icons at work.) To adjust offset, drag the dark gray icon up or down.

Now take a few moments to experiment with changes to both offset and spacing in combination. If you have short document names, many documents, and good eyesight, you might want to tighten up on both vertical and horizontal spacing and add only a small offset. If your docu-

ments tend to have long names, then a little more space and a greater offset may be in order. Also (as noted in Chapter 2), a large offset makes it possible to see more document names in a narrow window if you are willing to cut off from view most or all of the icons in the top row.

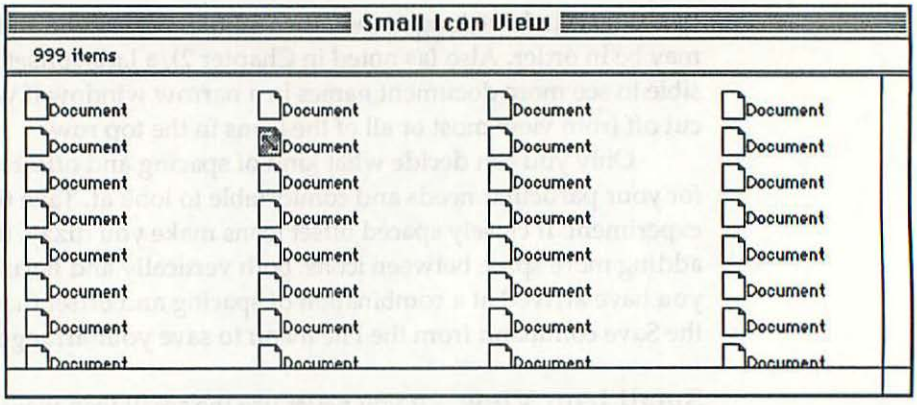
Only you can decide what kind of spacing and offset is both efficient for your particular needs and comfortable to look at. Take the time now to experiment. If closely spaced offset icons make you dizzy, for example, try adding more space between icons, both vertically and horizontally. When you have arrived at a combination of spacing and offset that you like, select the Save command from the File menu to save your arrangement.

Small Icon View If you never use the small icon view you can skip this part. But if you do use it, select **Small Icon View** from the Adjust menu (or press Command-M).



This is the standard Macintosh small icon arrangement, as illustrated by Layout's *Small Icon View edit window.*

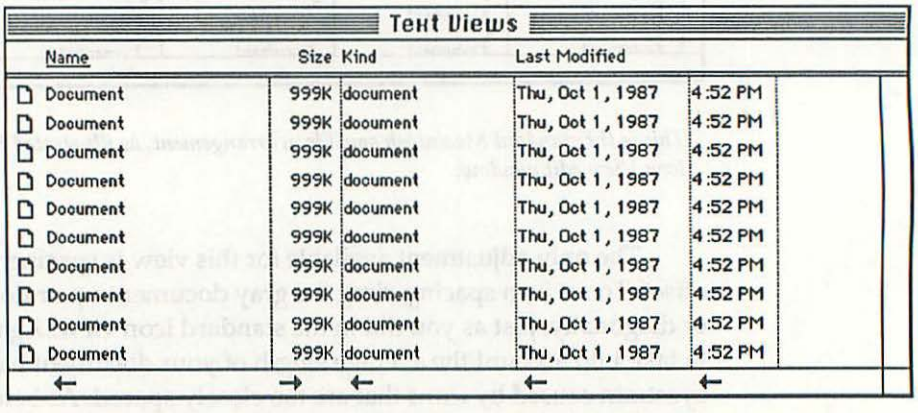
The only adjustment available for this view is spacing; there is no offset. To set icon spacing, drag the gray document up or down, right or left, or diagonally, just as you did in the standard icon view. Again, you'll want to take into account the average length of your document names and the eyestrain caused by icons that are too closely spaced. As before, when you have arrived at an icon spacing you like, select the Save command from the File menu to save your arrangement.



Drag the dark gray icon in any direction to increase or decrease the space between adjacent small icons.

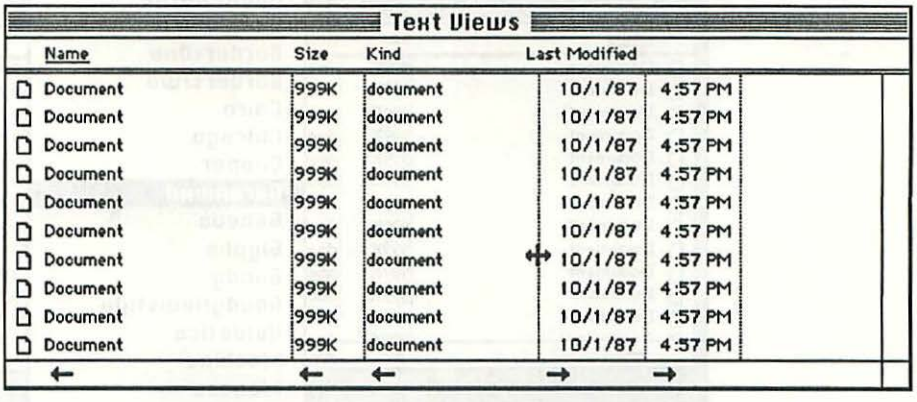
List Views

Although small and standard icon view arrangements are set independently, the list or text views (those By Name, By Date, By Size, and By Kind) are customized as a group. In contrast to the icon views, however, the number of attributes under your control is relatively large. To see just what you can change, select the **Text Views** command under the Adjust menu, or press Command-T to open the generic list view.



The various list views are customized collectively. This is the standard Macintosh list view arrangement, as illustrated by Layout's *Text View edit window.*

The most useful attribute you can change is the width of a column. You do so by dragging the dotted line to the right of the column you want to change (the other columns will be pushed right or left as needed, but their widths will not change). You can also right or left justify a column by clicking on the arrow underneath. You can also change the style of the date to short (as in 7/19/88), medium (as in Sun, Jul 19, 1988), or long (as in Sunday, July 19, 1988) by double-clicking on the date column until the desired format appears.



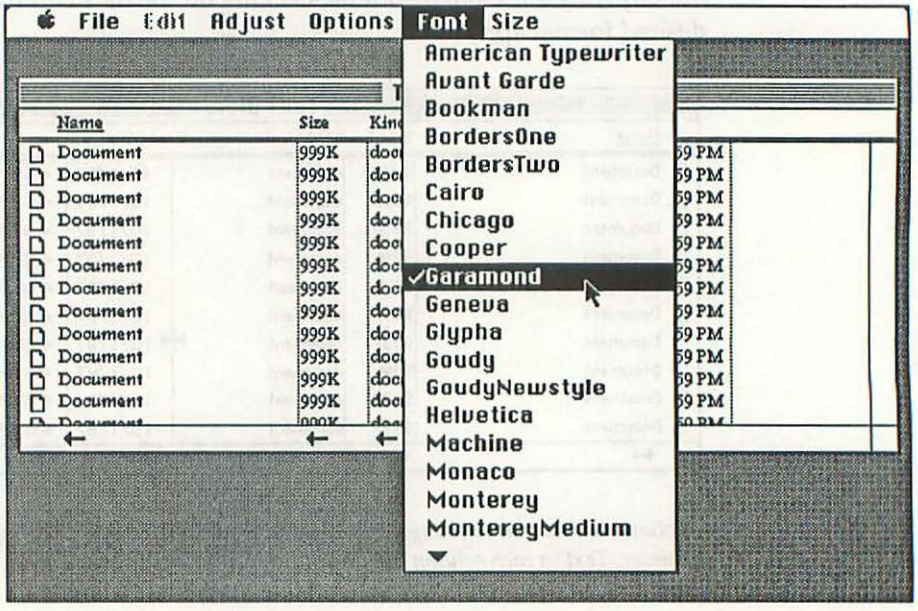
Column widths can be changed by dragging the dotted column dividing lines with the pointer. Text in each column can be justified right or left by clicking on the arrow beneath each column.

Note that you cannot change the order of the five columns or delete any of the columns. Also be aware that the Finder will simply ignore any text that extends beyond the limits of your columns (in *Layout*, on the other hand, the text simply runs into the next column). So be sure that your columns are wide enough for the information they must contain. Finally, when you have settled on a text view design that fills your needs, select the Save command from the File menu to save it.

Font and Font Size

With the Font and Size menus, you can specify how you want disk and document icon names and the information in the list views displayed. To see the effect of your choices, keep the generic list view on the screen. Look

through the Font menu; you should see listed all the fonts on your System disk (except the ones you made invisible in Chapter 4). Experiment now with different fonts and sizes (making sure, of course, that you pick only native sizes) and see if there is one that you like better than Geneva, the default.



List views and disk and document icon names can be displayed in any font in your System file and in any font size.

If you have different System disks with different collections of fonts, you'll either have to select different fonts for the Finders on those disks or choose a font that is installed in all your System files (you may have New York in all your System files, for example). If you do choose a font that is not in all your System files and use the same Finder everywhere, some of your Finder windows may end up using a resized version of a different font and be difficult, if not impossible, to read.

Also be sure to take into account how font size influences the space required by the text in all three types of views (icon, small icon, and list).

Once you have decided on a font and size, review the spacing in your list and icon views. If you chose a font or a size which is radically different from Geneva-9 (the default with which we started), you may need to go back and readjust the settings in all three windows. Finally, once you've got everything the way you want it, select the **Save** command from the File menu.

Other Window Options

Layout's Options menu gives you control over two other window attributes and a third item which is only marginally related to windows. The first of the three commands is **Use Zoom Rects**. With it, you can turn on or off the zoom-style animation that occurs every time you open or close a window or open an application. Animation is on by default, as shown by the check mark next to the menu listing. Since this is a toggle command, select the command to remove the check and turn animation off, and select it again to turn it back on. I can see no particular reason why one might want to turn animation off, so whether you turn this off or keep it on is a matter of personal preference.

The second command in the Options menu is **Always Grid Drags**. This command applies only to icon views and its default setting is off. When **Always Grid Drags** is off, you may place either small or standard icons wherever you wish in a window, regardless of the spacing and offset you established earlier. (When **Always Grid Drags** is off, those arrangements apply only to the **Clean Up** command.) If **Always Grid Drags** is on (checked), however, icons will "snap to" correct positions whenever you drag an icon to a new location. Your windows will be neater and more easily put in order if you turn **Always Grid Drags** on.

The final Options command is **Skip Trash Warnings**, and the default setting is off. This is a potentially dangerous command, since it eliminates the warning that you are trashing a possibly important file, such as an applications or a System file. Turn **Skip Trash Warnings** on and you will never again be faced with the query, "Are you sure you want to throw away the file WhatHaveYou?" It's an appealing idea if you are getting tired of such overweening solicitude.

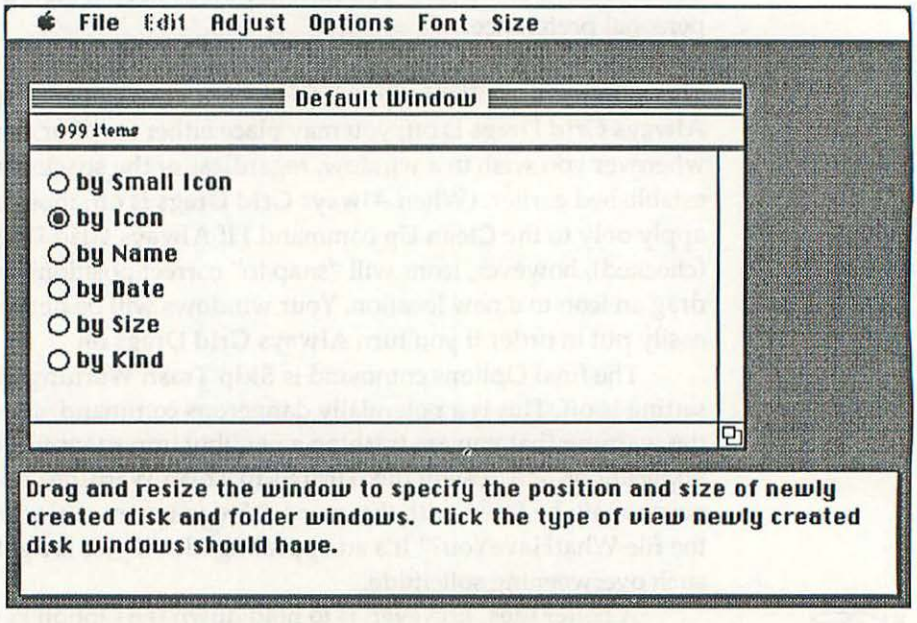
A better idea, however, is to hold down the Option key whenever you throw away files that might prompt this warning. If you become distracted and throw away the wrong file, you'll probably forget to hold down the

Option key as well. The warning will thus appear, wake you (and perhaps shake you) up a bit, and give you the chance to recover gracefully from what could have been a small (or large) disaster. Turning on **Skip Trash Warnings** provides no such opportunity. Therefore, I strongly recommend that you leave this command off (unchecked).

Having set the Options menu commands the way you want them, select the Save command to copy them to disk.

The Default Window

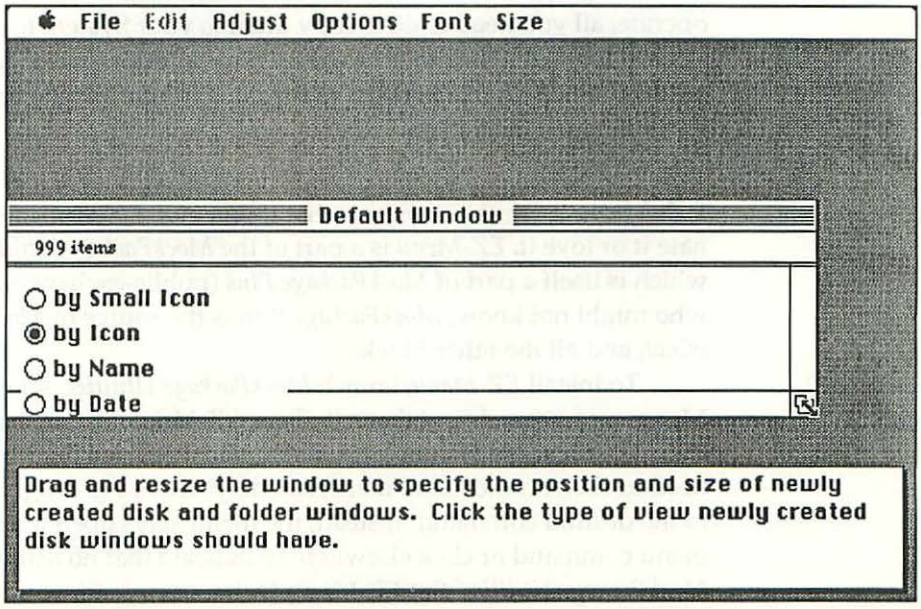
One final set of decisions you can make with *Layout* concerns the default window. This is the window which first opens from newly created folders and newly initialized disks. The standard default window is relatively large, located in the upper left portion of the screen, and set to the standard icon view. Choose the Default Window from the Adjust menu to see an illustration of the default size and position.



The standard default window (and view) is illustrated by Layout's *Default Window edit window.*

To change the size and position of the default window, drag the title bar to position the window where you would like it to appear, and drag the size box to size the window the way you want it. If you think back to the window placement discussion in Chapter 2, you will recall that I recommend that most floppy disks open their main window as a narrow strip across the middle of the screen (for hard disk owners, this keeps in view the two main windows of a hard disk). Thus, I recommend setting the default window size and position the same way: a narrow strip, slightly more than one standard icon in height, across the middle of the screen. Be sure to leave enough room along the right for the disk icon column. To change the default view to something other than the standard Icon view, click on the desired option inside the default window. Since all my windows start out in icon view (though they change to list view when things get crowded), I keep the default where it is, on icon view.

One last selection of the Save command fixes all your changes in the Finder. You can now quit *Layout*, replace the old Finder with the new one, and go to work on your new Desktop.



One possible default window size and location is as a narrow strip in the center of the screen.

Custom Menus and Menu Command Shortcuts

We have already seen, in the FKey section of Chapter 3, how the Desktop's menu bar can be altered (by adding a new menu for FKeys, using *Pop-Keys*) and new keyboard commands can be created (commands, in this case, to call up FKeys). This, however, is just the tip of the iceberg. There are many other ways in which you can alter the menus on the Desktop (and elsewhere, for that matter), add new items to the menu bar, or add new commands to perform new tasks, such as opening applications with a single keyboard command.

Customizing the Desktop Menu Bar

Here are three programs with which you can add a clock to the menu bar, add a screen-saver function that is controlled, in part, from the menu bar, and alter the way menus drop down and commands are selected. Two of the three programs are Init resources, which means that to get them to operate, all you need to do is copy them to your System folder and restart your Macintosh. The other has its own installer program. All three programs operate no matter what program is running at the time: the Finder, an application, and so on.

EZ-Menu I can assure you that if you give *EZ-Menu* a try, you'll either hate it or love it. *EZ-Menu* is a part of the *MockPackage Utilities* program, which is itself a part of *MockPackage Plus* (public-exchange). For those of you who might not know, *MockPackage Plus* is the source of *MockWrite*, *MockTerminal*, and all the other Mocks.

To install *EZ-Menu*, launch *MockPackage Utilities*, select the **Install EZ-Menu...** command, and that's it. Once *EZ-Menu* is installed in your System file, menus drop down as soon as the pointer passes over them; there is no need to click on the menu name, nor do you need to drag the pointer down to the desired command. Instead, the menu stays open until you click on a menu command or click elsewhere to indicate that no action is desired. With *MockPackage Utilities*' **Set EZ-Menu Delay** command, you can set the number of seconds a menu will wait for a click, after which it will close automatically.

Some people like their menus to work this way. It is, after all, the modus operandi for menus on the Amiga. But others hate having menus drop down when the pointer passes unintentionally over the menu bar. Since my pointer tends to wander all over the screen, I favor the original method. You might prefer *EZ-Menu*. The only way to know is to experiment.

JClock *JClock*, an Init file which you copy to your System folder, comes in two flavors, but the idea behind both is the same. To the right end of the menu bar, *JClock* adds a small digital clock indicating the correct time. One version of *JClock* displays hours and minutes in 12-hour time. The other version displays hours, minutes, and seconds in 24-hour time. I prefer the former because I dislike 24-hour time (my mind doesn't work that way) and more importantly, I find the constantly changing seconds distracting. Others seem to enjoy watching the seconds tick past as they attempt to overcome writer's block. The choice is up to you.

JClock is pretty good about moving aside when something else wants to occupy its space (such as MultiFinder's mini-application icon). Sometimes, however, *JClock* becomes indecisive and garbles its display when something else is in the way. Usually, this is only a cosmetic problem that does not affect program operation. In cases like these, you can turn *JClock* off for the current session by holding down the mouse button while the pointer is over the time display. In two or three seconds the clock will disappear, at which point you can release the mouse button. You can get it to reappear by repeating the process.

Be sure you test *JClock* carefully before using it during regular business hours. Most programs have no quarrel with it, but some do, and some other Init resources in combination with *JClock* can cause problems.

AutoBlack* and *Pyro! If you use your Macintosh day in and day out, you are no doubt aware that it is best to turn down the screen (using the dial beneath the overhang on the case of the Mac Plus, for example) when you are not actually at work. Otherwise, the image on the screen may "burn in" and haunt you forever after as a faint ghost lying behind whatever is supposed to be on the screen. Clever people have devised screen-saver desk accessories, FKeys, and you name it, to make it possible to turn the screen off from the keyboard or to have it turn off automatically after so many minutes of inactivity. But as far as I'm concerned, no programs have done this as effectively as *Pyro!* or *AutoBlack*.

Installing *AutoBlack* is a little different from the way you install most Init files. This is, in part, because it must be loaded into memory first, before any other Inits. It is also because *AutoBlack* comes in several versions, which can be installed in different ways. The method I prefer uses the "Init/application" version of *AutoBlack*, the one whose icon is a hand writing on a black Macintosh screen. To use this version, simply rename it `MacsBug`. (Renaming the file `MacsBug` ensures that it is loaded into memory first. For developer purposes, the System file loads any file in the System folder named `MacsBug` before installing any Inits. Usually the MacsBug file is a debugger. In this case, it's not.) Now copy *AutoBlack* to your System folder; restart your Mac, and *AutoBlack* is ready to go. (Refer to the AutoBlack manual for directions for the other methods of installing *AutoBlack*.)

Once installed, *AutoBlack* turns off the screen after five minutes lack of exercise and bounces a small, white (and functional) analog clock around the screen to remind you that the Macintosh is still on. Move the mouse, insert a disk, or press a key and the screen instantly reappears. In the "Init/application" version of *AutoBlack*, you can reset the blackout delay to any desired time, in minutes and seconds. To do so, double-click on the *AutoBlack* icon any time after the program is installed. (If the computer replies that "an application cannot be found to open this document," hold down the Option key as you double-click.) A window will open asking you to type in a new blackout interval. Click on OK to set the new delay for this and all future work sessions. To turn *AutoBlack* off temporarily, set the interval to zero. Reset the interval to any desired time to turn it back on.

Unlike some screen savers, if you are downloading a file over the phone, or are in the midst of printing, *AutoBlack* will not interrupt those activities but will merely turn the screen off. Furthermore, if you want to turn the screen off right away, you can do so by placing the pointer in the upper right corner of the screen (up in the menu bar, which is why screen savers are described in this part of the book). If you want the screen to remain on even though inactive, simply place the pointer in the lower right corner of the screen. (The screen will eventually black out, however—after about two hours.)

I have used several versions of *AutoBlack* for well over a year and am addicted to it. I leave my Macintosh on all day (the experts claim that this is better than turning it on and off all the time). When I get up for a break, to run off a long printout, or to take off on a short errand, *AutoBlack* takes over

and saves my screen from unwanted ghosts. During the period that I have used the program, I have found only a few incompatibilities, even with the large number of programs I try out over the course of a year. Just to be safe, however, you should be sure to test your setup for incompatibilities (especially with other Inits) if you decide to install *AutoBlack*.

Pyro! is a screen-saver utility that every purchaser of *Suitcase* or *PowerStation* (both Software Supply) receives as a bonus. To install *Pyro!*, you simply copy it to your System folder and restart your Macintosh. When *Pyro!* blacks out the screen, it displays, as its name suggests, some simple fireworks animation to remind you that the machine is still on. If you have a color monitor, the pyrotechnics will be in color (sorry, no sound).

To set *Pyro!*'s options, you use the Control Panel, clicking on the *Pyro!* icon to open its configuration screen. With the Control Panel, you can turn *Pyro!* on or off and set the delay time (the time that the Macintosh must be idle before the screen blacks out) to an interval ranging from one minute to two hours. You can also set both the "sleep now" corner and the "never sleep" corner to any of the four corners of the screen. (If you place the pointer in the sleep now corner, the screen blacks out immediately; if you place it in the never sleep corner, *Pyro!* is suspended.) You can also elect to suspend certain background tasks such as print spooling and alarm notification during screen blackout, or to run them as usual when the screen blacks out.

In addition to its use of the Control Panel for configuration and the range of the configuration settings themselves, *Pyro!* differs from *AutoBlack* in one other significant way. During long file transfers using a telecommunications program and a modem connected to the modem port, *AutoBlack* can blank the screen without disrupting the transfer process. *Pyro!*, on the other hand, interprets signals passing through the modem port as a sign of activity and, as a result, does not black out the screen (if it did, the transfer would cease).

Whether or not this difference between the two programs matters to you depends on how often you send or receive large files via a modem. Since I frequently download large files, I like the fact that *AutoBlack* blanks the screen for me without interrupting the telecommunications process. You, on the other hand, may like the different options *Pyro!* offers or the way it works (or the screen display). As in so many things, it is a matter of personal choice as much as anything else.

New Menu Command Shortcuts: Getting Ready

As one becomes more proficient with the Macintosh, the usefulness of keyboard commands becomes increasingly apparent. Once you've gotten to the point where you can remember without effort what key combination performs which task, keyboard shortcuts can significantly speed up the way you work. Indeed, the more keyboard commands available, the faster your work will go (up to a point, anyway). Thus it is easy to imagine how nice it would be if you could give keyboard shortcuts to oft-used commands which lack them, or if you could call up desk accessories with the press of a key or two.

Although most people think of keyboard shortcuts in terms of application menus, they are just as important with the Finder; maybe more important, since the Desktop is so much a part of our everyday work routine. Thus this section and the following two sections consider the Desktop's commands and show how to add keyboard equivalents where they are lacking. The last of the three also looks at the subject of entirely new Desktop commands, commands that do not correspond to any single menu item.

As you read through the following sections, keep in mind that, to use keyboard shortcuts effectively, it is important that they be easy to remember. Perhaps even more important, however, is the notion that the same or similar commands in different programs (your major applications, the Finder, and so on) be assigned the same key combination. Since the universality of keyboard command assignments is a significant aspect of program integration, it will be discussed at some length in the next chapter. For now, however, remember that the key combinations you assign to Finder shortcuts should be both easy to recall and, where applicable, fit within the scheme of existing shortcuts in your major applications.

Drawing a Menu Map The first step in adding or changing keyboard shortcuts, whether they be in the Finder or in an application, is to take stock of all the menu commands and their existing keyboard assignments. This is very important, if for no other reason than the fact that you must avoid assigning the same keyboard equivalent to two different commands: Command-P for **Print** as well as for **Plain Text**, for example. If two different commands in the same application are assigned the same keyboard shortcut, one of two things will happen. What usually happens is that the first command with that assignment, going from left to right and top to bottom

through the menus, is always invoked and the other command is always ignored. Sometimes, however, the application gets confused and crashes as a result.

A good overview of all your commands is also important in helping you choose the most appropriate shortcuts. You might be tempted, for example, to give Command-C to the Desktop's **Close** command (which closes a window) if the only menu you were considering was the File menu. But if you had all the menus in mind, you would remember that Command-C is an almost universal shortcut for **Copy**, and thus you might assign Command-W (for window) instead.

A useful way of taking stock of an application's menus is to create a Paint file containing pictures of all its menus. Creating such a file is a bit laborious, but it has several advantages over other methods:

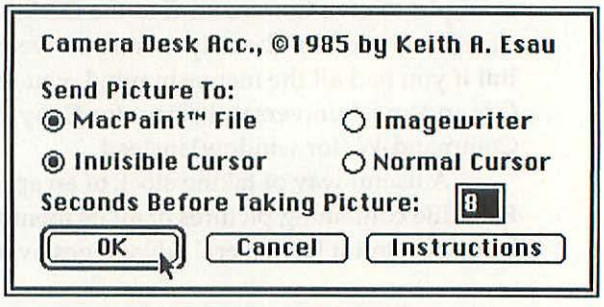
- It avoids the kinds of errors that can creep in when you attempt to copy down menu commands and shortcuts by hand.
- Since you can print as many copies as you like, your Paint document can serve as a scratch pad for testing different combinations of shortcuts until you arrive at one that you like.
- When editing menus, it provides an easy way to determine a menu's order and hence its ID number.
- If you carefully record all your new keyboard shortcuts, your menu map will make a handy and handsome reference.

To create a Paint document full of open menus, you need to be able to make a "snapshot" of the screen while a menu is open. Unfortunately, you can't use the Finder's screen shot FKey (Command-Shift-3) to do this, since it will wait for the menu to close before recording the screen. To take a snapshot of an open menu, you can use a desk accessory such as *Camera* (public-exchange), which was described in Chapter 3.

To make a menu map, install *Camera* in your System file or use a font/desk accessory extender to install it temporarily, and follow along:

1. First make sure you are in the application whose menus are of interest, the Finder, in this case, and that *Camera* is listed in the Apple menu. Also make sure that there is about 100K of disk space on the Startup disk (and the application disk, if the two are not the same; *Camera* will choose one or the other to store its snapshots).

2. Select *Camera* and in the dialog box that follows, enter eight seconds (or thereabouts) in the entry box labeled "Seconds before taking picture". Click on the **MacPaint File** and **Invisible Cursor** options to turn them on.



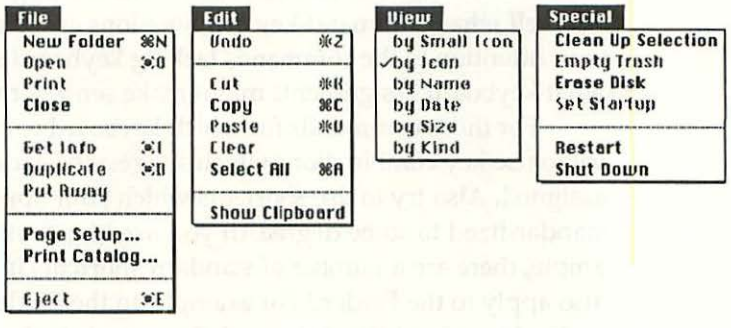
Tell Camera *how long to wait before it takes the screen shot and whether or not to show the cursor, then click on OK.*

3. Now click on the OK button. When the dialog box disappears, put the cursor over the word File on the menu bar and press the mouse button. Do not drag the pointer down into the menu itself. Hold the mouse button down until you hear first one beep, then two beeps.

   If you hear three beeps in a row, *Camera* was unable to complete the snapshot, possibly because it couldn't find enough disk space for the resulting MacPaint file. *Camera* stores the files it makes on the "default drive." If you have a program such as *DiskTop*, you can set the default drive, before using *Camera*, to a disk with plenty of free space. If you don't have a program that can do this, simply make sure once again that the disks in all your drives have at least 100K of space available.

4. Select *Camera* again, make sure that the settings are the same as before, and take a snapshot of the next menu, the Edit menu. Repeat this procedure until you have a snapshot of every menu. (If you intend to assign shortcuts to your desk accessories, be sure to take a screen shot of the Apple menu as well. If the menu is so long that it scrolls, you may have to take several screen shots, a segment at a time, and piece the menu back together in your Paint program.)

5. Now open your Paint program. Any Paint program will do, but what follows will be easier if it can keep at least two windows open at once.
   a. From within your Paint program, locate and open the first *Camera* snapshot. It should be labeled `Screen 0` and may be found on the Startup disk and not in any folder, in the System folder, or in the folder or on the disk containing your application. You may have to search a bit, but once you find one, you should find them all.
   b. With your Paint tools, erase everything in the snapshot but the File menu and its highlighted name. Place the menu in the upper left corner of the document and save the document (using **Save As...**) with a new name, such as `Desktop Menus`.
   c. Now open the next screen shot and, using the marquee or the lasso, surround the Edit menu. Copy it, switch to the Desktop Menu window, and paste it in next to the File menu. Clean up around the edges of the Edit menu as needed.



Use Camera *and your Paint program to create a guide to all the Finder's commands and their shortcuts.*

   d. Continue cutting menus from the other screen shots. Paste them into Desktop Menu one by one, in order, from left to right. Continue on a second or third row as needed.

6. When you are done, save Desktop Menu, make a backup copy and put it somewhere safe, and throw all the screen shots in the trash. Finally, print several copies of Desktop Menu.

Deciding on Appropriate Keyboard Combinations Now that you are armed with a complete set of opened Desktop menus, you need to consider which commands need shortcuts and what those shortcuts should be. First you should rule out those commands which do not currently have shortcuts and should remain that way. I am referring to the commands which have potentially unpleasant consequences: **Erase Disk**, **Restart**, and **Shut Down**, for example. Since it is relatively easy to press an incorrect Command-key combination, you should not assign shortcuts to hazardous commands. Cross these three commands off your Desktop menu printout.

Next you should delete commands which lack keyboard shortcuts and which might as well remain that way, simply because they are not very useful. If selecting a certain command by mistake from the keyboard is more of a pain than having to use the mouse to activate it when it is needed, then the command falls into this category. I include in this group **Print Catalog** and **Show Clipboard**, at the very least. Window views which you never use, such as **By Kind** or **By Size**, for example, should also be added to this group. Cross these off your Desktop menu printout as well.

This should leave you with all the commands that already have keyboard shortcuts, other useful commands which do not, and some or all of your desk accessories. First, review the existing shortcuts to remind yourself what Command-key combinations are already in use. Then turn your attention to the commands lacking keyboard shortcuts and consider what keyboard assignments might make sense for them.

For those commands for which keyboard commands are appropriate, try to use key combinations which suggest the commands to which they are assigned. Also try to use shortcuts which your applications have already standardized to some degree. (If you use Microsoft applications, for example, there are a number of standard shortcuts in these programs which also apply to the Finder.) For example, in the Finder you might want to assign Command-P to **Print** and Command-W (for window) to **Close**, as many applications do. Other assignments you might want to consider include:

By Small Icon	Command-S
By Icon	Command-L
By Name	Command-M
Clean Up	Command-U
Empty Trash	Command-T

As this lists shows, some Command-key assignments are easy to think of, but others are more difficult. Command-E for **Empty Trash** would have made sense, for example, but it is already used for **Eject**. Command-I would have been a logical choice for **By Icon**, but it is a familiar shortcut for **Get Info**. So second or third choices must be assigned instead, or, if you just can't come up with something memorable, you may choose to make no keyboard assignment at all.

If you do decide on a shortcut, however, be sure that it makes sense to you and that it is easy to remember. Command-S and -L, for example, are easy to remember for Small and Large icon views, Command-T for emptying the Trash, Command-U will clean Up the window. The hardest of all the ones on the list above is clearly **By Name**; Command-N creates a new folder, so that assignment is out, and Command-B could apply to any **By... view** command. I finally opted for a close relative to Command-N—Command-M. Perhaps you can think of something better.

Two other menus, the Apple and Font menus, are unlike the standard command menus considered so far, since their contents change according to your choice of fonts and desk accessories. Only the Apple menu is present in the Finder, unless a desk accessory with a Font menu is open. So that we may consider the entire menu picture, however, let's ignore that "small detail" for the moment. Because font and desk accessory menus are special ones, assigning keyboard shortcuts to fonts and desk accessories requires some special techniques. But, as we'll see, it can be done.

What with the great number of fonts and desk accessories, just how you assign particular key combinations to individual desk accessories and fonts can be quite a puzzle. My Apple menu, for example, has five desk accessories starting with the letter A (*Acta*, *AffiniFile*, *Amortization* [public-exchange], *Appointment Diary*, and *Art Grabber+*). Clearly, Command-Option-A will only work for one of them. Which one? What should I do about the others?

There are several ways you might resolve a problem like this:

- You might assign shortcuts only to the desk accessories you use frequently. I, for example, can easily forego a shortcut for *Amortization*, and use the mouse those few times I need this accessory. If you are a real estate agent, on the other hand, it would obviously be most helpful to have a keyboard shortcut for *Amortization*. In that case, you'd want to consider foregoing shortcuts for one or more of the other "A" desk accessories.

- You could use other, equally meaningful parts of each desk accessory's name as inspiration for your shortcut: Command-Option-F for the File in *AffiniFile*, Command-Option-D for the diary in *Appointment Diary*, that sort of thing.
- You could resort to renaming your desk accessories (using the technique described later in this chapter) to better associate shortcut and name. Change *Acta* to Outliner and assign it the shortcut Command-Option-O, for example; or change *Art Grabber+* to simply Grabber and assign it the shortcut Command-Option-G.

A similar problem can also arise in a well-stocked Font menu. Consider New York, New Baskerville, New Century Schoolbook, and Narrow Helvetica, all in the same Font menu. The possible ways of resolving font name problems such as this are much the same as the ones listed above for desk accessories.

Finally, remember to take care not to assign the same keyboard shortcut to two different menu items in the same program, be it the Finder or an application. The consequences of such an inconsiderate act can be frustrating at least, most unpleasant at worst. Always go back over your shortcut plans one last time before you actually assign them, to check for inadvertent duplications.

ResEdit: Some Comments for the Uninitiated *ResEdit* was created by Apple Computer for programmers and all those other people who think they know what they are doing. It lets you go anywhere and do anything, all the time assuming that you know best. As a result, *ResEdit* is perfectly willing to let you make mistakes. No matter how grievous the blunder, *ResEdit* will in most cases silently and without complaint do as you request. As a result, it is as easy to mangle a program with *ResEdit* as it is to put a little extra polish on it.

This doesn't mean that only programmers and similarly omniscient people should use *ResEdit*. What it means is that members of the "non-cognoscenti" (and I include myself in that group) should use *ResEdit* with great care and respect. After all, even someone who is all thumbs can successfully use a sharp knife to accomplish a difficult task, as long as he is mindful of the danger and follows directions. Carelessness, a disregard for

what a sharp knife can do, a failure to follow instructions—these are what lead to unexpected and painful consequences.

From this point on, you will occasionally run into instructions which require the use of *ResEdit*. If you dutifully follow the warnings and other comments which follow, the worst thing that can happen is that you'll destroy an expendable copy of a program. If worse comes to worst, you may also deep-six the disk containing the copy. But such not-so-dire consequences should be rare or altogether absent if you carefully follow the instructions and always think of *ResEdit* as a programmer's sharp knife.

All this negativism, by the way, is not meant to frighten you away and put you forever in fear of what might happen if you do no more than copy *ResEdit* onto your hard disk. These scare tactics are necessary to keep my liability insurance rates at a reasonable level, and more importantly, they are intended to make sure that you are wide awake and perfectly sober when you take *ResEdit* for a spin. Remember, it's your responsibility to keep both hands on the wheel, all four tires on the road, and watch where you're going.

***ResEdit* Do's and Don'ts** Because *ResEdit* gives you the power to destroy as well as create, it is most important that you always make proper preparations for a *ResEdit* work session and follow a few simple rules during your work. Here are some do's and don'ts that apply to any *ResEdit* session. You may want to reread them each time you start a new *ResEdit* project.

- *Always work on a back-up copy* of the file or application to be altered. Never touch the original with *ResEdit*.
- *Perform your work on a floppy disk*, as much as possible, anyway. This means that the copy of the file to be altered should be on a floppy. If you have two floppy disk drives, create a *ResEdit* startup floppy disk and restart your Mac with that; put the disk containing the file to be working in the other drive (and ignore your hard disk if you have one).
- *Do not work on an active file* unless you have no other choice. If you are at work on a System file, it should not be the System file of the current Startup disk. If you are using *Switcher* or MultiFinder, do not work on any program previously launched and still open. If MultiFinder is turned on, do not work on the Finder on your Startup disk.

- *If you see something you do not understand, leave it alone.* If you see rows and rows of numbers and letters, don't insert a few of your own. If you think you changed something that should not have been changed, select *ResEdit*'s **Revert** command (from the File menu), or close all your windows (or as many as you can) until you are asked if you want to save your changes. Then click on No and start over again from the beginning.

Some *ResEdit* Basics Since we will be using *ResEdit* a number of times, it makes sense to cover just once some of the basics of using it: starting the program and finding the right file, navigating *ResEdit*'s windows and menus, and finishing a *ResEdit* work session. These instructions will not be repeated for each set of *ResEdit* instructions so, as with the previous section, you may want to reread this list each time you begin working until you are comfortable with the way *ResEdit* works.

1. Launch *ResEdit* just like any other application. The *ResEdit* menu bar will appear and a window will open for each floppy disk inserted in a drive and for each hard disk (if you have one or more). The title bar for each window displays the name of the corresponding disk, and within the window is a list of all the files and folders on that disk. If you look carefully, you will notice that hidden files, such as the Desktop file, are listed as well as all the familiar ones.
2. To locate a file, simply double-click on whatever folders you need to open until you reach the file you are looking for. If the disk you want is not yet in a drive, insert it (if a drive is available) and its disk window will open automatically. If you need to eject a disk to free up a drive, simply click the close box of the window whose disk you want to eject.
3. The first step in altering the innards of a particular file is to "open" that file within *ResEdit*. To do so, double-click on the name of the appropriate file. A new window will open containing a list of all the file's resources (which is what individual program segments are called). Make certain at this point that you have opened the correct file on the correct disk and that the file you are opening is a copy and not the original.

4. To locate the resource you want to change, either scroll through the list or type the first letter or two of the resource's name. Then follow the directions for the particular alteration you want to perform.

5. When you have completed your editing, close all the windows you have opened during your work. Continue to close windows until *ResEdit* asks if you want to save your changes. Click on Yes and *ResEdit* will copy your work to disk. You may now begin work on another file or quit *ResEdit*.

Although there is no such thing as a bonafide *ResEdit* manual, two rather heavy-duty references can help you to understand the many things that *ResEdit* can do. One is the bible of Macintosh programmers, the mammoth *Inside Macintosh* (Addison-Wesley). The other is the slightly more friendly *Macintosh Revealed* (Howard W. Sams). Neither qualifies for bedtime reading, but one or the other may, with some effort on your part, make *ResEdit* a little more comprehensible.

Assigning New Shortcuts to Existing Menu Commands

In this section, we will be adding keyboard shortcuts to familiar Desktop menu items. The instructions in this section will work, however, with almost any application. Indeed, in the next chapter, we will be using these same methods to make the necessary changes so that the keyboard shortcuts for similar commands are the same in all your applications. Since not all of the material here is repeated in Chapter 6, even if you do not wish to add any keyboard shortcuts to the Finder, you may want to skim the remainder of this section in preparation for the material in the next chapter.

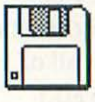
Shortcuts for Standard Menu Items Having already decided which commands need shortcuts and what those shortcuts should be, we can now perform a little cosmetic surgery on the Finder. First, we'll consider how to add a shortcut to a standard menu command, such as Command-P for the **Print** command.

The best way to accomplish this sort of change is with *MenuMaster* (public-exchange; available in both FKey and desk accessory format) or the *MenuKey* desk accessory (part of the *TopDesk* set of desk accessories), or

with *ResEdit*. You can also use *QuicKeys*, *Tempo II*, or a similar program; they, however, will not display the shortcut on the menu the way *MenuMaster*, *MenuKey*, and *ResEdit* will. Also, if you remove *Tempo II* or *QuicKeys* from your System folder, your shortcuts will no longer work, but shortcuts assigned with *MenuMaster*, *MenuKey*, and *ResEdit* are permanent (unless you are notified otherwise by the program during the assignment process).

On the minus side, if you use *MenuMaster*, *MenuKey*, or *ResEdit* to create your menu command shortcuts, they consider shifted and unshifted characters to be the same in most cases. Shift-Command-A and Command-A, for example, are identical shortcuts as far as these programs are concerned. The only exceptions to this rule are the number keys. Standard FKey shortcuts, Command-Shift-number combinations, are not confused with any Command-number shortcuts you may assign to menu commands.

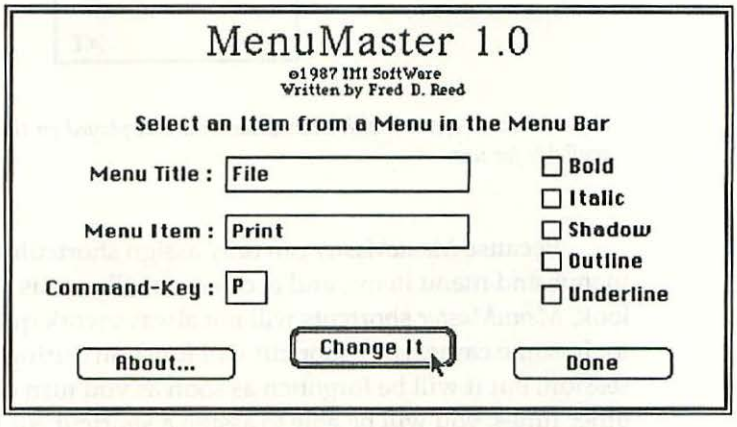
The problem then is not with FKey conflicts but with the fact that *MenuMaster*, *MenuKey*, and *ResEdit* eliminate an entire class of potential shortcuts by ignoring the Shift key. Indeed, they ignore a lot of other possible combinations as well. Those of you who feel constrained by the limited number of key combinations these programs accept may wish to switch to a program, such as *QuicKeys* or *Tempo II*, that makes a distinction between Command-key and Shift-Command-key combinations and permits quite a few other combinations as well (more on *QuicKeys* and *Tempo II* shortly).



Using *MenuMaster* The easiest way to assign shortcuts to standard menu items is to use *MenuKey* or *MenuMaster*. For example, the following shows how to use *MenuMaster* to assign Command-P to the Finder's **Print** command and Command-W for the **Close** (as in close a window) command:

1. Launch *MenuMaster*; the exact method for doing this will depend on whether you are using the FKey or desk accessory version. The *MenuMaster* window should now appear. (If you are using *MenuKey*, you may need to turn MultiFinder off before launching *MenuKey*.)
2. Select the **Print** command from the File menu. The Menu Title and Menu Item text entry boxes in the *MenuMaster* window should now contain the appropriate menu and command names: File and Print. (As you'll see later in the chapter, you can change command and menu names as well simply by editing what appears in these text entry boxes.)

3. If the flashing cursor is not already there, click inside the text entry box labeled Command-Key. Press Shift-P to enter an uppercase P. Remember, you are entering an uppercase P only for legibility and appearances; the actual keyboard shortcut is Command-P, not Command-Shift-P. Also, note that you do not need to hold down the Command key when you fill in the Command-key box; that part of the key combination is assumed. If all this sounds a little too complex, don't worry; *MenuMaster* will not allow you to enter invalid (nonstandard) key combinations, such as Option-key combinations.



After selecting the command to which you want to assign a shortcut, enter the key you want to use in the Command-Key combination.

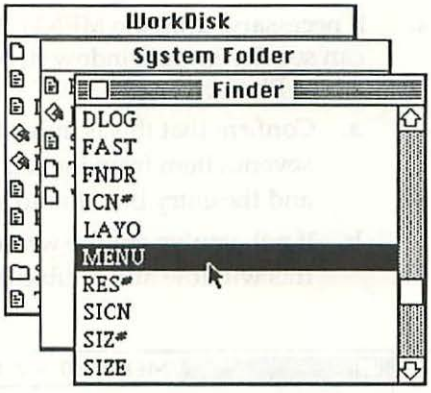
4. Click on the Change It button or press Return.
5. Now select the **Close** command from the File menu. The top two text entry boxes should now display the names File and Close.
6. If the flashing cursor is not already there, click again inside the text entry box labeled Command-Key. Press Shift-W to enter an uppercase W.
7. Click on the Change It button or press Return.
8. Click on the Done button or press Command-D. The new shortcuts are now available for use. (See graphic next page.)

New shortcuts created with MenuMaster *are displayed on the menu and are immediately available for use.*

Because *MenuMaster* can only assign shortcuts to so-called standard menus and menu items, and because not all menus are as standard as they look, *MenuMaster* shortcuts will not always work quite as you expect them to. In some cases, your shortcut will function during the current work session, but it will be forgotten as soon as you turn off your Macintosh. At other times, you will be able to assign a shortcut, and it will appear as usual opposite the command name in the menu, but pressing the key combination gets you nowhere. These problems are annoying but, by and large, harmless.

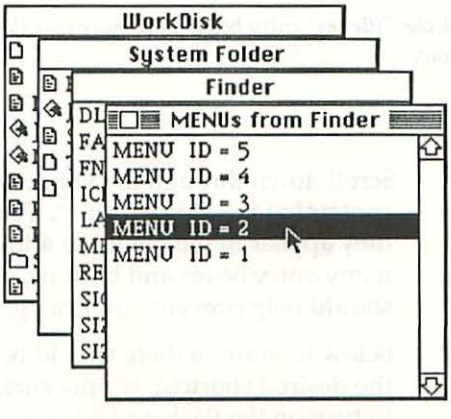
Using *ResEdit* For those who like to see what they're tinkering with, here's how to make the same shortcut assignments as before, this time using *ResEdit*:

1. Open *ResEdit* and then open (a copy of) the Finder, following the directions given in the ResEdit introductory sections.
2. Locate the MENU resource in the Finder window and double-click on it to open the MENUs From Finder window.



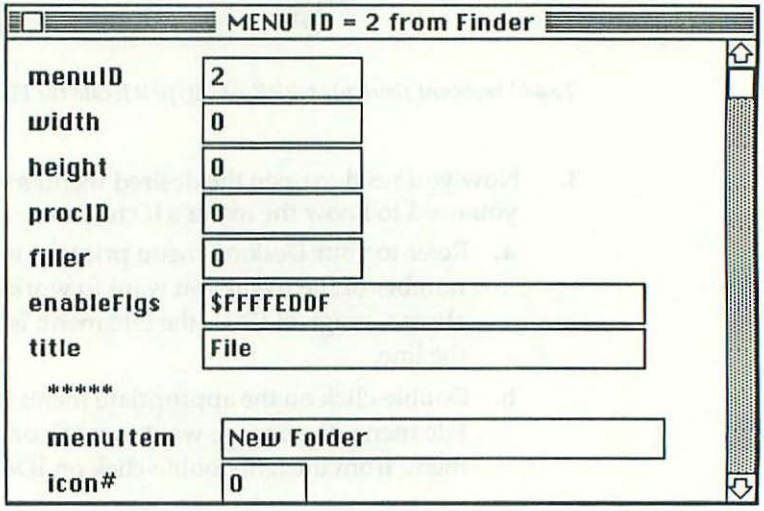
To add keyboard shortcuts with ResEdit, *first locate the Finder's MENU resource.*

3. Now you need to open the desired menu's window. To do this, you need to know the menu's ID number.
   a. Refer to your Desktop menu printout to determine the ID number of the menu you want to work on. The Apple menu is always assigned ID=1, the File menu is ID=2, and so on down the line.
   b. Double-click on the appropriate menu ID number. Since the File menu, the one we want to work on first, is the second menu from the left, double-click on ID=2.



Double-click on MENU ID=2 to open the editing window for the File menu.

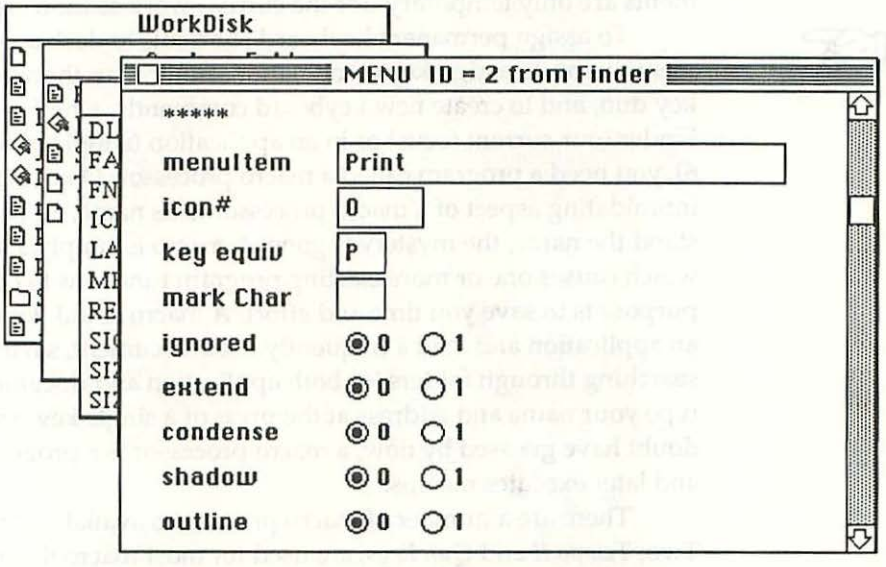
4. If necessary, drag the MENU ID=2 From Finder window up so you can see the entire window (it opens with the bottom clipped off on a Mac Plus screen).
   a. Confirm that this is indeed the correct menu by checking the seventh item from the top. The item name should be "Title" and the entry box should state "File" for this example.
   b. If not, you've got the wrong menu, and you'll have to close this window and double-click on the correct ID number.



Check the Title text entry box to make sure that the window is indeed the File menu editing window.

5. Scroll down through the window until you find a menuItem box containing the word "Print". Commands are listed in the order they appear in the menu, so although the window contains a great many entry boxes and buttons, your Desktop menu printout should help prevent you from getting terribly lost.
6. Below menuItem there should be a box labeled key equiv. To add the desired shortcut, simply click the pointer in the key equiv box to turn on the flashing text entry marker and type an uppercase P

(Shift-P). Remember, type an uppercase P for consistency and appearance's sake only. Command-P (and not Command-Shift-P) is the actual keyboard command. Also as before, the Command key is already assumed to be part of the shortcut and need not be typed here.



Locate the section for the Print command and enter the desired shortcut key in the key equiv box.

7. Now scroll down the list until you reach the Close command, and place a W in its key equiv text entry box.
8. Finally, click on the ID=2 window's close box.
9. Now close the MENU and Finder windows by clicking on their close boxes. When you are asked if you want to save your changes, click on Yes or press Return. Quit *ResEdit*.

Having used *ResEdit* on a copy of the Finder, the one thing remaining for you to do is to replace your old Finder with the new one. When you next use the Finder, the File menu should now display your new keyboard shortcuts.

Shortcuts for Desk Accessories and Fonts The desk accessory and font menus are not standard menus, and thus items within them cannot be assigned shortcuts with *MenuMaster* or *ResEdit*. In *MenuMaster*, for example, both menus will be dimmed (and thus not selectable). Unlike *MenuMaster* or *ResEdit*, *MenuKey* can assign shortcuts to items in font menus, and to desk accessories. In most instances, however, these assignments are only temporary (for the current work session only).

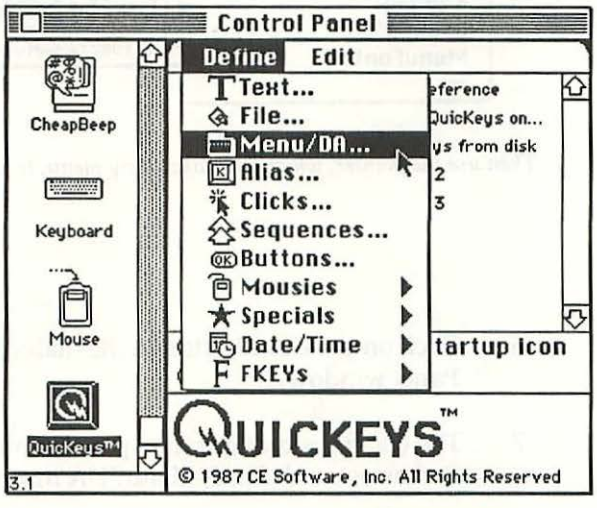
To assign permanent keyboard shortcuts to desk accessories and Font menu items, to assign key combinations other than the simple Command-key duo, and to create new keyboard commands, whether they be in the Finder (our current focus) or in an application (a topic covered in Chapter 6), you need a program called a macro processor. Don't blanch; the only intimidating aspect of a macro processor is its name, and once you understand the name, the mystery is gone. A macro is simply a key combination which causes one or more existing program functions to occur and whose purpose is to save you time and effort. A macro could, for example, launch an application and load a frequently used document, saving you from searching through folders for both application and document. Or it could type your name and address at the press of a single key, and as you no doubt have guessed by now, a macro processor is a program which creates and later executes macros.

There are a number of macro processors available for the Macintosh. Two, *Tempo II* and *QuicKeys*, are used for most macro illustrations in this book. You have, in fact, already seen *QuicKeys* in action (it was used to assign keyboard shortcuts to FKeys in Chapter 3). If you already own a different macro processor, or intend to buy a different one, that's no problem. The details in the examples throughout the book apply specifically to *Tempo II* or *QuicKeys*, but the general principles apply to all macro processors. With your manual in hand to guide you in the exact details, you should be able to create most, if not all, of the macros I recommend in this and the following chapters.

Demo

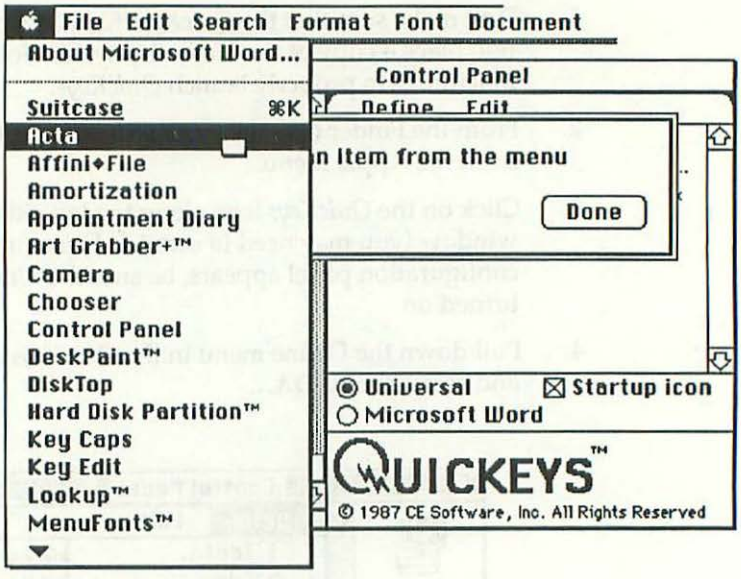
Both *Tempo II* and *QuicKeys* can be used to assign keyboard shortcuts to fonts and desk accessories. The following example shows how you would use *QuicKeys* to assign shortcuts for a desk accessory and a font: Command-Option-O (for outline) to launch *Acta* and Command-Option-B to select New Baskerville.

1. First make sure that the *QuicKeys* file is in your System folder. If not, place a copy of the file in the System folder and restart your Macintosh to properly launch *QuicKeys*.
2. From the Finder or from any application, select the Control Panel from the Apple menu.
3. Click on the *QuicKeys* icon along the left side of the Control Panel window (you may need to scroll to locate it). When the *QuicKeys* configuration panel appears, be sure the Universal radio button is turned on.
4. Pull down the Define menu in the *QuicKeys* configuration panel and select **Menu/DA...**



Select QuicKeys' *Menu/DA... command to assign a keyboard shortcut to a desk accessory or font.*

5. Now, as the ensuing dialog box directs, select *Acta* from the Apple menu. Then go to the Font menu and select New Baskerville. Even if the Font menu is dimmed and appears nonfunctional, drag down to the desired font and release the mouse button.



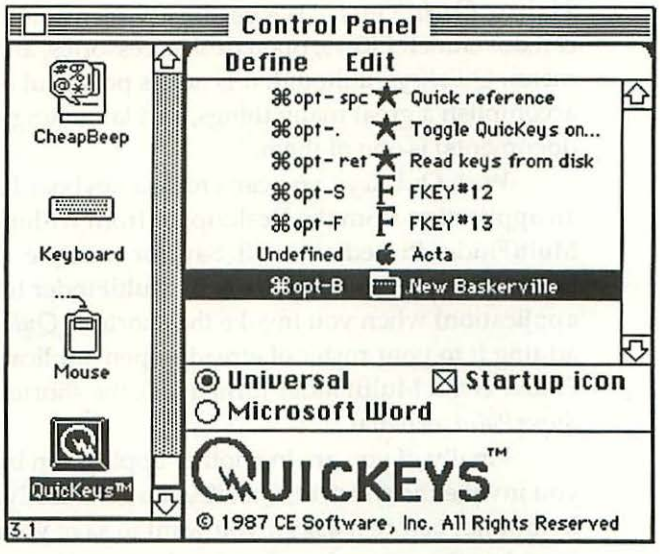
Then use the pointer, which looks like a tiny menu, to select the desired desk accessory.

6. Click on the Done button in the dialog box to return to the Control Panel window.

7. The *QuicKeys* configuration panel should now list the two new, and as yet undefined, Menu/DA macros.

   a. Click on the one labeled *Acta* to select it, then press Command-Option-O. The key combination Command-Option-O should now be listed in front of the *Acta* entry.

   b. Repeat this process for the New Baskerville shortcut: select the New Baskerville entry and press Command-Option-B.



Back at the QuicKeys *configuration panel, highlight your new shortcut and press the key combination you want assigned to it.*

8. Click on the Control Panel close box to return to work.

Your new desk accessory and font shortcuts will now be available from within any application, including the Finder, as long as *QuicKeys* is launched at startup. Don't forget that you can also use *Suitcase*, or *Font/DA Juggler Plus* as described in Chapter 4, to call up desk accessories by name. The number of key presses required is more than that needed to press Command-Option-O, for example, but it still keeps your hands on the keyboard, and when all is said and done, a desk accessory name may be a lot easier to remember than a key combination.

Creating New Commands and Their Shortcuts

Fortunately, we don't have to stop adding keyboard shortcuts just because we've run out of menu items. We can, in fact, fashion all kinds of keyboard commands to accomplish all sorts of Desktop activities, from launching applications to throwing files in the trash.

Demo

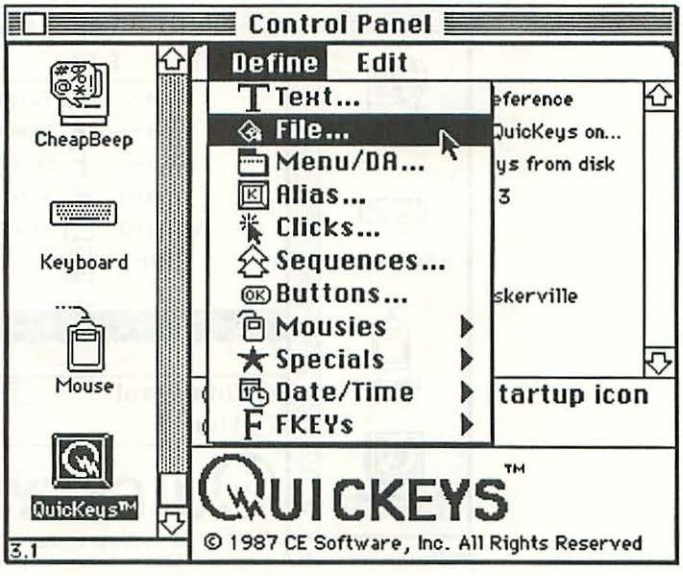
Using *QuicKeys* We've already seen a number of things that *QuicKeys* can do: launch FKeys, open desk accessories, and select fonts from the Font menu. *QuicKeys*, although it is not as powerful as *Tempo II*, can in fact accomplish a great many things, and launching applications (and opening documents) is one of them.

With *QuicKeys*, you can create a keyboard shortcut which will launch an application from the Desktop, or from within any other application, with MultiFinder turned on or off. Say, for example, you create a shortcut that launches *SuperPaint*. If you are in MultiFinder (either at the Desktop or in an application) when you invoke the shortcut, *QuicKeys* launches *SuperPaint*, adding it to your roster of already-open applications. If you are in the Finder (with MultiFinder turned off), the shortcut simply launches *SuperPaint* as usual.

Finally, if you are in another application but not in MultiFinder when you invoke the shortcut, *QuicKeys* automatically quits the program you are in (but not before it asks if you want to save your documents) and then launches the one assigned to the shortcut. You can thus use *QuicKeys* to create a collection of "transfer utility" shortcuts which allow you to pass directly from one application to another without stopping at the Finder. In Chapter 6 we'll talk more about bypassing the Finder with *QuicKeys* shortcuts, transfer utilities, and other programs.

The following example shows how you would create the macro to launch *SuperPaint*.

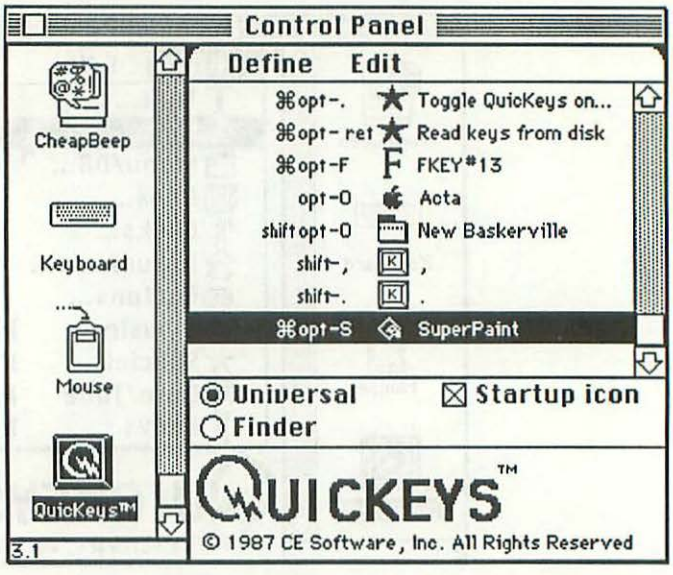
1. As in the earlier *QuicKeys* example, make sure that the *QuicKeys* file is in your System folder. If not, place a copy of the file in the System folder and restart your Macintosh to properly launch *QuicKeys*.
2. From the Finder (or from any application), select the Control Panel from the Apple menu.
3. Click on the *QuicKeys* icon along the left side of the Control Panel window (you may need to scroll to locate it). When the *QuicKeys* configuration panel appears, be sure the Universal radio button is turned on.
4. Pull down the Define menu in the *QuicKeys* configuration panel and select the **File...** command.



Select QuicKeys' ***File...*** *command to create a keyboard shortcut for launching an application.*

5. Now in the ensuing file selection box, locate and select *SuperPaint* (or a *SuperPaint* document, if you want to be able to launch *SuperPaint* and open a particular document all at once). Click on the Open button and then, when the file selection box reappears, click on the Cancel button to return to the Control Panel window.
6. The *QuicKeys* configuration panel will now display the new, and as yet undefined, File macro.
   a. Click on the *SuperPaint* macro to select it if it is not already selected.
   b. Press the key combination you wish to assign to the macro. For example, for the *SuperPaint* macro, press Command-Option-S. (See illustration on following page.)
8. Click on the Control Panel close box to return to your application.

Your new shortcut will run from within any application, including the Finder, as long as *QuicKeys* is installed at startup. As described previously,



At the QuicKeys *configuration panel, highlight the* SuperPaint *macro and press the key combination you want assigned to it.*

from the Finder it will launch *SuperPaint* as usual. If invoked from within an application, the macro will either close the current application for you before launching *SuperPaint* (if MultiFinder is turned off) or add *SuperPaint* to your list of open applications (if MultiFinder is turned on).

Using *Tempo II* *Tempo II* is the newest release of what used to be called simply *Tempo*. Because only *Tempo II* works reliably with MultiFinder, and because *Tempo II* contains so many enhancements over the original, all of the information and instructions in this book refer to the new version, *Tempo II*. If you have on old version of *Tempo*, I strongly recommend that you upgrade to *Tempo II*. (I should also note that a prerelease version of *Tempo II* was used for all of the illustrations in this book. Thus there may be minor discrepancies between what you see here and what the final version of *Tempo II* actually displays on the screen.)

Tempo II can either be launched on startup (using *Tempo II*'s Set Startup file or the techniques described in Chapter 7) or launched when needed by selecting *Tempo II* from the Apple menu. If you have created

Desktop macros (or application macros, as described in the next chapter) and expect them to be available whenever you need them, then you should have *Tempo II* launched automatically at startup. If you do intend to run *Tempo II* all the time, however, be sure to test *Tempo II* with all of your most frequently used applications: Init files, desk accessories, FKeys, and so forth. *Tempo II*, like any so-called resident program, may be incompatible with some programs, and it is best to know this beforehand.

With *Tempo II*, you could, if you wanted to, create any of the menu command key shortcuts described in the previous section. As was noted previously, *Tempo II* may not be the best choice for assigning shortcuts to existing menu commands; remember that *MenuMaster* and *MenuKey* actually put the shortcut up on the menu itself, in case you forget it. But simple menu command shortcuts are only one of the many things that *Tempo II* can do. It can, for example, invoke a number of commands in sequence. It can type in frequently needed pieces of text. It can make choices between alternatives. It can create macros that are specific to one application or universally applicable. It can, in short, automate almost anything you do repeatedly.

For those of you with better memories for names than key combinations, one particularly nice feature in *Tempo II* is the Play By Name option. When Play By Name is turned on, you can select macros by typing their names, instead of using arcane key combinations. *Tempo II* also has a pop-up menu option, so that people who own big screens can call up the *Tempo II* menu no matter where the pointer may be; just press the correct key combination and click the mouse, and the *Tempo II* menu appears beneath the pointer.

While the possibilities for *Tempo II* macros are virtually unlimited, we'll consider just two variants of one example for the moment (there will be others in later chapters). As with the macro we created with *QuicKeys* in the previous section, this example involves the opening of an application from the keyboard. Let's say for the sake of illustration, that you want to create a keyboard command to launch *MicroPhone* from the Desktop. The following is what you would do:

1. Make sure that *Tempo II* is already launched and the *Tempo II* symbol (the Command key's cloverleaf) is in the menu bar. Although we will assume that you are in the Finder (since that's what this chapter is about), you can create this macro in any application.

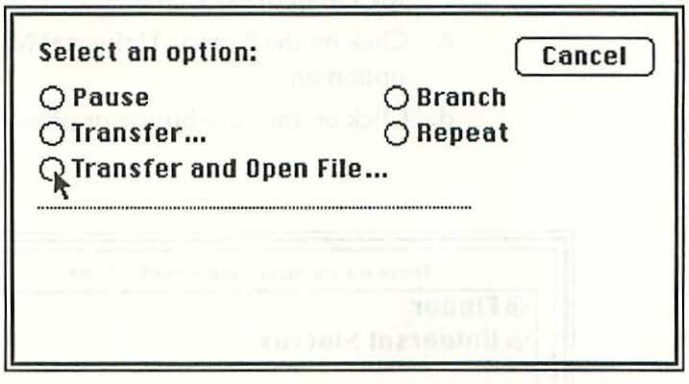
2. Select **Start Recording** from the *Tempo II* menu or press Command-accent grave (`).

To start creating a Tempo II *macro, select the* ***Start Recording*** *command.*

3. Go to the *Tempo II* menu again and select the **Options...** command.

Then choose the ***Options...*** *command.*

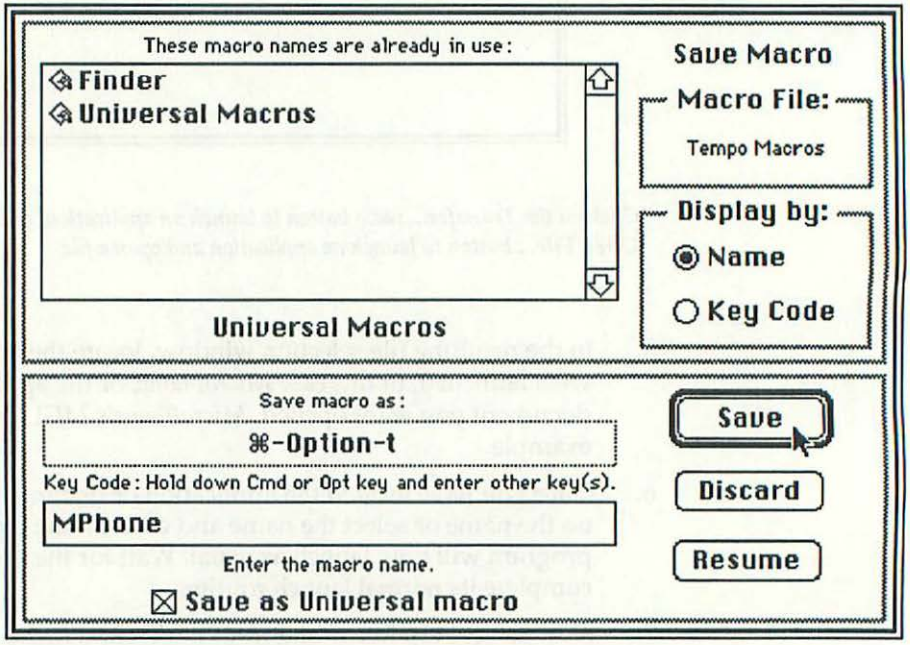
4. In the resulting dialog box, you must click on one of two radio buttons. Which one you click on depends on whether you want to open only the application or open the application and a document.
   a. If you want to launch the application only, click on the **Transfer...** radio button.
   b. If you want to open both the application and one of its documents, click on the **Transfer and Open File...** radio button.



*Click on the **Transfer...** radio button to launch an application, or the **Transfer and Open File...** button to launch an application and open a file.*

5. In the resulting file selection window, locate the application you wish launched, in this case *MicroPhone*, or the application's document you want opened, *MicroPhone*'s MCI Mail file, for example.
6. Once you have located the application or document, double-click on the name or select the name and click on the Open button. The program will now launch as usual. Wait for the program to complete its normal launch routine.
7. Now select Stop Recording from the Tempo menu or press Command-accent grave (`) again. This will open the Save Macro window.

8. If you made a mistake in recording your macro and want to try again, click on the Discard button. Otherwise:
   a. Press the key combination you want to assign to the macro; in this example, you might assign Command-Option-T (for telecommunications).
   b. In the Enter Macro Name box, give your macro a brief, descriptive name such as `MPhone` or `MCI` (depending on whether you chose to launch MicroPhone only or to open the MCI Mail file as well).
   c. Click on the **Save as Universal Macro** check box to turn that option on.
   d. Click on the Save button or press Return.



After you have stopped recording, press the desired key combination and enter a name for the macro.

Now every time you press Command-Option-T, *MicroPhone* will launch and if you programmed it to do so, open a specific file or invoke a certain command. If you are in MultiFinder, the macro will add *MicroPhone* to the applications you already have open. If MultiFinder is turned off and you are in another application, *Tempo II* will quit that application before launching *MicroPhone*.

You can, of course, continue to assign keyboard commands that open other applications, making it possible to launch virtually any program from the keyboard, and other Desktop activities can be automated in the same way. With a little imagination, you can use *Tempo II* (or *QuicKeys*, for that matter) to convert nearly all the Desktop activities you regularly perform into simple keyboard commands.

Although the function keys of the extended keyboard offer all sorts of additional key combinations that may be easier to remember and simpler to type than the standard Command-Shift-Option-letter key combinations, *Tempo II*'s Play By Name option may make the search for suitable key combinations a thing of the past. With the Play By Name option, you don't have to remember all sorts of key combinations and the macros they are attached to; instead, you need remember only one, the one that turns on Play By Name.

The MCI Mail macro created in our example can be invoked by pressing the Play By Name key combination (initially set to Command-Option-M, but you can change it to anything you like with the **Configure...** command), typing the macro name `MCI` and pressing Return. Thus if you are careful about how you name your macros, you can quickly invoke any macro by calling it by its English name. Since *Tempo II* needs only enough of the name to distinguish the requested macro from any other macros you may have created, you don't really even need to use the whole name; just the first few letters will do.

The advantage of such a system is threefold, at least: carefully chosen names are easier to remember than odd key combinations, the number of possible names is far greater than the number of possible key combinations, and the Play By Name method does not conflict with any of the key combinations your applications may already have assigned. True, Play By Name requires a few more key presses than the old way, but you can always use key combinations if you want to. I predict, however, that many *Tempo II* owners will switch almost exclusively to Play By Name, since well-chosen macro names are so much easier to remember.

Custom Keyboards

Not everyone likes Apple's Macintosh keyboards. For some it is the shape, for others it is the height or the angle or the placement of the keys. For these people, there is little I can recommend except to consider a Mac-compatible keyboard from some other manufacturer. But there are some people, perhaps many people, who are not happy with the keys or key combinations to which certain characters are assigned. Some, for example, are constantly frustrated by the fact that Shift-period yields a right angle bracket, not another period, and Shift-comma a left angle bracket and not another comma. Many LaserWriter owners are less than happy that printer's quotation marks (the curled kind) require an obscure key combination, but typewriter quotation marks (the straight kind) are right there on the keyboard. For these people, there is something we can do.

The answer to dissatisfactions such as these is to remap the keyboard—or at least that part which offends. If remapping the keyboard interests you, keep in mind that, by and large, Mac keys are not amenable to moving to new locations on the keyboard. Although the differences are not easy to see, Macintosh keys are specially sculpted according to where they are found on the keyboard. As a result, moving keys around can produce an odd-feeling keyboard.

If you make radical changes to the keyboard (such as converting it to Dvorak layout), moving the keys to match their new assignments may not be a good idea. Instead, you should make a keyboard map so that you and anyone else using your Macintosh can figure out which key types what characters. I'll show you a quick and easy way to do that later in this chapter.

Using *MacKeymeleon*

One way to remap the keyboard is to use a keyboard customization program such as *MacKeymeleon*. *MacKeymeleon* allows you to choose the characters assigned to any number of keys, any way you like. With this program, you can even create a variety of specialized keyboards and, with the *MacKeymeleon* desk accessory, switch from one to another.

Although *MacKeymeleon*'s abilities are impressive, I don't recommend going whole hog and creating an abundance of custom keyboards for every

eventuality. That would just make things too confusing. Rather, I suggest that you take stock of what it is you don't like about your current keyboard and then reassign keys as needed to eliminate the aspects you don't like. Finally, take the new keyboard assignments and install them permanently, so that they are in effect at all times. Not only is this less confusing, but it also means that you do not have to install the *MacKeymeleon* desk accessory in the System file.

There are many reasons why one might want to reassign characters on the keyboard. You may already have a list of such changes in your head, if not written down. I know I certainly do. Here are some of the "problems" I would like to see fixed:

- The printer's quotation marks moved to the key with quotation marks on the label (instead of Shift-Option-bracket, etc.).
- The em-dash (the extra long dash, equivalent to a double-dash on a typewriter, obtained on the keyboard with Shift-Option-dash) changed to a more readily accessible key assignment.
- Periods and commas produced when the period and comma keys are shifted (as well as unshifted).
- The copyright symbol moved to Option-C (so I can remember and find it).

You, no doubt, can think of others, varying in complexity from simple character swaps to complete keyboard remapping (to create a Dvorak layout, for example).

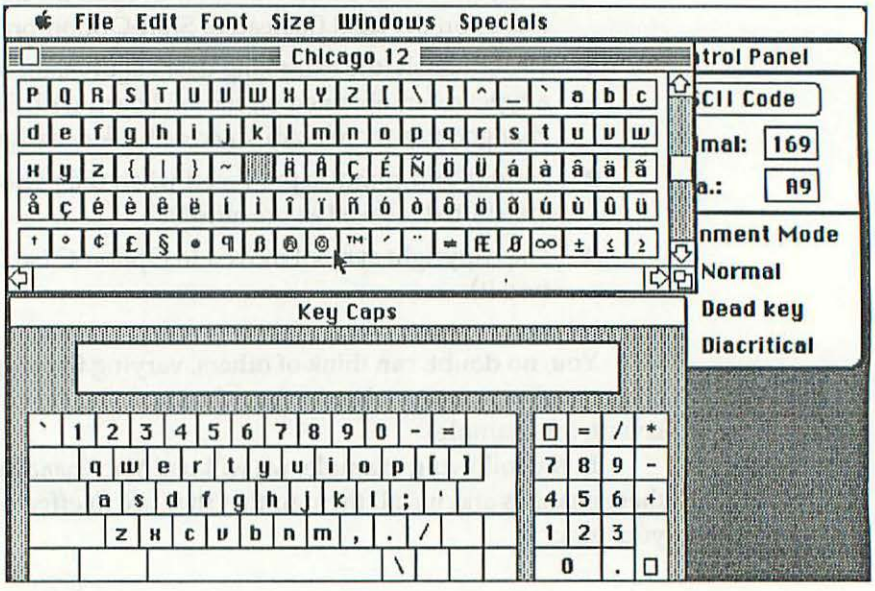
In the following example, we will use *MacKeymeleon* to make some of these changes and install them so that they are in effect whenever you use your Mac.

1. Make a copy of your System file on a floppy disk. For now, put the disk aside.
2. Launch the *MacKeymeleon* keyboard editor. When the main editor screen is in view (it should contain a font window, a standard Key Caps window, and a *MacKeymeleon* Control Panel), expand the Font window so that you can see the whole length of each character row without having to scroll left or right. (Don't be concerned if you cover up part of the Control Panel.)

3. Open the USA keyboard file from the Apple Keyboards folder on the *MacKeymeleon* disk. This file contains the standard US keyboard assignments, the ones with which we are all familiar. This is the file on which we will work our changes.

4. For starters, let's try a simple, single character reassignment: moving the copyright symbol from Option-G to Option-C.

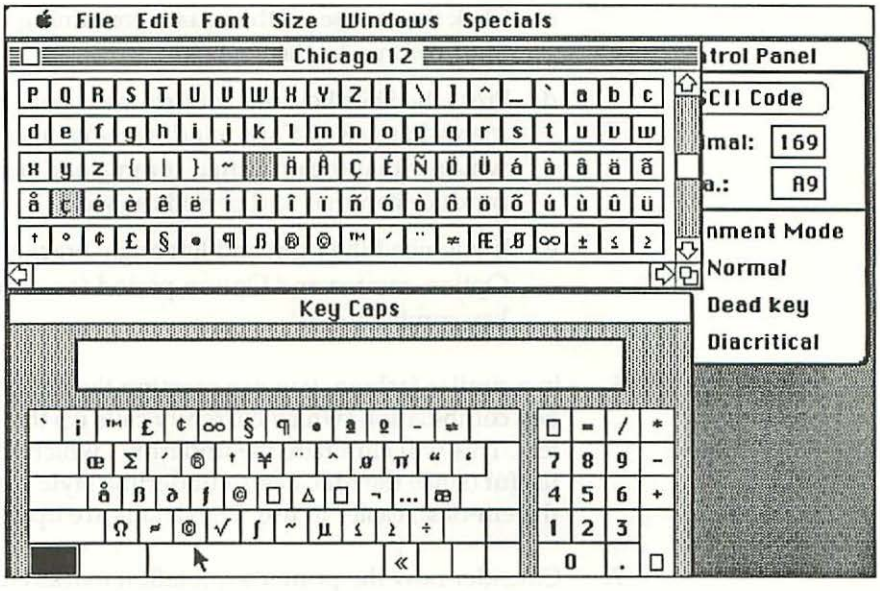
   a. Click on the Font window title bar to make the Font window active. Locate the square containing the copyright symbol in the Font window (you may have to scroll down to see it). When you find it, click on it once with the mouse. The copyright symbol square should now flash on and off.



In MacKeymeleon's *Font window, click on the square containing the copyright symbol.*

   b. Turning to the keyboard (the real one, not the one in the Key Caps window), press the key combination you want assigned to the copyright symbol: Option-C. The square in the Font window should stop flashing.

c. Now check Key Caps to confirm that the new assignment you just made has indeed been recorded. When the Option key is depressed, the C key in the Key Caps window should now display the copyright symbol, not the ç (which was what Option-C used to be).



Hold down the Option key and check the Key Caps window to confirm that the copyright symbol is now assigned to Option-C.

d. Finally, return to the Font window and note that the square containing the ç is now gray. This means that the ç is no longer available on the keyboard. If you use the ç, then you should reassign it to another key combination—to the copyright's original position, for example. To do so, first click on the gray box containing the ç; it should then start flashing. Then press Option-G on the keyboard. That's all there is to it. The ç square will no longer be gray, and the Key Caps window should show the ç assigned to Option-G.

5. Now let's make Shift-comma and Shift-period produce commas and periods, instead of angle brackets.

   a. Locate the squares containing the comma and the period in the Font window.

   b. Click the mouse on the square containing the period. Press Shift-period on the keyboard.

   c. Click the mouse on the square containing the comma. Press Shift-comma on the keyboard.

   d. Press the Shift key and confirm your new keyboard assignments in the Key Caps window. Note also in the Key Caps window that the unshifted comma and period keys still produce commas and periods as well.

   e. If you need the right and left angle brackets, reassign them to Option-comma and Option-period (or some other convenient key combination).

6. In a similar fashion, you can reassign the em-dash to the Shift-dash key combination (which currently calls up the underline character). This will eliminate the underline, which really isn't very useful (since the Mac has an underline style command), and make the em-dash easier to use. The details are up to you.

7. Consider now the printer's quotation marks currently assigned to Option and Option-Shift combinations of the right and left bracket keys. Reassigning these characters is a somewhat complex maneuver, since there are actually four different quotation marks: right and left for both single and double quotation marks. Complex though it may be, this can be a very useful change for LaserWriter owners who want easy access to these professional-looking characters. Assigning them to the quotation mark key has the drawback, however, of making the quotation mark key more difficult to use; whenever you use it, you must always be sure that you pick the correct quotation mark, either the right one or the left one. For ImageWriter owners, this change also has its good and bad aspects. On the ImageWriter, the printer's quotation marks in some fonts do look better than the typewriter-style marks; but in others they actually look worse, and in some ImageWriter fonts there are no printer's quotation marks.

If, after considering the pros and cons, you decide to move the printer's quotation marks, the following is one way to do it:

a. Locate the printer's quotation marks in the Font window. Remember, these are the quotation marks which curl to the right or the left. You should find them near the end of the character list.

b. First reassign the single quotation marks. Click on the opening single quotation mark (') character box in the Font window, then press the quotation mark key (by itself, not with any other key). Now click on the closing single quotation mark (') character box and then press Option-quotation mark.

c. Now reassign the double quotation marks. Click on the opening double quotation mark (") in the Font window, then press Shift-quotation mark. Now click on the closing double quotation mark (") and then press Option-Shift-quotation mark.

d. If you want to retain access to the typewriter-style quotation marks, you can reassign them to two of the four key combinations originally used for the printer's quotation marks: Option-right bracket and Option-left bracket. If you want to retain access to æ and Æ (which were also lost in the change), you can reassign them to the other two key combinations originally assigned to printer's quotation marks, Option-Shift-right bracket and Option-Shift-left bracket.

8. If you have any other keyboard changes you would like to make, do them now. Remember, simply click on the character you want to reassign in the Font window, then press the key or key combination to which you want to assign it.

9. Finally, you must install your new keyboard in the System file. To do this so that it is activated every time you use your Mac, follow these steps:

   a. First, save your keyboard as a document for backup purposes, in case you want to alter it in the future, or in case you want to install it elsewhere. To do so, select **Save as...** from the File menu, give the new document a name, such as `USA.custom`, choose a disk and folder to save it in, and click on the Save button.

b. Now install your keyboard so that it is in effect whenever you use your Macintosh. Select **Install at Startup...** from the File menu. Insert the disk containing the copy of your System file; the name of the disk should appear above the Eject button. If not, click on the Drive button until the name of the disk with the System file appears. Click on the Install button.

If you now replace your old System file with the new one containing the custom keyboard map and then restart your Mac, your new character assignments will be available to you at all times, from that point on. To change the assignments, simply repeat the process on the backup keyboard document and install it as before to replace the older keyboard with the revised one.

Using *QuicKeys*

If you feel that the key changes you want to make are not sufficient to justify buying a keyboard customization program, you could use your macro program instead. With *Tempo II* this is not very practical, since you would have to reassign the characters you want to move to Command-Key or Option-key combinations; you could not, for example, make the printer's quotation marks available with the press of a single key. With *QuicKeys*, however, you can reassign any character to any key you like. With some effort, in fact, you could remap the entire keyboard with *QuicKeys*. I don't suggest it however, since after a while it would be easy to get lost among all your new keyboard assignments; unlike *MacKeymeleon*, *QuicKeys* does not remap Key Cap's keyboard layouts so that you can keep track of what you've accomplished so far. But for a manageable number of changes, *QuicKeys* works quite well.

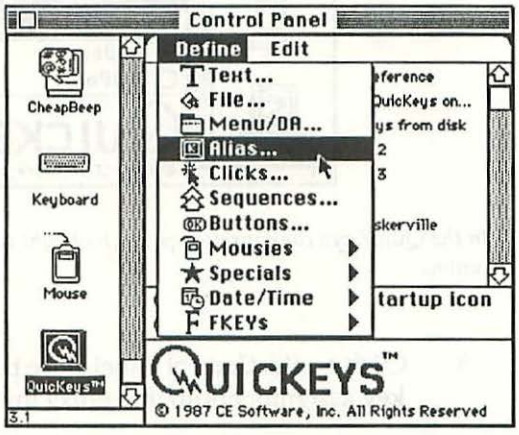
Demo

For example, the following steps show how you would use *QuicKeys* to make the right angle bracket (Shift-period) produce a period and the left angle bracket (Shift-comma) produce a comma:

1. As always, make sure that the *QuicKeys* file is in your System folder. If not, place a copy of the file in the System folder and restart your Macintosh to properly launch *QuicKeys*.
2. From the Finder or from any application, select the Control Panel from the Apple menu.

3. Click on the *QuicKeys* icon along the left side of the Control Panel window (you may need to scroll to locate it). When the *QuicKeys* configuration panel appears, be sure the radio button labeled Universal is turned on.

4. Pull down the Define menu in the *QuicKeys* configuration panel and select the **Alias...** command. A dialog box will appear.

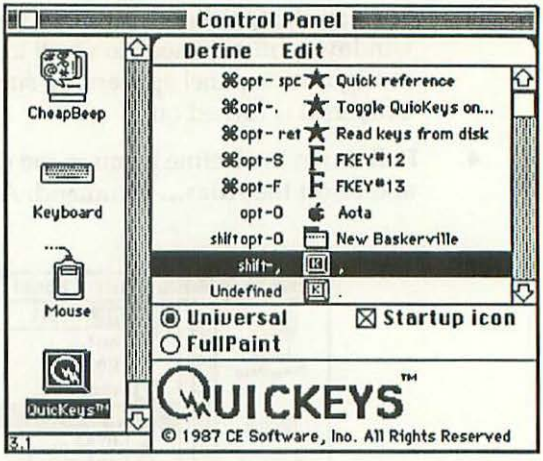


Select QuicKeys' **Alias...** *command to reassign the characters on the keyboard.*

5. Now press the key (or keys) displaying the character you want to assign to a different key. For example, if you want to replace the right angle bracket with a period and the left angle bracket with a comma, you would press a period and then a comma (*QuicKeys* defines these as two separate alias requests).

6. Click on the Done button in the dialog box to return to the Control Panel window.

7. The *QuicKeys* configuration panel should now list the two new and as yet undefined alias keys.
   a. Click on the listing containing the comma, then press the left angle bracket (Shift-comma). The key combination, Shift-comma, should now be listed to the left of the comma.
   b. Repeat this process for the period: select the entry displaying the period and press Shift-period. (See graphic next page.)



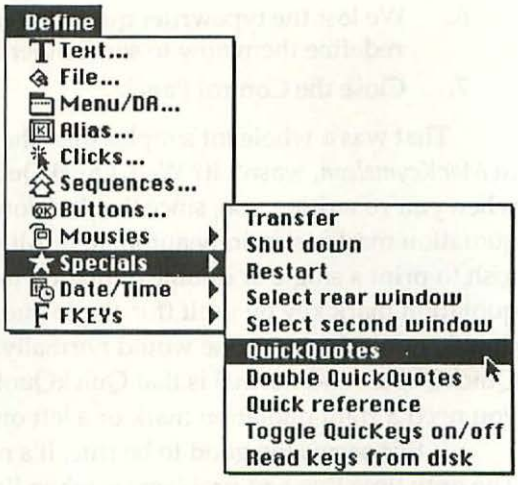
In the QuicKeys *configuration panel, highlight the comma entry and then press Shift-comma.*

8. Click on the Control Panel close box to return to work. Your new key assignments go into effect immediately.

Of course, you can continue to reassign keys, as in the *MacKeymeleon* example, to remap your keyboard however you wish. Your new keyboard assignments will now be universally available, as long as *QuicKeys* is installed.

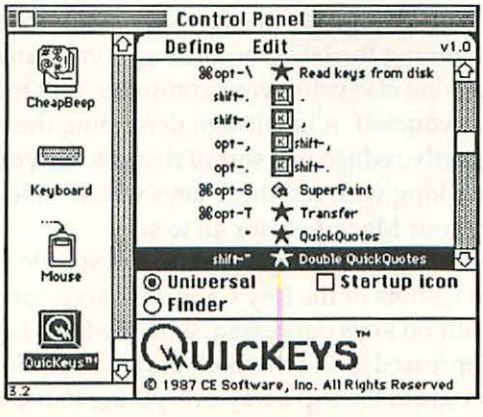
An important and unusual difference between *MacKeymeleon* and *QuicKeys* is the special way in which *QuicKeys* handles the printer's quotation marks. Instead of remapping them with the **Alias...** command, in a manner parallel to that employed in *MacKeymeleon*, you use a special command expressly designed for printer's quotation marks. First let's see how it's done, then we'll take a look at how it works:

1. Run through steps 1, 2, and 3 of the previous set of instructions to open the *QuicKeys* configuration panel.
2. Pull down the Define menu in the *QuicKeys* configuration panel and select the **Specials** command. A submenu will appear to the right; drag the mouse into the submenu and select **QuickQuotes**. Release the mouse button.
3. Repeat step 2, except this time select **Double QuickQuotes**.



To assign printer's quotation marks, select the QuickQuotes and Double QuickQuotes from the Specials submenu.

4. Back at the configuration panel, select entry for QuickQuotes. Press the quotation mark key.
5. Now select the Double QuickQuote entry. Press Shift-quotation mark.



In the QuicKeys *configuration window, select the Double QuickQuotes entry and press shift-quotation mark.*

6. We lost the typewriter quotation marks in this process, so you can redefine them now to some other key or keys if you like.
7. Close the Control Panel.

That was a whole lot simpler than the shenanigans we went through in *MacKeymeleon*, wasn't it? Well, QuickQuotes are a whole lot easier to use when you're writing, too, since the decision of whether to enter right or left quotation marks is entirely automated. All you have to do is tell the Macintosh to print a single or double quotation mark by pressing either the quotation mark key by itself (for single quotes) or Shift-quotation mark (for double quotes), just as one would normally. The difference between QuickQuotes and normal is that QuickQuotes is smart enough to know if you need a right quotation mark or a left one.

If this seems too good to be true, it's not. It works almost every time. The only time I've had problems is when I'm editing text and inserting missing quotation marks; then QuickQuotes gets confused, since the immediately preceding letter is not always the one you just typed. But this minor failure in no way diminishes the value of QuickQuotes. In fact, until QuickQuotes, I had resolved that it was back to typewriter quotes for me, since the hassles of entering printer's quotes from the keyboard was more than I could bear. So for those who use printer's quotation marks frequently, I can't recommend anything better.

Making a Custom Keyboard Map

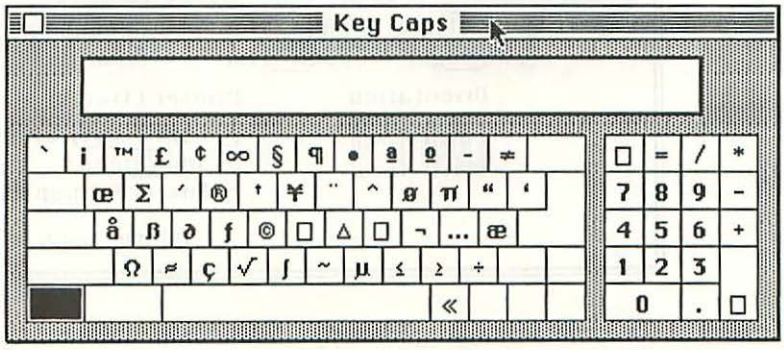
If you change (even a few) key assignments without moving the keys or changing the labels accordingly, you can cause considerable confusion for anyone else using your computer, and maybe even cause a little confusion for yourself. A small sign describing the relevant keyboard changes can greatly reduce this sort of distress. Or you could create a keyboard map, marking with red those keys which have new assignments, and post it next to your Macintosh for all to see.

Making a keyboard map is simple. All you need to do is take screen snapshots of the Key Caps desk accessory window in each of four states: with no keys depressed, with the Shift key depressed, with the Option key depressed, and with both Shift and Option keys depressed. Use a Paint program to strip away everything in the screen shots but the Key Caps windows themselves, and then combine them in one (if you're good at packing) or two Paint documents. Print the documents and voila, a keyboard map.

The problem is that capturing the Key Caps window in the correct state is difficult. If you attempt to take a normal screen shot (using Command-Shift-3), Key Caps intercepts the Command and Shift keys and, as a result, you invariably get a picture depicting the Key Caps window with the Command and Shift keys depressed. This is not a very useful Key Caps window, actually.

There are two ways around this problem. One is to use the *Camera* desk accessory as described in the menu map description earlier in this chapter. The other method involves no special software. Instead, it employs the following elegant and simple trick.

1. First open Key Caps.
2. Take the mouse pointer and place it over the Key Caps window's title bar. Press the mouse button down and continue to hold it down until you reach step 5.
3. Press the screen-shot key combination, Command-Shift-3, long enough to get the message through. Then release the keys.
4. Now press and hold the keys you want to register in the Key Caps window. If that is no keys at all, then press nothing. If it is the Shift key, then press that key. You get the idea. Continue to hold the keys down; do not release them until step 5 is complete.
5. Now release the mouse button. If you've followed the directions to the letter, the keys you are currently pressing will register in the Key Caps window, the correct screen shot will be made (you'll hear your disk drives whir, or if you have a hard disk, see its light flash), and then, only then, should you release the keys you've been holding down.



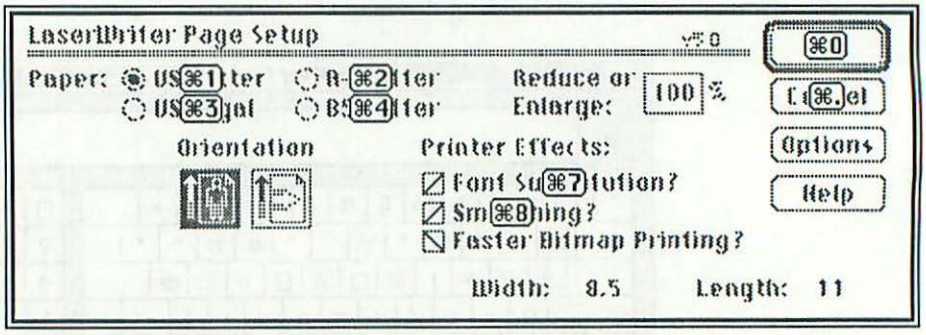
To capture a screen shot like this, you need to use Camera, *or to trick the screen shot FKey.*

Repeat this procedure with different keys depressed for the four Key Caps windows you'll need. Then launch your Paint program to assemble and put the finishing touches on your keyboard map.

Custom Dialog Boxes

Dialog boxes are one of those things that can make a dedicated keyboard user tear his or her hair out. Just when you thought it was safe to go back to the keyboard, up pops a dialog box that refuses to accept any keyboard input except perhaps the Return key as a substitute for the OK button. And, of course, you don't want to click on OK, or, if you do, you want to select a few options beforehand. So return to the mouse you must.

A few programs provide keyboard shortcuts for the buttons in some dialog boxes; a few are so well designed that nearly every conceivable choice is assigned an equivalent keyboard command. *Word*, for example, allows you to navigate almost all of its dialog boxes, complex as some of them are, using keyboard commands corresponding to the first letter of each button or option. Even better, in a *Modern Jazz* (Lotus) dialog box, if you hold down the Command key, then command-key combinations are superimposed over the various buttons and options to show you exactly what keys you need to press. But most programs, the Finder included, are far less accommodating to keyboard enthusiasts.



In Modern Jazz *dialog boxes, if you hold down the Command key, keyboard shortcuts are displayed over the various buttons and options.*

One resolution of this age-old Macintosh conflict is an Init resource called *DialogKeys* which is part of the *QuicKeys* package. As with any Init, all you need to do is place a copy in your System folder and restart your Macintosh to get it running. Then, the next time a dialog box appears, here's what you do:

1. Press Command-space bar or Command-Tab to turn on *DialogKeys*.
2. One of the buttons in the dialog box will start blinking. If that's not the button you want to click on, press Command-Tab. The next button will begin to blink.
3. Continue moving from button to button (and check box to check box), by pressing Command-Tab to move forward and Shift-Command-Tab to move backward.
4. When the button or check box you want to click on is blinking, press Command-space bar again (or Command-Return or Command-Enter).

If a dialog box is complex, this hopping from button to button can involve quite a few keystrokes, enough to make you wonder if there isn't an easier way. Well there is, and it's right in *QuicKeys*. With *QuicKeys*, you can assign a specific key combination to any button or option in a dialog box, either for all your applications, or for a specific application. The key combination you choose becomes part of your macro collection, so it can't duplicate existing macros or other useful keys and key combinations. (As a result, you cannot assign Return to all OK buttons, for example, since that would eliminate the Return key's usual and most important function, ending Paragraphs. You can assign some other key combination, however, such as Command-Return.)

For example, some Cancel buttons, such as those in *Word*, already recognize the keyboard shortcut Command-period (which is also used in many programs to cancel printing), but many do not. You can thus use *QuicKeys* to standardize the Cancel button shortcut over all your applications. By way of illustration, the following steps show how you would assign Command-Option-period to the Cancel buttons in all your dialog boxes. We're assigning Command-Option-period rather than Command-period, since the latter might eliminate the cancel printing function in some applications.

Demo

1. As before, open the *QuicKeys* configuration panel. Make sure that the **Universal** radio button is turned on.
2. Pull down the Define menu and select the **Buttons...** command.
3. In the dialog box which follows, enter the name of the button or option to which you want to assign a keyboard command. You must enter it exactly as it appears in the dialog box. For example, if one or more of your applications do not have a shortcut for the Cancel button, type in `Cancel`.
4. Click on the OK button to close the dialog box.
5. Select the Cancel button entry in the *QuicKeys* configuration window if it is not already highlighted.
6. Press Command-Option-period.
7. Quit the Control Panel by clicking on the close box.

From this point on, any dialog box in any application containing a Cancel button, including the Finder, can be canceled from the keyboard by pressing Command-Option-period.

For the many relatively simple dialog boxes that exist, *DialogKeys* is a convenient way of avoiding the mouse and keeping your fingers on the keyboard. But complex dialog boxes can require too many keystrokes with *DialogKeys*, and not all programs work with *DialogKeys*; *DialogKeys* 1.0, for example, did not function properly with *Word* 3.01 and *FullPaint* 1.0 on my Macintosh Plus. For programs that don't work with *DialogKeys*, for more complex dialog boxes, or to create specific responses to specific dialog box buttons or options, *QuicKeys*' **Buttons...** command is a better choice.

Custom Icons

Just as standard icons send messages which take only a split second to receive, icons that are altered in some meaningful way can convey useful information quickly and efficiently. They can tell you the version number of the application, for example. They can inform you that this application is the one with the custom menus. They can send almost any kind of telegraphic signal to the person viewing the screen.

Icons serve many functions in the Macintosh world, and each kind can be altered however you please. Unfortunately, icons are tricky little numbers to work with and require some patience. But the effort is often worth the trouble, especially if, like me, you use Icon View as often as you can. Since it isn't possible to devote an entire chapter to the ins and outs of icons and icon editing, we will consider only one kind of icon customization here, the simplest (if you can call it that) and one of the most useful: the alteration of an application icon.

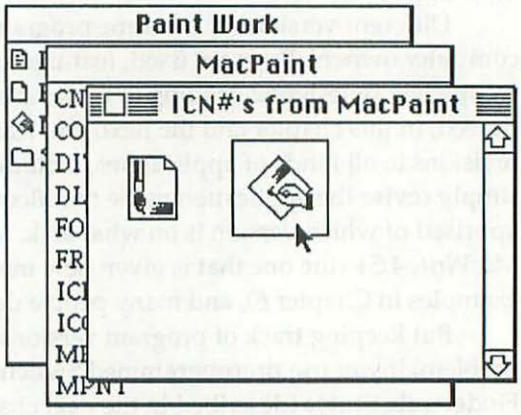
Different versions of the same program are a fact of life for personal computer owners. Bugs are fixed, features are added, or applications are completely redesigned, leading to updates of major or minor proportions. Indeed, in this chapter and the next, you learn how to make your own revisions to all kinds of applications, including the Finder. Now you could simply revise the application name to reflect these changes and to keep you apprised of which version is on what disk: *MacWrite* 2.3, or *MacWrite* 4.5, or *MacWrite* 4.5+ (the one that is given new menu commands following examples in Chapter 6), and many people do just that.

But keeping track of program versions in this way can present a problem. If you use preprogrammed Switcher sets, macro programs, or Finder-substitutes (described in the next chapter), all of which call up specific programs by searching for their names, you would have to change all references to a given program each time you changed its name to reflect a new update or a new customization. An easy way around this problem is to alter the application's icon so that it, rather than the program name, indicates the program's version number or the presence of special custom features. Since icons are used only by people to locate applications, none of your macros, Finder-substitute buttons, or Switcher sets will need to be changed as a result.

Although the procedure for customizing application icons is a little complex, it is not difficult. For the following illustration, assume that you own *MacPaint* 1.5 and that you want to change the icon to show its version number, as well as the fact that your working copy includes special menu commands (added using techniques to be described in the next chapter). Before we start, put *MacPaint* on a floppy disk, which we will call *MacPaint* Work, containing nothing else of importance. Here's what to do:

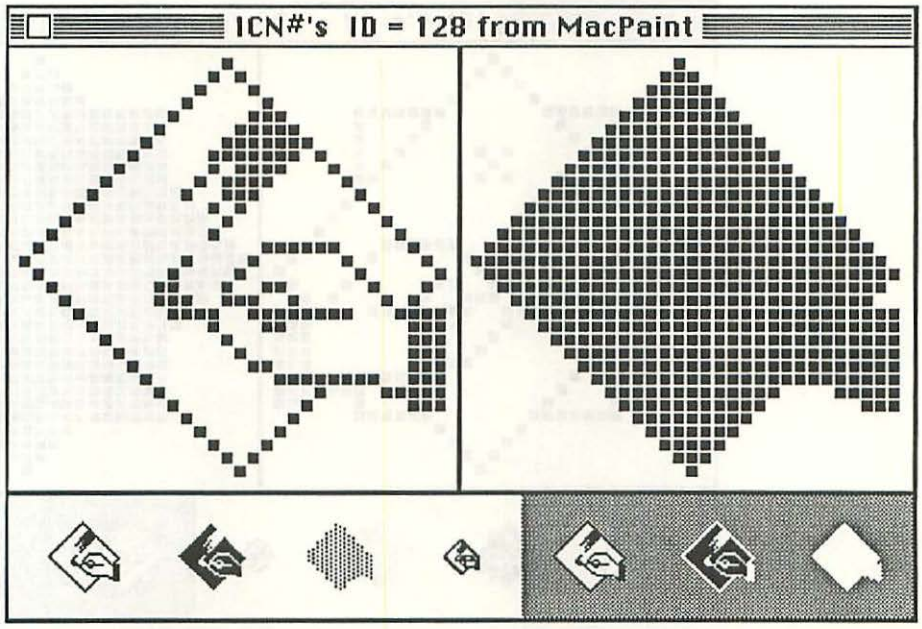
1. Open *ResEdit* and then double-click on the *MacPaint* listing in the *MacPaint* Work window.

2. In the *MacPaint* window, locate the ICN# resource and double-click on it (do not select ICON by mistake). This will open a window containing two icons, *MacPaint*'s application and document icons. The document icon is an upright page with a brush emblazoned on the left side; the application icon is a hand poised over a tilted square.



Open MacPaint*'s ICN# window to see the program's two icons, one for documents and the other for the application itself.*

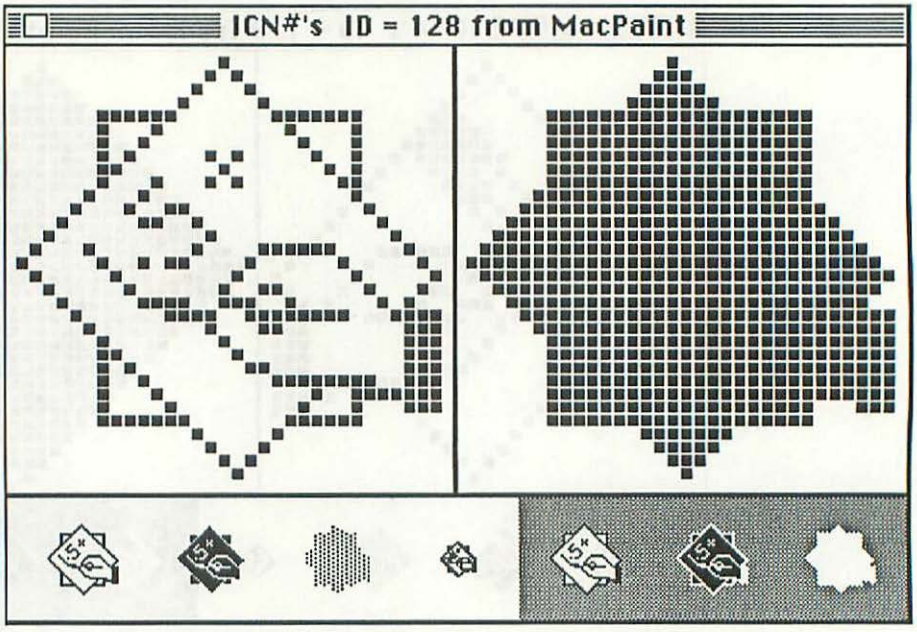
3. Double-click on the *MacPaint* application icon, since this is the one you want to edit. This will open the icon editing window (it will be labeled "ICN#'s ID=128 from *MacPaint*"). If necessary, drag the window higher on the screen so that it is not cut off at the bottom. The major portion of the window is occupied by two enlargements. The one on the left is the actual icon, the one on the right is what is called the mask. (A mask contains instructions for drawing its icon on the screen: at every white bit, the Finder leaves the background pixel unchanged; at every black bit, the Finder replaces the background pixel with the corresponding icon pixel.) Beneath them is a row of real-size icons in various states (not selected, selected, and so on), one in small icon size, the rest in standard icon size.



The icon editing window has two editing areas for the icon and its mask, and it also displays icons in different states and on different backgrounds.

4. Go to work on the large icon on the left using the pointer and standard fatbits techniques (click on a white bit to turn it black, a black bit to turn it white). As you work, keep an eye on the results of your efforts in the real-size icons at the bottom of the window.

5. If you have changed the shape of the icon in any way, choose the Data->Mask command under the ICN# menu to transfer the new shape to the mask. (You can make what is called an animated icon, one that changes into an entirely different icon when it is selected, by playing further with the mask. But this is a very difficult task without the right tools, and *ResEdit* is not the right tool, so I don't recommend it.)



If you have changed the icon's shape, invoke the Data->Mask *command so that the mask exactly matches the new outline.*

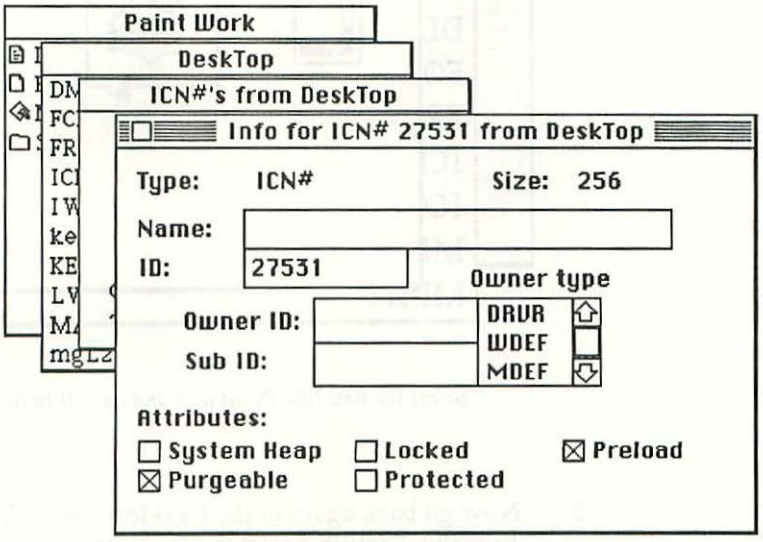
6. Once you have the icon the way you want it and a mask to match, close the editing window.

If you want *MacPaint*'s new icon to appear on this disk (and you very likely do), then you must install the new icon in the disk's Desktop file, which means there is a bit more work remaining. That work can be accomplished manually or automatically. The automatic method is easier, but it has some important drawbacks. Instead, I suggest you use the manual method.

To do it automatically, you simply hold down the Command and Option keys as you quit *ResEdit*. This will rebuild the Desktop file, thereby throwing out the old icon and inserting the new one. The problem is, this procedure will likely rebuild the Desktop file on other disks as well, and when you rebuild the Desktop file, all the information written in Get Info boxes is erased. Thus this method is only recommended if your hard disk (if you have one) is turned off and no important floppy disks are currently in any drive.

The safer route, and one that reveals a bit about how the Macintosh Desktop file works, is to do the work yourself. The following is the manual method:

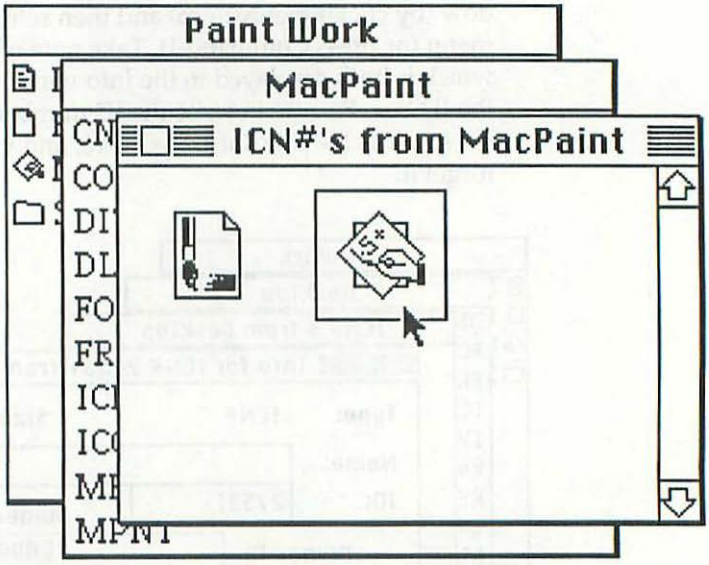
1. Go back to the *MacPaint* Work disk window and locate the Desktop file. Remember, every disk has a Desktop file, so be sure that you have the one on the correct disk.
   a. Double-click on the Desktop file to open its window.
   b. Locate the ICN# resource in the Desktop file window and double-click on it to open the ICN# window. Note that there is a copy of the original, unaltered *MacPaint* application icon here.
2. Select the original *MacPaint* icon in the Desktop file's ICN# window (by clicking only once) and then select **Get Info** from the File menu (or press Command-I). Take note of the icon's ID number, which is both displayed in the Info window's title bar and listed in the ID box. Be sure to write the ID number down; it's probably not the same as the one illustrated here, and you're in trouble if you forget it.



The Info window will tell you the ID number of the original MacPaint *application icon still in the Desktop file.*

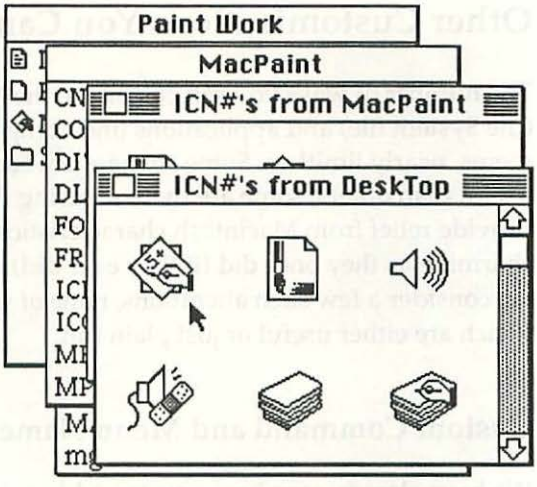
3. Close the Info window. Still in the Desktop file's ICN# window, select the original *MacPaint* application icon again (if need be) and choose **Clear** from the Edit menu to delete it from the Desktop file.

4. Leaving all other windows open for now, return to *MacPaint*'s ICN# window. (Pay close attention now to which window you're in; keep in mind that there is a difference between *MacPaint*'s ICN# window and the Desktop file's ICN# window.) If you have closed *MacPaint*'s ICN# window, reopen it so you can see the new *MacPaint* icon.
   a. Select the new icon (a black box will surround it).
   b. Select **Copy** from the Edit menu (or press Command-C).



Select the new MacPaint *icon and copy it to the clipboard.*

5. Now go back again to the Desktop file's ICN# window; click on it to make it active. Select the Paste command (or press Command-V). The new icon should now appear in the window.



Paste the new MacPaint *icon into the Desktop's ICN# window.*

6. Click on the *MacPaint* application icon you just pasted in to select it, and then select the **Get Info** command.
   a. In the resulting Info window, double-click on the ID number box to select the number that appears there.
   b. Enter the ID number which you so carefully noted in step 2.
   c. Click on the Info window's close box to close it.
7. Close *ResEdit's* windows one by one. When it asks you if you want to save your changes, answer Yes or press Return. Quit *ResEdit*.

When you return to the Desktop, your new *MacPaint* icon will be there to greet you. Be forewarned, however, that custom application icons such as these are finicky beasts. If you attempt to put two different versions of the same program on the same disk (each must have a slightly different name, of course, so one doesn't copy over the other), and one version has the old icon and the other the new one, the new icon will vanish and the old icon will return to its rightful place. If you copy a version of the program with the new icon onto a disk which, unbeknownst to you, still has the old icon in its Desktop file, you will not see the new icon unless you go through one of the Desktop file reconstruction techniques just described.

Other Customizations You Can Do

The number of ways you can customize the Macintosh operating system (the System file) and applications (including the Finder) are, it sometimes seems, nearly limitless. Some changes are easier than others; some are more useful than others; some are more amusing than others; still others may provide relief from Macintosh characteristics which may no longer seem as charming as they once did (if they ever did). In this last section of Chapter 5, we consider a few such alterations, none of which are essential, and all of which are either useful or just plain fun.

Custom Command and Menu Names

We have already seen how you can add or change keyboard shortcuts for menu commands in the Finder (and we will look at this notion again, in reference to applications, in the next chapter). But keyboard shortcuts are just one of the attributes of a menu that you can tinker with. You can, for example, change the name of the menu or the name of the commands in the menu. Or you can put the name of the command as it appears on the menu in boldface or italic.

The technique for customizing menus depends on which menu you want to alter. Thus the sections that follow are organized by specific types of menus. The first section covers the Apple menu—a menu which, of course, appears in the Finder and all your applications as well—and the desk accessories it contains. The second section concerns Font menus, another special kind of menu which (usually) does not respond to standard customization techniques (and which usually does not appear in the Finder). The third section covers all other menus—standard menus which reside in an application, be it the Finder or your graphics program, and which follow the rules menus are supposed to follow.

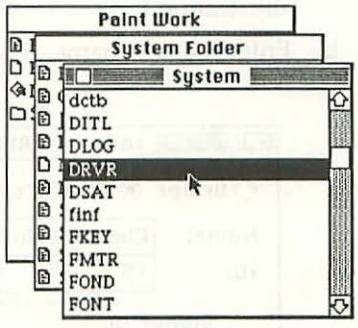
The Apple Menu The Apple menu is a very different beast as compared to most menus. The desk accessory names which are listed in the Apple menu are not the same as command names, and thus they cannot be located or changed using the same methods. Therefore, *MenuMaster* will not work with the Apple menu, and you must turn to *ResEdit* for help. (Unless you have *Font/DA Juggler Plus*. That program's *Font/DA Utility* makes renaming desk accessories a snap.)

Note that this discussion is limited to changing the names of the desk accessories in the Apple menu. Changing other Apple menu items that crop up regularly or irregularly, such as the **About...** command, is something else again. Something for which you will have to turn elsewhere.

In the illustration that follows, you'll see how to change the names of desk accessories in the System file. If you want to work on desk accessories installed in applications or stored in Font/DA Mover files, merely apply the following instructions to what you find within your application or desk accessory file. This is not really all that difficult (or risky, since you will be using a copy, right?); you just need to use your head, since you'll be navigating through relatively uncharted resources.

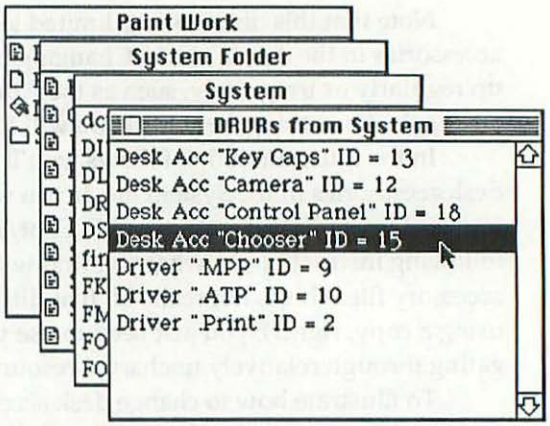
To illustrate how to change desk accessory names with *ResEdit*, let's assume that you want to change Chooser back to its original name, Choose Printer, and you also want to change the name of *Camera* to Screen Shot. The System file we will be working on is on floppy disk for safety reasons. The disk's name is Paint Work, and it is a copy (of course).

1. Open *ResEdit* and then double-click on the System file entry in the Paint Work window.
2. Locate the DRVR resource in the System window and double-click on it to open the "DRVRs from System" window.



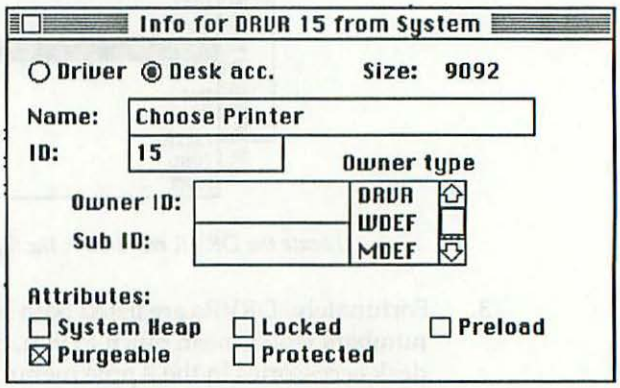
Locate the DRVR resource in the System file window.

3. Fortunately, DRVRs are listed both by name and number. The numbers won't mean much to you, but the names are those of the desk accessories in the Apple menu. To change a name, first select the line (click on it *once*) that contains the name you want to change. For example, click on the line containing the name Chooser.



Select the entry containing the name Chooser.

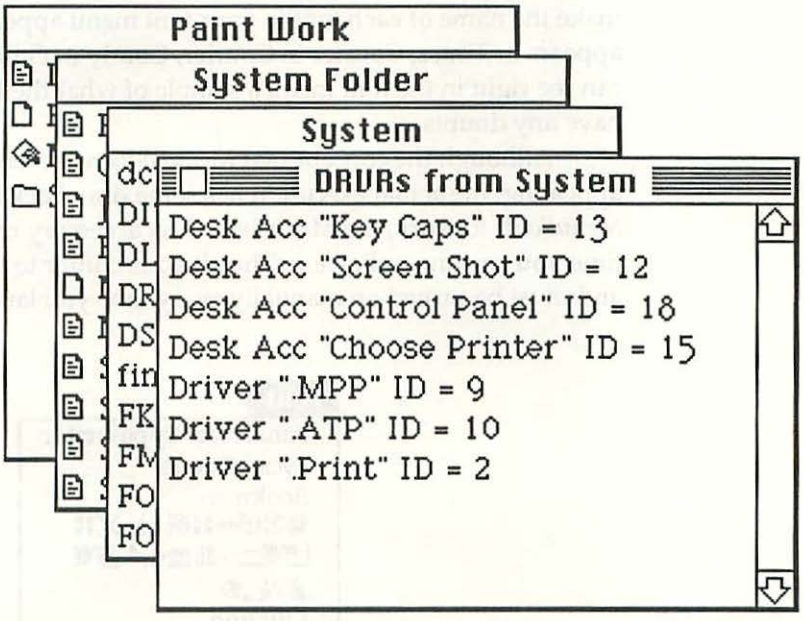
4. Now select the Get Info command from the File menu (or press Command-I).
   a. In the Info window, the name Chooser should already be highlighted. If it is not, double-click on the Name box. (Don't worry, by the way, if various aspects of your Info window, such as Chooser's ID number, don't match those in the illustration.)
   b. Enter the new name `Choose Printer`.



Enter the new name of the desk accessory in the Name text entry box.

c. Close the Info window. As you can see, the new name is now listed in the DRVR window.

5. Now repeat the process to change *Camera*'s name to Screen Shot.



Desk accessory name changes are displayed immediately in the DRVR window.

6. As usual, close windows one by one and save your changes. Restart your Mac with this System file to see the new desk accessory names.

Font Menus In Chapter 4 we discussed how to change font names in order to organize your font menu the way you want it organized. Refer back to the techniques outlined in that discussion if you want to change a font name: to convert Option to Optima, for example. But changing font names is not the only way you can customize your font menus. There is (at least) one other way you can make your font menus more useful and informative, and it requires a desk accessory called *MenuFonts*. What *MenuFonts* does is make the name of each font in your font menu appear in that font. Times appears in Times, Courier in Courier, Goudy in Goudy, and so on. Thus you can see right in the font menu a sample of what the font looks like if you have any doubts.

Although the concept that *MenuFonts* represents is one I find highly appealing, the actual execution has some drawbacks. For one thing, for *MenuFonts* to work, the *MenuFonts* desk accessory must be selected each time you run an application. Thus it adds clutter to the desk accessory menu and must be turned on manually every time you launch a new application.

With MenuFonts, *there is a sample of each font right in the Font menu.*

True, you could create a macro that turns on *MenuFonts* after launching an application. But how much more elegant it would have been had *MenuFonts* been an Init file that took care of everything automatically. (The manufacturer reports that a new version of *MenuFonts*, not yet released when this went to press, is in Init file format.)

Another drawback is that *MenuFonts* works only with standard font menus containing fonts and nothing else (such as font sizes or styles). Thus nonstandard font menus, such as those in *MacDraw* and *Word*, and font lists are out of luck. A final problem is that when *MenuFonts* is in effect, picture fonts produce a line of pictures on the menu. For those who know their picture fonts inside out, this may not be a problem; for those of us who do not, we'll just have to guess which is Cairo and which is Doodads according to the position of the font on the list.

These problems aside, however, if you have trouble remembering just what all those fonts on the menu look like, and you work with programs which have standard font menus, then *MenuFonts* may provide a quick and convenient reminder.

Standard Menus Since the emphasis of this chapter is the Desktop, it is appropriate to consider here how you would change the name and appearance of Desktop (and hence, Finder) commands. The concepts that follow apply to almost any application, however. If you want to change the name of a menu item in your graphics program, for example, or change the style in which it is displayed, the following instructions, in their general form, will work for that situation as well.

As an illustration, assume that you have decided to work on the Finder's View menu. Referring back to your Desktop menu printout, you have decided to change the name of the **by Icon** command to **by Large Icon**, so that it more properly parallels the **by Small Icon** command above it. Further, you want to highlight the new command so that it stands out, since it is responsible for the icon view you use most often.

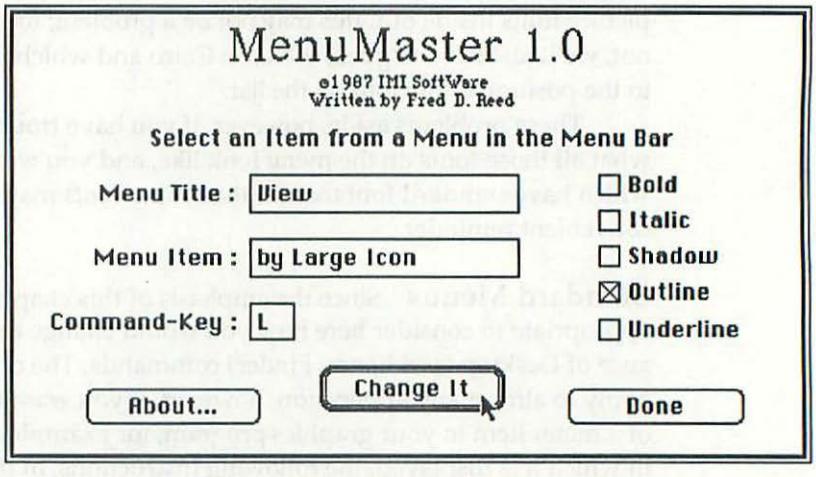
You may go about this task in one of several ways. Consider the following easy way first, using *MenuMaster*.

1. From the Finder, launch *MenuMaster* (precisely how you do so will depend on which version you are using, FKey or desk accessory).
2. With the *MenuMaster* window open, select the **By Icon** command from the View menu. The Menu Title text box should now display "View" and the Menu Item text box should display "by Icon".

3. If you want to change the name of the View menu, you could do so at this point by entering a new name, such as `Window`, in the Menu Title box.

4. To rename the "by Icon" menu item, simply enter the new name: `by Large Icon`.

5. To change the style of the new menu item name, click on the Outline check box to turn it on. (The styles in the list on the right side of the MenuMaster window apply only to menu items, not to menu titles.)



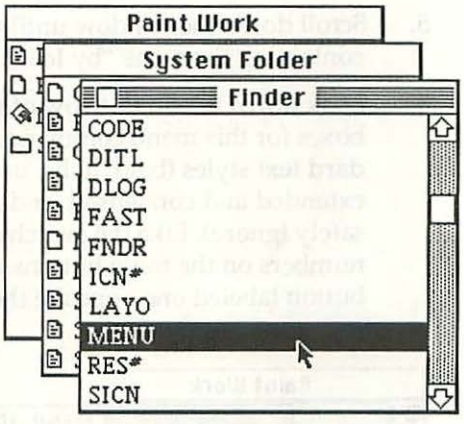
Edit the By Icon command name and click on the Outline option. Then click on the Change It button to save your changes.

6. Click on the Change It button or press Return to save your changes. Click on the Done button or press Command-D to close the *MenuMaster* window.

If you prefer, or if *MenuMaster* runs into trouble for one reason or another, you can use *ResEdit* to change command names. The following steps explain how you would perform the same changes just described using *ResEdit*.

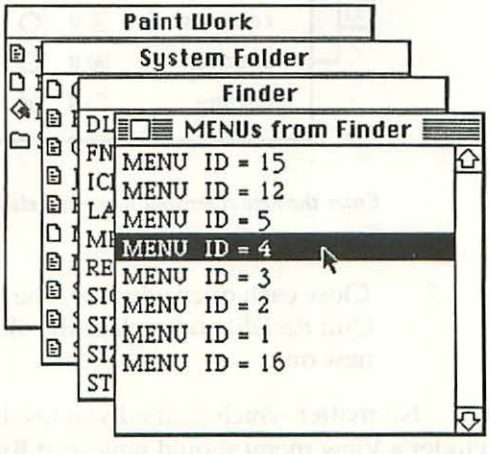
1. Open *ResEdit* and then double-click on the entry corresponding to the Finder you want to change (a copy, of course).

2. Locate the MENU resource in the Finder window and double-click on it to open the "MENUs from Finder" window.



Double-click on the MENU resource in the Finder file window.

3. Recall that the View menu is the fourth menu from the left on the Desktop, so double-click on MENU ID=4 to bring up the proper window. Check the name in the Title box to be sure that you have the View menu.

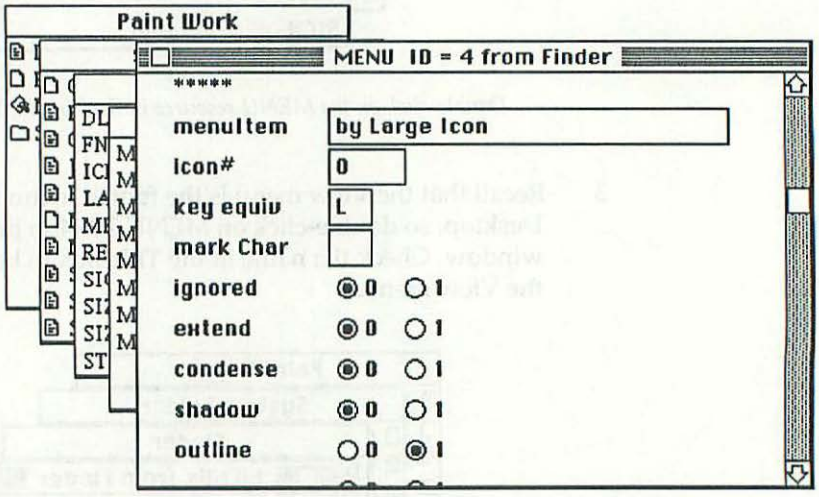


Double-click on the MENU ID=4 resource to open the View menu editing window.

4. At this point, if you wanted to change the name of the menu, you could do that simply by replacing the current name in the Title box.

5. Scroll down the window until you reach the menuItem box containing the name "by Icon". Enter the name `by Large Icon`.

6. Look down the double row of radio buttons beneath the text entry boxes for this menu command. These are labeled with the standard text styles (bold, italic, underline, outline, and shadow), plus extended and condensed, and at the top, ignored (which you can safely ignore). Like the switch on the back of your Mac, the numbers on the radio buttons are 0 for off and 1 for on. So click the button labeled one opposite the word outline.



Enter the new command name and click on the radio button labeled **1**.

7. Close each open window, one by one, and save your changes. Quit *ResEdit* and replace the old version of the Finder with the new one.

No matter which method you used, the **By Icon** command in the Finder's View menu should now read **By Large Icon** and appear in outline style.

The View menu now displays the new command name in its new style.

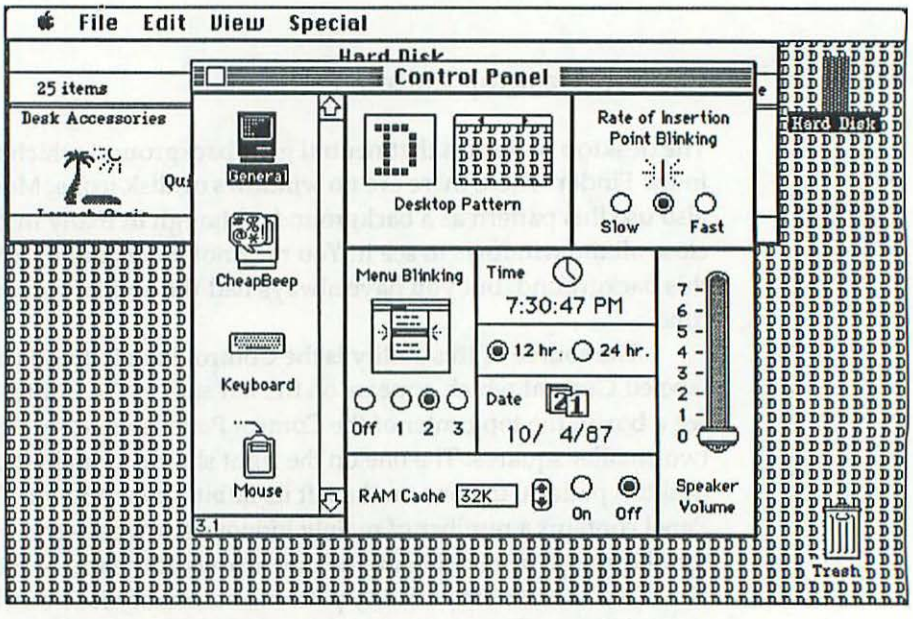
Custom Desktop Patterns

The desktop pattern is that neutral gray background which fills the screen in the Finder where there are no windows or disk icons. Most applications also use this pattern as a background, although in many instances you must close all the windows to see it. You may not have paid much attention to this background, but you have always had the ability to change it at any time.

The source of this ability is the Control Panel. If you click on the icon labeled General which appears on the left side of the Control Panel, you'll see a box in the top center of the Control Panel's window which contains two smaller squares. The one on the right should contain the current desktop pattern, the one on the left its fatbits enlargement. The Control Panel contains a number of mainly hideous stock patterns which you can scroll through by clicking on the tiny arrows at the top of the right-hand box. Or you can start with any pattern and modify it as much as you like, using fatbits techniques, in the enlargement on the left. Once you've selected or created the desired Desktop pattern, click in the center of the right-hand pattern box to record it in the System file.

Frivolous as "artsy" desktop patterns may sound, the fact that desktop patterns are stored in the System file can actually be put to good use. Let's say you have different master System disks specially organized around different sets of downloadable fonts and/or different types of projects. To remind you of which System disk is currently in use, you could customize the Desktop pattern of each System disk so that it is immediately recogniz-

able simply by looking at the pattern. For example, your text font System disk could have a background consisting of the letter T, and your display font System disk could sport a background of Ds. You could create a small "pattern translation guide" to keep by your side for just such purposes, or if the organization of your disks is amenable and your imagination well developed, you could create patterns that speak for themselves. Either way, you need only check the Desktop pattern, either on the Desktop itself or behind the document window of your word processor, for example, to know what your options are.



Custom desktop patterns are created with the Control Panel General configuration window.

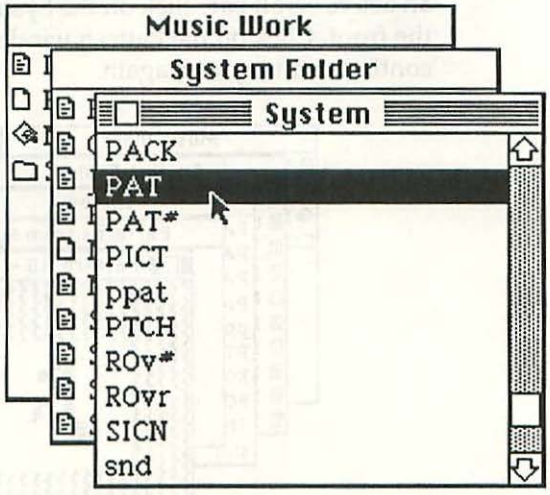
Of course, patterns being what they are (repetitious), your creativity is limited to an eight-by-eight pixel box. If that is too confining for you, then the very last section of this chapter may be what you're looking for. In that section, I describe how a Startup screen may be used as a Desktop pattern, or more precisely, a Desktop picture.

Custom Scroll Bars

Now here's an almost totally frivolous change you can make to the System file: altering the pattern in the scroll bar. It applies to all the scroll bars in operation under the altered System file, both on the Desktop and in applications. Thus, like the Desktop pattern, scroll bar patterns do have at least one possible use: they can be used to indicate which System file is currently in control, as long as your application uses scroll bars. Unlike Desktop patterns, however, the scroll bar pattern must be changed with *ResEdit*.

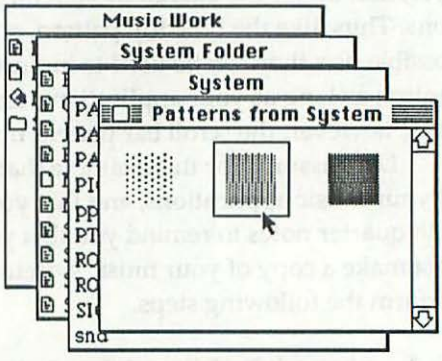
Let's assume for this example that you have a special System disk for all your music applications, and that you want to have scroll bars decorated with quarter notes to remind you that your music System disk is in use. First make a copy of your music system disk and call it Music Work. Then perform the following steps.

1. Launch *ResEdit* and then, in the usual manner, double-click on the System file in the Music Work disk window.
2. Locate the PAT resource in the System file window and double-click on it to open the Patterns from System window.



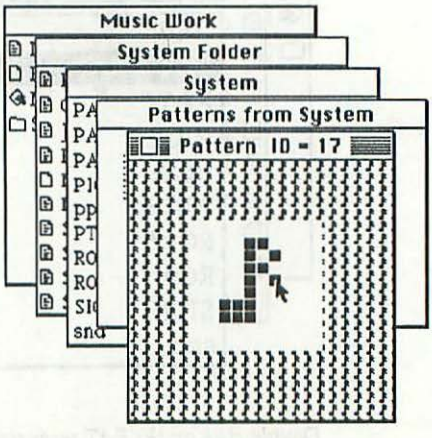
Double-click on the PAT resource in the System file window.

3. In the Patterns from System window, double-click on the medium-gray pattern. To confirm that you have selected the correct pattern, be sure that the window which then opens says "Pattern ID=17" in the title bar.



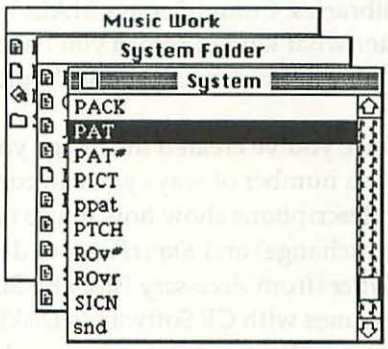
Double-click on the medium-gray pattern to edit the scroll bar pattern.

4. Edit the pattern enlargement in the center of the Pattern ID=17 window in the usual fatbits style. The pattern around the edges of the window will change to reflect your editing and show you how you're doing. If you want to see how the pattern actually looks on an active scroll bar, click on the System file window to bring it to the front. Click on the Pattern window to bring it forward and continue editing once again.



As you edit the enlargement in the center of the window, the surrounding pattern reflects your changes.

5. When you have a pattern that you like, close the open windows one by one until you are asked if you want to save your changes. Click on the Yes button and quit *ResEdit*.



The scroll bars in ResEdit illustrate exactly how your new scroll bar pattern will look.

All that's left now is to restart your Mac using your new System file and admire your handiwork.

Custom Startups

If you've seen it once, you've seen it a thousand times: "Welcome to Macintosh." Oh boy. It's about time for something completely different, don't you think?

It is possible to dip into the System file and edit the welcoming message, but your options are few using this method. An easier route and one that offers unlimited possibilities involves the creation of a startup screen. A startup screen is a standard Paint graphic saved in a special format. If you put this startup screen in your System folder and give it the name `StartupScreen`, your Mac will display this new picture instead of its usual message every time you turn on or restart the computer. In this way, you can start your computer day with a favorite picture, a company logo, seasonal greetings, offensive comments, you name it. If you get tired of your new startup screen, just replace it with another one.

Creating a Startup Screen The most difficult part of making a startup screen is, of course, creating the image. A screen which mimics the standard welcoming message and merely changes the words is not that

difficult. But a sophisticated and elaborate drawing requires considerable artistic skill.

If you have the talent, great. If not, you can borrow images from commercial clip art collections or public-exchange images offered by user group libraries, CompuServe's MAUG, and similar sources. Just remember, no matter what kind of screen you have in mind, the image must initially be in Paint document format, and it should be large enough to fill a Mac Plus screen.

Once you've created the image you want to use for a startup screen, there are a number of ways you can convert it to the proper format. The following descriptions show how to use two different programs, *Screen Maker* (public-exchange) and *SuperPaint*, to do the job, but there are other ways. *Paint Cutter* (from *Accessory Pack I* by Silicon Beach Software) and *Widgets* (which comes with CE Software's *DiskTop*) can also create startup screens, for example. The concept, however, is by and large the same no matter what program you use.

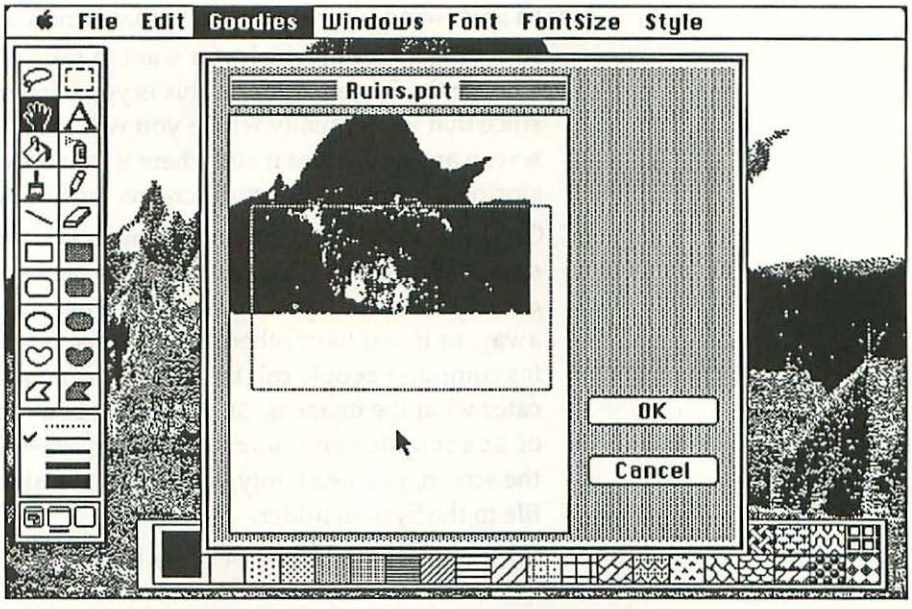
Note that the techniques and programs described in this section create startup screens appropriate for the standard Mac Plus and Mac SE video display. If you have a Mac II or a large screen attached to your Plus or SE, the following instructions may not work for you; or if they do, the resulting Startup image may not fill the entire screen.

Using *Screen Maker* We pick up the discussion at the point where you have just put the finishing touches on the startup image, using any standard Paint program; that is, a program which can save a document in standard Paint or *MacPaint* format:

1. First, before quitting your Paint program, move the image so the upper left corner of the picture is in the upper left corner of the document (since that is where *Screen Maker* expects your image to be). Pushing it into the corner is not difficult; just be careful not to push it so far that you cut off the top or the side.

   What can be difficult is determining whether or not the image is too small (in which case it will not fill the entire screen at startup) or too big (and thus will be clipped):

   a. If your application has a full-screen view, simply place the image in the corner, move the screen view into the corner of the document as well, and turn on the full-screen view to see how your startup screen will look.



To create a startup screen, first put the image in the upper left corner of the Paint document. (Scanned image by Martha Duff.)

b. If your Paint program does not have a full-screen view, then you'll have to guess as to whether or not your image is big enough or too big, and whether it is properly positioned. Some experimentation may be necessary until you get it right. Once you do get it right, however, make note of what you did so you can repeat the process later, without the trial and error.

2. With the image properly positioned in the upper left corner, save the document and quit your Paint program.

3. Now open *Screen Maker*.

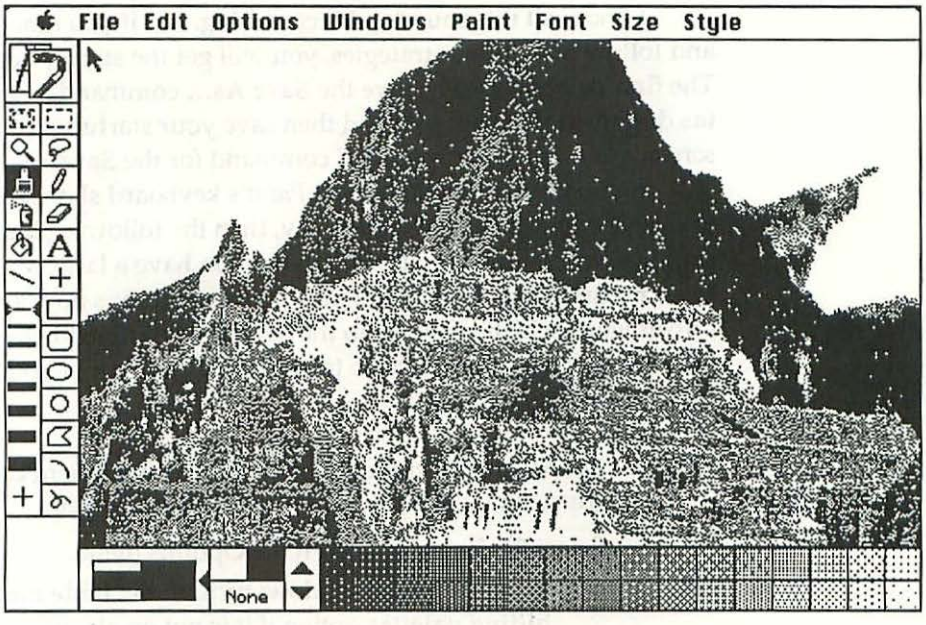
4. Go to *Screen Maker's* File menu and select the **Doc To Screen** command. A file selection box will open; locate and select the Paint document you just completed and saved. Do so using normal file selection techniques and double-click on the document once you find it.

5. Now a Save window will appear. In that window:
   a. Select the folder in which you want to save your screen. Choose the System folder if this is your first startup screen, since that is eventually where you will want to put the startup screen anyway. Place it elsewhere if you are creating and storing a number of startup screens for future use.
   b. Give the new startup screen a name. If you want to use this screen immediately, the correct name, `StartupScreen`, is already filled in. If you don't want to use this screen right away, or if you have other startup screens, add an extension (as computer people call them) to `StartupScreen` that indicates what the image is. `StartupScreen.logo`, for example, or `StartupScreen.warning`. Later, when you want to use the screen, you need only delete the extension and copy the file to the System folder.
   c. Click on the Save button or press Return.

As you can see, choosing a destination folder or disk and giving the document a name is really all there is to making a startup screen with *Screen Maker*. Should you wish to make changes or improvements to one of your startup screens, *Screen Maker* can also convert startup screens back to Paint documents with its **Screen to Doc** command.

Using *SuperPaint* *SuperPaint* is, by itself, able to save files in startup screen format. The only problem is that *SuperPaint*'s manual is not especially clear about that part of the current document is used for the startup image. The one thing that is clear is that unlike *Screen Maker*, *SuperPaint* does not always use the upper left corner of the document for the new image.

The trick is this: *SuperPaint*'s startup image begins at the upper left corner of the image as it is currently displayed and continues as far to the right and down as is required to fill a standard Macintosh Plus screen. As a result, if menu bar and palettes are visible, then the startup screen will extend beyond the right and bottom edges of the window. On the other hand, if you are viewing the image in full-screen view with only the menu bar visible and no scroll bars, the startup image will extend a short way below the bottom edge of the window (but not to the right, since the upper left corner is against the left edge of the screen).



If you were to save your startup screen in SuperPaint *at this point, the upper left corner of the startup image would be the point in the corner of the image directly opposite the pointer. The resulting startup image would extend beyond the right and left borders of the visible portion of the image. (Scanned image by Martha Duff.)*

If you are in true full-screen view—no palettes, no menu bar—then what you see is what you will get for a startup image. The problem is that you usually cannot save a startup screen in true full-screen view, since the **Save as...** command, which is the command you must choose to create a startup screen, has no keyboard shortcut and thus is only available when the menu bar is visible. (You can save an image as a startup screen using the **Save** command, which does have a keyboard shortcut, only if the document has never been saved before.) Since the menu bar must be visible to select the **Save as...** command, the startup image will necessarily extend below the bottom of the window (because, with the menu bar in place, the upper left corner of the image is no longer in the upper left corner of the screen), and what you see will therefore not be what you get for your startup screen.

I know all this sounds a bit confusing. But if you ignore the reasons and follow one of two strategies, you will get the startup screen you want. The first strategy is easy: give the **Save As...** command a keyboard shortcut (as described in Chapter 6) and then save your startup screen in true full-screen view using the keyboard command for the **Save As...** command.

If you have yet to alter *SuperPaint*'s keyboard shortcuts but are eager to install a startup screen right away, then the following instructions should produce the desired result. Note that if you have a large screen there is no **Hide menus when hiding palettes** option, and thus no way to create a full-screen view, which is critical to the technique outlined below. If you cannot hide the menu bar, you'll have to adjust the position of your startup-screen-to-be accordingly.

1. Launch *SuperPaint* and then open the document containing the image you want to use for your startup screen.
2. Select **Preferences...** from the Options menu.
   a. In the resulting window, turn on the **Hide menus when hiding palettes** option if it is not on already.
   b. Click on the OK button to close the Preferences window.
3. Next surround the entire image with the selection rectangle.
4. Select **Hide Palettes** from the Windows menu or press Command-H.
5. Now hold down the space bar to change the pointer into a hand. Continue to hold down the space bar and use the hand to pull the whole document down and to the right as far as it will go. When it will go no farther, stop dragging and release the space bar. The screen view should now be in the upper left corner of the document.
6. Your image should still be surrounded by the selection rectangle.
   a. Using the pointer, drag your image into the correct startup-screen position. Since you are in full-screen view, the image on the screen is precisely what the startup image will look like. There is no need to compensate for cut-off edges: what you see is what you will get. Be careful not to "de-select" the image before you are finished moving it into place; you may chop off a piece of the picture by mistake if you do.

b. When you have positioned the image as you want it, look carefully at the upper left corner of the picture and remember what you see.

7. Turn the palettes and menu bar back on by pressing Command-H.

8. Once again, use the hand by pressing the space bar, and this time push the document down and to the right as far as it will go.
   a. When you are done, stop dragging and release the space bar.
   b. Now check out the upper left corner of the visible window; it should correspond precisely to the upper left corner of the full-screen view (which you carefully committed to memory in step 6, part b). If it does not, you haven't moved the document as far as it will go, or something has gone awry.

   This is the critical part of the exercise; no matter what view you are in (palettes visible, palettes hidden, menu visible, menu hidden, etc.), the upper left corner of the visible image *must* be the same as the upper left corner of the desired startup image. Any parts of the desired image extending beyond the right or bottom edges of the window will be properly saved, and any parts of the desired image extending beyond the left or top edges of the window will not be saved.

9. Now that the upper left corner is in the correct position, all that is left is to save the image. Select **Save as...** from the File menu, click on the StartupScreen radio button and give the file an appropriate name. Finally, click on the Save button.

Make sure that your new document is named `StartupScreen`, place it in the System folder, and it will greet you each time you turn on or restart the computer.

Startup Sounds, Beeps, and Surprises If you thought that simply replacing your startup screen was all you could do to the startup process, think again. You can replace the tone that sounds at startup with a digitized excerpt from David Letterman, Monty Python, Rocky J. Squirrel, or who knows what. You can replace the beep that signifies an error (the replacement occurs during startup; the beep sound itself does not usually occur until later). You can replace the desktop pattern with the startup screen image, and you can arrange it so that all three—the startup screen, startup

sound, and error beep—change every time you turn on your Macintosh. Admittedly, this is not a starchy, white-collar worker sort of thing to do, but it is fun. So if it appeals to you, just follow along.

First, you need to assemble the necessary utilities and files. For utilities, you will need the following:

- *Randomizer* (public-exchange)—Place this Init in your System folder and your startup screen will also be used as the desktop pattern. In addition, if you add the proper folders and files, *Randomizer* will randomly change startup screens, startup sounds, and error beeps each time you turn on or restart your Mac.

- *SoundInit* (public-exchange)—After the standard startup tone, this Init plays a digitized sound which is stored in a separate file.

- *BeepInit* (public-exchange)—This Init turns off the standard error beep and replaces it with a digitized sound which is stored in a separate file.

- *CheapBeep* (public-exchange)—Like *BeepInit*, this utility replaces the standard error beep with a digitized sound file. Unlike *BeepInit*, it does not work with *Randomizer*. Also unlike *BeepInit*, it works from the Control Panel, so you can preview sounds, change their speed, change the sound file used for the beep, and even turn *CheapBeep* off entirely.

For files, you will need:

- Startup screens—If you want to replace the standard welcome message or use a startup screen as the desktop pattern, you will need at least one startup screen in the proper file format. If you want *Randomizer* to change screens, you will, of course, need two or more startup screens. As mentioned earlier, you can create your own startup screens, or select from the many Paint or Startup format documents available from users groups, such as Boston Computer Society, CompuServe's MAUG, and similar sources.

- Digitized sound files—These files are used for startup sounds and error beeps. They can be created with MacNifty's *SoundCap* (a software and hardware combination) or a similar program. If you

don't have *SoundCap*, however, (and not many of us do) you can obtain a wide variety of sound files from the same sources as those listed for startup screens: users groups, CompuServe's MAUG, and the like.

Having assembled all the required software, you are ready to begin. How you use each of the files, and in what numbers and combinations, depends on what you want to do. What follows is a list of the possibilities, and how to put them into effect.

Single new startup screen You already know how to do this. Simply place a startup screen format file in the System folder and rename it `StartupScreen`. Henceforth, the old welcome message will be replaced with your startup image.

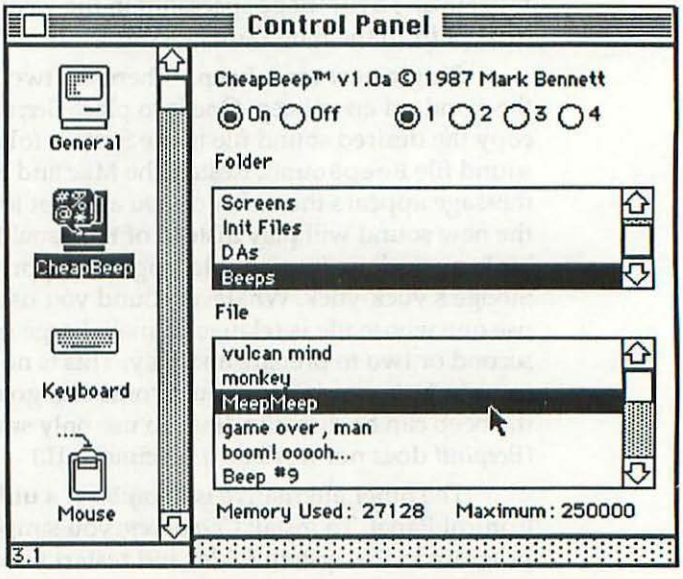
Single new startup sound Place SoundInit in your System folder. Place a sound file in your System folder and name it `StartUpSound`. From this point on, your digitized sound will play on startup shortly after the usual tone. For added effect, you might try matching the startup sound and the startup screen: a picture of a Rocky J. Squirrel and the sound of "And now here's something you'll *really* like!"; a futuristic spaceship in the vastness of space and an excerpt from the Blue Danube Waltz.

Single new error beep There are two alternatives for replacing the standard error beep. One is to place *BeepInit* in your System folder, copy the desired sound file to the System folder, and rename the sound file `BeepSound`. Restart the Mac and every time an error message appears thereafter or you attempt to perform an illegal act, the new sound will play instead of the usual beep. David Letterman's broken window, for example, might be appropriate, or a Three Stooge's yuck-yuck. Whatever sound you use, however, it is best to use one whose file is relatively small. Large sound files can take a second or two to prepare and play. This is no big deal for startup sounds, but error beeps should come and go quickly; the pause before the beep can be disconcerting. So use only small files for beep sounds. (*BeepInit* does not work on a Macintosh II.)

The other alternative is *CheapBeep*, a utility that runs from the Control Panel. To install *CheapBeep*, you simply place a copy of the program in the System folder and restart your Macintosh. Collect the sounds you want to use for beep sounds in a folder and place that in the System folder as well.

To select a beep sound once *CheapBeep* is installed, open the Control Panel and select the *CheapBeep* icon; the *CheapBeep* configuration panel will then appear:

1. Turn *CheapBeep* on by clicking the **On** radio button.
2. From the upper scrolling list, locate and select the folder containing your prospective beep sounds in the usual fashion.
3. When the correct folder is selected, its contents will appear in the lower scrolling list.
   a. Select a sound file in the lower list by clicking on it.
   b. Preview the selected sound by pressing the Enter key.
   c. Modify the speed at which the sound is played by clicking on the buttons numbered **1** through **4**. Button **1** is fastest, button **4** is slowest. Press the Enter key to test the different speeds. Most sounds are meant to be played at either speed **1** or **2**, but **3** and **4** can produce some interesting effects.
   d. Continue experimenting with sounds and speeds until you have a combination that pleases you.



With CheapBeep, *you can test and select beep sounds from the Control Panel.*

4. Once you have the desired sound and the correct speed selected, close the Control Panel by clicking on the close box.

5. To put your selection into effect, you must restart your Macintosh. Whatever beep sound was in use before you configured (or reconfigured), *CheapBeep* will continue to sound until you restart the computer.

Startup screen as desktop pattern Copy *Randomizer* to your System folder. If you don't already have one, place a startup screen format document in the System folder as well, and name it `StartupScreen`. When you restart your Macintosh, the startup screen will be used as the desktop pattern (or desktop picture, to be more precise) in the Finder and in many applications. You will probably find it easier on the eyes if your startup screen is fairly simple in design and neutral in tone. Very busy images, largely black images, or largely white images are all a strain on the eyes and best avoided.

All of these modifications can be mixed or matched in any combination. In concert, however, they can take up quite a bit of disk space, so the group as a whole is not recommended unless you have a hard disk. The additional variations that follow can also be used in any combination and add an element of surprise to your startup screens, sounds, and beeps. But they require even more disk space and hence are also not recommended unless you have a hard disk with plenty of room to spare.

Randomly changing startup screens Place *Randomizer* and a startup screen format file named `StartupScreen` in your System folder. Since this startup screen file is merely a "place holder" (it will be used only during the next startup sequence and never again), it doesn't matter what the image is in the file. Just make sure it is not your only copy. Now create a new folder within your System folder and name it `Screens`. Inside this folder place all the startup screen files you want *Randomizer* to choose from; what their names are is of no consequence. Now, every time you start or restart your Macintosh, *Randomizer* will randomly select one of the screen files in the `Screens` folder for the startup screen. Note that since *Randomizer* also uses the current startup screen as the desktop pattern, your desktop pattern will randomly change as well. There is no way to avoid this.

Randomly changing startup sounds Place *SoundInit*, *Randomizer*, and a startup sound file named `StartUpSound` in your System folder. As before, since this startup sound file is merely a place

holder (it will be used only during the next startup sequence and never again), it doesn't matter what sound is in the file. Just make sure it is not your only copy of a useful sound file. Create a new folder inside your System folder and name it Sounds. Inside this folder place all the sound files you want to be used as startup sounds; their names are, again, of no consequence. From this point on, every time you start or restart your Macintosh, *Randomizer* will randomly select one of the files in the Sounds folder for the startup sound.

Randomly changing error beeps Place *BeepInit* (not *CheapBeep*), *Randomizer*, and a beep sound file named BeepSound in your System folder. Again, since this beep sound file is merely a place holder (it will be used only during the next startup sequence and never again), it doesn't matter which sound is in the file. Just make sure it is not your only copy. Create a new folder inside your System folder and rename it Beeps. Inside this folder place all the sound files you wish to use as beep sounds; as before, they may be named anything. Now, every time you start or restart your Macintosh, *Randomizer* will randomly select one of the files in the Beeps folder for use as an error beep during the current work session. (Since *BeepInit* does not work on the Macintosh II, this procedure for randomly changing error beeps won't work on a Macintosh II either.)

If you've already started using *Randomizer*, you may have noticed that it displays a small window at startup which lists the files it will use the next time you restart the computer. To turn this display off for the current and all future startups, press the Command key after the startup sound has played and hold it down until the Desktop appears. (To turn the display back on, repeat the procedure.)

Summary

- Because the Finder is, for many of us, the application we turn to more often than any other, it makes sense to fine-tune it to our personal working style. You should not feel compelled to do everything suggested in this chapter, however. Whatever you do, be sure to test your particular combination of customizations before trusting it during regular working hours.
- Use *Layout* to change the arrangement of icons and lists in your windows. Use it also to set the font and font size of disk and

document information, to turn on an invisible "snap to" grid to keep your icons neatly ordered, and to set the size and position of windows of newly created folders and newly initialized disks.

- Install a screen saver such as *AutoBlack* or *Pyro!* to prevent your screen from burning in. They will turn the screen off automatically after a set idle period, or you can turn the screen off yourself simply by placing the pointer in the corner of the screen.
- Fine-tune your menu bar with:
  - ◊ *EZ-Menu*, so that your menus drop down automatically when the pointer passes over them.
  - ◊ *JClock*, to add a digital clock to your menu.
- Add keyboard shortcuts to often-used menu commands which lack them, such as **by Icon** and **Empty Trash**.
  - ◊ Create a menu map first, using *Camera* to take screen shots of open menus and to get an overview of all existing commands and shortcuts.
  - ◊ Choose menu shortcuts which are easy to remember and which you aren't likely to confuse with keyboard shortcuts in your applications. Don't assign keyboard shortcuts to commands such as **Shut Down**, which have potentially dangerous consequences. Don't assign the same shortcut to two different commands by mistake.
  - ◊ If you use *MenuMaster*, *MenuKey*, or *ResEdit* to make the shortcut assignments, the new shortcuts will be permanent and will appear up on the menu (in case you need a quick reminder).
  - ◊ To assign keyboard shortcuts to desk accessories and fonts, you'll need to use a macro program such as *QuicKeys* or *Tempo II*.
- You can also use *MenuMaster* or *ResEdit* to change the name of a menu or a menu command, or to change the style in which a command name appears (to italic or outline, for example).
  - ◊ If you want to change desk accessory or font names, you'll need to use *ResEdit*, or *Font/DA Juggler Plus*'s *Font/DA Utility*, since both Apple and Font menus are non-standard menus.

- ◊ *MenuFonts* will make the font names in a Font menu appear in their respective fonts.

- Augment your repertoire of convenient keyboard commands with *QuicKeys* or *Tempo II*. For example, you could create keyboard shortcuts for launching applications and/or specific documents.
- If your dialog boxes lack keyboard shortcuts, add them with *QuicKeys*' **Button** command or with *DialogKeys* (a program which comes in the *QuicKeys* package).
- If the characters that certain keys call up are not to your liking, use a keyboard customization program such as *MacKeymeleon* or *QuicKeys*' **Alias...** command to change them.
  - ◊ For example, you can make Shift-period produce a period instead of a right angle bracket.
  - ◊ To make printer's quotation marks easy to type, assign *QuicKeys*' QuickQuotes and Double QuickQuotes to the quotation mark key.
  - ◊ If you do alter your keyboard, make a keyboard map to keep track of the new key assignments. To do so, use *Camera* or use a special trick with Apple's screen shot FKey to take pictures of the Key Caps window.
- Instead of changing an application's name to indicate a new or customized version, change its icon with *ResEdit*.
- Custom desktop patterns (created with the Control Panel) and custom scroll bar patterns (created with *ResEdit*) can be used to indicate which of several System disks is in control.
- For fun, you can create custom startup screens, add a startup sound, change the error beep, and use the startup screen as the desktop pattern with (among other things) *Screen Maker*, *SoundInit*, *BeepInit*, and *Randomizer*.

6

Creating Your Own Integrated Program

Standardizing Your Applications *315*
Order out of Chaos *316*
Assigning Shortcuts and Creating Macros *324*
Changing the Application Font *334*
Integrating Your Applications *335*
Limited-Memory Integration *336*
Ample-Memory Integration *377*
Summary *397*

In the world of the IBM PC, integrated programs (all-in-one combinations of spreadsheet, word processing, database, and other applications) have a certain cachet. Part of the reason for this is that you can switch quickly and easily from one program to another, completely avoiding ungainly and unfriendly DOS, the PC's pallid alternative to the Finder. Another reason for their appeal is that integrated programs are designed (or are supposed to be designed) so that the same or similar commands are invoked in the same way in every part of the program. These were (and to a great degree, still are) revolutionary ideas to the IBM PC community, ideas whose allure made all-in-one, integrated programs for the PC famous for more than Andy Warhol's proverbial 15 minutes.

But integrated programs have their drawbacks, even on an IBM PC. Often one module, such as the spreadsheet or the database manager, is the dominant one, to which all other modules play second fiddle in terms of quality and features. Another problem is that integrated programs have a hard time being all things to all people. The spreadsheet may meet your requirements, for example, but the word processor may not, leaving you with no choice but to buy a stand-alone word processor to fill in the gaps. Or the integrated program may be lacking an important module, such as communications, again forcing you to go out and buy another, separate program.

If you think about it, though, each one of us already has the building blocks with which to make an integrated program that overcomes these and other problems. For one thing, Macintosh owners already have one advantage: the Finder is far less intimidating than DOS, and thus the desire to avoid it is less compelling (although you can easily avoid it if you want to). We also have MultiFinder, which makes it possible to run a number of different applications at the same time. But perhaps most important of all is the fact that independently produced Macintosh programs are already very similar (due to similar menu structures, similar commands, and so on). Thus, within any collection of Macintosh applications lies the potential for an integrated program whose modules are your favorite applications. You can't buy such a beast, but you can make it yourself from software you already own.

This chapter is about converting your favorite applications into what amounts to a single, smoothly running megaprogram: a megaprogram that is perhaps not as slick and seamless as *Modern Jazz*, but one that may be

molded to your specific needs better than anything created by someone else (*Modern Jazz* included). Need I add that working with a set of building blocks already at your disposal is considerably cheaper than buying a whole new set?

The first half of the chapter covers the modules of your megaprogram—your applications—and the alterations and additions you may need to make them work together as smoothly as possible. The second half of the chapter describes a number of strategies by which you can integrate these individual applications. To help you sort through these strategies, they are divided into two types. Those who have a limited amount of RAM (one megabyte or so) and thus are likely to run into memory limitations should turn to the section on limited-memory integration. Those who have ample RAM (at least 2 or 2.5 megabytes) should turn to the section on ample-memory integration. Within each of these two sections, a variety of integration strategies, which differ more in style than anything else, are illustrated so you can decide which one makes most sense to you and the way you work.

Standardizing Your Applications

Possibly the greatest drawback to the software for IBM PCs and compatibles is that every application goes about its business differently. Learning one program in no way prepares you for operating the next one; keyboard commands are different, screen displays are different, which-part-of-what-you-see-is-what-you-actually-get-on-paper is rarely the same, and on and on. Macintosh programs, on the other hand, have a greater number of similarities than differences. Although some differences are best left alone (unless you are a hot-shot programmer), other differences and conflicts can be resolved with relative ease.

Before we consider the changes and additions you can accomplish on your own, you need to do some important preparation. The most important thing is to decide which applications you want to use for the modules of your megaprogram: Microsoft *Word*, *Excel*, and *File* (that would make for easy integration); *MacWrite*, *FullPaint*, *OverVue*, and *MicroPhone*; or *WordPerfect*, *Omnis 3*, *SuperPaint*, *Illustrator*, *PageMaker*, Hayes *SmartComm*, and *Excel*? Whatever your favorite applications are, choose one of each type.

It doesn't really matter how many there are altogether. For one thing, although you may never use them all in a single work session, you may well use them in a wide variety of subsets which ultimately combine every application with every other application. For another, this chapter illustrates strategies that allow you to integrate a potentially infinite number of programs.

Once your candidates for integration are chosen, you should make a menu map for each application. You'll find these make handy references as you work, of course, but more to the point, you'll need them as you consider existing and yet-to-be assigned keyboard shortcuts. So before proceeding to the next section, flip back to Chapter 5, reread the instructions for making a graphic menu summary, and create menu summaries for each of your megaprogram modules.

Order out of Chaos

How often have you had to *think* before using a keyboard command to cut out a piece of text or copy something to the clipboard? If you're like me, you just press Command-X or Command-C, without a thought concerning which program you are in and what its keyboard shortcuts are. We can do this because Cut, Copy, and Paste are virtually always assigned the same keyboard shortcuts: those which are used in the Finder. The result is that **Cut**, **Copy**, and **Paste** are probably the easiest, most readily used keyboard commands around.

On the other hand, how often have you tried to quit an application by pressing Command-Q, only to find that nothing happens; or to print a file by pressing Command-P, only to discover that you've changed your text to "plain" instead. Frustrating, isn't it? Without keyboard shortcut standards, you have to consciously think about mechanics (which program you are in, what its keyboard shortcuts are, and so on) when faced by such inconsistencies. If you *knew* that Print is always Command-P in all your applications and that Quit is always Command-Q, then the mechanics would slip into the background, and you could concentrate completely on the important aspects of your task. That's what integration is really all about.

Assigning Standard Menu Shortcuts The first step in standardizing your applications is to eliminate all the menu command surprises and,

as much as possible, to make the keyboard shortcuts in all the modules of your megaprogram the same. Command-Q should always invoke the Quit command, for example; Command-P should begin the printing process; and so on.

We'll review the procedures by which you make the actual changes shortly. For the moment, however, I want to consider the menu commands themselves, and how you would go about developing a standard set of menu shortcuts.

Common Commands A number of basic commands are universal (or nearly so) no matter what programs are in your collection. These commands include:

New
Open...
Close
Save
Save As...
Print...
Quit
Undo
Cut
Copy
Paste
Select All

Of course, not every application will use exactly the same names. Your communications program, for example, may have a Print Selection command, rather than a Print command. But most major applications will have, in one form or another, these twelve menu commands. You may find a few others in your application collection—some worth assigning a shortcut to, and some not (such as the Page Setup... command, which most people use only rarely).

The first step in menu shortcut standardization is to comb through the menu summaries for all your megaprogram applications and make a list of the shortcuts used for the common commands. For example, assume that we have decided to use *MacWrite* 4.5, *PageMaker*, *SuperPaint*, *Excel*, and *MicroPhone* as the modules of our megaprogram. After surveying the menus

in our menu summary documents and reviewing our list, we note the following:

- Several commands, including Save As..., do not have keyboard shortcuts in any of the five applications.
- For those commands over which there seems to be general agreement on what shortcut to use, not every program assigns a shortcut. Close, New, and Quit, for example, usually sport the same shortcut when one is assigned, but sometimes they lack a shortcut altogether.
- There are a number of commands over which there is some (or considerable) disagreement over the appropriate shortcut. *PageMaker*, for example, uses Command-W for the Fit in Window command (while other programs use it for the Close command), and *MacWrite* 4.5 uses Command-P for the Plain Text command (and not for Print, as most programs do).

Now comes the hard part: deciding how to standardize keyboard shortcuts for the common commands on your list. The following, for example, are my suggestions for the previous list:

New	(Command-) N
Open...	O
*Close	W
Save	S
*Save As...	A
Print...	P
Quit	Q
Undo	Z
Cut	X
Copy	C
Paste	V
*Select All	E

Of course, many of my choices are no surprise (such as Save, Quit, Undo, Cut, Copy, and Paste). But you may find some of the others of

debatable value. If you use *PageMaker*, for example, you would have to change *PageMaker*'s Fit in Window shortcut in order to assign Command-W to the Close command. Even more problematic, perhaps, is the use of Command-A for **Save As...** and Command-E for **Select All**. Very few programs assign a shortcut to the **Save As...** command, while a fair number use Command-A for **Select All**. But I use **Save As...** a lot, and find Command-A the most obvious shortcut. Assigning Command-A to **Save As...** means that I have to change the **Select All** command shortcut to something else, such as Command-E (for Everything).

The choice of standard keyboard shortcuts is a personal matter, even among the more common commands. It can also turn out to be a complicated matter. Although most of the twelve commands listed have fairly standard shortcuts, attempting to assign or change shortcuts to some commands, especially those marked with an asterisk, can lead to a domino effect of ever-increasing shortcut conflicts. (Changing **Save As...** to Command-A, for example, may mean that **Select All** must be changed to Command-E, which then means that Footnote must be changed to Command-F, and so on, ad infinitum.) Clearly, assigning Command-P to the Print command in *MacWrite* 4.5, and as a result deleting or changing the shortcut for Plain Text, is not such a serious matter. But a similar situation in *Word* can get you tied up in knots, since *Word* uses virtually every letter in the alphabet for its many shortcuts and thus leaves you very little room for maneuvering.

Setting up standards is rarely easy (that's why there are so few of them in the IBM PC world), and you can begin to see just how hard it is. Thus I entreat you to be cautious and careful not to make a bad situation worse. If it looks like making one change is going to necessitate a whole string of changes, then I strongly suggest that you leave well enough alone.

To complete this first step in standardization, make your final decisions about standard shortcuts for common commands. Check it carefully to make sure that you have not assigned the same shortcut to two commands or introduced more confusion than there was before. Then with list in hand, go over the menu summary for each of your programs and write in the new commands as required. If any of the new standard shortcuts conflict with existing shortcuts (such as Plain Text), mark the existing shortcuts that need to be changed or deleted.

Uncommon Commands The other major group of commands that must be considered are those which either are used by only a few applica-

tions or are unique to individual programs. It is important to check keyboard shortcuts for commands which occur in only one program or a few programs, for several reasons. One has already been mentioned: because new shortcut standards sometimes conflict with shortcuts already assigned, and those conflicts must be resolved.

An equally or perhaps more important reason for considering unique commands is the problems that arise when a single keyboard combination is used for a variety of different commands in a variety of different programs. One such problem is the difficulty in remembering what a particular keyboard shortcut does, when it does something different in each of ten programs. Another and potentially more serious problem is the fact that some integration schemes work so smoothly that it is conceivable that you could temporarily forget (or incorrectly identify) which program is currently running. In such circumstances, you could mistakenly invoke a keyboard shortcut which is destructive in function, when you were intending to invoke a different command in a completely different program.

Command-D, for example, is used by many applications for many different purposes: to dial the phone in a telecommunications program; to delete a row or column of cells in a spreadsheet; to duplicate a selection in a graphics program; to bring up the font, size, and style dialog box in a word processor; to move a headline down in an outliner; to repeat a previous action in another graphics program; or to place text or graphics in a page makeup program. The question is, is this use of one keyboard shortcut for so many unrelated commands confusing or, even worse, potentially dangerous?

The answer to both questions is probably yes. Turning to the confusion issue first, there is clearly a limit to the number of keyboard shortcuts employing the Command key and a single, unshifted character. It would be wonderful, of course, if *Word*'s Insert Graphics command, *PageMaker*'s **Place** command (which allows you to import both graphics and text), *More*'s **Receive From** command, and similar commands in other programs all had the same keyboard shortcut; or if the shortcuts for the **Rotate** and **Flip** commands in your paint, draw, and PostScript graphics programs all matched.

But the fact is, with a limited number of shortcuts to draw from, program designers have had to make compromises, and these compromises often mean that similar commands in different programs do not have the same shortcut. They also mean that, unless you want to jump right in and

redesign the menu shortcut part of the program yourself, it is probably best just to live within this imperfect world as best you can—with one exception.

That exception revolves around the potential for accidental data loss. Take Command-D, for example. Most of the previously mentioned commands use Command-D as a shortcut for a "positive" action. The spreadsheet, however, uses it in a "negative" way, to delete a row or column of cells. If you are not careful, you could use Command-D absent-mindedly and lose some valuable data in the process. To avoid such unpleasantness, a destructive command in one application should not be assigned a shortcut which is used in other applications for benign commands. In other words, considering all your applications as a group, there should be no shortcuts associated with both constructive and destructive commands.

For example, virtually all programs have a Cut command, and virtually all of them use Command-X to call it up (and if they don't, you'll soon be changing it so that they do). Command-X is thus *always* a shortcut to be used with care, and you no doubt know this well enough already. Other destructive commands are actually relatively rare, but when they do occur, they should be assigned shortcuts that *always* and *only* perform destructive functions. Or they should have no shortcut at all.

Thus, although Command-D appears to make sense as a shortcut for deleting a spreadsheet column, there are too many commands in other programs that use Command-D for more commonplace and less harmful actions. To prevent inadvertent deletions in your spreadsheet program, I would suggest that you change the shortcut for the **Delete Column** command to one of the few you have reserved for destructive commands. Or, if that doesn't work, remove the shortcut and make the command available only from the menu.

Finally, I should repeat that as desirable as widespread keyboard shortcut changes may sometimes seem, large-scale changes can in fact create more confusion than they are supposed to eliminate. Don't overdo it. If you decide to implement a grand scheme, try doing it in stages. In the end, of course, how you standardize your keyboard shortcuts is up to you, and the only "right" way is the one that works for you. I encourage you to be conservative, at least at first, in the changes you make. But if a more liberal attitude and a greater number of shortcut reassignments suits you better, then by all means go ahead. There's always a backup (or there should be backup) to rescue you if you make a mess of it.

A General Macro and Menu Shortcut Strategy With *ResEdit*, *MenuMaster*, or *MenuKey*, you can assign keyboard shortcuts to almost any standard menu command. With a macro processor such as *Tempo II* or *QuicKeys*, and the extra function keys on the extended keyboard, you can make any program overflow with all sorts of keyboard shortcuts. What with the hundreds of keyboard shortcuts that are possible, it is clear that a little thought and planning is in order if you want to avoid total chaos. Thus before we turn to the actual methods for assigning shortcuts and creating macros, we need to pause a moment to consider how they should be organized.

There are, of course, many different strategies you could devise to impose some order on your combined keyboard shortcuts (both menu shortcuts and macros). One such strategy is suggested in the following paragraphs. If you like it, use it. If you don't like it, alter it or create your own. But whatever you do, don't start adding shortcuts higgledy-piggledy, before you have some organizational principles in mind. If you do, you will soon find yourself buried in a monstrous tangle of long-forgotten keyboard commands.

The essence of the following strategy is to restrict certain kinds of commands or functions to certain keyboard combinations. The primary purpose of these restrictions is to help you remember what's what, and where you'll find it. The secondary purpose is, simply, to help keep your menu shortcuts and macros in order, and to prevent you from losing track of things. In a nutshell, the overall strategy looks like this:

- Command-key and Command-Shift-key combination Macros with these keyboard assignments are restricted to unadulterated menu commands and modest elaborations on menu commands. Most (but not all, as you will see) of the basic, two-key Command-key shortcuts can be assigned with *ResEdit*, *MenuMaster*, or *MenuKey*. You can considerably expand the number and variety of key combinations, with a program such as *QuicKeys* or *Tempo II*, to include all Command-Shift-key combinations; Command-key pad key shortcuts (keys on the key pad are interpreted as being distinct from the keys on the main section of the keyboard); Command-function key shortcuts on the extended keyboard for the Mac SE and Mac II; and the few standard Command-key

shortcuts that *ResEdit* and the others cannot get to function properly. (More on this shortly).

Command-key combinations should be used for major, frequently used menu commands, and Command-Shift-key combinations should be reserved for lesser menu commands and/or modest elaborations (such as Command-Shift-P for a macro that both selects the Print command and clicks on the OK button in the Print dialog box). Beyond this major and rather self-evident guideline, you are the judge of what makes sense and what doesn't.

A look at how a program such as *Word* (or *More*) uses these two key combinations may help you decide how to organize your own shortcuts. If you intend to use *Word* (or *More*) in your integrated megaprogram, then following its lead makes even more sense.

- Command-Option-key combinations These keyboard shortcuts should be restricted to macros which perform the same function in every one of your applications: such things as calling up desk accessories, FKeys, or fonts from the keyboard and *QuicKeys* commands which transfer to other applications.

  Also in this category fit more complex sequences which, though different perhaps in their details, perform similar functions in several different programs. For example, assume you have a macro (Command-Option-T) in your word processor which selects the entire document currently on the screen, formats it in 12-point Times, and prints it. You also have other, similar macros which print your document in 12-point Palatino (Command-Option-P), 12-point New Baskerville (Command-Option-Shift-B), and so on. These macros use Command-Option-key combinations because in your outliner and your spreadsheet as well, you have macros assigned to the same key combinations which perform the same tasks.

- Command-Option-Shift-key combinations Macros with these knuckle-breaking keyboard assignments are restricted to macros which perform the unique functions in each application. In your spreadsheet, for example, you may have a macro that swaps two columns of data (Command-Option-Shift-S). In your word processor, you may have a macro that assigns both bold and italic styles

to the selected text (Command-Option-Shift-B). In your draw program, you may have a macro for rotating the selected object five degrees or seven degrees or some such thing (Command-Option-Shift 5 or 7), and so on.

Of course, for this strategy to work, you must do more than just put it to use in a few frequently used applications. You must follow this strategy in *every* module of your integrated megaprogram. Only if your menu shortcuts and macros are well-organized both within each application and across all applications, only then will you begin to get the full benefit of homemade integration.

Assigning Shortcuts and Creating Macros

Only when you have a finished master plan of keyboard shortcuts and macros for all your megaprogram modules should you make changes to each of the applications involved. If you are at all disorganized about this, you'll wind up assigning and reassigning shortcuts without end. So before proceeding any further, make sure you have done your homework.

Menu Shortcuts: a Reprise To assign new menu shortcuts and reassign old ones, you can use *ResEdit*, *MenuKey*, or *MenuMaster*. In Chapter 5, we saw how to use these three utilities to customize the Finder's keyboard shortcuts. The same techniques and principles apply when working on the keyboard shortcuts of any other application. This is especially true with *MenuKey* and *MenuMaster*, which are straightforward, forgiving, and uncomplicated programs. Thus rather than repeat myself once too often, I refer you to Chapter 5 for illustrations of how to change or create keyboard shortcuts with *MenuKey* and *MenuMaster*.

Because *ResEdit* is neither particularly forgiving nor especially uncomplicated, I give an additional *ResEdit* example in this section illustrating how to add or change menu shortcuts in an application. In the example, the keyboard shortcuts are added to *MacWrite* 4.5, but the techniques apply to most any application. Simply substitute the application you want to work on for *MacWrite* 4.5, and the shortcuts you want to add or alter for the ones described here.

The following is how you would go about adding six new keyboard shortcuts to the File menu in *MacWrite* 4.5 using *ResEdit*. (The newest version of *MacWrite* from Claris Corporation already incorporates most of these additions.)

1. Launch *ResEdit* and then locate the copy of *MacWrite* 4.5 that you want to work on. Open the *MacWrite* window by double-clicking on the program name in the disk window.

2. In the *MacWrite* window locate the **MENU** resource and double-click on it.

3. Refer to your *MacWrite* menu summary to determine the ID numbers of the menus you want to work on. The Apple menu is always assigned ID=1, the File menu is ID=2, and so on down the line.

4. Since the File menu is the menu we want to work on, locate the menu ID=2. Double-click on it to open the MENU ID=2 window. If necessary, move the MENU ID=2 window up so you can see the entire window. Confirm that you have the correct menu by checking the name in the Title entry box (the seventh item from the top); it should be "File."

5. Scroll down through the window until you find the first menuItem entry box, the one containing the name of the **New** command.
   a. Click inside the **New** command's key equiv box (the second box below menuItem).
   b. Enter the new shortcut—N. (Type an uppercase N for appearance's sake only.)

6. Scroll down to the next menuItem entry box, the one for the **Open...** command. Click inside the key equiv box beneath it and enter the new shortcut—O.

7. Scroll down to the menuItem entry box for the **Save** command. Click inside the key equiv box beneath it and enter the new shortcut—S.

8. Scroll down to the next menuItem entry box, the one for the **Save As...** command. Click inside the key equiv box beneath it and enter the new shortcut—A.

9. Scroll down to the menuItem entry box for the **Print...** command. Click inside the key equiv box beneath it and enter the new shortcut—P .
10. Scroll down to the next menuItem entry box, the one for the **Quit** command. Click inside the key equiv box beneath it and enter the new shortcut—Q.
11. Now we have to delete a few shortcuts that conflict with the ones we just assigned. Close the MENU ID=2 window and open the MENU ID=5 window. Confirm that you have the correct menu by checking the name in the Title entry box; it should be "Format." If necessary, move the MENU ID=5 window up so you can see the entire window.
12. Scroll way down to the menuItem entry box for the **Align Left** command. Double-click inside the key equiv box beneath it to select the current shortcut (N) and press the Backspace key to delete it.
13. Close the MENU ID=5 window and open the MENU ID=7 menu window. Confirm that you have the correct menu by checking the name in the Title entry box; it should be "Style." If necessary, move the MENU ID=7 window up so you can see the entire window.
14. The first menuItem entry box should contain the **Plain Text** command. Double-click inside the key equiv box beneath it to select the current shortcut (P) and press the Backspace key to delete it.
15. Scroll down to the menuItem entry box for the **Outline** command. Double-click inside the key equiv box beneath it to select the current shortcut (O) and press the Backspace key to delete it.
16. Scroll to the next menuItem entry box, the one for the **Shadow** command. Double-click inside the key equiv box beneath it to select the current shortcut (S) and press the Backspace key to delete it.
17. Close the MENU ID=7 window and any other menu windows if any remain open. Now close the MENU window by clicking on its close box. Finally, close the *MacWrite* window. When you are asked if you want to save your changes, click on Yes or press Return. Quit *ResEdit*.

The final step, of course, is to launch your revised version of *MacWrite* 4.5 and test drive your new shortcuts. Sad to say, your test run will reveal that not everything worked as planned. Command-N and Command-O do not call up the New and Open commands—they don't do anything anymore. And Command-Q only works when a window is open. Well, such are the occasional problems we tinkerers face when trying to "fix" someone else's program. Other programs, such as *MacPaint*, seem to tolerate our tinkerings without complaint. Some, such as *MacWrite* 4.5, do not.

This is a problem you will run into occasionally using *ResEdit*, *MenuMaster*, or *MenuKey*. The difficulty (at least in this case) arises because some programs do not "look for" menu commands or do so only at certain times. Thus, valid shortcuts for valid menu commands may not function at all, or only under certain circumstances, if the application is not programmed to look out for them. *MacWrite* is just such a program, and that is why only some of the shortcuts we created in the above illustration worked as expected, and others worked only some of the time. As we will see shortly, however, problems such as these can be easily rectified with a macro processor such as *Tempo II* or *QuicKeys*.

Macros: Some Examples Some people are reluctant to use macros and macro programs. Part of this reticence is due to simple lack of familiarity with such things. This lack of familiarity is something that I hope the illustrations in this section will cure.

But before we turn to some examples, I should point out that if you plan to use macros and a macro program as part of your regular routine, it makes sense to have the macro program up and running as soon as you start your Macintosh. With *QuicKeys*, that's easy. Just place the *QuicKeys* file in the System folder and restart your Macintosh. From that point on, *QuicKeys* will always be awaiting your commands.

As for *Tempo II*, however, simply putting the *Tempo II* file in your System folder will not turn it on at startup. If you want to have a number of things, including launching *Tempo II*, done for you at startup, then refer to Chapter 7. But if the only thing you want accomplished at startup is launching *Tempo II*, then you can use *Tempo II*'s special Set Startup file to do the job. The method for doing so is a little tortuous (since Set Startup is

really meant to install *Tempo II* and also to run a specially designated macro), but it works. Here's how to do it:

1. In the Finder, record a macro that doesn't do anything.
   a. In the *Tempo II* menu, first select **Start Recording.**
   b. Then immediately select **Stop Recording.**
   c. In the Save Macro dialog box, give the macro a name such as `Startup Dummy`, leave the keyboard shortcut blank, and click on the Universal Macro check box.
2. Put a copy of Set Startup in the System folder.
   a. Double-click on the Set Startup icon while pressing the Command key.
   b. In the resulting file selection box, select the Startup Dummy macro and click on the Select button.
3. If you have not done so already, set the default macro file with the **Configure...** command in the *Tempo II* menu. (For simplicity's sake, keep the default macro file in the System folder.)
4. Select *Tempo II*'s Set Startup file and with the Finder's **Set Startup** command, make it the startup application.

From this point on, every time you start or restart your Macintosh, *Tempo II* will be launched automatically. Once launched, *Tempo II*'s menu, which is normally labeled with a Command-key symbol, will follow you around as you hop from application to Finder to application (if MultiFinder is off) or application to application (if MultiFinder is on). You can record macros which are specific to the application you are currently in, or ones which are universal and work in any application. If you are using MultiFinder, *Tempo II* recognizes which application is the current one and uses the corresponding application-specific macros.

We've already mentioned a number of ways in which *Tempo II* and *QuicKeys* macros can be put to good use and described how to create a few of them. Now let's look at some others. The first three examples below refer to the previous section's failed attempts to modify several of *MacWrite*'s menu shortcuts. The fourth illustration is based on a suggestion of a few pages back. The last two examples show how you might use macros in other simple but useful ways.

Example One For this macro, we use *Tempo II* to make a keyboard shortcut that selects *MacWrite*'s 4.5 **New** command. Before beginning, therefore, be sure to launch *MacWrite* 4.5.

1. If *Tempo II* has not already been launched, select it from the Apple menu. This will place the *Tempo II* menu on the menu bar. You should now be in *MacWrite*. If a window is currently open, close it.
2. Select **Start Recording** from the *Tempo II* menu (or press Command-accent grave). The *Tempo II* menu symbol will now change to a flashing R.
3. Using the mouse, open the File menu and select the **New** Command. A new, untitled window will open.
4. Select **Stop Recording** from the *Tempo II* menu (or press Command-accent grave again).
5. After the Save Macro window opens, press Command-N (*not* Command-Shift-N) to set the macro's keyboard combination. Then type a descriptive name for it, such as `New Window`. Make sure the Universal Macro check box is turned off.
6. Click on the Save button, now your New Window macro, which functions identically to most other program's Command-N menu shortcut, is ready for use.

We could have used *QuicKeys* to create this macro as well, using that program's **Menu/DA...** command.

Example Two Now let's do the same thing for *MacWrite*'s 4.5 **Open** command. In this illustration, you'll see how you can end a macro at a dialog box, at which point you then make a choice, depending on your needs at the moment.

1. If *Tempo II* has not already been launched, select it from the Apple menu. This will place the *Tempo II* menu on the menu bar. You should be in *MacWrite*. If a window is currently open, close it.
2. Select **Start Recording** from the *Tempo II* menu (or press Command-accent grave).
3. Using the mouse, open the File menu and select the **Open** command. The standard file selection box will open.

4. Select **Stop Recording** from the *Tempo II* menu (or press Command-accent grave again). Note that even though a file selection box is open, the *Tempo II* menu still works (and it is the only menu that works).

5. When the Save Macro window opens, press Command-O to set the macro's keyboard combination. Then type in a name like `Open Window`. Make sure the Universal Macro check box is turned off.

6. Click on the Save button and your new macro is complete and ready to run.

As before, we could have used *QuicKeys* to create this macro as well, using its **Menu/DA...** command. Because an application's menus are not functional after selecting the **Menu/DA...** command, you don't have to deal with the open file selection box; otherwise, the process is much the same.

Example Three Now let's use *Tempo II* to make a **Quit** keyboard shortcut for *MacWrite* 4.5 which works whether a window is open or not.

1. If *Tempo II* has not already been launched, select it from the Apple menu. You should be in *MacWrite*. If any windows are open or documents need saving, perform all your housekeeping now. There should be no windows open when you start recording the macro (since you don't want to record responses to dialog boxes).

2. Select **Start Recording** from the *Tempo II* menu (or press Command-accent grave).

3. Using the mouse, open the File menu and select the **Quit** command.

4. Wait for *MacWrite* to quit and for the Desktop to appear. When everything has settled down, select **Stop Recording** from the *Tempo II* menu (or press Command-accent grave again).

5. When the Save Macro window appears, press Command-Q to set the keyboard combination. Then enter the macro name: `Quit MacWrite`. Make sure the Universal Macro check box is turned off.

6. Click on the Save button and your new **Quit** command is fully operational.

As before, we could have used *QuicKeys* to create this macro as well, using its **Menu/DA...** command. Again as before, because an application's menus are not functional after selecting the **Menu/DA...** command, you won't actually quit *MacWrite*; otherwise, the process is much the same.

Example Four In this example, we create a *Tempo II* macro in *Word* that formats a document in 9-point Palatino and sends it to the LaserWriter. The reason we've switched from *MacWrite* to *Word* is that *Word* contains a great variety of built-in keyboard shortcuts with which one can scroll a document, move the cursor, select text, select a font, and so on. These shortcuts make the construction of complex macros quite a bit easier. Indeed, creating a macro such as the following one, which works under all conceivable circumstances, may be impossible in *MacWrite*.

1. If *Tempo II* has not already been launched, select it from the Apple menu. Of course, *Word* must also be launched and running before you begin recording. So you can see what you are doing, open any document containing a page or more of text.
2. Select **Start Recording** from the *Tempo II* menu (or press Command-accent grave).
3. Press Command-Option-M to select the entire document. (Alternatively, you could press Command-key pad-3 to move the text insertion point to the bottom of the document, then press Command-Shift-key pad-9 to move the insertion point to the top of the document and select all the text in between.)
4. Press Command-Shift-E to select the **Font** command. Type `pala` and press Return to select Palatino (these four letters are probably more than enough to specify Palatino; depending on the fonts in your System file, you may want to type more or fewer letters).
5. Press Command-D to bring up the Character Format dialog box. Type `9` and press Return to select 9-point type.
6. Press Command-P to bring up the Print dialog box. Press Return.
7. If you do not want to print your current document at this time, select **Suspend Recording** from the *Tempo II* menu (or press Command-comma), and then press Command-period to cancel printing.

8. Select **Stop Recording** from the *Tempo II* menu (or press Command-accent grave again).
9. When the Save Macro window appears, press Command-Option-P as the keyboard combination for this macro. Then enter a name like `Print Pala-9`. Make sure the Universal Macro check box is turned off.
10. Click on the Save button and your macro is ready for use.

Although it may be possible to create this macro with *QuicKeys*, the process would be laborious, since you'd have to create separate macros for each step, then link them together using the **Sequences...** command. Indeed, any time a macro gets at all complex, *Tempo II* is clearly a better choice. But, as I've noted elsewhere, *QuicKeys* has its own special charms as well.

Demo

Example Five Keyboard shortcuts for zooming windows in and out and selecting windows buried in a stack are standard issue in *Modern Jazz;* with *QuicKeys* and *Tempo II*, we can make our own window-management shortcuts. For example, the following is how you create a shortcut to zoom a window in or out with *QuicKeys*:

1. Open *QuicKeys* via the Control Panel, as we have done in earlier examples.
2. Make sure the Universal radio button is turned on.
3. Open the Define menu and drag down to the Mousies command. Pause to open the Mousies submenu and drag down that menu to the **Zoom Windows** command. Release the mouse button while **Zoom Windows** is highlighted.
4. Back at the *QuicKeys* configuration panel, select the Zoom Window macro you just created and press the key combination you want to assign it.
5. Close the Control Panel Window.

Oddly enough, although *QuicKeys* (version 1.0) and *AutoBlack* are generally compatible with each other, this particular macro does not work if you have *AutoBlack* installed as well as *QuicKeys*. If you are using *AutoBlack*, you can get around this problem by creating the macro in *Tempo II* instead. Simply record a universal macro consisting of a click on a zoom box on any window. *Tempo II* is smart enough to know that when it plays the macro back, it should click on the zoom box of the active window, no matter where the window is on the screen or what shape it is.

Demo

Example Six In MultiFinder, no matter which of the three switching methods you use (click on the menu bar, click on the application icon in the Finder, or select the application from the Apple menu), you have to use the mouse to get from one program to another. In this final example, we use *QuicKeys* to record a click over the small icon in the menu bar, thus creating a macro which allows you to rotate through your open applications with the press of a key or two.

1. Make sure that MultiFinder is turned on. You can be in the Finder, or in an application; it doesn't matter. Open *QuicKeys* via the Control Panel, as we have done in earlier examples.
2. Make sure the Universal radio button is turned on.
3. Open the Define menu and select the **Clicks...** command.
4. The menu bar should now state in highlighted text, "Click on location...."
5. Click once over the small application icon on the right end of the menu bar. Try to click dead center; the icon tends to move a little from one application to the next, so a click on the edge of one icon may be over the edge and thus have no effect on another.
6. In the resulting dialog box, leave all settings as they are except for the text entry box at the top, the one labeled Description. In that box, enter the name `MF Click`. Click on the OK button.
4. Back at the *QuicKeys* configuration panel, select the macro you just created and press the key combination you want to use to call it up.
5. Close the Control Panel Window.

As with the previous macro, this one may not work if you are running *QuicKeys* (version 1.0) and *AutoBlack* together. If you are experiencing problems, the solutions are the same as before: remove *AutoBlack* or use *Tempo II* to create the macro instead.

Perhaps the most important concept behind all these examples is that a macro processor can be used to automate almost anything that you do more than occasionally. Whether you perform the task once a day, every day, or once a week, every week, or intend to perform it many times today and never again—if you do it often, a macro processor can save you a lot of

time and effort. It's easy to forget how much a program such as *Tempo II* or *QuicKeys* can help. Thus it pays to stop and think about your work habits from time to time, and to ask yourself if there is anything you do regularly that *Tempo II* or *QuicKeys* couldn't do just as easily. You may be surprised at the answer.

As you can see, macros are not all that difficult. Nor are they much different from the menu command shortcuts programmers insert in their programs, except that you're in control over what they are and what they do. If you are judicious in your choice of macros and generous in their use, you can transform needless rote into light work indeed.

Changing the Application Font

Finally, there is one other thing you can do to standardize your applications. This is to set (or reset) what is called the default application font. The default application font is the font many applications select when you first open a new document. *MacWrite*, for example, opens a new document with the default application font already selected. If you immediately start typing, your words will appear in that font.

The default application font is normally Geneva. With the Control Panel file *ApFont* (public-exchange), you can change the default application font to any font in your master System file, and to any standard point size. Since I like to do my initial work in New York, I have set mine to New York-12. You may prefer to set it to the LaserWriter font you normally use, for example, or to Boston font if you use an ImageWriter. In any case, once set to the new font, all applications that use the default application font information will open all new documents with that font automatically selected.

Like many of the settings in the Control Panel, the default application font is stored in battery-backed ROM and thus is independent of the System file you are using. If you have several master System files, each with a different set of fonts, be sure that the font you select for the default application font is installed in all of your master System files; otherwise, you may end up with Geneva anyway. Unless you want to work with a resized font, also be sure that the size you have selected is installed as well.

Finally, you should know that some applications do not use the default application font. Some use a particular font (often Geneva or Monaco or a similar lackluster font) no matter which font you prefer. Others

allow you to set the default font for that program. *Word*, for example, always opens a new document with New York selected, but you can change to any other font simply by redefining the Normal style with the **Define Styles...** command. (To do so, select **Define Styles...**, select Normal in the style selection box, select the font of your choice from the Font menu or with the **Character** command from the Format menu, then click on the Define button.) In some other programs which ignore the default application font, you can set the default font by selecting the desired font when no documents are open.

Integrating Your Applications

Now that we've done as much as we can to standardize the modules of our megaprogram, it's time to put the modules together and go to work. The many ways of accomplishing this can be placed in one of two basic subdivisions. In one subgroup, applications are run one at a time and a special utility program, simpler and faster than the Finder, facilitates the switch from one application to the next. In the other subgroup, the Macintosh's RAM is subdivided and several applications are kept in memory at the same time, one per memory subdivision; switching between them is as simple as clicking the mouse and just about as fast.

The first integration subgroup requires no more memory than that needed by a single application. It is thus most suitable for those whose computers have a limited amount of RAM. The second subgroup usually requires quite a bit more RAM, since a number of programs are loaded into memory at the same time. The first method is described in the section that immediately follows. The second method is discussed later in this chapter, under "Ample-memory Integration."

The question is, of course, how much memory is "ample?" Which integration method should you use if you have a certain amount of RAM? The answer, as usual, is that it depends. It depends on how many programs you want to switch between in a single work session; it depends on the size of your chosen applications and how much memory they require; it depends on how much an application's performance deteriorates as available memory approaches the program's minimum requirements; it depends on how large your documents are, and how well your applications handle

large documents in a limited amount of memory. It depends on whether or not you have installed a RAM disk or a RAM cache (see Chapter 7 for more on these two items); and on and on.

Although generalizations are difficult, the following are offered as suggestions only, not as hard and fast rules:

- If your Mac has only one megabyte of RAM, memory constraints are a significant issue. Most likely, you will find it difficult to load more than one good-sized program using MultiFinder, or two good-sized programs using *Switcher*. Thus, with a single megabyte of RAM, you will most likely want to use one of the schemes outlined in the following section.
- If you have two or more megabytes of RAM, which integration method you should use depends on your overall RAM use strategy. If you use a large RAM disk or RAM cache (again, see Chapter 7 for more on these two topics), you may be able to load only two full-sized applications into the remaining amount of memory. If you want a megaprogram with four or more modules and not two, then you should use the integration methods described in the limited-memory section. If, on the other hand, you do not use either a RAM disk or a RAM cache (or perhaps only a small RAM cache), then you probably can load four or five applications into memory, and the methods described in the "Ample-Memory" section would then make more sense.
- If you have four or more megabytes of RAM, you can safely skip the next section and proceed directly to the "Ample-Memory" discussion. For even if you have a large RAM disk or cache, you should have enough RAM left over to keep quite a few programs in memory simultaneously. The combination of a RAM disk or a RAM cache and one of the schemes detailed in the "Ample-Memory" section should provide you with a super-fast and highly versatile megaprogram.

Limited-Memory Integration

Under limited-memory constraints it is often difficult to run more than one good-sized application at a time. (MultiFinder, for example, really needs at least two megabytes of RAM in order to function the way it was meant to.)

Thus the basic idea behind limited-memory integration is to run programs much as you have in the past (sequentially, one after the other) but to avoid returning to the Finder when you switch from one application to another. The reason for Finder-avoidance is, basically, speed. The Finder is not known for the quickness with which it opens and closes, and returning to the Finder merely to launch another application is clearly an unnecessary extra step.

There are two ways in which you can avoid using the Finder in a limited-memory setup. One is to use a utility that allows you to choose the application that is launched immediately after you quit the current one. (Normally, of course, the Finder is the application that is launched when you quit a program; the utility program simply substitutes the application of your choice in place of the Finder.) The other method is to use a program which, in essence, replaces the Finder with a smaller, faster, more efficient temporary stopover. You lose some things and gain others with such Finder-substitutes, but what you gain, greater speed and easier access to your major programs, usually outweighs what you lose.

Note that although the utilities covered in this section provide greater speed (when switching from application to application) than is possible with the Finder and other advantages as well, they are not the fastest way to go. For that, you need lots of memory so you can utilize the strategies described in the "Ample-Memory" section. But if you have even a little RAM to spare, you can speed up the programs discussed in this section through the judicious use of a RAM disk or with a RAM cache. For more on that subject, refer to Chapter 7.

Using Direct-Transfer Utilities Direct-transfer utilities allow you to go directly from one application to another, completely bypassing the Finder in the process. Although the details differ, the basic idea is that once activated, the utility opens a file selection box when you quit an application. You make your selection and the new application is launched immediately. Despite their outwardly simple nature, the speed and facility with which direct-transfer utilities allow you to slip from program to program should not be ignored.

Because direct-transfer utilities require that you close your current document before you leave the application, they make most sense in situations where you switch from application A to application B to application C, and so on. They are least efficient if you are switching from application A

to application B and back again, for you would have to close, then reopen, then close the same document over and over again.

Remember also, these are no-frills utilities. If you want to move files, delete them, create new folders, or otherwise clean house, you must return to the Finder or use a desk accessory such as *DiskTop* (CE Software) or *DiskTools II* (Electronic Arts) to do so. Finally, keep in mind (you knew this was coming, didn't you) that direct-transfer utilities occasionally bomb in their attempt to quit the current application. If your applications are well-behaved, and if you are careful to save and close all your documents before you quit (rather than waiting to be reminded to do so after you select the Quit command), you shouldn't have any problems. Even so, it is nonetheless wise to test a transfer utility thoroughly with the programs you intend to use it with.

There are many direct-transfer utilities available from a variety of sources. The two described here, *SkipFinder* and *SafeLaunch*, are among the most reliable. One is a desk accessory, the other both a desk accessory and an FKey (your choice). They also differ in other ways as well. Which one suits you best is a purely personal decision.

SkipFinder *SkipFinder* is a desk accessory; you activate it from within any application or from the Desktop simply by selecting it. Actually, when you select it from the Apple menu, nothing obvious happens. Instead, *SkipFinder* waits quietly and unobtrusively until you quit your application. Only when you quit is it apparent that you're not in Kansas anymore.

The following, by way of illustration, is how *SkipFinder* would be used during part of a typical work day:

1. Let's say you've begun the day by working up a report in Microsoft *Word*. You now want to insert an illustration in your report. The illustration was sent to you by electronic mail the night before, so you need to switch to *Desktop Express* to download it. *SkipFinder* is not yet active, so you select it from the Apple menu. Then you close your report document and quit *Word*.

2. Instead of going to the Finder, *SkipFinder*'s file selection box appears with the request that you select an application. You locate *Desktop Express* in the usual manner, select it, and click on the Open button (or you can double-click on the name *Desktop Express*).

3. *Desktop Express* now opens and you do what is required to download the illustration. Once the illustration is on your disk, you quit *Desktop Express*. The *SkipFinder* file selection box returns (since *SkipFinder* remains active until you turn it off).

4. You want to do some file reorganization—delete a few files to free up disk space and move the illustration document to the report folder—before doing any further work. So instead of selecting an application, you click on the *SkipFinder's* Finder button to return to the Desktop. Once there, you perform the required housecleaning.

5. You suspect that the illustration requires some finishing touches before you can insert it in your document, so you decide to launch *SuperPaint*. You click on the *SuperPaint* icon in the Finder, just as you always do to open an application. You then go to work on the illustration.

6. When you finish your touch-up job on the illustration, you select it with the marquee and copy it to the clipboard. Then you quit *SuperPaint*.

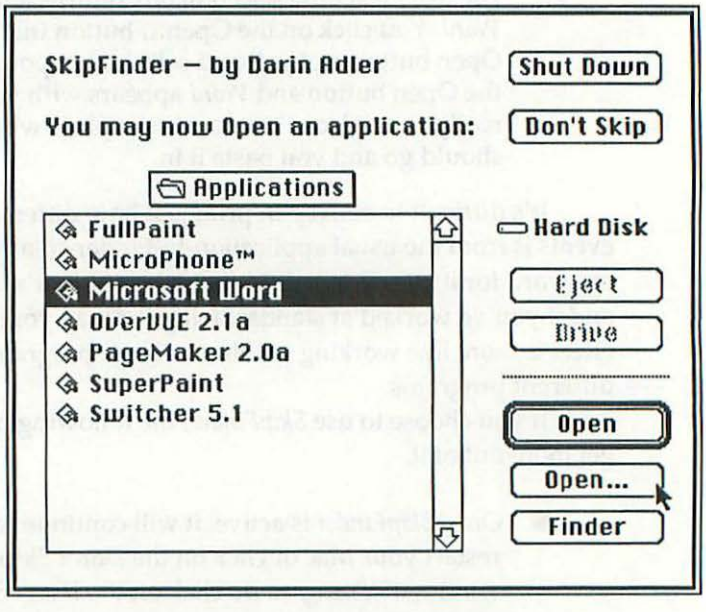
7. The *SkipFinder* file selection box returns again, and you select *Word*. You click on the Open... button (*not* the similarly named Open button, not yet) and select the report document. You click on the Open button and *Word* appears with your report open and ready to work on. You locate the place where the illustration should go and you paste it in.

It's difficult to convey in print just how different this sequence of events is from the usual application-to-Finder-to-application routine. Take my word for it; it's different. More to the point, it's faster and more efficient, and if you've worked at standardizing the way your programs function, the effect is more like working within one large program, rather than many different programs.

If you choose to use *SkipFinder*, the following pointers may help you get more out of it.

- Once *SkipFinder* is active, it will continue to function until you restart your Mac or click on the Don't Skip button in the *SkipFinder* window. When you do click on the Don't Skip button, either click on the Finder button to return to the Finder or click on one last application to open.

- If you need to do some housecleaning, click on the Finder button in the *SkipFinder* file selection box to get to the Desktop without turning off *SkipFinder*. When you have finished your work in the Finder, launch whatever application you want to work in next, just as you normally would. *SkipFinder* will reappear when you quit from that application.
- Alternatively, you can use a desk accessory such as *DiskTop* or *DiskTools II* for ordinary housecleaning tasks (such as moving, copying, and deleting files, creating new folders, and so on). *SkipFinder's* file selection box is not like most such boxes in that it allows you to use desk accessories and FKeys when the box is on the screen. Of course, you could do your *DiskTop* or *DiskTools II* housecleaning within an application as well.
- Use the Open button to launch the selected application. Select an application and click on the Open... button to select a document—*then* click on the Open button to launch them both.



With SkipFinder, *you open documents of the selected application with the Open... button. You can also use desk accessories when the* SkipFinder *window is open.*

- The Shut Down button actually performs the same function as the Finder's **Restart** command. (In other words, it is not equivalent to the Finder's **Shut Down** command.)

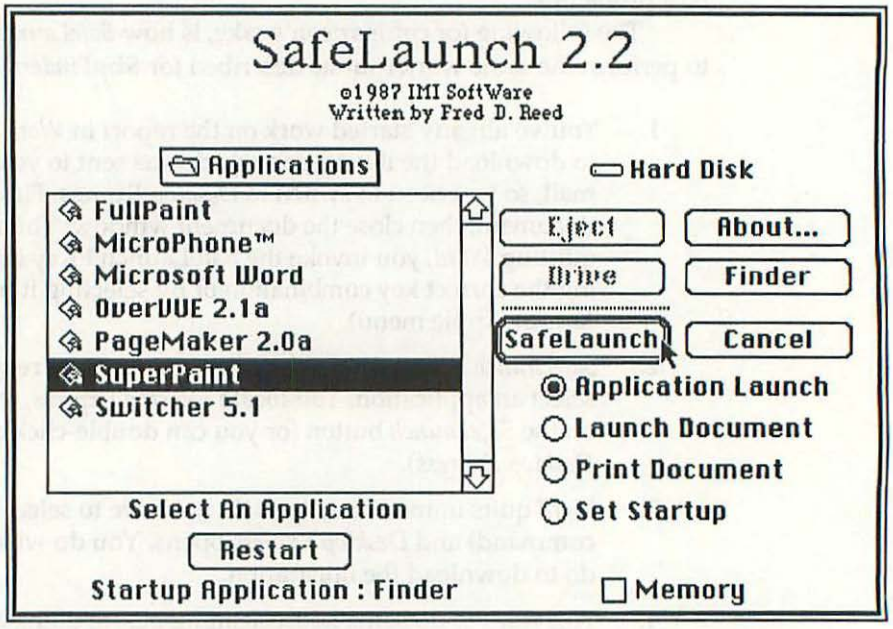
SafeLaunch SafeLaunch comes in both FKey and desk accessory format; to use it, you must first install it either with an FKey installer such as *FKey Manager* or Font/DA Mover. Once it is installed, you can use *SafeLaunch* from within any application, or from the Desktop to save you the time required to dig through open windows to find your application. (Of course, if your hard disk is organized the way it should be, this ought not to be a problem.)

The following for comparison's sake, is how *SafeLaunch* would be used to perform the same work routine described for *SkipFinder*:

1. You've already started work on the report in *Word*. You now want to download the illustration which was sent to you by electronic mail, so you need to switch to *Desktop Express*. First you save your document, then close the document window. Then without quitting *Word*, you invoke the SafeLaunch FKey (either by pressing the correct key combination or by selecting it from the *PopKeys* or Apple menu).
2. *SafeLaunch's* file selection box appears with the request that you select an application. You locate *Desktop Express*, select it, and click on the *SafeLaunch* button (or you can double-click on the name *Desktop Express*).
3. *Word* quits immediately (you do not have to select *Word's* **Quit** command) and *Desktop Express* opens. You do what you need to do to download the illustration.
4. You want to do some housecleaning before doing any further work. So, instead of invoking *SafeLaunch*, you quit *Desktop Express* in the usual manner and return to the Desktop. Once there, you perform the required file maintenance.
5. To give the illustration some finishing touches, you decide to launch *SuperPaint*. You click on the *SuperPaint* icon in the Finder, just as you always do to open an application. (Alternatively, you could invoke *SafeLaunch*, select *SuperPaint*, and click on the *SafeLaunch* button. This might save you the trouble of searching through a bunch of poorly organized windows.) You then go to work on the illustration.

6. When you finish your touch-up job on the illustration, you select it with the marquee and copy it to the clipboard. Then you invoke *SafeLaunch*.

7. The *SafeLaunch* file selection box appears, and you select *Word*. You click on the Launch Document radio button to turn it on and then click on the Select button. Then you select the report document, click on the *SafeLaunch* button and *Word* appears with your report already open. You locate the place where the illustration should go and you paste it in.



SafeLaunch *must be selected each time you want to transfer to a new application.*

As with *SkipFinder*, a written description cannot adequately convey just how different the above sequence of events is from the usual application-to-Finder-to-application routine. You have to see it to believe it. But believe me, using *SafeLaunch* is faster and more efficient, and as with *SkipFinder*, more like working within one large program rather than many different programs.

If you choose to use *SafeLaunch* instead of *SkipFinder*, the following pointers may be of help when using *SafeLaunch*.

- *SafeLaunch* does not sit in the background once it is invoked, waiting for you to quit your application. You must select *SafeLaunch* every time you want to switch to another application. If you quit in the usual way, you will end up, as usual, back in the Finder. If you invoke *SafeLaunch* and then decide you want to go to the Finder instead of another application, just click on the Finder button (or press Command-F).
- *SafeLaunch* uses the arrow keys for scrolling and Command-key combinations for nearly all options and commands. To scroll through the selection box and move from folder to folder, use the arrow keys just as you normally would in a file selection box. To press a button or select an option, use the Command key in combination with the first letter of the button or option (Command-L for Launch Document, for example). Only the About..., Select, and *SafeLaunch* buttons do not work this way. About... must be clicked on, and both Select/*SafeLaunch* can be invoked (as you can see from the bold button border) simply by pressing Return.
- Since *SafeLaunch* automatically quits the current application when you press the *SafeLaunch* button, you should always make certain that you're ready to make the switch before you invoke *SafeLaunch*. For safety's sake (and this applies to everyday quit practices as well as those with *SafeLaunch*), save all your documents and close all your windows before you call up *SafeLaunch*. It's nice to know that *SafeLaunch* will close windows and ask if you want to save your documents if you forget to do so yourself, but it's much safer to use *SafeLaunch*'s abilities as a backup defense, rather than to rely on them all the time.
- To launch an application and a document together, click first on the Launch Document radio button (or press Command-L). Locate the desired application, select it, and click on the Select button (formerly the *SafeLaunch* button) or press Return. Select the document you want opened and then click on the *SafeLaunch* button (or press Return). The document selection list displays only

those documents created by the selected application. If you want to open a document not created by the selected application (a *MacWrite* file when you are launching *Word* or a *MacPaint* document when you are opening *FullPaint*, for example), press Command-V (for View all documents) before selecting the application you want to open to get a display of all the documents on the disk. Be sure to select a document which your application can, indeed, open.

- If you invoke *SafeLaunch* and then decide you really don't want to quit your current application, click on the Cancel button (or press Command-C).
- *SafeLaunch*'s Restart button (or pressing Command-R) does the same thing that the Finder's Restart command does. Just to be safe, be sure all your documents have been saved and closed before clicking on Restart.
- On rare occasions, after you click on the *SafeLaunch* button (or press Return), the word *SafeLaunch* appears on the menu bar and nothing else happens. If this happens to you, *SafeLaunch* has most likely failed to get your application to quit. To get back on the track, quit the application as you normally would; *SafeLaunch* will then take over.
- *SafeLaunch* is billed as being compatible with *Oasis*, which is described later in this chapter. By using these two utilities in combination, you can have at your disposal two quick methods for switching applications. Which one you would want to use at any given moment will depend on how your *Oasis* screen is configured, your state of mind, and other imponderables.
- Desk accessories that are still open when you invoke *SafeLaunch* are closed during the subsequent automatic quit. However, if you are using a large desk accessory, such as a spelling checker or a thesaurus, it is wise to close it (or them) before calling up *SafeLaunch*.
- In my experience, I have found that *SafeLaunch* may occasionally crash. This may be due to the fact that my System file is loaded with all sorts of desk accessories, FKeys, and fonts, and/or that I have a partially incompatible Init file in my System folder. Or it

may be due to some other problem. In any event, you should be sure to test *SafeLaunch* thoroughly before using it in your everyday work. If you find any problems, you should try to isolate the cause. You can then use *SafeLaunch* safely simply by avoiding those situations which cause problems. Two functions which you should be sure to test thoroughly before using are the program's Memory option (which is explained in the *SafeLaunch* documentation) and the document launching routine.

Using Finder Substitutes Finder-substitutes, obviously enough, take the place of the Finder. They are designed to give you quick and easy access to all your applications, often via a screen full of customizable buttons. No longer do you need to flip through windows or open folders to locate applications; the Finder-substitute already knows in which folders your applications and documents reside. Instead, you just click on the button representing the application (or application and document) and, much more quickly than it would from the Finder, the new application is launched. Each time you quit an application, the substitute reappears instead of the Finder.

To achieve a significant speed advantage over the Finder, Finder-substitutes are usually limited in one or more ways. The number of buttons, and hence the number of applications you can open via a button, may be fairly small. Some or all of the Finder's housekeeping functions may be missing, or the ease with which those functions are performed may be lacking as well. (You can't drag a file to move it from one folder to another, for example; if it is possible to do so at all, you may have to select the file in a selection box, then select the new folder, then click on the Move button.) Also, as with direct-transfer utilities, since you must quit one application to go to another, the use of Finder substitutes makes more sense if you are going from application A to application B to application C, and less sense if you are switching from A to B and back again.

By way of introduction, the following is how a Finder-substitute such as *Oasis* or *PowerStation* would be used to accomplish the same sequence of events as was described in the *SkipFinder* example. You may wish to refer back to that example to see how the two types of programs differ.

1. You've begun the day by working up that report in *Word*. As before, you want to insert an illustration in your report, an illustration that was sent to you by electronic mail. Thus you need to

switch to *Desktop Express* to download it. You close your report document and quit *Word*.

2. The Finder-substitute's screen quickly appears. You easily locate the *Desktop Express* button, which is in the telecommunications row of buttons, and click on it.
3. *Desktop Express* now opens and you download the illustration. Once you have the illustration, you quit *Desktop Express*. The Finder-substitute's screen reappears.
4. You want to do some file reorganization—delete a few files to free up disk space, move the illustration document to the report folder—before doing any further work. So you pull down the Utilities menu, select the **Delete** command and delete the desired files. Then you select the **Move** command and move the illustration file into the report folder.
5. Since you suspect that the illustration requires some finishing touches, you decide to launch *SuperPaint*. You click on the *SuperPaint* button and go to work on the illustration.
6. When you finish your touch-up job on the illustration, you select it with the marquee and copy it to the clipboard. Then you quit *SuperPaint*.
7. The Finder-substitute screen returns and you click on the *Word* button. You open the report document, locate the place where the illustration should go and paste it in.

There are a number of well-executed Finder substitutes available from a variety of sources. The three described here, *Oasis* (which was the model for the previous illustration), *SuperFinder*, and *PowerStation* are among the best. They differ, however, in flexibility, in setup, and in other ways as well. In this section, I also show how *DiskTop*, *DiskTools II*, and *QuicKeys* can be used as Finder substitutes (although, to be more accurate, each in its own way is a kind of cross between a transfer utility and a Finder substitute).

It may be appropriate to note here that programs such as *Oasis* and *SuperFinder* are also useful if you have a hard disk loaded with files and folders, and other people need to use your Mac. You could set up *Oasis*, for example, so that everything the outside user needs is available from a button on the *Oasis* screen. As a result, the outside user will never need to search unfamiliar folders in an attempt to find applications and documents; everything he or she needs will be as close as the click of a button.

Oasis *Oasis* can be customized in all sorts of ways. You create the buttons on the *Oasis* screen yourself, and thus you are free to determine their size, location, organization, and abundance. Once created, buttons can be resized, moved, and reassigned. The size of the text in the buttons can be changed with a menu command (and the font can be changed, if desired, with *ResEdit*). Buttons can be used to open just applications or applications and specific documents. *Oasis* also has a Utilities menu which contains most of the Finder's housekeeping functions. It also has a built-in screen saver which can be set to activate at one of a number of different idle intervals.

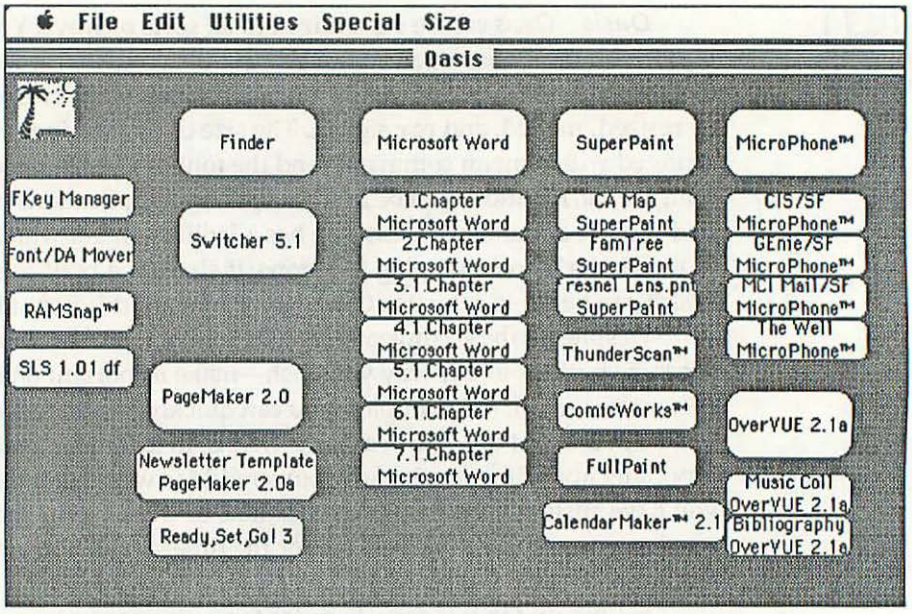
Despite all these conveniences, *Oasis* is fast; and because you can organize buttons in any way you wish—make important ones big and minor ones small, for example—you can quickly locate the button you want and be in and out of *Oasis* in a flash. That is, in fact, what makes *Oasis* especially appealing for the fully powered Mac with limited memory: just as you have customized all your applications to work well together, and just as you have customized the Desktop for the times when you are at work there, *Oasis* can be custom designed to closely fit your needs.

Setting up *Oasis* is easy. In order for your programs to return to *Oasis* automatically (instead of to the Finder), *Oasis* must be in your System folder. To get *Oasis* to take the reins from the Finder, you simply launch it. You can launch *Oasis* by double-clicking on the *Oasis* file itself. Or to avoid the hassle of having to open the System folder and look for *Oasis*, you can put the *Launch Oasis* file somewhere readily accessible, then simply double-click on *Launch Oasis*. Or if you set *Oasis* as the startup application in the Finder every time you start your Mac, from that point on *Oasis* will be in control.

Oasis has a great many features and options. Thus it is impossible to cover every aspect of the program. What I can do, however, is first go over some of the basics involved in creating and adjusting *Oasis* buttons. After that, I'll review some pointers on how to organize an *Oasis* screen so that it is fast and efficient.

First some *Oasis* button basics:

- To create a button, merely drag the pointer to create its outline, just as you would with the rectangle tool in your Paint program. As soon as you release the mouse button your new *Oasis* button will appear.



In Oasis, *you create buttons of whatever size you want and position them wherever you like on the screen.*

- To assign an empty button to an application, just click on it. A file selection box will appear. Locate the desired application, select it, and click on the Open button. The name of the application will now appear in the button.
- To assign an empty button to an application and document combination, click on the button while holding down the Option key. Select the application first, then the document. The name of the document (above) and the application (below) will appear on the button.
- If the name or names on a button run off the edges, you can change the size and shape of the button to accommodate them. First, hold down the Command key. Then place the pointer over the button and press the mouse button. Drag the pointer until you have enlarged the *Oasis* button as much as needed, then release the mouse button. The names will recenter themselves in the resized button.

- To move a button to a new location, first select the **Move Button** command from the File menu, or press Command-M. The pointer should change to a hand. Place the hand over the button and drag it to the new location. The pointer will remain a hand, and you can thus continue to move buttons, until you click on something other than a button.
- To remove an application assignment from a button, select the **Clear Button** command (or press Command-W). The pointer will change to an X in a circle. Point to the offending button and click.
- To remove a button entirely, select the **Remove Button** command (or press Command-R). As with the **Clear Button** command, the pointer will change to an X in a circle. Point to the button you want to remove and click.
- Cut down on button creation time by using the **Duplicate Last Button** command (or pressing Command-D immediately after creating a button). Create the size and shape button you want, then make a whole column of them (if you want) by pressing Command-D half a dozen times or so. Create a different size button and repeat the process. When you have all the buttons you want, then move them into position.

Now for some pointers on creating an effective *Oasis* screen:

- Neatness counts! So always work with the Grid option (found in the File menu) turned on. With the grid on, you will find it easier to create buttons of a uniform size and shape, and easier to line your buttons up in rows and columns.
- Organization counts, too. There are many ways in which you could organize the buttons on your *Oasis* screen. The following are a few useful ones:
  - ◊ If you use only a small number of major applications, put buttons for the modules of your megaprogram across the top of the screen and put smaller buttons for all your secondary programs and documents underneath.
  - ◊ If you have a large number of megaprogram modules, clump related modules together: graphics programs here, text programs there, database material over there. Secondary

applications could be included in the clumps as well but displayed in smaller buttons.

◊ If you rarely use anything other than your main applications, you might want to set up a column of buttons for each program. The application alone is at the top (in a big button) and underneath are smaller buttons for a variety of commonly used documents, such as templates for letters, memos, reports, and so on.

The important point is to organize the buttons so that everything you normally need is in a button, and you can find what you want fast. Whether you choose one of these arrangements or some other one, the main thing is that it works for you.

- Don't clutter up your *Oasis* screen with too many applications. Less frequently used applications should be launched using *Oasis'* **Transfer...** command (in the File menu). This command is essentially a built-in *SkipFinder*, except that you cannot select documents to open along with applications. When you are finished with the application you transferred to, you will return to *Oasis*, not the Finder.
- There are two ways to get to the Finder from *Oasis* (if you need to get to the Finder; don't forget that *Oasis* Utilities menu contains many of the Finder's housekeeping functions). One method terminates *Oasis* and one does not:

  ◊ If you select **Quit** from the File menu (or press Command-Q), *Oasis* will relinquish control and return you to the Finder. Unless you launch *Oasis* again, your applications will quit to the Finder as usual.

  ◊ You can put the Finder in a button just like any other application. Click on the button to go to the Finder. When you are finished with the Finder, launch any application as usual. When you quit that application, you will return to *Oasis*, not the Finder.

- You can change the size of the text in all of the buttons with the Size menu. Obviously, the larger the text, the larger the buttons need to be. What size you choose will be determined by good

taste, the number of buttons you want to fit on the screen, and how soon you want to see your optometrist again.

- You can also change the font used for the buttons, but only by getting your hands dirty with *ResEdit*. Follow the directions in *Oasis* documentation to do so.

SuperFinder Although the button assignment process is different, *SuperFinder* (public-exchange) is much like *Oasis* in performance and features. You must launch *SuperFinder*, like *Oasis*, to have it take over from the Finder. Each time you quit an application, *SuperFinder* appears (quickly). You then click on the button corresponding to the application you want to use next. An application alone or an application and a specific document can be assigned to a button. A File Utils menu contains many of the housekeeping functions of the Finder, and a Shut Down menu contains the standard Restart and Shut Down commands. To launch an application which is not assigned to a button, you click on the large Other... button, which is at the top of every *SuperFinder* screen. To return to the Finder and quit *SuperFinder*, you click on the large Finder button, also at the top of every screen.

Installing *SuperFinder* is similar to installing *Oasis*. To have your programs return to *SuperFinder* automatically, you must put *SuperFinder* in your System folder. To get *SuperFinder* to take the control from the Finder, you must launch it. You can launch *SuperFinder* by double-clicking on the *SuperFinder* file itself, by double-clicking on the *Run SuperFinder* file (which can be placed somewhere convenient on the Desktop), or by setting *SuperFinder* as the startup application. If you choose the last option, *SuperFinder* will appear automatically every time you start your Macintosh.

The one major difference between the two programs is that if you need more than the number of buttons that fit on a single screen, you can create several *SuperFinder* button sets, each with a different name and each with its own set of buttons. This is also useful if you want to segregate buttons by type or function, each on a different *SuperFinder* screen. Button sets are reached via the Set menu, which contains a list of all the different sets.

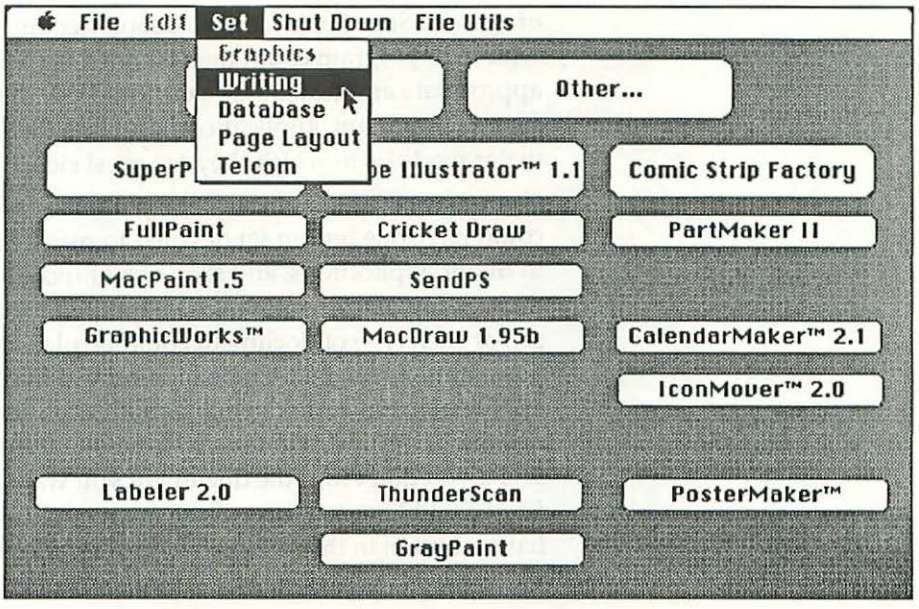
As in the previous section, the following are some of the program's basics, especially those that are different from *Oasis*. Then I review some pointers on how to organize *SuperFinder* so that it is as fast and efficient as possible. For the complete story on *SuperFinder*, I refer you to the *SuperFinder* documentation.

First, some *SuperFinder* basics:

- A *SuperFinder* set starts out with only two buttons, the Other... and Finder buttons. To add a file (and a button to go with it), select **Add Docs or Apps...** from the File menu (or press Command-A). In the resulting file selection box, select the desired application or document (you do not need to select an application before selecting a document) and click on the Add button. The file selection box will then reappear; continue to select files and click on the Add button until you've made all the desired button assignments. Then click on the Cancel button. Buttons are assigned in order from left to right and top to bottom.
- To clear a button assignment (and remove the offending button), select **Remove Docs or Apps...** from the File menu (or press Command-R) and click on the button or buttons you want to clear. To cancel the Remove command, go back to the File menu and click on **Stop Removing** (which should be where **Remove Docs or Apps...** used to be).
- To add a new set of buttons, select **Add Set...** from the File menu. Give the set a name and press Return or click on the OK button. The new set of empty buttons will be created, displayed on the screen, and added to the Set menu.
- To remove a button set, select **Remove Set** from the File menu, then select the set to be removed from the Set menu. Although **Remove Set** allows you to delete only one set each time it is invoked, you can remove any set, even one with a full complement of buttons. So take care using this command. You cannot remove the only remaining button set.

Now that you know a little bit about how *SuperFinder* works, the following are some ideas on how to use it effectively.

- To move a button, hold down the Option key, click on the button and drag it to its new position. A hollow rectangle will show the new location of the button as you drag it into position. Release the mouse button and the button pops into place.
- To resize a button, hold down the Command key, click on the button, and drag the pointer. A small right angle shows where the lower right corner of the newly resized button will be (the upper



With SuperFinder, *you can create as many button sets as you want. You could, for example, create a set for each major type of application.*

left corner will stay where it is). Release the mouse button and the button shrinks or grows as desired.

- When you assign a document to a button, *SuperFinder* (unlike *Oasis*) does not display the name of the application to which it belongs. To help you determine what kind of document you're dealing with, you may want to list documents beneath their corresponding application button. Or you could assemble documents in button sets, each set for a particular application. Or you could institute a naming convention for documents used in buttons, in which each document name contains a reference to the application to which it belongs (such as Letter.wrd for a letter template for Microsoft *Word*).
- There are many ways in which you can organize your *SuperFinder* buttons and button sets. Here are just a few of the possibilities:
  - ◊ If you have just a few megaprogram modules, you could line up the modules (applications) across the top or down the side

of a single *SuperFinder* screen. Useful documents associated with each program could then be lined up below or beside the appropriate application. Or you could use the extra buttons for less important applications, and line them up with the major modules to which they are most closely related.

- ◊ If you have a large number of important applications, you could have one button set devoted to major applications, one to minor applications, and then one or more sets devoted to documents. Documents could be lumped together in a single set, or each type of document could be placed in a special set. Since buttons can be repeated, you could have each major application appear in the main application set, and in the set devoted to its documents as well, so you don't have to switch sets if you don't find the document you want. (For example, *Excel* could have a button in the main application set and also have a button in the set devoted to *Excel* files.)
- ◊ You could have each button set devoted to a specific application and its documents, and have no main application set at all. With this scheme, to find the application you want, you would have to select a new set from the Set menu virtually every time you were in *SuperFinder*. In some circumstances, however, it might still make sense.
- ◊ Or you could devote your sets to specific tasks (newsletters, for example) or devote your buttons to document templates (business letter, personal letter, newsletter template, etc.).
- ◊ Finally, if there are several people who use your Macintosh, you could set up a set for each user, tailoring the button choices to their specific needs.

■ Once a button set is named, there is no way to change its name without destroying the set. So be sure the name you give a set when it is created is a good and logical choice. If you don't want a button set called Primary Set (this is the one set that comes with *SuperFinder*), create at least one other set (since there must always be at least one button set), then remove the Primary Set.

Demo

PowerStation *PowerStation* (Software Supply) has some of the features of both *Oasis* and *SuperFinder*, and a number of unique features as well. It is, overall, the most powerful of the three. Each of *PowerStation*'s

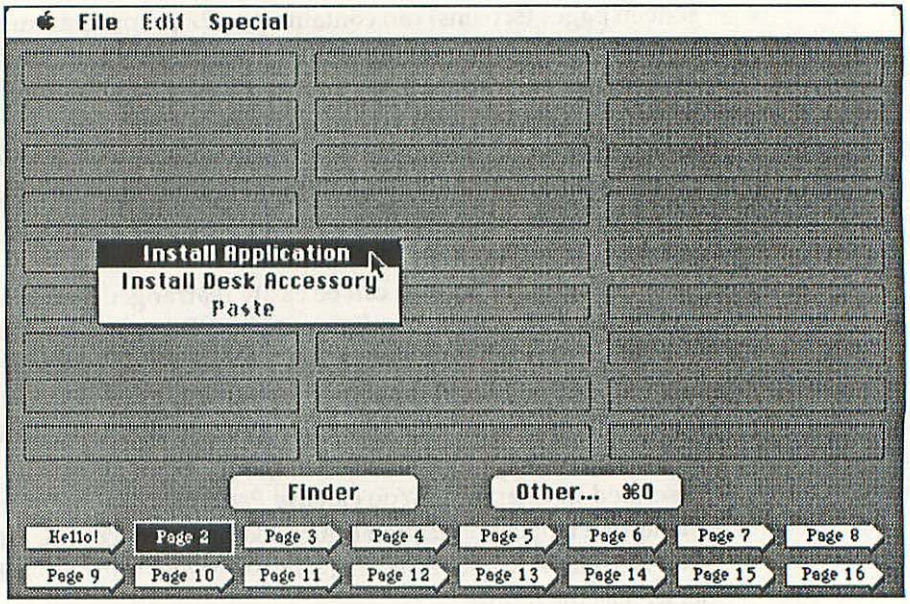
sixteen pages (screens) can contain up to 27 preprogrammed buttons. The buttons can call up an application, an application and one or more specific documents, or a desk accessory. Two rows of page buttons at the bottom of the screen provide quick access to each set of buttons. All buttons can be renamed in whatever way is most meaningful to you (without altering the name of the corresponding application or document). One button can open either an application or that application and a specific document, and thus it can act as if it were two different buttons.

PowerStation buttons can be easily rearranged within the button grid on that page or moved to a different page. *PowerStation* also contains a number of useful Clean Up options for automated button rearrangement, and you can "anchor" those buttons you don't want moved by the **Clean Up** command. *PowerStation* incorporates a screen-saver utility much like *Pyro!* (The package also includes the stand alone version of *Pyro!*, which is described in Chapter 5.) You can use *PowerStation* to print multiple documents of a single application (much like the print command in the Finder). You also can navigate *PowerStation*'s pages and buttons with the keyboard as well as the mouse.

PowerStation does not contain any of the Finder's housekeeping functions. Thus in order to move, delete, or copy files and create new folders, for example, you'll have to go to the Finder or install *DiskTop, DiskTools II,* or a similar desk accessory (which can be run from a button called "housekeeping" if you like). Another limitation is that *PowerStation* buttons cannot be made larger or smaller, nor can they be moved to any location other than those in the preset button grid. These, however, are small quibbles compared to what *PowerStation* can do.

So you can directly compare *PowerStation* with *Oasis* and *SuperFinder,* the following are some *PowerStation* basics:

- To assign an application to a button, first click on an empty part of the screen to display the button grid which shows the outlines of all empty buttons. Press and hold the mouse button when the pointer is over the button you want to work with, select **Install Application** in the resulting popup menu and release the mouse button. Select the desired application in the usual way.
- To assign one or more documents to a button, first assign the appropriate application. Then click on the small document icon to the right of the main button. Click on the Install button in the



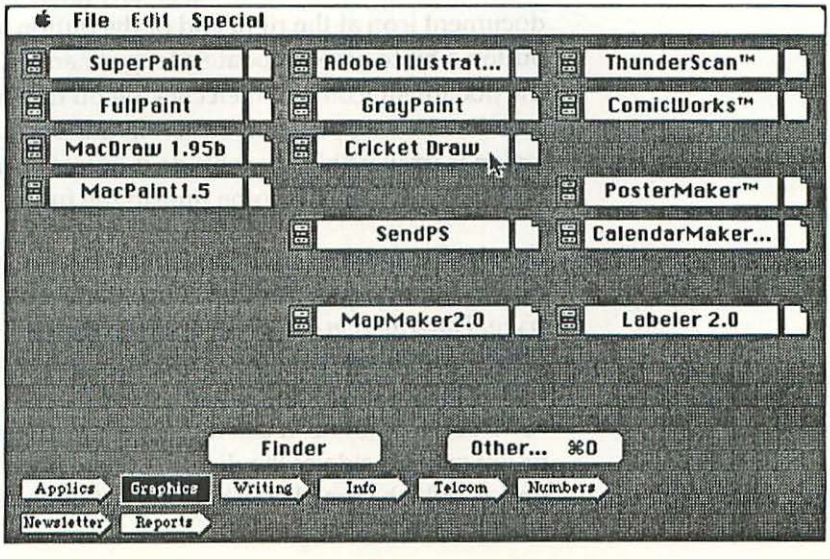
To install an application in a PowerStation *button, click on a button outline and select the* ***Install Application*** *command in the resulting popup menu.*

resulting window, select a document you want to install, and click on the Install button again. At this point you can make the *PowerStation* button always open the selected document by clicking on the Dedicate the Application Box to this Document button. Or you can repeat the sequence and select other documents, so that you will have a number of documents to choose from. (There are other options for handling documents as well.)

- Once your *PowerStation* button is set up, you click on the main button to open the application (or application and document, if you have dedicated the button to a single document) or click on the document icon to the right of the main button to open the document selection box and choose from the documents you have installed.
- To rename a button, click on the menu icon to the left of the main button to select the **Rename/Anchor...** command. The new name applies only to the button to make it easier for you to recognize and does not rename the application or document. If you click on

the **Anchor...** check box to turn it on, the button will not move when you select the **Clean Up** command.

- To move a button, place the pointer over the button, press the mouse button, and drag the button to its new location. It will "snap to" the nearest location in the grid.
- To remove a previously assigned button, click on the small menu icon and select the **Cut** command.
- To move to a new page, click on the box displaying the number or name of the page you want to go to.
- You can keep the numbering scheme by which pages are identified or change the numbers to names which are more meaningful. To do so, you simply press on the button for the current page, select the **Rename...** command, and enter the new name in the ensuing dialog box. Page selection buttons are fairly small, so your name must be relatively short.
- If you want to delete the contents of a page and start over again, press on the button for the current page and select the **Cut Page** command.



You can have sixteen different "pages" or sets of buttons in PowerStation. *Just click on an arrow button at the bottom of the page to go to a different page.*

Many of the more general tips and techniques described for *Oasis* and *SuperFinder*, especially those dealing with button organization and placement, apply equally well to *PowerStation*. The following are some additional tips that apply specifically to *PowerStation*.

- *PowerStation* can be operated almost exclusively from the keyboard, as well as with the mouse. Simply typing the first few letters of a button name will select the button, and once it is selected, pressing Return is the same as clicking on it. *PowerStation* has many other keyboard shortcuts, many of which are displayed by the **Shortcuts...** command in the program's Special menu. Take advantage of them.

- In order to make operating *PowerStation* from the keyboard easier, use a button's **Rename/Anchor...** command to make button names more distinctive. Strip off all the "Macs," for example, in program names—MacPaint, MacWrite, MacWhatever—so that you don't have to run your M, A, and C keys into the ground.

- If you have no documents installed for a particular application button and don't intend to install any later, you can remove the document icon at the right end of the button and thus lengthen the button a bit to accommodate a longer name. Do this by clicking on the document icon and then clicking on the Remove the Document Icon For This Application button in the Document window. This action is irreversible, so don't do it unless you're sure you won't be adding documents some time in the future.

- Use the **Clean Up...** command and its many options to automatically reorganize your buttons once you've completed work on a page. *PowerStation* can order your buttons alphabetically by their on-screen names or by the actual application and document file names, or it can simply neaten up the current order. It can fill by rows or by columns. It can move your buttons to the top, bottom, or left or right side of the display. There is even an **arrange nicely** option, which spreads the buttons out on the screen if there is room.

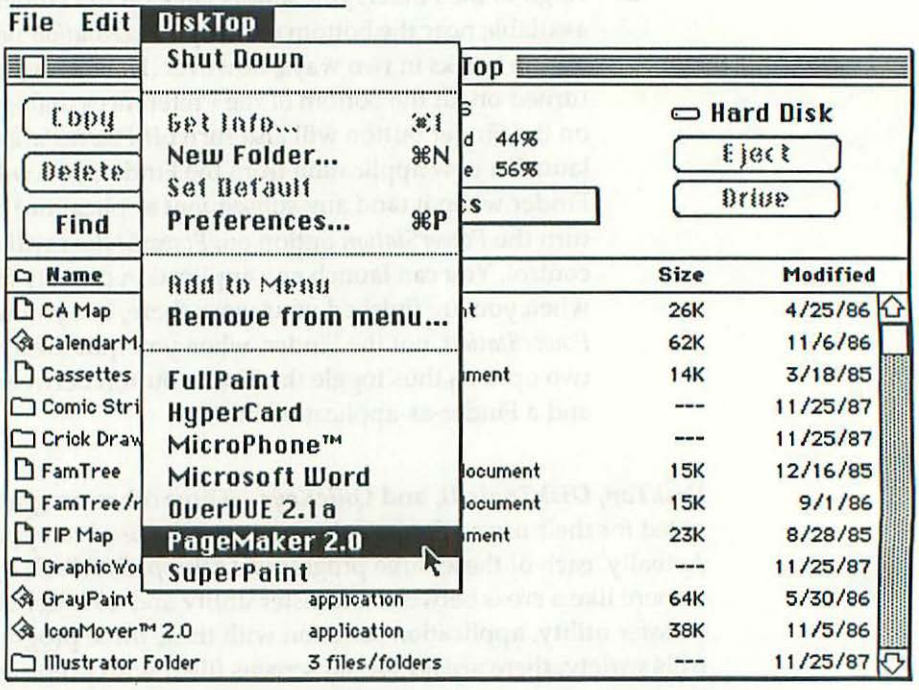
- Take advantage of *PowerStation*'s "type ahead" feature to bypass the *PowerStation* display and switch directly from one application to another. While your current application is quitting, if you type enough letters to select a *PowerStation* button and then quickly press Return, *PowerStation* will skip its own display and instead immediately open the application (and document, if any) indicated by the button.

- To go to the Finder, you simply click on the Finder button which is available near the bottom of every *PowerStation* page. The Finder button works in two ways, however. If you have the Finder option turned on (at the bottom of the Preferences dialog box), clicking on the Finder button will also turn off *PowerStation*; when you launch a new application from the Finder, you will return to the Finder when it (and any subsequent application) quits. But if you turn the *PowerStation* option on, *PowerStation* will remain in control. You can launch any application directly from the Finder when you are finished your work there, but you will return to *PowerStation*, not the Finder, when you quit the application. These two options thus toggle the Finder button between a Quit button and a Finder-as-application button.

DiskTop, DiskTools II,* and *QuicKeys Three other programs, although not noted for their use as Finder substitutes, can be used to bypass the Finder. Actually, each of these three programs, *DiskTop*, *DiskTools II*, and *QuicKeys*, is more like a cross between a transfer utility and a Finder substitute. As in a transfer utility, application selection with these three programs is of the no-frills variety; there are no special screens filled with preprogrammed buttons to click on. But like a Finder substitute, each program offers its own unique set of extra features which can enhance your computing environment.

I have already mentioned how useful *DiskTop* can be if you want to copy, move, or delete files (or perform some other housekeeping function) from within an application. But you can also use *DiskTop* to transfer from one application to the next (or to open a new application in MultiFinder), using any one of the following three methods.

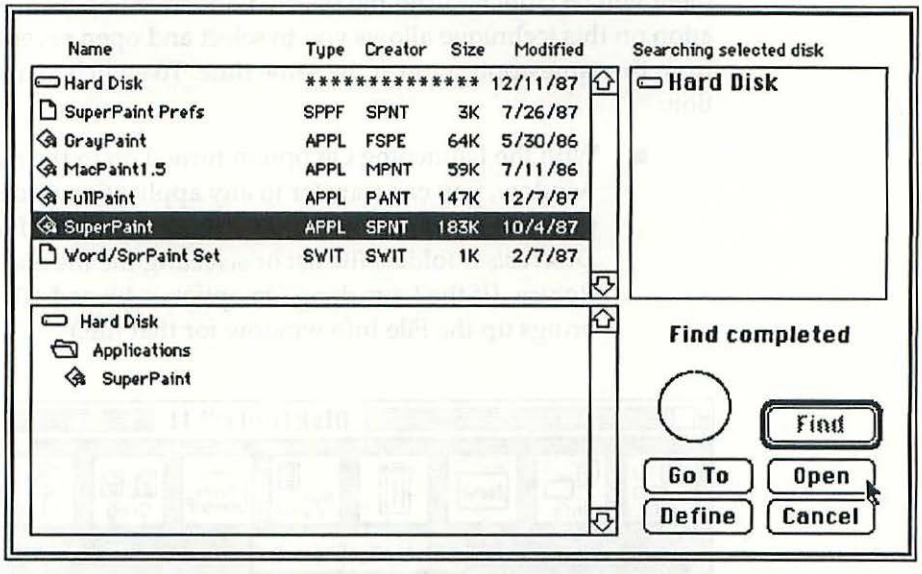
- With the **Add to Menu** command, you can add up to twenty applications or documents to the *DiskTop* menu. To transfer to any of these, you simply open *DiskTop*, pull down its menu, and select the desired application or document. If a document is chosen, *DiskTop* will launch the application that created it and then open the selected document.



You can transfer from one application to another by selecting a program from the DiskTop *menu.*

- You can launch any application or launch an application and open a document simply by double-clicking on the name of the appropriate application or document in any standard *DiskTop* file listing. Or you can select the file and press Command-O.

- If you used the Find button to locate a file (either a document or application), you can select the file in the found files list and click on the Open button to transfer to the desired file. As before, if a document is chosen, *DiskTop* will launch the application that created it and then open the selected document.



After DiskTop *finds a file for you, you can launch it (or the application that made it) by clicking on the Open button.*

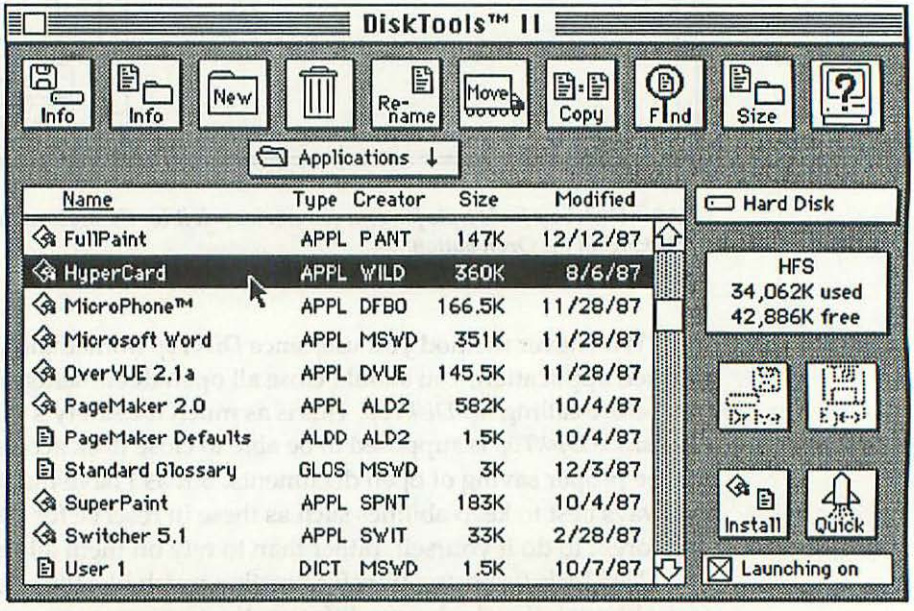
Whichever method you use, since *DiskTop* immediately transfers to the selected application, you should close all open documents and desk accessories before calling up *DiskTop*. This is as much for safety's sake as anything else, since *DiskTop* is supposed to be able to close desk accessories and allow for the proper saving of open documents. But as I have mentioned before, it is always best to keep abilities such as these in reserve for the times when you forget to do it yourself, rather than to rely on them all the time.

DiskTools II sports a transfer function much like the one in *DiskTop*. In fact, although they look very different, the two programs are otherwise quite similar in their abilities. But there is one significant difference worth mentioning in regards to the present discussion. This is the fact that

DiskTools II does not work with MultiFinder. (According to the manufacturer, by the time you read this, a new, MultiFinder-compatible version of *DiskTools II* should be available.)

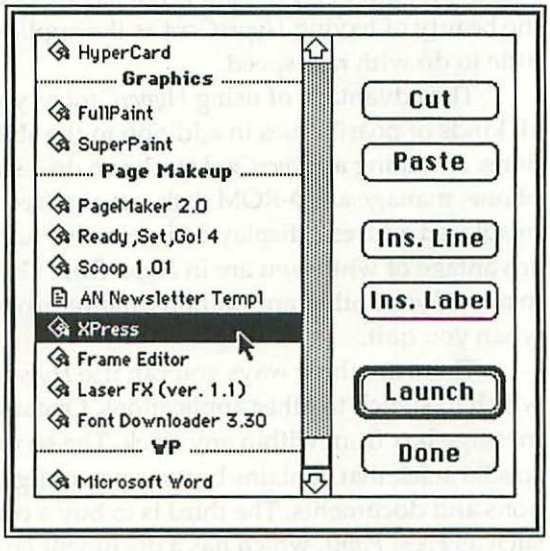
As with *DiskTop*, you can use any one of three different methods to transfer to another application or document in *DiskTools II*. Also as in *DiskTop*, if you select a document, the application that created that document will be launched and the selected document opened. A useful variation on this technique allows you to select and open several documents from the same application at the same time. To switch to another application:

- With the Launching On option turned on in the main *DiskTools II* window, you can transfer to any application or document (or group of documents) by double-clicking on the file name in any *DiskTools II* folder/file list or selecting the file and then pressing Return. (If the Launching On option is turned off, double-clicking brings up the File Info window for that file.)



When Launching On is turned on, double-clicking on a file name in any DiskTools II *file list will transfer you to that application or document.*

- If you have used the Find button to locate a file, double-clicking on the name of the file in the found file/folder list will return you to the main *DiskTools II* window with that file highlighted. If Launching On is turned on, double-clicking again (or pressing Return) will transfer you to the found document or application.
- You can also click on the Quick Launch button (or press Command-Q) in the main *DiskTools II* window, select a document or application from the resulting list, and click on the Launch button (or press Return). *DiskTools II*'s Install button adds applications or documents to this Quick Launch list. The list itself starts out in alphabetical order, but it can be rearranged in any meaningful way, and labels and dividing lines added as desired. There is no practical limit to the number of documents and applications you can place on the Quick Launch list.



The DiskTools II *Quick Launch list can be rearranged and labeled to fit your needs.*

Finally, as mentioned before, *QuicKeys* can be used to construct a type of transfer utility which operates entirely with keyboard shortcuts. To assign a macro that launches a particular file (either an application alone or

a document and the application that made it), use the **File** command. To create a macro that opens a transfer file selection box (which allows you to transfer to any application), use the **Transfer** command from the Specials submenu. Together these two types of macros cover the field.

By the way, this ability to assign universal macros which transfer to other applications (or add them to your list of open applications, if you use MultiFinder) is especially useful if you have an extended keyboard. With that keyboard, you could assign specialty macros such as these to the real function keys on the keyboard and add labels or post a keyed list to help you keep them all straight. Of course, with the extended keyboard, there are all sorts of useful macros you could assign to your function keys.

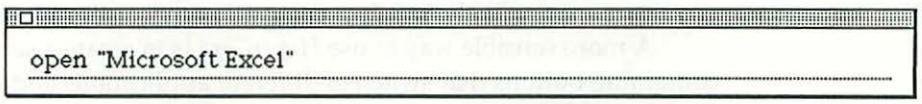
Using *HyperCard* If you haven't had the time yet to fully explore *HyperCard*'s abilities (or for those of you who have had your Macs for a while, if you have yet to obtain a copy of *HyperCard*), you may be surprised to learn that *HyperCard* can serve as a Finder substitute. Although transferring in and out of *HyperCard* is not as fast as with *PowerStation*, for example, the beauty of having *HyperCard* as the application you always return to has little to do with raw speed.

The advantage of using *HyperCard* as your home base is that it offers all kinds of possibilities in addition to the ability to switch between applications. Anything a *HyperCard* stack can do—store phone numbers, dial the phone, manage a CD-ROM disk, act as a free-form database, locate a misplaced address, display an interactive tutorial or help file—you can take advantage of while you are in *HyperCard*. On top of all that, you can switch to any of your other applications and return to *HyperCard* automatically when you quit.

There are three ways you can use *HyperCard* as a home base from which to switch to other applications. One uses the Open command and the message box from within any stack. The second involves the creation of a special stack that contains buttons preconfigured to open specific applications and documents. The third is to buy a commercial piece of stackware such as *Focal Point*, which has a document launcher as one of its modules.

The first method involves the smallest amount of up-front time and energy (or monetary investment) but is the least satisfying and the most arcane in actual operation. With this method, no matter what *HyperCard* stack is currently open, the following steps will take you to the desired application.

1. Select the **Message** command from the Go menu, or press Command-M from any card in any stack in *HyperCard*.
2. Type `open "ApplicationName"`, where ApplicationName is the name of the application you want to switch to. Be sure to type the quotation marks. Note the following points about the use of the Open command:
   a. The application name, as you have typed it, must duplicate exactly the name of the application as it appears on the disk. This means that any symbols which may occur in the name, such as trademark (™), registered trademark (®), copyright (©), and so on, must be reproduced just as they appear.



You can use HyperCard*'s Message Box and the Open command to transfer to an application at any time.*

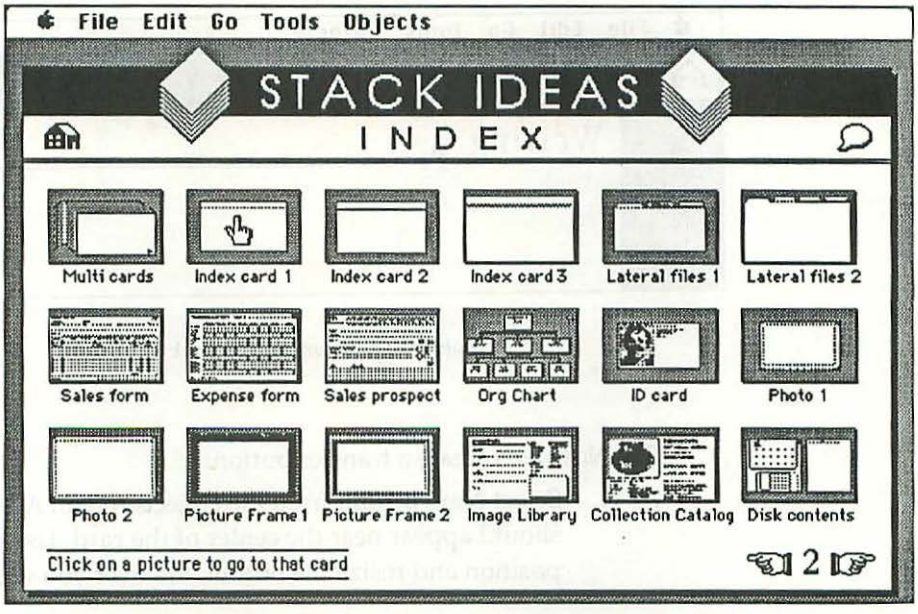
   b. You can launch a document as well as an application by typing `open "DocumentName" with "ApplicationName"` instead. As with applications, the name of the document must duplicate the name on your disk exactly.
   c. If the Look For Applications In or Look For Documents In Card of the Home stack does not contain directions to the folder containing the application or document you want to launch, you will have to enter not just the application name, but the entire search path name for that application or document. For example, if you want to switch to *MacDraw*, which is in the Draw folder of your Graphics folder on your hard disk, you would type `open "HardDisk:Graphics:Draw:MacDraw"`. (For more on search paths, see pages 29-32 of Danny Goodman's *The Complete HyperCard Handbook*.)
3. Press Return or Enter to invoke the Open command and switch to the application.

When you quit the application you have switched to, you will be returned to the exact same point (the same card in the same stack) where you left off.

Obviously, the Open command method has its drawbacks. If you work from a hard disk, you would have to be very familiar with the organization of its contents to be able to use the Open command with any degree of facility (or have all potential search paths laid out in advance in the Look For Applications In and Look For Documents In cards). You would have to remember precisely how the names of your applications appear on disk and be able to reproduce them, arcane symbols and all, when you want to switch. You have to do a lot of typing and invoke commands in a very "un-Mac-like" way. In addition, anyone not familiar with *HyperCard* would have no idea how to use this method; even relatively complete instructions on your part are not likely to help a computer novice very much.

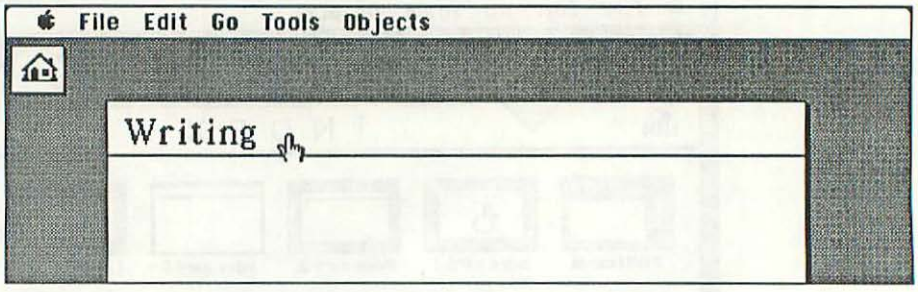
A more sensible way to use *HyperCard* is to create a separate stack containing buttons that switch to different applications and documents. Create the *HyperCard* version of an *Oasis* or *PowerStation* screen, in other words. For those who have yet to master *HyperCard*, the following is a brief set of instructions to get you started. They show you how to create a new stack containing a single button that switches to the application of your choice; after that, you're on your own.

1. Launch *HyperCard.*
   a. Go to the Home Card if you are not already there.
   b. Click the left arrow button near the bottom of the screen to go to the User Preferences card. Click on the Scripting radio button.
   c. Click on the right arrow button to return to the Home card.
2. To select a background for your new stack:
   a. Click on the Stack Ideas stack to open it.
   b. Look through the available graphics and select (click on) one that seems appropriate. For example, you might select Index Card 1.



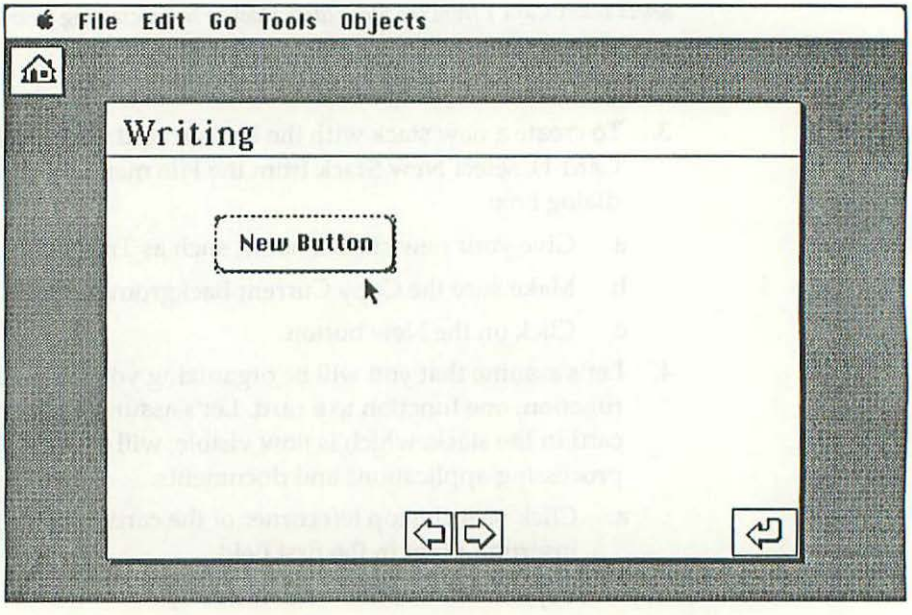
Select Index Card 1 from the Stack Ideas stack to begin creating your Transfer stack.

3. To create a new stack with the background now in view (Index Card 1), select **New Stack** from the File menu. In the New Stack dialog box:
   a. Give your new stack a name, such as Transfer.
   b. Make sure the Copy Current Background option is turned on.
   c. Click on the New button.
4. Let's assume that you will be organizing your transfer buttons by function, one function to a card. Let's assume further that the first card in the stack, which is now visible, will contain all your word-processing applications and documents.
   a. Click near the top left corner of the card to place the text insertion point in the first field.
   b. Type the title of the card: `Writing`.



Label the first card in the stack `Writing`.

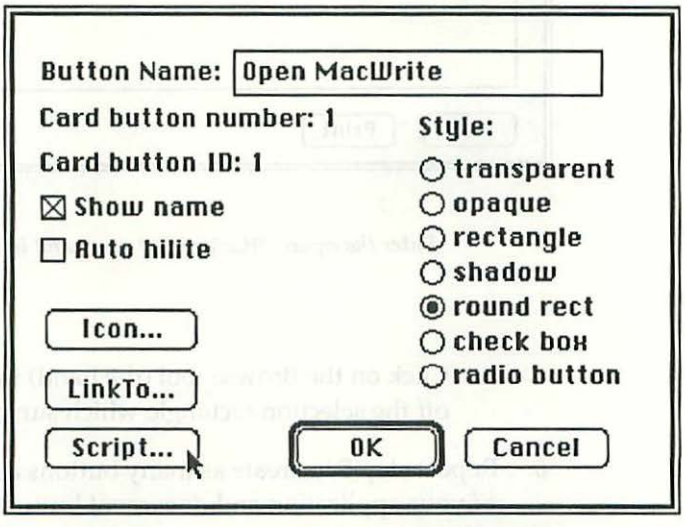
5. Now let's create a transfer button:
   a. Select **New Button** from the Objects menu. A "default button" should appear near the center of the card. Use the pointer to position and resize the button however you wish.



Use the pointer to position and resize the new button.

b. To give the new button a name and a function, double-click on the button itself to bring up the Button Info dialog box.

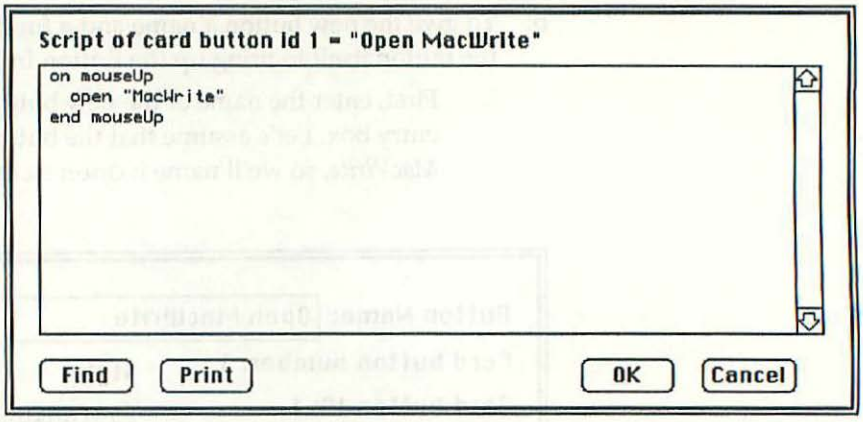
   i. First, enter the name of the new button in the Name text entry box. Let's assume that the button will switch to *MacWrite*, so we'll name it `Open MacWrite`.



Enter the name of the button, then click on the Script button.

   ii. Now Click on the Script button to give the button a function. After the Script window opens, type `open "MacWrite"` (the rest of the necessary language is already entered for you). Don't forget, the name of the application must exactly match the name as it appears on your disk. If the name of your working copy of *MacWrite* is *My MacWrite*, or *MacWrite Special*, or *MacWrite 4.5*, be sure that you enter that name, not just *MacWrite*. Note also that to open a document as well as an application, you would enter some variation on `open "WriteDocument" with "MacWrite"`.

   iii. Click on the OK button.



Enter the `open "MacWrite"` *command in the Script window.*

c. Click on the Browse tool (the hand) in the Tools menu to turn off the selection rectangle which surrounds the button.

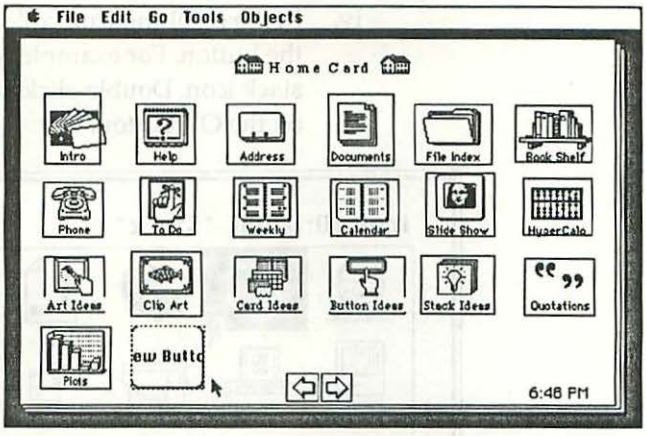
6. Repeat step 5 to create as many buttons as you like. If the number of your application and document buttons or your organization scheme requires more than one card (and it most likely will), add cards as needed with the **New Card** command in the Edit menu.

7. Now you need to put a button on your Home Card which will send you to your new Transfer stack:

   a. Return to the Home Card (the first card in the Home Stack). You can get there by selecting **Home** from the Go menu, pressing Command-H, or clicking on the house icon in the upper left corner of the screen (if you used Index Card 1 as a template).

   b. Select **New Button** from the Objects menu.

   c. Resize and move the new button as required.



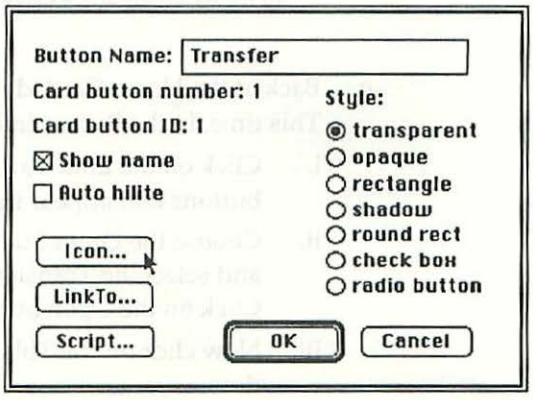
Use the pointer to position and resize the new button on the Home Card.

d. Double-click on the new button. This should open the Button Info dialog box:

   i. Enter a name for the button, such as `Transfer`.

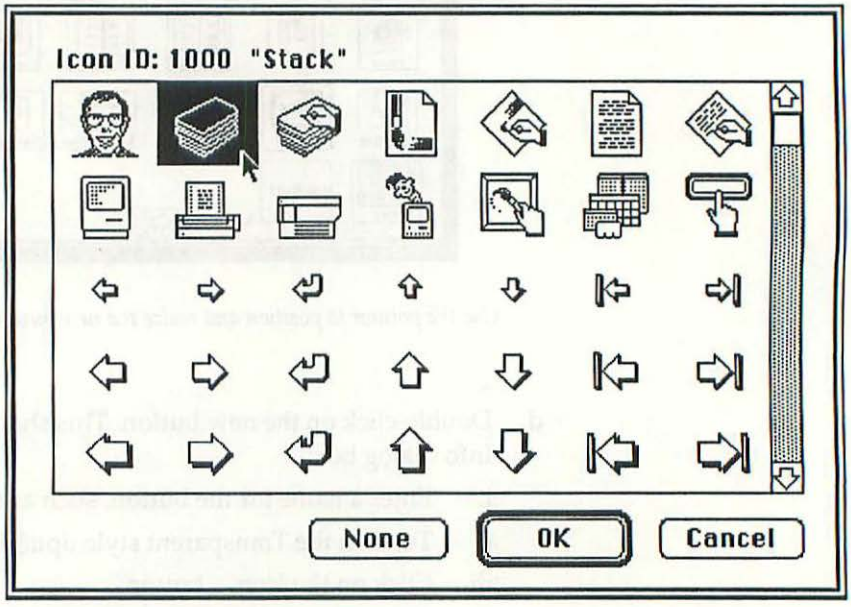
   ii. Turn on the Transparent style option.

   iii. Click on the Icon... button.



In the Button Info dialog box, enter the button name, select the ***Transparent*** *style option, and click on the Icon... button.*

iv. In the resulting icon collection window, select an icon for the button. For example, you might select the generic stack icon. Double-click on the icon or select it and click on the OK button.



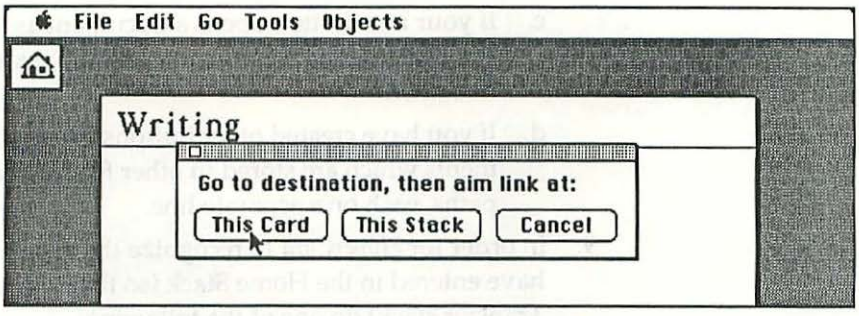
In the icon collection window, select the generic stack icon.

e. Back at the Home Card, double-click on the new button again. This time, in the Button Info dialog box:

i. Click on the Link To... button. Do not click on any of the buttons that appear in the resulting Link To window yet.

ii. Choose the **Open Stack...** command from the File menu and select the Transfer stack in the file selection box. Click on the Open button.

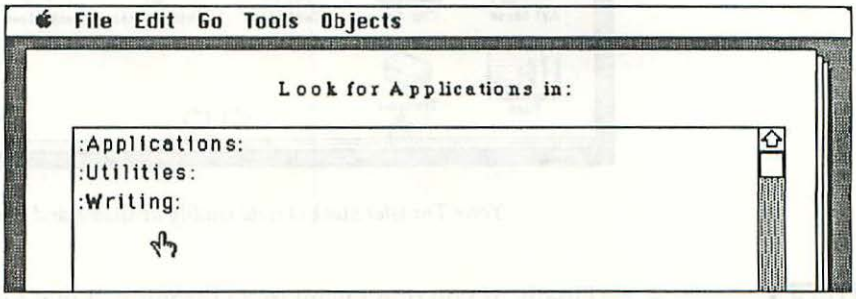
iii. Now click on the This Card button in the Link To window.



Click on the This Card button in the Link To window.

iv. Click on the Browse tool (the hand) in the Tools menu to turn off the selection rectangle which surrounds the button.

8. If the application your transfer button switches to is stored in a folder:
   a. Click through the cards in the Home stack until you come to the one labeled Look For Applications In.
   b. Enter the search path which will lead *HyperCard* to MacWrite. For example, if your copy of *MacWrite* is in a folder named Writing, then you would type in `:Writing:`. (As I mentioned before, for more on search paths, turn to *The Complete HyperCard Handbook.*)



Enter the search path to MacWrite *in the Look For Applications In card in the Home Stack.*

c. If your new button opens a document as well, go to the Look For Documents In card and enter the search path to the document.

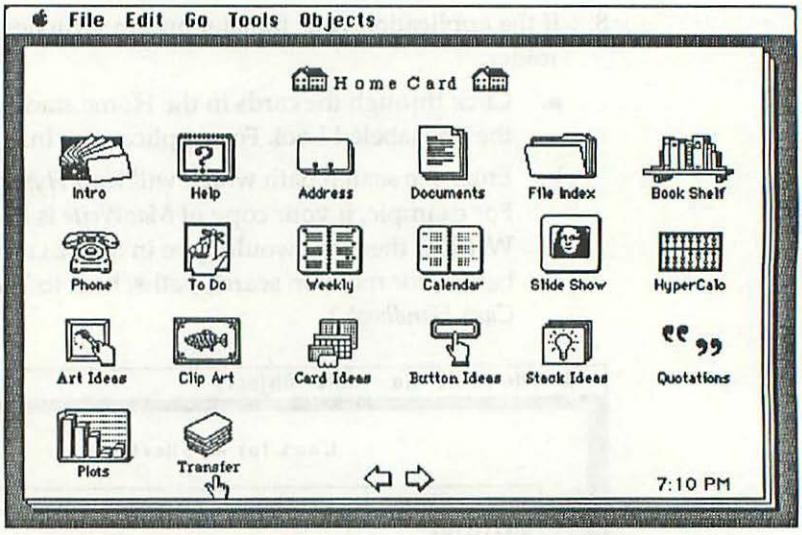
d. If you have created other buttons for applications or documents which are stored in other folders, enter their search paths, each on a separate line.

9. In order for *HyperCard* to recognize the search path (or paths) you have entered in the Home Stack (so that you can use your new Transfer stack) do one of the following:

   Quit *HyperCard* and then launch it again, or

   Select the Message command from the Go menu or press Command-M. Type `startup` in the Message Box and press Return or Enter.

Your Transfer stack is now ready to roll.



Your Transfer stack is now readily available and ready to use.

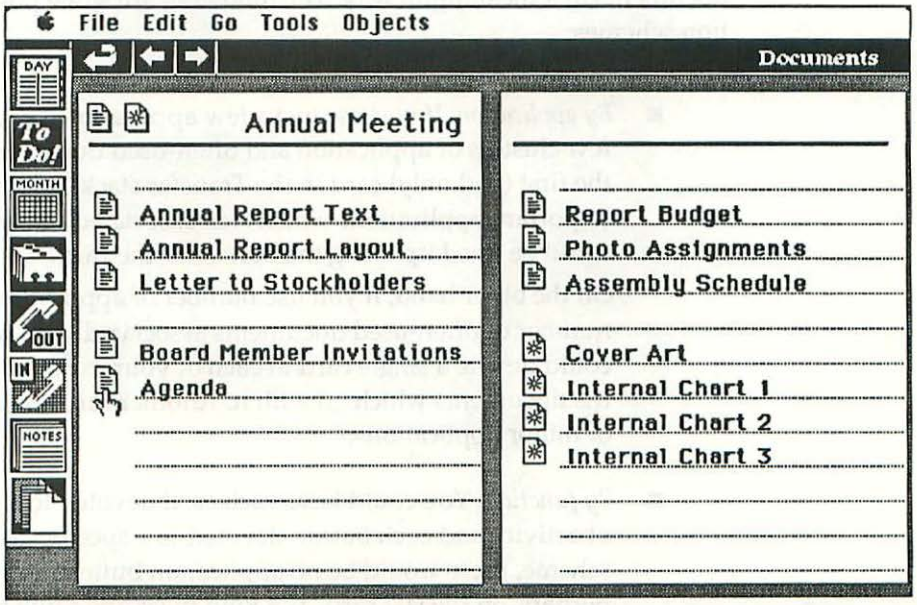
Demo

Finally, if you don't want to go through all that work, you can obtain stackware that comes ready to run, such as *Focal Point* (Activision). *Focal Point* is an elegantly designed set of stacks which place a variety of re-

sources at your fingertips: a daily appointments calendar, a monthly calendar, a "to do" list, a phone directory and dialer, incoming and outgoing phone logs, a free-form note pad, a variety of business-related, record-keeping modules (client records, time sheets, expense records, deadlines, and so on), and a Document Launcher.

Actually, *Focal Point*'s Document Launcher comes not quite ready to run. This is because the Document Launcher module has no way of knowing what applications you work with or what documents you might want to make accessible via *HyperCard* buttons. Thus you must set up the Document Launcher with the appropriate application and document buttons. Fortunately, *Focal Point* leads you through the button assignment process step by step, so that the process is quite simple and there is little chance for error.

Once you have set up *Focal Point*'s many modules with its Installation stack and set up the buttons in the Document Launcher, you have about as sophisticated a Finder substitute and home base for all your Macintosh work as you can get. The only thing you might want to add to *Focal Point* is



Focal Point *offers, among many other things, an easy-to-use application and document launcher (screen shot courtesy of Danny Goodman).*

a housekeeping desk accessory such as *DiskTop* or *DiskTools II*, unless you prefer to switch to the Finder when you need to copy, move, or delete a file, create a folder, and so on.

There is, by the way, a simple trick you can use to go to the Finder when you quit an application, instead of returning to your *HyperCard* Finder-substitute stack. The trick is to hold down the Option key when you quit your application and to continue to hold it down until the Finder appears. You can now go about your business in the Finder as usual. When your work there is finished, you can double-click on *HyperCard*, and you'll be returned to your Transfer stack (not the Home Card). Or you can go directly to another application by launching it in the usual way. The only drawback to this technique is that pressing the Option key closes all the windows you keep open in the Finder. Thus when you get to the Finder, you'll have to reopen all your now closed windows to do your housekeeping.

Whether you create your own document launching stack, or work with the one in *Focal Point* or some similar piece of stackware, it is, as always, important to organize the buttons that switch to applications and documents in a meaningful way. The following are some possible organization schemes:

- *By application.* If you use just a few applications, you could create a few clusters of application and often-used document buttons on the first (and only) card in the Transfer stack. Buttons for less important applications which lack associated document buttons could be lined up along the bottom of the card.

  On the other hand, if you use number of applications and have a number of often-used documents associated with each one, you could devote a single card to each of your major applications and the documents which go with it. Another card could be a grab bag of minor applications.

- *By function.* You could have each card devoted to a particular type of activity and each button devoted to a specific task. With this scheme, there would be no application buttons per se (except perhaps on the last card, as a kind of escape route for those times when you can't find the task you are looking for).

For example, you could set up a card for information storage and retrieval. Some of the buttons, such as those labeled Phone Book and Miscellaneous Notes, might refer to other *HyperCard* stacks. Others, such as client lists and inventory records, might lead to documents created with a database program such as *Omnis 3*. Still others, such as financial records, might open *Excel* documents. As you can see, the important point is not what application is used to keep the records but the nature of the information itself. Once your buttons are properly configured, you need never think about which program is used to store what kind of information. You just click on the information you want and there it is.

- *By user.* If there are a number of people who use your Macintosh, you could set up a card for each user, tailoring the button choices to their specific needs by function, by application, by document, or whatever makes sense for each person.

Ample-Memory Integration

Although *SkipFinder, SafeLaunch, Oasis, PowerStation,* and *HyperCard* all have their charms, the most convenient form of integration is to have all your program modules loaded in memory at the same time. Then the switch from program to program takes but an instant. Your work in each program is exactly where you left it a minute or an hour ago, since there is no need to close documents before switching. Because switching is so fast and seamless, you can shift from one application to another and back again on a whim. This is what *Switcher*, MultiFinder, and *Servant* are all about.

The price you pay for this form of integration is the cost of the memory needed to accommodate two or three or more programs all at the same time. Two or 2.5 megabytes of RAM may well suffice, and 4 (or more) megabytes should be enough for all but the most demanding combination of applications and RAM disk or RAM cache (of which I have more to say in Chapter 7). Attempting to run such an integrated setup on 1 megabyte of RAM, on the other hand, is the computer equivalent of trying to squeeze into Cinderella's shoe: as much as you would like to think your foot is that dainty, it's really quite hopeless.

So the first thing you need to ascertain before attempting one of the following strategies is whether or not your Mac has sufficient RAM for what

you want to do. If it doesn't have enough RAM, the results may be more of a hassle than anything else.

Experimentation is really the only way to tell how much RAM is enough in your particular case. Even if your Mac has a limited amount of RAM, it makes sense to at least experiment with one of the setups described in the following sections. But keep in mind that the limited-memory strategies in the previous section are probably more useful and more efficient in the long run when RAM is limited. If you keep running into snags as you attempt to integrate your megaprogram modules, then back up, try *SkipFinder* or *Oasis*, and see if they don't do the job better. If your experiments tell you that you don't have enough RAM, then it's time to try another integration scheme—or buy more RAM.

If you do have enough RAM, then, with one of the three following programs, you'll be able to keep several programs in memory at the same time. By this I do not mean that the programs actually run concurrently (although with MultiFinder, limited multitasking, as it is called, is possible). Rather, RAM is divided into smaller "virtual computers," and a single program resides in each one of these subdivisions. The program whose main window is foremost on the screen is the one in charge, the one which is "running" at that time. All the others are essentially dormant, waiting for their turn to take control and respond to orders.

As you may have gathered from your reading in the computer press, there are those who have elevated multitasking (the idea of having two or three programs running at the same time) to the status of computerdom's Holy Grail. In terms of the fully powered Macintosh, however, this capacity is of minimal use, and unless it is handed to us on a plate, it is hardly worth the trouble for most of us. Virtually all the work I do on the computer happens as I press the keys. To have a program work "in the background" makes little sense for me, since my interaction is an integral part of a program's operation. The same is true, I suspect, for most people.

Printing and telecommunications are two of the few exceptions to this generalization, however. Certainly, we all would like to have printing and downloading or uploading files handled as background operations while we go back to work; but true multitasking is not required to put either one of them out of sight and out of mind. For printing, print spoolers have been around for a while and remain a good way to make printing a background task (for more on print spoolers, see Chapter 7). One of the better ways to avoid wasting time uploading and downloading files is to use a telecommu-

nications program that can do the job for you while you sleep. Not only is the work done when you're not around, the phone rates are cheaper then too.

In any event, while the integration we are striving for here is extremely useful, it is not multi-tasking. Although the basic idea of putting multiple programs in memory so all are available instantly is behind each of the following three programs, in many other respects the three programs are quite different. As with the Finder substitutes and direct-transfer utilities previously described, which one you choose to work with is as much a personal decision as anything else.

Using MultiFinder At first glance, using MultiFinder doesn't seem all that much different from using the Finder in its normal, non-MultiFinder mode. Indeed, there's very little that you need to know to use MultiFinder that you don't already know from using Finder, except that you don't have to close one application before opening another one. It sounds so mundane, yet what a change from the old days.

That's not all; MultiFinder will allow certain kinds of tasks to continue "in the background" while you work at something else "in the foreground," such as printing with the LaserWriter using the Print Monitor program which accompanies MultiFinder. (See Chapter 7 for more on print spooling, as this feature is called.) Applications, as they are upgraded or developed, will permit other tasks to occur in the background as well, such as telecommunications sessions, making hard disk backups, and performing long and complex calculations.

The basics of using MultiFinder are covered in usual Apple fashion in the MultiFinder manual, and the print spooler aspect of the program is discussed in the next chapter. (If you don't have a MultiFinder manual, you can get one as part of the *Macintosh System Software Update* package from your Apple dealer.) Besides, since working with MultiFinder really is so much like working with the plain vanilla Finder, what little there is to explain you probably already know. Thus I won't repeat MultiFinder basics here.

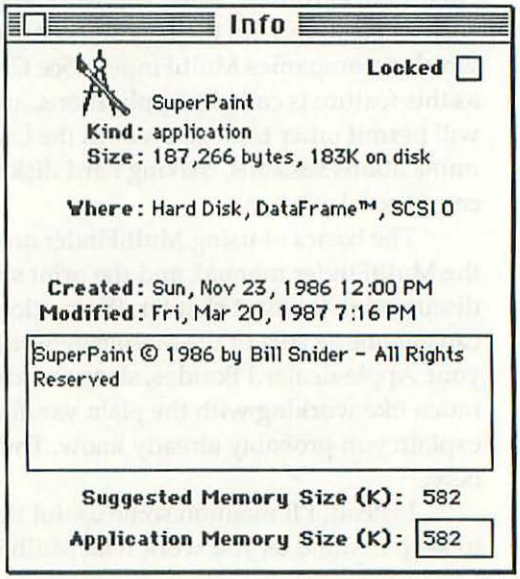
Instead, I'll mention some useful tips and techniques you might want to keep in mind as you work with MultiFinder.

- Although it is almost impossible to run MultiFinder and two reasonably sized applications within one megabyte of RAM,

MultiFinder still can be useful under such memory constraints. Even if you run just one application at a time with MultiFinder, the fact that you can switch between the Finder and your application at any time and that the Finder's functions are always available makes the Finder much more helpful and convenient. Because the Finder is always present in the background, quitting from an application takes but a fraction of the time it would with MultiFinder turned off.

- If you want to squeeze an application into a limited RAM partition by changing the Application Memory Size (in the file's Get Info window) to something smaller than the original value, you could choose the Suggested Memory Size and leave it at that. But often the Suggested Memory Size is too small. So how big is big enough, yet still small enough?

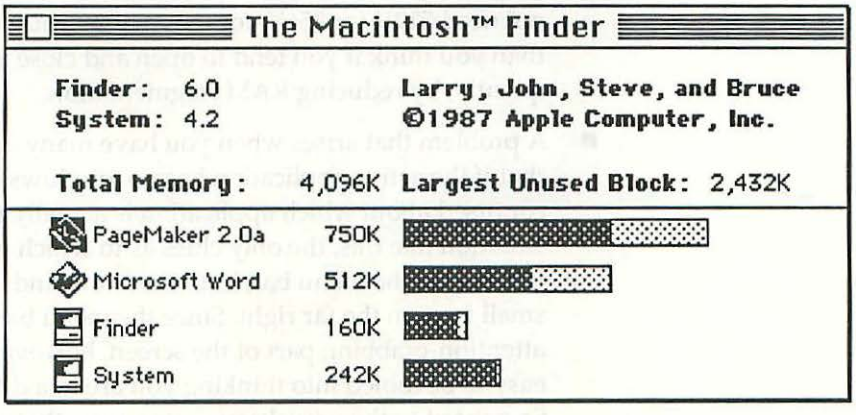
  One answer to this question is to experiment with a number of trial memory allocations. As you request smaller and smaller



MultiFinder memory allocations are set in each application's Get Info window.

amounts of memory (but never smaller than the Suggested Memory Size), your application may at first become somewhat sluggish or balk at opening large documents. When it refuses to launch (not likely, if you stay above the Suggested Memory Size) or crashes when you try to do something that requires extra memory, such as copying a large picture or piece of text to the clipboard (more likely), you know you've gone too far. Obviously, an allocation that causes a program to crash is unacceptably small. The point at which performance degradation becomes unacceptable is something you'll have to decide for yourself.

- On the opposite side of the coin is how much memory is too much? You may find that you need to assign more memory to accommodate an especially large document. Or it is not uncommon to discover that an application performs better with more memory than the amount set back at the factory. To find out at what point are you wasting RAM, use the memory graph in the About Finder... window. The gray bar after each application name represents the total amount of memory in that application's memory partition. The dark gray segment is what is currently in use. The light gray segment (if there is any) is what is not being used.



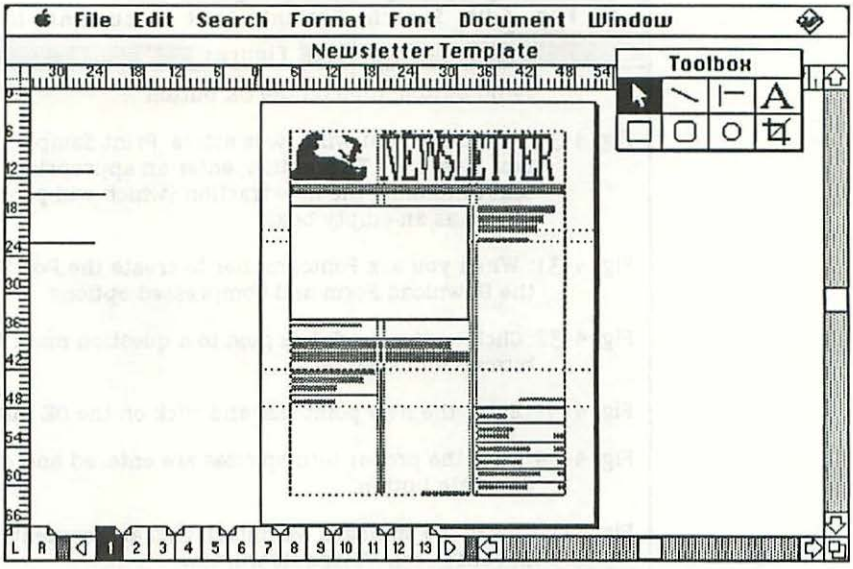
Use the About Finder... window to monitor memory usage.

If you check this window frequently and the memory bar for the application in question consistently shows some (or a lot of) light gray, then its memory allocation is too large. Reduce the allocation little by little (prior to subsequent uses of the application) until only a small wedge of light gray remains during an average work session.

- One problem with opening and closing many applications with different memory allocations under MultiFinder is that it causes RAM "fragmentation": bits and pieces of free RAM become wedged between application partitions (much as in a fragmented disk, a problem described in Chapter 8), all of them too small to be of use, since an application must reside in a single, contiguous chunk of RAM. Apple's suggested solution to the problem, periodically quitting all your applications and restarting them again, is effective but tedious.

  Another way around the problem is to allocate the same amount of RAM to all your major applications. For example, you might consider allocating 512K to each of your applications (some memory hogs may require even more). Then, when you quit one and open another, the second is (just) able to use the empty space left behind by the first. If some of your applications need a lot less than 512K, you might assign them 256K, so two can fit where one once was. This strategy may waste RAM for programs which do not need 512K (or 256K, for that matter), but it may waste less than you think if you tend to open and close applications frequently, by reducing RAM fragmentation.

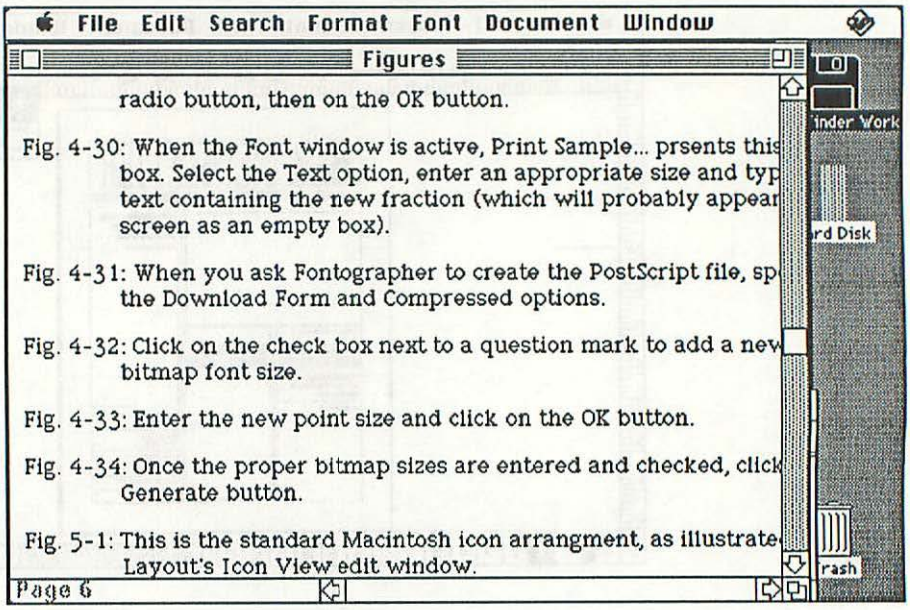
- A problem that arises when you have many applications open is that if the active application has no windows open, you can get confused about which application is actually in control. In a situation like this, the only clues as to which is the active application are on the menu bar, both the commands themselves and the small icon on the far right. Since the menu bar is often the least attention-grabbing part of the screen, however, it is remarkably easy to be fooled into thinking you are in a different application. So a word to the wise: keep your eye on that menu bar.



Surprise! This isn't PageMaker, *it's Microsoft* Word *in MultiFinder with no open windows and* PageMaker *idling in the background.*

- Most document windows fill the whole screen and cover the disk icons along the right side of the display. When you select the Finder, the windows from your other applications still cover the disk icons, making it impossible to copy a file to a disk, eject a disk by dragging it to the trash, and so on. If you have a lot of documents open, resizing them all to gain access to the disk icons can be a real pain.

  To get around this problem, resize all your windows as soon as you open them so that they leave the row of disk icons uncovered. When you want to work in a window, click on the Zoom box (or make a macro that does the same thing) to make it full size. When you are ready to switch to another window, click the Zoom box again to uncover the disk row.



Resize your current document window before you switch to another window; only in this way will you be able to see your disk icons when you switch to the Finder.

- If you have a large screen and you manage your windows carefully, you can keep the documents from several applications in view all at the same time. Thus, to take but one example, you could refer to an open spreadsheet document while writing a summary report in your word processor. This isn't quite as slick as *Modern Jazz*'s Hotview function, but it's better than having to flip through a bunch of windows every time you want to check a number.

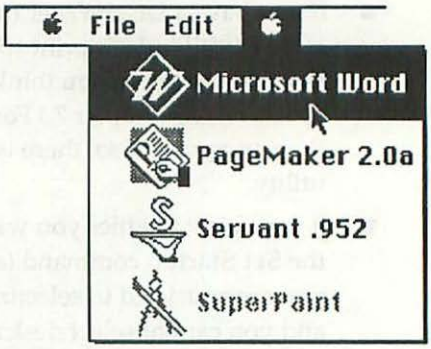
  A corollary to this technique is that you should arrange your windows carefully on a big screen to avoid the cluttered (and confusing) look that MultiFinder is so prone to. For example, if you have a two-page display and four programs running, you could segregate the tool palettes and windows of each program into the four quadrants of the screen. Expand the window of the active program to cover one or two of the quadrants not currently active, then zoom the window back down to quarter-screen size just before switching to another application.

- If you have a LaserWriter (or other PostScript printer), you should see if MultiFinder's print spooler, Print Monitor, is adequate for your needs before you think about buying one. (For more on print spoolers, see Chapter 7.) For many, Print Monitor should be more than adequate; if so, there is no need for you to buy a print spooler utility.
- If you select the files you want opened at startup before choosing the **Set Startup** command (and using the Selected Items option), you are restricted to selecting files which are all in the same folder, and you cannot select desk accessories. If you want to open applications in different folders or open both applications and desk accessories, launch the desired applications and desk accessories before you choose the **Set Startup** command, and use the Open Applications and DAs option. Unfortunately, with this latter method, you cannot select documents.
- Remember; if one program crashes, your entire set of open applications may be dragged down with it. So save your documents frequently. Indeed, it is a good rule of thumb to save your documents every time you switch to a different application.

Using *Servant* On the face of it, *Servant* (public-exchange) looks and acts a lot like MultiFinder. In fact, in terms of their ability to keep several applications in memory at once, *Servant* and MultiFinder are like two peas in a pod. Both programs allow you to keep open as many applications as memory permits. You can switch to any open application to make it active by clicking on a window or selecting the application from a menu. The desktop functions of both programs retire to the background whenever an application is launched, but they are always available if you need them. The amount of memory used by an application in both programs is set in the application's Get Info box.

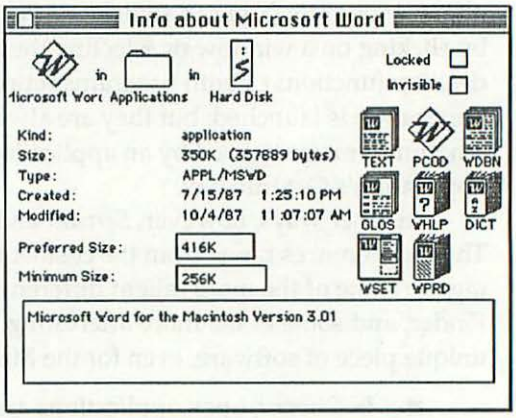
In other ways, however, *Servant* and MultiFinder are quite different. These differences range from the cosmetic to the fundamental. The following are some of the more salient differences between *Servant* and MultiFinder, and some of the more interesting features that make *Servant* such a unique piece of software, even for the Macintosh:

- In *Servant*, open applications are listed in a menu that is added to the end of every menu bar. (The equivalent menu in MultiFinder is appended to the bottom of the Apple menu.)



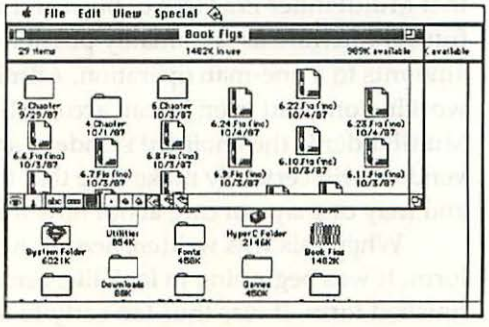
Applications open under Servant *are available from the* Servant *menu.*

- *Servant* has an "emergency eject" keyboard command: Command-Shift-Option-period. Should the active application crash, pressing this key combination will, if you are lucky, return you to the desktop and allow you to save the documents in your other open applications.
- You change the Application Memory Size (in *Servant*, it's called the Preferred Size) for an application the same way you do in MultiFinder. There is, however, a lot more to a *Servant* Get Info box than a MultiFinder Get Info box.



Use an application's Get Info box to set memory allocation, and to find out all sorts of other things.

- You can launch several applications all at the same time by selecting them and then choosing the **Open** command in the File menu.
- The multiapplication feature of *Servant* may be turned on and off at will during a work session. Thus you can devote all of the computer's memory to a single application at any time. You do not need to restart the computer to put your change into effect, as you do with MultiFinder.
- *Servant* windows do not scroll in the usual fashion. Instead, you use a hand to push around the "page" which lies within the window, much as you would a document in *MacPaint*. If you click on the eye icon at the bottom of the *Servant* window, *Servant* will display a "Show Page" view of all the icons in the window.
- *Servant* windows are always in icon view. Icons can be larger or smaller (there are six possible icon sizes) to accommodate greater or smaller numbers of icons, or poorer or better eyesight. The text below each icon can display either the file's (or folder's) name alone, or its name and size, or name and creation date.
- The size of a folder is displayed in the information bar of that folder's open window.
- Application, folder, and disk icons can be customized extensively in *Servant*. You simply select the icon, choose the **Custom Icon** command from the Special menu, and rework the fatbits version of the existing icon. You can also take an existing icon (such as an application icon), shrink it, and paste it onto the front of a folder icon.



Servant *windows are in icon view only, but they can show file and folder size, and date last modified beneath each file and folder name.*

- A file in a window can be selected from the keyboard by using the Tab key to select the next icon moving from left to right, from top to bottom.
- A *MacPaint* or *ThunderScan* image can be used as the background pattern simply by selecting the file containing the desired image and choosing the *Set Background* command from the Special menu.
- If MacinTalk (the Apple speech synthesizer file) is in the System folder, *Servant* can speak the name of a file when it is selected and the name of a command when it is chosen.
- *Servant* allows you to view the building blocks of an application—its resources—and move them, copy them, or duplicate them as if they were individual files in a folder. In general, this feature should only be used for casual, hands-off browsing unless you know what you are doing.
- Desk accessory icons can be moved onto the desktop and subsequently launched just like an application.

These are only the most significant differences between *Servant* and MultiFinder (and even more were slated to appear at *Servant*'s official release—if there is one). It would takes pages to describe all the other things that make *Servant* so distinctive, features which are often elegant, but just as often more a matter of appearances than function.

Appealing as these features are, it is important to remember that as a complete replacement for Finder or MultiFinder, *Servant* has some very big shoes to fill. Apple can commit whatever resources are necessary to ensure that MultiFinder is as free of bugs and as compatible with existing (and future) software as is humanly possible, but *Servant* is the product of what amounts to a one-man operation. Admittedly, a very smart and hard-working one, but even he can accomplish only so much in a day. Since MultiFinder is the anointed standard and *Servant* the upstart kid, software vendors will certainly make sure that their products run with MultiFinder and may or may not care about how they work with *Servant*.

When this was written, *Servant* had not yet appeared in final, finished form. It was beginning to look like *Servant* would never appear in final, finished form. It was thus too early to tell how well *Servant* would stand up against MultiFinder—if it ever does. But a good guess is that although

MultiFinder may have a few compatibility problems, *Servant* will probably have a great many more. *Servant*'s compatibility problems are also more likely to persist. If your current print spooler does not work with MultiFinder, for example, either a new upgrade soon will, or you can use MultiFinder's own spooler (if you have a LaserWriter). If your print spooler does not work with *Servant*, however, you'll probably have to give up on either *Servant* or the spooler.

Servant offers the power of MultiFinder and the customization abilities of a variety of utilities in a single package. It is the epitome of a "neat" program, and although "neatness" counts, it isn't everything. Regardless of whether or not a final version of the program eventually appears, it is unlikely that *Servant* will ever become even a second standard in Macintosh operating systems, a real alternative to MultiFinder. If its features, its "neatness," appeals to you, by all means try it out. But be sure to test it carefully with all your applications, desk accessories, and so on, before committing to it. Also test it thoroughly with any new programs or new versions that you acquire later on.

For most of us, however, I suspect MultiFinder will remain the creative force behind our desktop.

Using *Switcher* *Switcher* was Apple's first attempt at the idea that eventually became MultiFinder. In its day, some people loved *Switcher* and some people hated it. Now that we have MultiFinder, many people in both camps have abandoned it. Yet, although *Switcher* has been set aside by Apple and is now slowly fading from view, *Switcher* does have a few good points, even in the MultiFinder era.

Switcher has been around for a long time and been through many versions. It had a hard time keeping up with the many applications it was used with, and many applications had a hard time keeping up with *Switcher*. As a result, some people never managed to find a *Switcher* version that was completely compatible with their collection of applications; these are the people who hated *Switcher*. Others were luckier, and found *Switcher* a reliable, almost miraculous way to integrate a group of programs; these, of course, are the people who loved *Switcher*. I have always, somehow, been a member of the latter group, which is perhaps why I think *Switcher* still has some life in it.

When it first came out, *Switcher* used to wow everyone who saw it. Click on the switching arrow and a new program literally rolled into view. It looked as though program screens were pasted on a drum, and the drum

rotated either right or left with each click on the arrow. Although the animation is appealing (you can turn it off, by the way), what's important about *Switcher* is not that applications roll around on the screen. What's important about *Switcher* is how sensible a solution it is to program integration: *Switcher* (like MultiFinder and *Servant*) allows you to add and subtract program modules as you work. *Switcher* allows you to decide how much memory to give to each program you launch, at the moment you launch it. *Switcher* also provides you with a way of creating any number of preconfigured megaprograms (called *Switcher* sets), tailored to specific tasks, which can be launched with a double-click of the mouse button.

Unfortunately, with the coming of MultiFinder, it appears that *Switcher* has begun to die slowly and quietly of neglect. This is unfortunate, for in some ways I prefer *Switcher* to MultiFinder. What do I like so much about *Switcher*?

- It doesn't require nearly as much memory as MultiFinder. Therefore, it is much easier to squeeze two applications (possibly even three) into one megabyte of RAM.
- You can create *Switcher* sets, and each set can be comprised of an entire integrated megaprogram. Simply double-click on a *Switcher* set, and the entire collection of applications is opened automatically.
- Each program or module in *Switcher* controls the entire screen, and the windows of other programs do not appear in the background. As a result, I find the *Switcher* screen far less confusing to work with than MultiFinder's.
- Because there are no extraneous windows in the background, a misplaced mouse click during a moment of inattention cannot put you in a different program. Nonetheless, switching from program to program is easy; you simply click on the arrow at the right end of the menu bar or press Command-[or Command-], depending on which direction in the program sequence you want to go.
- When you open an application, you can let *Switcher* set the memory allotted to it automatically, or you can set the memory allotment yourself (should you foresee extra memory requirements

due to a large number of documents, for example). If you set the memory allotment manually, the change can be either temporary (for the current session only) or permanent (for all sessions).

- If you want to run a single application using all the computer's RAM, you can do so easily by quitting *Switcher* and then launching your application.
- Like *Servant*, *Switcher* has a keyboard emergency-eject command which can extricate you from potential disaster if one of your applications crashes—at least some of the time.

Despite these good points (and, to be fair, it must be pointed out that *Switcher* has some not-so-good points), *Switcher*'s days are numbered, since Apple has ceased updating it to meet the needs of new and different applications. As a result, *Switcher* will slowly but surely become less and less compatible with the programs we are all using. Thus, inevitably, *Switcher* will become less and less reliable (those who hate it will no doubt ask at this point if it ever was), less and less usable, and harder and harder to find at your local software emporium.

But until these problems become overwhelming (and they haven't for me—yet), some people may continue to find *Switcher* a viable alternative to MultiFinder. I still use it on occasion, partly from nostalgia, partly because I still have a number of *Switcher* sets already set up for some of the regular chores I perform. Although I certainly don't recommend that people who already own and use MultiFinder turn back the clock and hunt up a copy of *Switcher* for everyday use, those of you who have a copy from the old days may still find it useful, as I do.

I'm not going to cover *Switcher* basics here; you can get those in the (admittedly brief) *Switcher* manual or from the paraphrases that used to be found in many other application manuals (such as those from Microsoft). What I will cover, however, is two ways you can use *Switcher* to run your megaprogram, and some useful *Switcher* tips.

Open-ended *Switcher* Sets One way to use *Switcher* to advantage is to employ what I call the open-ended technique. This method is especially useful for those with relatively limited RAM. It's also a good technique for those who use one or two programs most of the time and a large number of other programs only occasionally, often on the spur of the moment.

With the open-ended method, you start your day with *Switcher* loaded with the one or two programs you use most frequently. You can do this either manually or by means of a *Switcher* set. (This set could be launched automatically at startup, by using the technique described in the *Switcher* tips section that follows or by using the automatic startup techniques described in Chapter 7.) For example, until MultiFinder came along, I began most mornings by starting *Switcher* with a *Switcher* set containing only Microsoft *Word*.

For as long as the initially loaded applications are all you need, you're all set. When you need to work in a different secondary program, you simply return to the *Switcher* window and load that program in the next vacant *Switcher* slot. Use it as long as you need to, switching to your other programs as required. When you are done with it, quit the secondary program. Go back to work in your major applications until you need to work in some other secondary program and repeat the process. In this way, you can work in all manner of programs during the course of a day, yet never have to quit your main applications.

Consider the following example:

1. You began the day by launching *Word* in *Switcher* and then started writing a report. An illustration was sent to you by electronic mail, and you need it for the report. So after writing for a while, you decide to take a break and use *Desktop Express* to download the illustration.
2. You return to *Switcher*'s main window and launch *Desktop Express.*
3. You click on the switching arrow in the menu bar and *Desktop Express* appears. You download the illustration and then quit *Desktop Express* in the normal way. You are now back at *Switcher*'s main window.
4. You want to do some file reorganization—delete a few files to free up disk space, move the illustration document to the report folder—before doing any further work. So you open *DiskTop*, a file utility desk accessory I've mentioned before, and take care of these matters in the *DiskTop* window.
5. Since you want to put some finishing touches on the illustration, you decide to launch *SuperPaint* in the *Switcher* slot which had just held *Desktop Express*. *SuperPaint* appears and you go to work on the illustration.

6. When you finish your touch-up job on the illustration, you select it with the marquee and copy it to the clipboard. Then you hold down the Option key (since you have the Always Convert Clipboard option turned off for speedier switches; this is necessary to carry *SuperPaint's* clipboard over to the next application) and click on the switching arrow.
7. *Word* reappears just as you left it a while ago. The text insertion bar is already where the illustration should go, so you simply select **Paste** to insert it.

If this example sounds familiar, it should. It is the one we've used several times already to illustrate how other integration utilities work and how they differ. You might want to compare the scenario described here with the ones earlier in this chapter to see just how different using *Switcher* (or MultiFinder, or *Servant*, for that matter) is. Notice that to perform housekeeping functions in *Switcher*, I used the desk accessory *DiskTop*. Although it is possible to launch the Finder as a *Switcher* application, I've always found doing so a bit risky, and I prefer using a desk accessory such as *DiskTop*.

Preconfigured *Switcher* Sets Another useful *Switcher* strategy is to create a *Switcher* set for each useful combination of applications. Then at the start of a given task, all you need to do to launch all the necessary applications is double-click on the appropriate *Switcher* set.

The more *Switcher* sets you have, the more likely it is you will be able to start work with no more than a double-click. In the previous open-ended *Switcher* set illustration, the application launching and quitting described there would have been unnecessary had I already had a *Word/SuperPaint/Desktop Express Switcher* set. Of course, there is a point at which *Switcher* sets start to run your life, and overwhelm the other files on your disks. You need to strike a balance between too few and too many *Switcher* sets. Remember, as long as you have enough RAM, every *Switcher* set has the potential of being an open-ended set. Thus you need not make a set for every conceivable task and purpose; a broad-brush approach toward *Switcher* sets is often the best one.

For example, I still keep a half-dozen or so *Switcher* set documents in my main Application window. Because writing is my main occupation, most are based on my primary word processor, Microsoft *Word*. If I'm working on illustrations for a book, I can use my *Word/SuperPaint* set. If I'm

finishing a Macworld article in order to send it to the editorial office, I can open my *Word/AppleLink* set. Or if I'm spending the day or part of the day working on a newsletter, I can open my *PageMaker/Word/SuperPaint* set.

My *Switcher* set collection was, in the past, rather ameboid, expanding and contracting as different projects came and went, and the *Switcher* sets relating to them appeared and disappeared. Now many of these sets are being replaced by the ease with which MultiFinder allows me to slip into and out of almost any application. As time goes on, MultiFinder moves in on former *Switcher* territory, and my *Switcher* set collection is clearly expanding no more. But if you like the prefab nature of *Switcher* sets or if you perform the same routine operations day in and day out, then you may want to hold on to your *Switcher* sets as long as you can.

***Switcher* tips** The following are some tips for using *Switcher* in either of the two modes just described. They will likely apply as well to any other *Switcher* techniques you may have devised on your own.

- To achieve the fastest switching between applications, turn off all the options in the Options dialog box. Switching animation is sexy but unnecessary. Always converting the clipboard takes too much time; use the Option key when you need to transfer the Clipboard and otherwise, forego the convenience.

  Turning the **Save Screen** option on also makes switching faster, but it will cost you some RAM—about 22K per application. Turn it on either by clicking on the **Save Screen** check box in the Configuration dialog box or by clicking on the little Mac screen opposite the name of the application in *Switcher*'s main window so that the little international "No" symbol on the screen disappears.

- I almost never double-click (or use the **Install Application...** command, or press Command-I) to launch an application in Switcher. Instead I invariably use the **Configure Then Install...** command. This gives me the ability to set the program's memory allotment, using the Preferred Memory Size text entry box, according to the needs of the moment. To preserve the standard Preferred and Minimum Memory Size values (both of which are useful things to know), however, I am careful never to click on the Permanent button when I am ready to install the application.

Instead, I use the Temporary button or press Return to avoid changing the memory settings. Thus if I do double-click to launch that application later on, the standard Minimum and Preferred Memory Size values are still in effect and will be used by *Switcher*.

- In case I click on the Permanent button in the Configuration window by mistake, I keep a list of the standard Minimum and Preferred Memory Size values for all my applications. Whenever I first use an application with *Switcher*, I copy the values in the Configuration window onto my list. Of course, if these numbers are overwritten by mistake, you can determine the correct values by using *Switcher* to look at the backup copy of the application. Referring to a list, however, is much quicker and easier.

- If you use the **Configure Then Install...** command, how do you know how much memory to request? The answer to this question is basically the same as that for the classic "how do you get to Carnegie Hall?" Practice, practice, practice. If you request too little memory, your application may become agonizingly slow and deliberate, it may refuse to open large documents, or *Switcher* may not even be able to open it. You'll certainly recognize these symptoms when you see them.

  If you request too much memory, you've wasted RAM that another application could use. The way to check for an excessive memory allotment is to return to the *Switcher* window and select the **Show Info Window** command from the File menu. The gray bar after each application name represents the total amount of memory in that application's memory segment. The dark gray part is what is currently in use. The light gray part (if there is any) is what is not being used. Until you get a good sense of memory usage for each of your programs, check this window regularly (you can keep it open all the time and switch to it now and then) to see just how much memory your applications are actually occupying. Compare what you see in the Info window with the number and size of the documents you are working on. Eventually, you will get a feeling for how much memory each of your applications needs in order to work on documents of different sizes.

- Name your *Switcher* sets so you can easily identify what applications they load, and so you can find them easily in file selection boxes. If you start them all with `Set` (such as `Set.Word.SprPnt`, for example), they will all be grouped together in one place and segregated by the first application listed after the word `set`. If you start each one with the name of the major application (if the set has one), then these sets will group together by major application. Which one of these schemes, or some other scheme, makes most sense for you depends on how you work and where you store your *Switcher* sets.

- To launch *Switcher* and have it open a particular *Switcher* set at startup, set *Switcher* as the startup application in the Finder. Name the set you want it to open `Switcher.startup` and place the set on the Desktop (not in any folder). The next time you restart your Mac, *Switcher* will launch and open `Switcher.startup` automatically.

- You can install the Finder as an application in *Switcher*. (If you do, it must be the first application you install.) I, however, have found this a risky procedure, and do not recommend that you do so, no matter how careful you are. Instead, use a desk accessory file manager such as *DiskTop* or *DiskTools II*.

- When working with important documents, I make a habit of always saving my work just before switching to another application—just in case something unforeseen happens. But if you are running a group of applications under *Switcher*, have forgotten to save your files, and one of the applications crashes, all is not necessarily lost. Immediately after the crash, press Command-Option-Shift-period. Press it a number of times if nothing happens at first. With luck, your signal will get through and *Switcher* will return you to its main window. If you do succeed in returning to the main *Switcher* window, return to all the other open applications, save your documents and quit each one in turn, quit *Switcher*, and then clean your RAM of any stray stuff by restarting the Macintosh.

Finally, I should point out once again that those still using *Switcher* should take care to test it thoroughly as they acquire new applications and

new versions of old applications. Since it is no longer being kept up-to-date, *Switcher* is forever stuck in (the early part of) 1987. And as the months and years pass, *Switcher* will become less and less compatible with state-of-the-art Macintosh software (and hardware). Ready or not, *Switcher* fans will have to switch to MultiFinder sooner or later (probably sooner). So if you are a *Switcher* fan, there's nothing wrong with using *Switcher* if it still works well for you, but you should start the weaning process now, and use MultiFinder instead of *Switcher* whenever you are comfortable doing so.

Summary

- You can increase the speed, efficiency, and ease with which your Macintosh performs by turning your collection of favorite applications into what amounts to a homemade integrated program, a megaprogram. To do so, you need to standardize the keyboard shortcuts in your applications and use an integration scheme that fits both the RAM available in your Macintosh and the way you work.
- Before you become overwhelmed by keyboard shortcuts and macros, it is a good idea to develop a strategy by which certain broad kinds of commands and macros are always assigned to certain classes of key combinations, or function keys if you have an extended keyboard. For example, menu commands should always be assigned to Command-key or Command-Shift-key combinations.
- Be conservative in your attempts to unify the keyboard shortcuts in your applications. That said, try to establish at least the following few standards:
  - ◊ Except under very unusual circumstances, the most common dozen or so commands (mostly those found in the File and Edit menus) should all sport the same shortcuts in your applications: Command-P should always start the Print process, and so on.
  - ◊ Commands such as **Delete** should be assigned to one of a few shortcuts specifically reserved for such destructive commands. By doing so, you eliminate the possibility of unpleasant surprises.

- Shortcuts for standard menu commands can be assigned with *ResEdit*, *MenuMaster*, or *MenuKey*. On those rare occasions when shortcuts assigned with these programs fail to work, you can usually create functional shortcuts with programs such as *Tempo II* and *QuicKeys*.
- Use macros to automate almost any other task that you perform repeatedly. Such tasks can range from launching applications and clicking on a window's zoom box, to far more complex procedures. Since it is easy to forget how useful a program such as *QuicKeys* or *Tempo II* really is, take the time to review your work habits occasionally and analyze if there are ways that you can use them to perform what you currently do manually.
- Use *Ap Font* to change the default application font (the font first used by many applications in a new document) from Geneva to some other, more appropriate font. If you have multiple System disks, make sure that the font you choose is in the System file of every System disk.
- If your Macintosh has but one megabyte of memory, then you should consider using one of two techniques for speeding up the transfer from one application to another (and bypassing the Finder, except when you need to work with files):
  - ◊ File transfer utilities, such as *SkipFinder* and *SafeLaunch*, present a file selection box when you are ready to quit your current application. You select another program and the utility launches it immediately.
  - ◊ Finder substitutes such as *Oasis*, *SuperFinder*, and *PowerStation* can also speed up the transfer from one application to another. Their screens full of buttons also make it easier to locate and launch desired applications and documents, if you organize your buttons carefully. *HyperCard* (yes, *HyperCard*) can also serve as a way station between applications; *HyperCard*, of course, is also capable of quite a bit more.
  - ◊ Either *DiskTop*, *DiskTools II*, or *QuicKeys* can be used as a kind of cross between Finder substitute and transfer utility. With *DiskTop*, you can add up to twenty of the applications or documents you use most often to the transfer section of the

DiskTop menu. *DiskTools II* offers a customizable scrolling list of applications and documents from which to launch the next application. *QuicKeys* makes it possible to assign application and document launching to the key combinations of your choice.

- If your Macintosh has two or more megabytes of memory, then you will probably want to run your applications under MultiFinder. With it, you can keep several programs in memory at the same time and have the Finder available for routine file maintenance as well. It will also allow you to print LaserWriter documents in the background (while you work on something else), and if you have MultiFinder-compatible programs, you can download files over the phone and backup your hard disk, among other things, in the background. All of which makes for added speed and convenience.

- *Switcher* and *Servant* are two alternatives to MultiFinder. Although both have their advantages, neither are recommended for general use:

  - *Servant* has many unique features which allow you to create a working environment very much to your own taste. But because it is not an official standard, it is not likely that *Servant* will ever be as generally compatible and widely accepted as MultiFinder.

  - *Switcher* is MultiFinder's predecessor, and as a result, Apple no longer supports it. Thus with the passage of time, *Switcher* will become less and less compatible with ever-evolving Macintosh software and hardware. Those who have *Switcher* and who remain fond of it may want to continue using it until it no longer works with their applications. Others should head directly for MultiFinder.

7

More Speed and How to Get It

Automated Routines for Greater Speed *402*
- Using *StartMac* *403*
- Using *Tempo II* *410*

Keyboard Shortcuts in Dialog and File Selection Boxes *422*
- Universal Keyboard Shortcuts *423*
- Application Keyboard Shortcuts *424*

Shortcuts for Mouse-oriented People *427*
- Easier Folder and File Access *427*
- Creating "Icon Bars" for All Your Applications *433*

RAM Caches and RAM Disks *436*
- RAM Disks *438*
- RAM Caches *451*
- RAM Disks or RAM Cache? *467*

Print Spoolers *467*

Summary *473*

In previous chapters, we have looked at how to enhance the power and convenience of the Macintosh (by adding useful desk accessories and fonts, standardizing keyboard commands, and so on), and we have considered ways to create homemade, integrated megaprograms from your favorite applications. Although each one of these improvements—power, convenience, and integration—adds a measure of increased speed to the operation of your Macintosh, these are not the only ways or even the best ways to make your Macintosh run faster. There are, in fact, many techniques and software utilities whose sole purpose is speed. It is to these techniques and programs that we turn in this chapter.

The chapter begins with by looking at techniques with which you can speed up and automate your daily preparations for work. These techniques are but a prelude, however, to a number of important techniques which can turn even a sluggard of a program into a bit of a speed demon. Foremost among these are RAM disks and RAM caches; one or both of these utilities should be in the software library of anyone who is looking for a little (or a lot) more hustle from his or her applications.

The last part of the chapter considers print spoolers, a wonderful invention for anyone who uses an ImageWriter or LaserWriter, or who shares a LaserWriter over a network. Print spoolers don't actually make your Macintosh run any faster. Instead, they significantly decrease the amount of time your Mac is tied up by the printing process, thus giving you more time for productive work on the computer.

Automated Routines for Greater Speed

You arrive at your desk in the morning and even before that first cup of coffee, you turn on your Macintosh. You check your electronic mail, launch the usual application or two, and in general prepare yourself for the day ahead. Every day you perform the same ritual, press the same keys, launch the same applications, maybe even open the same documents. Does this sound like you? If so, you can reclaim those ten, fifteen, or twenty minutes a day by letting the Macintosh run these same routines automatically while you do something else, such as search out that morning cup of coffee.

Indeed, few things are as emotionally appealing (in computerdom, anyway) as a completely automated startup routine. Just turn on your Macintosh and all the things you used to do by hand to get ready for work are now done for you: the fonts you manually send to the LaserWriter each morning are downloaded for you, your electronic mail in-basket is checked and any new mail is copied to disk, the print spooler is turned on, a RAM disk is created and loaded, your *HyperCard* appointment book is opened to today's listing, and so on. What could be simpler, more elegant, or more satisfying?

Creating an automatic startup routine is not difficult, although obviously, the more complex the routine, the more effort required to program it and get the bugs out. But even if it takes you an hour or two to perfect an automatic startup sequence, your effort can be repaid in as little as a week or two. In the sections that follow, we look at ways in which you can instruct the Macintosh to perform startup procedures on its own, thus allowing you to do other things while the Macintosh chugs along. With such procedures, your Mac begins working as soon as you turn it on, and it stops only when it has accomplished all your usual startup tasks. This method works best, of course, if you tend to start every day more or less in the same way. Alternatively, you could be offered a choice of startup routines, each of which prepares for (or performs) a different task. This method is for those who have no set routine any given day but do have a limited number of opening gambits.

We will look at two ways to create automatic startup routines. One uses the public-exchange program *StartMac*, the other uses *Tempo II*. *StartMac*, though somewhat finicky, is a more versatile program in ways relevant to this discussion; *Tempo II*, however, is a far more powerful and useful application overall. Which one you should use will depend on what you want to do at startup and, naturally, whether or not you own a copy of *Tempo II*.

Using *StartMac*

StartMac 1.1 (public-exchange) is an application that follows the directions of a special "script file" immediately upon startup. You create the script file with any word processor, using a simple command language that *StartMac* understands. At startup, *StartMac* can show an image (which might contain

directions for novice users, for example), or a series of images which change from one to the next at the click of the mouse button. If you have a copy of *MacinTalk* (the Apple speech synthesizer file) *StartMac* can act as if it were reading aloud the text appearing on the screen, or it can request that the user press the mouse button to continue, or it can report on what *StartMac* is doing at the moment (such as installing a RAM disk). It can tell you the current date or time, or beep to get your attention. *StartMac* can also turn the Control Panel's RAM cache on or off and set the size of the cache as well. Most importantly, it can launch any program (or a sequence of programs).

As an illustration of what automated startups can do, this section shows how to create a fairly complex startup routine using *StartMac* 1.1. Although the particulars of this example may not apply to you, the general concepts illustrated here can be used in a myriad of ways to create an automated startup procedure tailored to your specific needs. Once your script is complete, the Macintosh can take care of many, if not all, of the preparations you perform each morning.

For this example, consider the following situation. You are leaving for vacation and your assistant, who has until now not paid much attention to the Macintosh, will be working on the computer while you are away. You have set up *PowerStation* (see Chapter 6) so that he or she can easily locate the applications and files needed for work. But now you need to automate the startup procedure you go through every day so that your assistant doesn't have to deal with it. Also, you want to add one or two introductory screens containing reminders about the startup process and some general comments about using the Macintosh. The following, in plain English, is what you want *StartMac* to do.

- Display an opening screen reminding your assistant to turn on the LaserWriter and similar mundane matters. This screen should remain in view until your assistant reads the notice and clicks the mouse button.
- Create a RAM disk (see later in this chapter for more on RAM disks).
- Launch *PowerStation*.

Although plain English descriptions work well between humans, to get *StartMac* to take care of this sequence of events, you must write a script

using the limited number of commands that *StartMac* can understand. You may write the script in any word processor that can save a file in text-only format. So start your word processor and follow along.

1. First, you want *StartMac* to display an opening screen and leave it on until the user clicks the mouse button. To make a big impression, you've also decided that the Mac should say a few words while the screen is displayed. Here's how:

    a. Displaying the screen until the user clicks the mouse is relatively straightforward. You use the **PICTURE** command to display the image and the **BUTTON** command to make *StartMac* pause until the mouse button is clicked:

    ```
    PICTURE=PaintDocName
    BUTTON
    ```

    Notice that the commands are all uppercase and that there are no spaces between the command and the equals sign, or the equals sign and the name of the Paint file. If the Paint file is not in the same folder as *StartMac*, you must specify the path to the file, starting with the name of the disk, through any intervening folders, and ending with the name of the file. Each landmark on the path (each disk, folder, and file name) must be separated from the others by a colon, as in `HardDisk:FirstFolder:SecondFolder:ActualFile`. There should be no more and no fewer spaces than there are in the actual names of the disks, folders, and files, and spelling must be exact.

    An important point to keep in mind when using the **PICTURE** command is that the document it calls up must be a standard Paint document, *not* a StartupScreen format file. The portion of the image used (the upper-left corner of the document) is the same as that used by *Screen Maker* when it creates a startup screen.

    b. Now you need to insert instructions for *StartMac* to issue a welcoming statement and ask the user to press the mouse button to continue.

        i. In order to have your comments heard, you want to be sure the speaker is turned on. So you should start the

sequence with the **VOLUME** command, just in case someone has previously set the volume too low (or too high):

```
VOLUME=3
```

The volume can be set anywhere from 0 (no volume, or volume off) to 7, just as in the Control Panel. Level 3 should be more than adequate, unless the office is very noisy.

ii. Next you should add the statements you want the Mac to enunciate, using the **TALK** command:

```
BEEP
TALK=Welcome to Robert's Macintosh.
TALK=Today is &date and it is now
&time.
TALK=Please read and heed the instruc-
tions on this screen.
TALK=Then press the mouse button to
continue.
```

The **BEEP** command merely sounds a beep (either the standard one or the custom one you installed according to the instructions in Chapter 5) to get the user's attention. The remainder of the commands are all **TALK** commands. Each sentence is set as a separate command, both for easier editing later on and to create short pauses between each phrase for better comprehension. (Mac-generated speech is not the easiest thing to understand.) To get the Mac to talk, you don't need to master phonetics first. You type in American English, and *StartMac* will speak Macintosh English in return. The only exceptions are the two codes seen in the second **TALK** command: &date and &time. These are shorthand instructions for the computer to enunciate the current date and the current time, as indicated by the Mac's internal clock. It's probably a good idea, by the way, to repeat the instruction, "Press the mouse button to continue," somewhere on the image used in the **PICTURE** command. Thus if the

user for some reason fails to hear the verbal directions, the written instructions are there to convey this important bit of information.

c. The final script for this segment should look like the following:

```
PICTURE=PaintDocName
VOLUME=3
BEEP
TALK=Welcome to Robert's Macintosh.
TALK=Today is &date and it is now
&time.
TALK=Please read and heed the instruc-
tions on the screen.
TALK=Then press the mouse button to
continue.
BUTTON
```

Note that all the commands associated with the speech segment have been sandwiched between the **PICTURE** and the **BUTTON** commands. This is because you want *StartMac* to speak after the picture appears but before the **BUTTON** command's pause is invoked. If you place the speech segment before the **PICTURE** command, *StartMac* will speak before the picture appears. Of course, if you place the speech segment after the **BUTTON** command, *StartMac* won't speak until after you press the mouse button. Neither is the effect you want.

2. Now it is time for *StartMac* to create the RAM disk. Let's assume you are using *RamSnap* (which is described later in this chapter), and that it is in the System Folder on your hard disk. We'll also assume that you have already configured *RamSnap* to create a RAM disk of the desired size and to copy the proper files to it, and that the *RamSnap* configuration file, SnapStart, is in the System folder as well. The command (which includes the proper path description) is:

```
OPEN=Hard Disk:System Folder:RamSnap
```

The only problem with simply running *RamSnap* like this is that it is often possible to halt programs such as *RamSnap* in midstream simply by pressing and holding the mouse button or pressing

Command-period. Anyone unfamiliar with your startup routine might panic, thinking that the computer has somehow gone out of control, and do just that. So to keep the user informed about what is going on and to forestall any potential confusion or anxiety, let's add a couple of phrases for *StartMac* to deliver before launching *RamSnap*:

```
TALK=I am now creating the ram disk.
TALK=Please stand by.
```

Thus the final sequence of commands for this segment is:

```
TALK=I am now creating the ram disk.
TALK=Please stand by.
OPEN=Hard Disk:System Folder:RamSnap
```

3. Finally, you need to launch *PowerStation*. To do so, you use the OPEN command again. *PowerStation* should be in your System Folder, and the Open PowerStation file should be in the main Hard Disk window (not in any folder). Your command should look like this:

```
OPEN=Hard Disk:Open PowerStation
```

If you like, you can add a comment or two before starting *PowerStation*, to round out the whole procedure:

```
TALK=I am now starting PowerStation.
TALK=You should find everything you
need on the PowerStation screen.
TALK=I am signing off.
TALK=So long until tomorrow.
```

4. The final script, then, should look like this:

```
PICTURE=PaintDocName
VOLUME=3
BEEP
TALK=Welcome to Robert's Macintosh.
TALK=Today is &date and it is now
&time.
TALK=Please read and heed the instruc-
tions on this screen.
```

```
TALK=Then press the mouse button to
continue.
BUTTON
TALK=I am now creating the ram disk.
TALK=Please stand by.
OPEN=Hard Disk:System Folder:RamSnap
TALK=I am now starting PowerStation.
TALK=You should find everything you
need on the PowerStation screen.
TALK=I am signing off.
TALK=So long until tomorrow.
OPEN=Hard Disk:Open PowerStation
```

5. Finally, save your script as a text-only file named `STARTMAC.COMMANDS`.

Now you have all the pieces of the *StartMac* puzzle. There are just a few more things to do back in the Finder, before all the pieces can fall into place:

1. Place the `STARTMAC.COMMANDS` file, *StartMac* itself, and the paint document you refer to at the beginning of the script in the System folder. Make sure that all the applications you refer to in the script are where they are supposed to be as well.
2. Since you are using the Talk command, you must be sure that MacinTalk is in the System folder; otherwise, your Mac will remain mute no matter what you want it to say.
3. Finally, select *StartMac* so that it is highlighted and then use the Set Startup command from the Special menu to make *StartMac* the startup application.

Your automatic startup is now ready to roll. The next time you or anyone else turns on or restarts your Macintosh, *StartMac* will take over and run through the procedure described in the script and then hand over control to *PowerStation*, since launching it is the last command in the script.

By the way, you do not need to restart your Mac just to test your script. With *StartMac*, the script file, and all the other files referred to in the script in their proper places, double-click on the *StartMac* icon and *StartMac*

will play the entire script, just as it would on startup. If the script fails to play properly, this is the time to fix it and retest it until you get it right.

One final note on using *StartMac* for startup procedures. *StartMac* 1.1 is fairly fussy about where your files are located (sometimes it wants them in a particular folder, sometimes not), and it does not work at all with some applications and/or with some System file configurations. For greater flexibility and reliability in startup procedures, use either *StartMac* 2.0 (available only to registered users of *StartMac* 1.1), *MacPS* (Neff Systems Group), or a similar program.

StartMac 2.0 can do everything *StartMac* 1.1 can, plus it can open documents as well as applications, launch desk accessories, and allow you to select from among more than one script file. *MacPS*, a product of *StartMac*'s creator, is a sort of super *StartMac*. Like *StartMac*, *MacPS* can be used to automate complex startup procedures. It can also be used for automatic slide shows and tutorials, for complex shutdown routines, to open a number of documents at once, to automate your daily backups, and so on. *MacPS*'s command language contains many more commands than *StartMac*, including a large selection of predefined variables, more Finder functions (trashing and copying files, ejecting disks), and conditional branches (IF this is true, *MacPS* will EXEC that).

Obviously, neither of these programs is particularly Mac-like. They look and operate more like a programming language than anything else. But, as you have seen, they aren't really all that difficult to master, and until a better program comes along with the same abilities (*Tempo II* could have accomplished some of the tasks, but not all) and a similar price (*Tempo II* is quite a bit more expensive), *StartMac* and *MacPS* will serve nicely.

Using *Tempo II*

Tempo II, as we have already seen, is a program with a wide variety of uses. An important feature of the program, its **Branch** command and the Set Startup file which accompanies *Tempo II*, make it especially attractive for automatic startup routines. *Tempo II*'s **Branch** command is particularly useful in constructing highly complex startup procedures from any number of simpler, individual macros.

Because *StartMac* was created specifically for automatic startups, it can do some things which *Tempo II* cannot (or which *Tempo II* can do only through tortuous means). *Tempo II*, for example, is unable to display an

opening graphic screen, nor can *Tempo II* read any of its messages aloud. *Tempo II*, on the other hand, is more versatile in other ways, and thus can, in turn, accomplish tasks too complex for *StartMac*. You could, for example, design a startup routine that launched your telecommunications program, called CompuServe or MCI Mail, checked your electronic mail, downloaded whatever mail was there, signed off, quit your communications program, and launched your word processor so you could read the downloaded files.

Indeed, reading your mail could be but a small segment of your entire automatic startup. If you have a standard set of fonts that you manually download to the LaserWriter each morning, you could program *Tempo II* to take care of that task for you as well. In fact, to illustrate some of the power of *Tempo II* as a startup procedure generator, we'll consider in this section how you can use it to download fonts to the LaserWriter on startup.

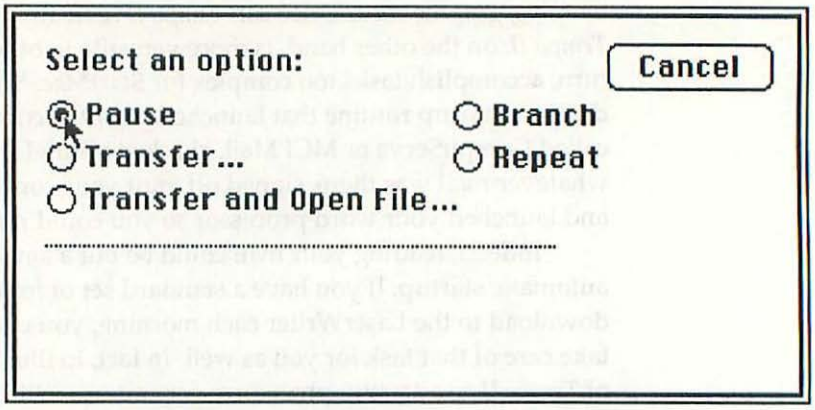
When creating complex *Tempo II* macros, it is always a good idea to construct them in segments and then assemble all the parts as the very last step. I should point out, however, that in the scheme of things, the macro we will create is not all that complex, and most people would make it in one fell swoop. We'll do it in pieces here, however, to illustrate how it's done. Thus, we'll break up the downloading procedure into its component parts (opening the font downloading program, selecting files and downloading them, and quitting the program), create macros which perform each individual task, and then assemble the individual macros into a single super-macro at the end of the whole process.

First, we need to put up an alert box which warns the user that the LaserWriter must be turned on and warmed up before proceeding:

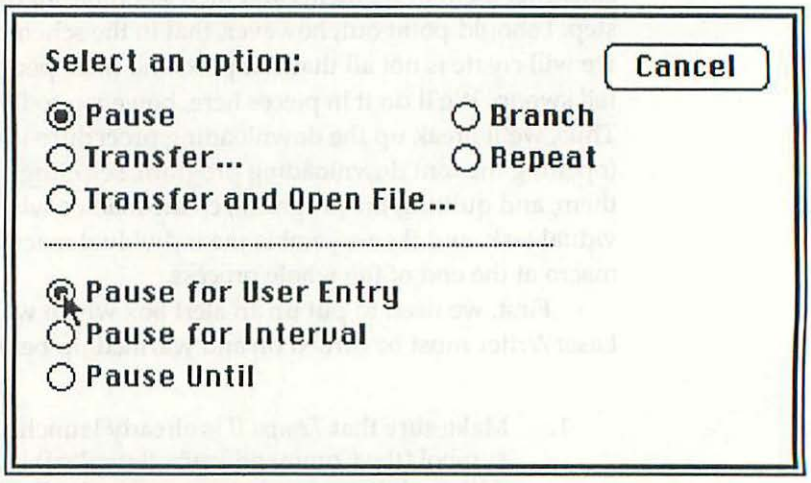
1. Make sure that *Tempo II* is already launched and the *Tempo II* symbol (the Command key's cloverleaf) is in the menu bar. Although I assume that you are in the Finder (since that's where a startup routine would be run), you can create this macro in any application as long as you turn on the **Save as Universal Macro** option.

2. Select **Start Recording** from the *Tempo II* menu (or press Command-accent grave [`]).

3. Select the **Options...** command from the *Tempo II* menu and click on **Pause**, then **Pause for User Entry**, then **Include Dialog Box**, and **Resume With "OK"**.



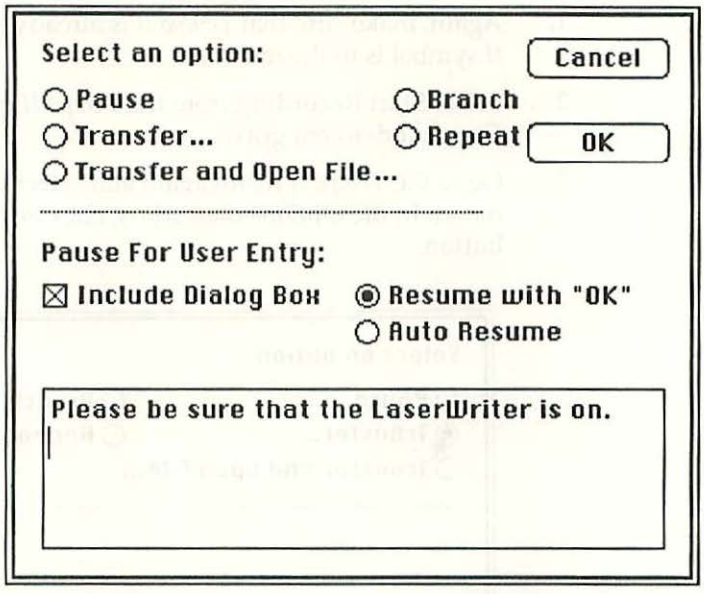
Select the Pause option in the Options dialog box.



Then click on the Pause for User Entry radio button.

4. In the text entry box, type in the text for the dialog box:

```
Please be sure that the LaserWriter is on.
If it is not on, please turn it on now and wait for
the steady green light.
Then click on OK and select Resume from the Tempo II
menu.
```



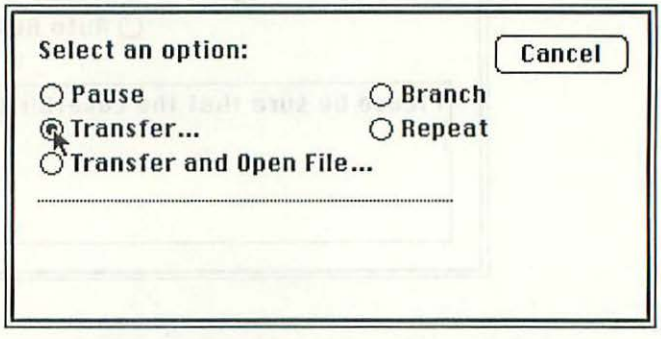
In the Pause For User Entry dialog box enter the LaserWriter message in the text entry box.

5. Click on the OK button.
6. Select **Stop Recording** from the *Tempo II* menu and in the Save Macro dialog box, name the macro `LW On Alert`. Do not assign a key code (unless you think you might have a use for one). Turn on the Save as Universal Macro option and click on the Save button.

Next we need to launch the font downloading application and download a font. Let's assume that we are using Adobe's *Font Downloader*, which is included with each of their fonts, and that *Font Downloader* has been set to recognize your printer (using Chooser) at some point in the past.

Because the font downloading process is not the same as launching an application and opening a document, we cannot use the **Transfer and Open File** technique described in Chapter 5. Instead, we'll have to launch the program, invoke the Download command, and then select a font file to download. To perform this more complex task, we will again break it into several smaller parts. Thus the next step is simply to launch *Font Downloader*:

1. Again, make sure that *Tempo II* is already launched and the *Tempo II* symbol is in the menu bar.
2. Select **Start Recording** from the *Tempo II* menu or press Command-accent grave.
3. Go to the *Tempo II* menu again and select the **Options...** command. In the Options dialog box, click on the Transfer... radio button.



*Select the **Transfer...** option in the Options dialog box.*

4. In the resulting file selection box, locate and then select *Font Downloader*. Click on the Open button. Wait for *Font Downloader* to complete its launch sequence.
5. Now within *Font Downloader*, select **Stop Recording** from the *Tempo II* menu (or press Command-accent grave again).
6. In the Save Macro dialog box, type in the macro name `Launch Font Downloader`. Click on the Save as Universal Macro check box to turn that option on, then click on the Save button.

Now we need to create a macro which will select *Font Downloader's* **Download** command and open the folder containing the fonts to be downloaded:

1. Select **Start Recording** from the *Tempo II* menu.
2. Press Command-D or select the **Download Font** command from the File menu.

3. Since *Tempo II* always resets file selection boxes to the "root" level (the lowest level possible in the chain of disk to folder to file), make sure that the name of your "root directory" (your hard disk or startup disk) appears in the box just above the file selection list. If the root directory does not appear in the box, and a folder name appears there instead:
   a. Suspend recording by pressing Command-comma (or if you prefer, select **Suspend Recording** from the *Tempo II* menu).
   b. Drag down back to the root level through the popdown menu that appears below the folder name when you click on it, which should be the last name in the popdown menu. With the root level selected, release the mouse button.
   c. Press Command-comma again to resume recording.
4. Assuming your printer font files are in your System folder (which they should be for automatic downloading), scroll through the file list until the System folder appears, and double-click on it.
5. Now select the **Stop Recording** command (or press Command-accent grave again).
6. In the Save Macro dialog box, name the macro `Open Font Folder`. Click on the OK button.

Next the heart of the matter: creating a macro which will select and download a specific font:

1. The file selection box should still be open and the System folder laid bare. If not, select the **Download Font** command again (or press Command-D) and/or open the System folder as appropriate.
2. Select **Start Recording** from the *Tempo II* menu or press Command-accent grave. Then select **Real Time** from the *Tempo II* menu (if you do not, *Tempo II* will replay the macro far too fast).
3. Now scroll through the System folder until you locate the file of the font you want to download. For example, if you want to download New Baskerville Roman, locate NewBasRom.
4. Select the font file and press Return, or just double-click on it. A dialog box will appear asking you to confirm your choice of font. Press Return again (or click on the OK button).

5. After a time, another dialog box will appear, informing you that the download was successful. Again, press Return or click on the OK button.

6. Select **Stop Recording** from the *Tempo II* menu, give the macro a name appropriate to the font it downloads, such as `Download New Bask Rom` for this example, and save it.

If you want to have *Tempo II* download more than one font, repeat the previous sequence as many times as necessary—or until the LaserWriter runs out of memory, whichever comes first—saving each macro with a concise name which clearly indicates the font which is downloaded.

> NOTE: Now that we've gone through all this work, I should point out that it would have been much easier to create a multiple font downloading macro had we used the desk accessory *LaserStatus* to download our fonts instead of *Font Downloader*. This is because you can create font sets with *LaserStatus* and put both frequently needed individual fonts and font sets in a special, configurable Downloads menu. To download an entire suite of fonts, all you would have to do is program *Tempo II* to first select *LaserStatus* from the Apple menu and then select a previously created and installed font set from the Downloads menu—and that's it. At the time this was written, *LaserStatus* did have one problem that must be taken into account, however. It worked poorly or not at all with some print spoolers. Thus if this problem hasn't been fixed and if your startup routine includes turning on a print spooler, you must place the *LaserStatus* part of the routine before the print spooler segment, not after.)

Finally, we need a macro to terminate the downloading process, quit *Font Downloader*, and return us to the Finder. This is easy enough.

1. You should still be in *Font Downloader*, which should display only a blank screen.

2. Select **Start Recording** from the *Tempo II* menu.

3. Press Command-Q or select Quit from the File menu to quit *Font Downloader*.

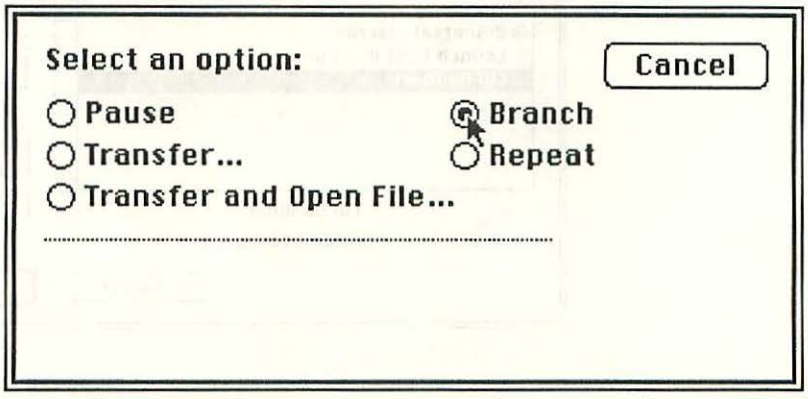
4. When the desktop reappears, select **Stop Recording** from the *Tempo II* menu. In the Save Macro dialog box, name the macro `Quit Font Downloader` and click on the Save button.

You should now have a collection of the following macros:

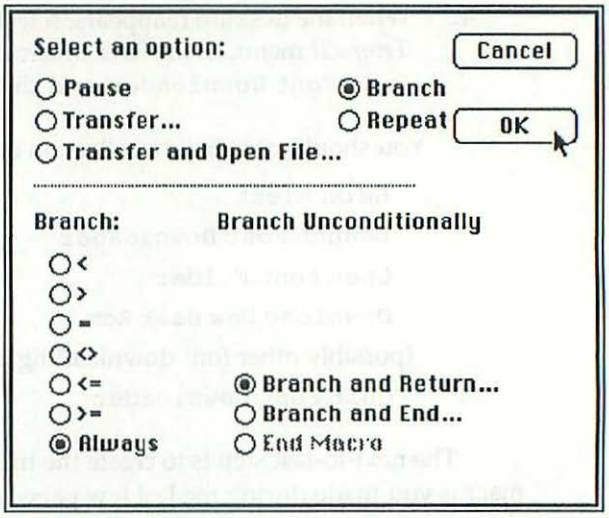
`LW On Alert`
`Launch Font Downloader`
`Open Font Folder`
`Download New Bask Rom`
(possibly other font-downloading macros as well)
`Quit Font Downloader`

The next-to-last step is to create the megamacro from the individual macros you made during the last few pages of instructions.

1. From the Finder, select the **Start Recording** command from the *Tempo II* menu.
2. To start your multiple-part downloading macro with the `LW On Alert` macro:
   a. Select the **Options...** command from the *Tempo II* menu.
   b. Click on the Branch option, then the Always and Branch And Return... options.



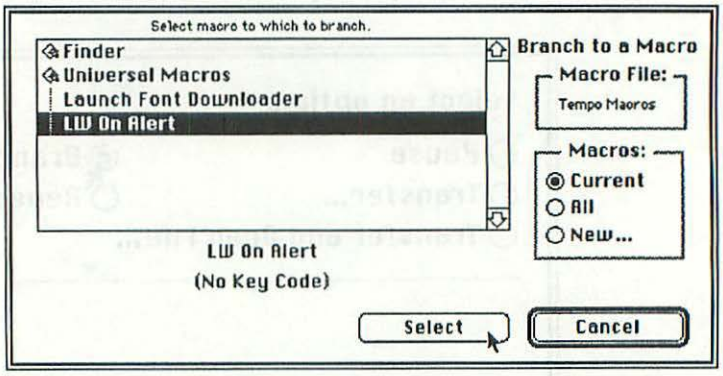
To create one macro out of several, first click on the ***Branch*** *option.*



Then select the Always and Branch and Return... options.

c. Click on the OK button.

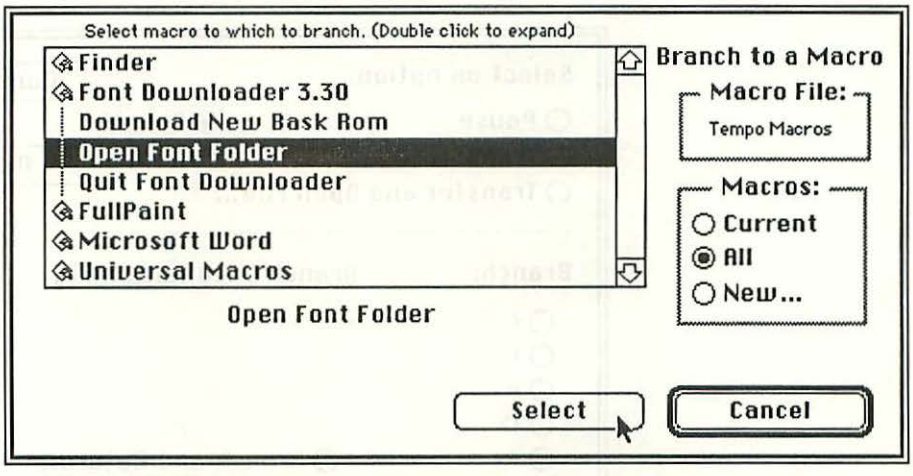
d. Select the `LW On Alert` macro from the Branch To A Macro file selection box and click on the Select button.



Finally, highlight the first macro in the sequence from the file selection box and click on the Select button.

3. Repeat step 2, except this time select the `Launch Font Downloader` macro.

4. The next step is to add the Open Font Folder macro. Since you probably saved this and the following macros as *Font Downloader* macros (with the Save as Universal Macro check box turned off), we need to use a slightly different technique:
   a. Select the **Options...** command again, choose the same set of Branch options as before, and click on OK.
   b. In the Branch To A Macro file selection box which then appears, turn on the **All** macros option. Since you are currently in the Finder, you need to do this in order to see the macros which you created within *Font Downloader.*

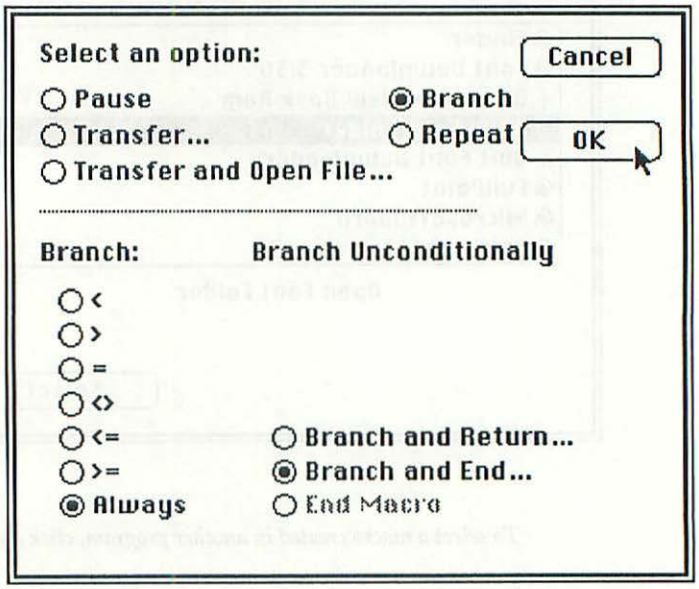


To select a macro created in another program, click on the All radio button.

   c. Select the `Open Font Folder` macro and click on the Select button.

5. Now add the macro which downloads the specific font, `Download New Bask Rom`. If you have several such macros, add them all at this time. The process for adding one or more such macros is the same as that outlined in step 4. Be sure, however, that you add only as many macros for as many fonts as the LaserWriter's memory can handle. A startup routine which both overloads the LaserWriter's memory and grinds to a halt as a result is of no use to anyone.

6. Finally, you need to add the `Quit Font Downloader` macro to the end of your super-macro. Since this is the last piece in the puzzle, the procedure for doing so is (again) slightly different:
   a. Select the **Options...** command from the *Tempo II* menu.
   b. Click on the Branch option, then the Always and Branch and End... options.



For the last macro in the sequence, select the ***Branch and End...*** *option.*

c. Click on the OK button.

d. In the Branch To A Macro file selection box which then appears, click on the Show All Macros button. Select the `Quit Font Downloader` macro and click on the Select button.

7. Select **Stop Recording** from the *Tempo II* menu. In the ensuing Save Macro dialog box, name the macro `Font Download Startup` and click on the Save button.

You can now employ this macro as an automatic startup procedure either by itself, or embedded in a more complex sequence. Whether you use it alone or as part of a larger macro, you must convert the final supermacro into an automatic startup procedure.

1. Place a copy of *Tempo II*'s Set Startup file in your master System folder.

2. Launch Set Startup by double-clicking on it while you hold down the Command key. Be sure to hold down the Command key until you see the Set Startup directions in the menu bar.

3. In the file selection box, click on the name of your startup macro (in this case, `Font Download Startup`), then click on the Select button.

4. Back in the Finder, select the **Configure...** command in the *Tempo II* menu, click on the Default Macro File... button, and make sure the *Tempo II* macro file on your startup disk is selected as the default file.

5. Finally, select *Tempo II*'s Set Startup file, then use the **Set Startup...** command in the Finder's Special menu to set that file as the startup application.

If all has gone as planned, you will be prompted to turn on the LaserWriter and New Baskerville will be downloaded automatically each time you turn on your Mac in the morning.

This, of course, is just one example of the sort of thing you do with *Tempo II* to create sophisticated, time-saving, automatic startup routines. But even if there isn't one single startup procedure you perform every morning,

Tempo II can still be used to make short work of your morning preparations and other time-consuming tasks as well. If there are common tasks you perform when you first start your Macintosh, you can create macros for each of these tasks or combinations of tasks, assign keyboard commands to them, and then once your Mac is started and *Tempo II* is turned on, you need only press a few keys to get going. Which macro you invoke will depend on your plans for the day and all the other factors that already influence your startup routine.

Keyboard Shortcuts in Dialog and File Selection Boxes

Another source of speed and efficiency is one which is already at your fingertips. Previously, we considered keyboard shortcuts for menu commands and the like, and we saw how these can be increased when in short supply with *ResEdit*, *MenuKey*, *MenuMaster*, *QuicKeys*, and *Tempo II*. We've also seen how you can use *DialogKeys* to operate dialog boxes which do not normally respond to keyboard input. But the keyboard shortcuts which are the subject of this section and are all too often ignored are the ones which already exist in standard dialog and selection boxes in the Finder and in your various applications. Some of these shortcuts are displayed in the dialog box buttons themselves. Many others, however, are documented poorly in the back pages of manuals or not at all. Yet if you have any facility with the keyboard, using quick and easy keyboard commands in dialog and selection boxes can be far faster than groping for the mouse.

Before all you mouse-lovers protest, let me assure you that this endorsement of keyboard shortcuts comes from one who, in most circumstances, would rather mouse than type. I still use the mouse for all sorts of word-processing tasks, for example, which are (to me at any rate) unduly awkward from the keyboard. Yet there are many other instances where the instinctive grab for the mouse has been productively replaced by the press of a key or two. The result is a faster and more efficient four-finger typist (if that isn't an oxymoron). So if it works for me, just imagine what a boon keyboard shortcuts could be to an accomplished typist.

Universal Keyboard Shortcuts

Here then is a list of some of the most useful universal shortcuts for file selection and dialog boxes. If you've been paying close attention, you may well know some or all of them already. If you find things here that you didn't know before, then you might want to try integrating them (a few at a time, perhaps) into your everyday routine and see if they help speed things up a bit.

- In a dialog box with more than one (large) button to choose from, the one outlined with a thick, black border can be invoked by pressing the Return key.
- Almost any dialog box that simply asks for confirmation—in other words, one which has only a single button labeled OK—can be closed by pressing the Return key. This is true even if the OK button is not outlined with a thick, black border.
- Many dialog boxes accept Command-period as a keyboard equivalent for the Cancel button (and it doesn't hurt to try it to find out).
- In a dialog box with several text input boxes, you can almost always jump from one input box to the next by pressing the Tab key. (Rarely, the Return key performs this function; in most, however, Return is equivalent to clicking on the OK button.)
- In file selection and save boxes there are many techniques you can use to avoid grabbing for the mouse:
  - ◊ On the Mac Plus, use Command-Shift-1 to eject the disk in the internal drive or Command-Shift-2 to eject the disk in the external drive, instead of clicking on the Eject button. (On a two floppy drive Macintosh SE, Command-Shift-1 ejects the upper disk, Command-Shift-2 the lower disk.)
  - ◊ Use the Tab key to change from one disk drive to the next. (Of course, if only one disk is inserted, the Tab key does nothing.)
  - ◊ Use the arrow keys to move from file to file and folder to folder. The arrow keys, by themselves, move the black high-

light bar up and down the current file and folder list. Command-Up Arrow closes the current folder and displays the contents of the folder (or disk) which contains it. Both Command-Down Arrow and Return open the currently selected folder.

◊ To select a file or folder in a file selection box, you can type the file or folder's name, or enough of the name to distinguish it from the other files and folders on the list. To narrow down a search in a long list, poor typists (like me) can type the first letter of the name to get close to the file in question and then use the arrow keys to nail down the precise file.

If you type a name such as `Letters` to get to the Letters Folder and end up at the Telecommunications Folder instead, you aren't typing fast enough. To avoid this sort of problem, either type faster or if both fingers are typing as fast as they can, select a longer Delay Until Repeat in the Keyboard section of the Control Panel.

◊ Of course, once a file is selected in a file selection box, pressing Return is the equivalent of clicking on the Open button.

Application Keyboard Shortcuts

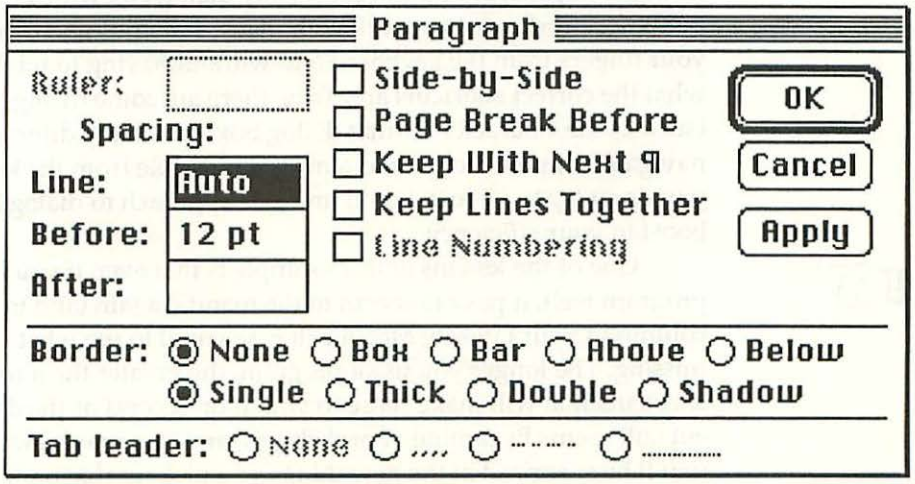
The First Commandment for any fully powered Macintosh owner is, quite simply, Know Thy Application. (I suppose that means that the Second Commandment is Read Thy Manual.) You'd be surprised how many people spend hours squeezing the last ounce of speed out of their Macintosh and then turn around and select a menu command, which has an obvious keyboard shortcut, with the mouse. They'd probably save more time if they memorized (and used) a few keyboard shortcuts instead.

Some programs are pretty stingy with keyboard equivalents. Other applications, however, have so many keyboard shortcuts that there's no way you could remember (or use) them all, unless you spent all day, every day, working with that application to the exclusion of everything else. Microsoft *Word* is one such program. Though I use it every day, I can't remember even half of all the keyboard shortcuts that *Word*'s designers managed to dream up. (Of course, *Word*'s designers never intended for one person to remember and use all of its many shortcuts, or at least I hope they

didn't. Rather, they provide enough variety that everyone will find at least a few shortcuts relevant to their particular needs and habits.) Some of them do little more than tie my fingers in a knot. But others perform functions that (apparently) can't be accomplished any other way. Still others make it possible to navigate *Word*'s many dialog boxes from the keyboard (not as elegantly as in *Modern Jazz*, but elegantly enough).

Indeed, one of the unsung and often-overlooked pleasures in *Word* is the way in which keyboard commands can be used to navigate any *Word* dialog box. Instead of clicking on a dialog box option or button, you can press Command and the first letter of the option or button name. (If there are several buttons which start with the same letter, you'll get the first one going from left to right and top to bottom.) For example: Command-C for **Change All**, Command-S for **Start Search**, Command-F for **Find Next**, Command-A for **Apply**, Command-S for **Set Default**, Command-C for **Cancel**, Command-S for **Start**, Command-B for **Box**, Command-O for **Open**, Command-D for **Double** (underline), Command-E for **Eject**, and on and on.

Two things make this use of keyboard shortcuts unusual. One is that the "standard" *Word* keyboard shortcuts, such as Command-C for **Copy**, Command-S for **Save**, Command-D for **Character...** (a format command),



You can use keyboard shortcuts for many of the options in this Word *dialog box: Command-K for* **Keep With Next Paragraph**, *Command-B for* **Box**, *Command-T for* **Thick**, *Command-P for* **Page Break Before**, *and so on.*

and Command-E for **Footnote**, are temporarily suspended whenever a dialog box appears. Because the shortcuts associated with a dialog box are directly related to the names of the options and buttons (something which is not so true with the standard shortcuts, as the last two examples amply show), and because one does not expect a standard shortcut to work when a dialog box is open, this temporary change is not as confusing as it initially sounds.

The second unusual aspect of all this is that each dialog box has its own unique set of keyboard shortcuts. Command-B gets you **Box** in the Paragraph format dialog box and **Bold** in the Character format dialog box. Command-C can mean **Change All** or **Cancel**. Command-S can mean **Start Search** or **Set Default** or **Save As** or **Start** (making a table of contents). Again, because the names of the Commands are directly related to the shortcuts, confusion is rare. Command-period, by the way, always means **Cancel**, so when Command-C is preempted, you can still cancel a dialog box from the keyboard.

And what a pleasure it is to reply to dialog boxes from the keyboard. Being able to press Return to invoke the highlighted button has always been a time-saver. But having nearly every button (and option) thus available is even better. You can do a complex search and change sequence entirely from the keyboard. You can cancel any dialog box with a quick Command-period. Indeed, you can work with many dialog boxes without ever taking your fingers from the keyboard and without having to think much about what the correct shortcuts are. True, there are some dialog boxes in *Word* (such as the Character Format dialog box) which are difficult enough to navigate with the mouse and almost impossible from the keyboard. But if you don't try to force it, you'll find this approach to dialog boxes a real boost to your efficiency.

One of the lessons of this example is that even if you think you know a program well, it pays to return to the manual again (and to read the tips column in your favorite Macintosh magazine) to see what you might be missing. The longer you use a program, the greater the number of keyboard shortcuts that will make sense to you. If on second or third reading, a shortcut still seems Byzantine, it probably is, and you may as well ignore it. But you'll be surprised at the new things you pick up that once seemed beyond the pale. I responded to *Word*'s secret way of handling dialog boxes on only the one-hundred and first reading of the program's 18-page *Quick Reference Guide*. I suppose keyboard shortcuts are where and when you find them.

Shortcuts for Mouse-oriented People

Mouse addicts, despair not. There is help for you, too. If you prefer to negotiate dialog boxes, file selection boxes, menus, and menu commands with the mouse, there are ways to speed up mouse movements as well. In fact, you don't have to be a mouse addict to take advantage of these enhancements. I'm one of those people who uses the mouse for some things and the keyboard for others, and I use both keyboard and mouse optimizers. You might find it useful to do the same.

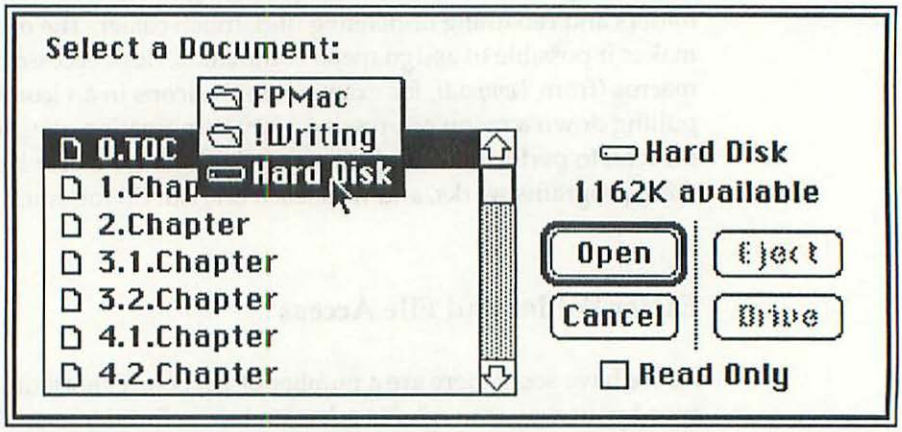
Two utilities are of particular interest to those who find the mouse a handy thing to have around. One, *HFS Navigator*, works with any Open or Save dialog box and makes finding existing folders, as well as creating new folders and renaming or deleting files, much easier. The other, *Icon-It!*, makes it possible to assign menu commands, desk accessories, FKeys, and macros (from *Tempo II*, for example) to the icons in an icon bar. Instead of pulling down a menu or pressing a key combination, you simply click on the icon to perform the desired action. Let's take a closer look at how each of these programs works, and what each one can do for you.

Easier Folder and File Access

As we have seen, there are a number of keyboard shortcuts you can use to speed your way through file selection boxes. But if you prefer to mouse through file selection boxes, *HFS Navigator* (THINK Technologies) can increase your efficiency. With *HFS Navigator*, you can place the folders you use most frequently within easy reach. You can also find files and folders, rename or delete files, and get information about files from within any application with a minimum of keyboard input.

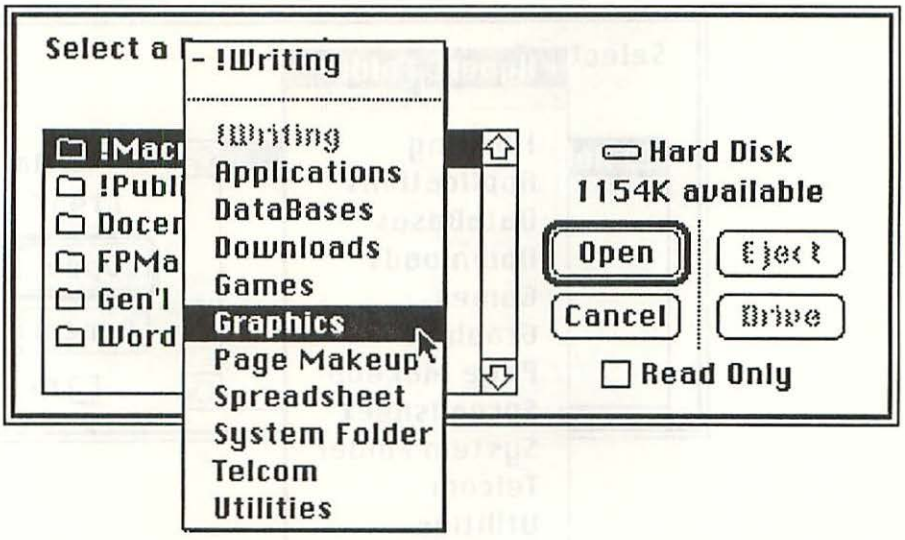
HFS Navigator is a special modification to your System file. To install it, you launch the *Install HFS Navigator* program and click on the Install button. Unfortunately, *Install HFS Navigator* can modify only the currently active System file. Thus to install *HFS Navigator* in more than one System file, you need to launch the program, install *HFS Navigator*, and quit, then restart with the new System file and repeat the process for each System you want to modify. If you have but one master System file, then this isn't a problem.

Once installed, *HFS Navigator* makes itself known only when you click on the current folder box (the small box above the file list) in Open or Save file selection boxes. Normally, when you click on the current folder box, a hierarchical menu—a menu containing the names of all the folders which contain the current folder and ending with the name of the disk all the folders are on—drops down. You can drag the mouse button to any folder on the menu to see that folder's directory (in other words, the files and folders it contains) or drag to the last name to see the root directory (the list of all the files and folders which would appear in the disk window). But when *HFS Navigator* is installed, something different happens when you click on the current folder box.



The standard hierarchical menu lists all the folders between the current folder and the "root" level, such as your hard disk.

Actually, one of three possibilities will happen when you click on the current folder box. A plain click (no keys presses at the same time) will bring up the navigation menu: a list of the folders you use most frequently, no matter where they are located on the disk. By dragging the mouse pointer to the desired folder and releasing the mouse button, the directory of the selected folder will appear instantly in the file list. If you hold down the Command key when you click on the current folder box, a menu of *HFS Navigator* commands will appear. Using this command menu, you can locate lost files or folders, create new folders, get information about a

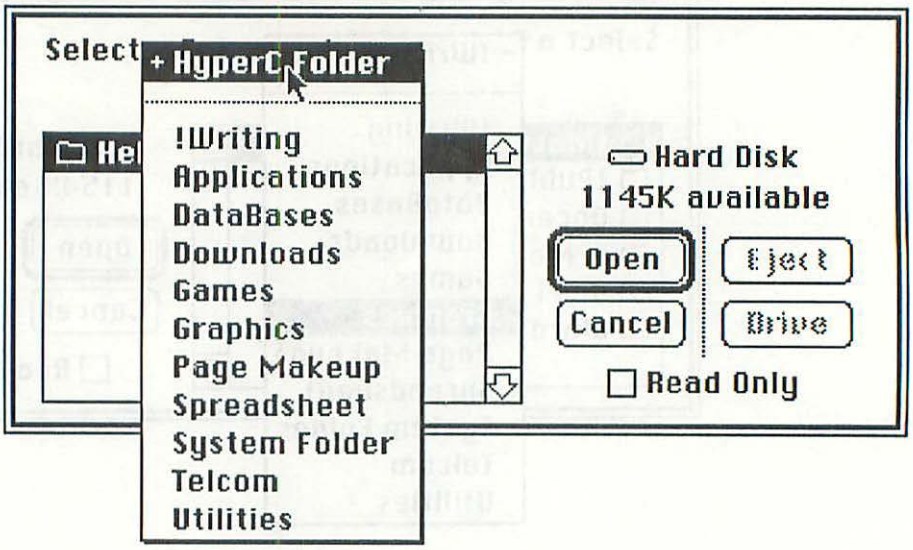


HFS Navigator *offers a different way of moving quickly from one folder to another with its navigator menu.*

specific file, or rename or delete a file. If you hold down the Option key when you click on the current folder box, the standard hierarchical menu appears.

When you first use *HFS Navigator*, the navigation menu will be empty. But adding folders to the navigation menu is easy. All you have to do is:

1. Select a command which opens a **Save** or **Open** dialog box.
2. Go to the folder you want to add to the navigation menu using the usual methods. (Hold down the Option key to get the hierarchical menu if you need it).
3. When the name of the folder you want to add to the navigation menu appears in the current folder box, click on the box (without pressing any keys). The name of the folder will reappear above the pointer (and the navigation menu, if it contains anything at this point, will drop down beneath it). The folder name should have a plus sign beside it. This means that if you drag the pointer up to the folder name to highlight it and then release the mouse button, the folder name will be added to the navigation menu.

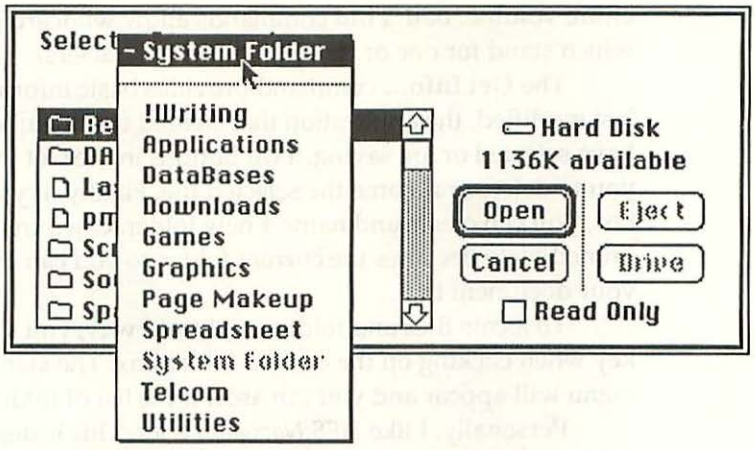


Add a folder to the navigation menu by opening the menu and moving the pointer up to the folder name, above the dashed lline.

Now, every time you click on the current folder box, the folder you just added to the navigation menu will be listed there, no matter what folder you are in. You can transport yourself to that folder instantly simply by dragging to the folder name to highlight it and releasing the mouse button. No longer do you need to retreat back to the root directory or some higher level folder and then descend into a new sequence of folders. Just click, drag, and release and you're there.

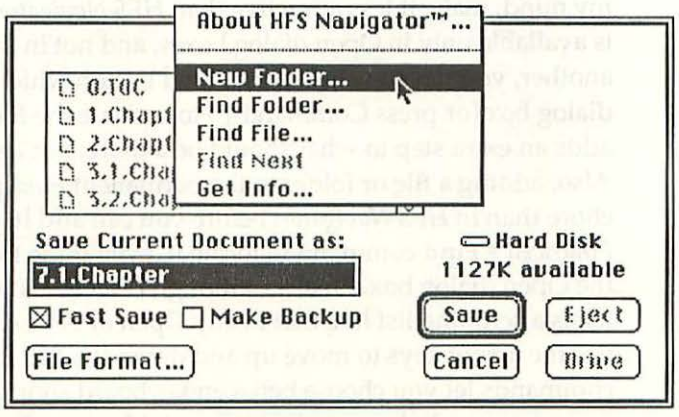
Deleting a folder from the navigation menu is just as easy as adding one:

1. Use the navigation menu to go to the folder you want to delete.
2. Click on the current folder box. Note that the folder name that appears above the pointer now has a minus next to it (as in the illustration on the next page).
3. Drag up to highlight the folder name and release the mouse button. This will remove the folder name from the navigation menu.



Add a folder to the navigation menu by opening the menu and moving the pointer up to the folder name, above the dashed line.

To reach the *HFS Navigator* command menu, hold down the Command key when you click on the current folder box. As with the navigation and hierarchy menus, you simply drag the mouse until the command you want is highlighted, and then release the mouse button. Two commands on this menu will search for files or folders. Using whole or partial names, they will search either the current folder and all the folders it contains, or the



With HFS Navigator*'s command menu, you can create a new folder, find files, and more.*

entire volume. Both **Find** commands allow wildcard characters (characters which stand for one or more unknown characters).

The **Get Info...** command provides basic information (file size, date last modified, the application that created the document) about the file you have selected or are saving. Two buttons in the Get Info dialog box allow you to delete or rename the selected file. Finally, if you are in a Save dialog box, you can create and name a new folder. Once created, the new folder immediately becomes the current folder so you can quickly and easily save your document in it.

To locate files and folders in the old way, you simply press the Option key when clicking on the current folder box. The standard hierarchical menu will appear and you can ascend the list of folders as required.

Personally, I like *HFS Navigator* a lot. This is despite the fact that until *HFS Navigator* came along, I tended to navigate file selection boxes fairly efficiently from the keyboard. (*HFS Navigator* obviously works only with the mouse.) The only drawback to it is that there is a limit of sixteen folders that can appear in the navigation menu at any one time. It would be better if there were no limit at all.

There is no limit to the number of folders you can list in *Findswell* (Working Software), a utility which is similar in concept, if not in execution, to *HFS Navigator*. *Findswell* has other advantages, too. It can list frequently used files as well as folders, and it is an Init resource, which means that to install it, you need only copy it to your System folder.

But unfortunately, *Findswell* has a number of design flaws which, to my mind, make it less attractive than *HFS Navigator*. For one thing, *Findswell* is available only in Open dialog boxes, and not in Save dialog boxes. For another, you have to click on the Find button which appears in every Open dialog box (or press Command-P) to activate the *Findswell* window, which adds an extra step to what should be a more, not less, economical process. Also, adding a file or folder to the permanent *Findswell* list is more of a chore than in *HFS Navigator*: before you can add it, you first have to use *Findswell*'s **Find** command to locate it, even if the file is already selected in the Open dialog box. Finally, although *Findswell*'s permanent file and folder list is a scrolling list like that in any Open or Save dialog box, you cannot use the arrow keys to move up and down the list. Even though most other commands let you choose between keyboard shortcuts and buttons, there is no way to scroll through the file and folder list other than by using the mouse.

Creating "Icon Bars" for All Your Applications

Icon-It! (Olduvai Software) makes invoking menu commands, desk accessories, and macros a point-and-click affair. With *Icon-It!*, so-called icon bars are specific to and installed in individual applications. The icons on each icon bar offer a quick and easy way—just point and click—to order up all sorts of functions.

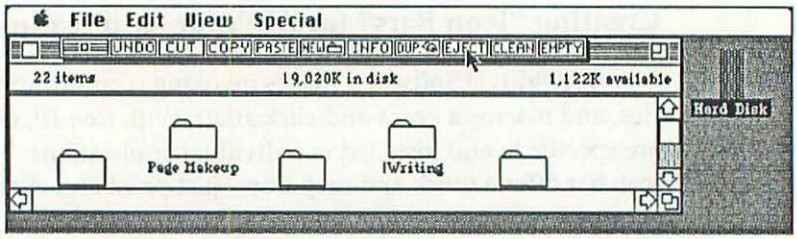
Icon-It! icon bars float on the screen, below the menu bar at the bottom of the screen, wherever you want them. On each icon bar is a row of icons which represent specific menu commands, desk accessories, FKeys, even *Tempo II* macros. *Icon-It!* comes with a wide selection of preprogrammed icon bars, or you can design your own. To activate whatever function is linked to an icon, you simply click on the icon.

Icon-It! is an Init resource, so to install it, you need only copy the *Icon-It!* file to your System folder and restart your Macintosh. Once installed, *Icon-It!* appears as a desk accessory under the Apple menu. If you want to install one of the program's preconfigured icon bars in your application, the process is a simple one.

1. Open the application in which you want to use the icon bar.
2. Select *Icon-It!* from the Apple menu.
3. Click on the Open... button in the *Icon-It!* editor window and, in the resulting file selection box, select the template appropriate to the application. Click on the Open button in the file selection box.
4. Back in the *Icon-It!* editor window, click on the Install button (or press Return or Enter). The icon bar will remain on the screen and the *Icon-It!* editor window will close.

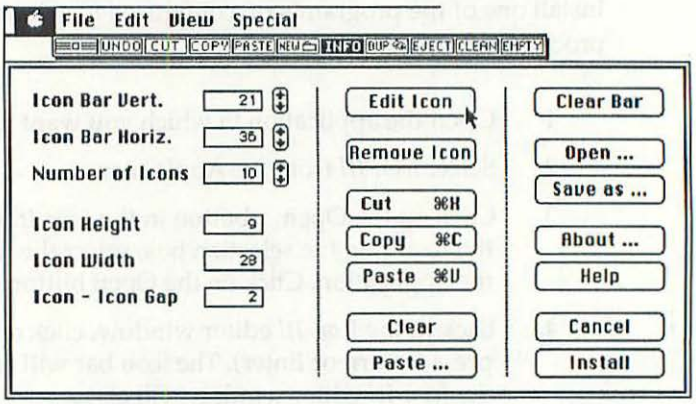
Now every time you start this application (which could be the Finder, or an application, or whatever), the icon bar will appear automatically.

Although the preconfigured icon bars which come with *Icon-It!* are often useful, the real power of *Icon-It!* lies in its Icon Bar Editor. With the Editor, you can modify existing icon bars or create your own from scratch (or from pieces of other icon bars). Although an icon bar can have only a single row of icons, that's about the only constraint on icon bar design. You can specify the size of the icons (all must be of a single, uniform size), their spacing, the number of icons (up to as many as can fit on an icon bar



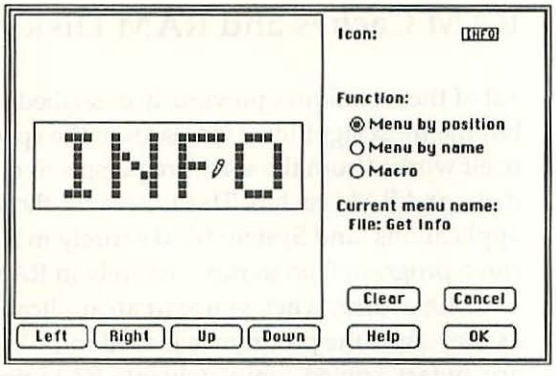
This simple icon bar in the Finder gives you point-and-click access to a number of the Finder's commands.

eighteen inches long, for very wide screens, I guess), and the location of the icon bar on the screen when the application opens (you can move it later if you need to). Each icon graphic can be drawn so it indicates what the icon does with letters, words, pictures, or a combination of the three.



The Icon Bar Editor is the place where the position and general form of the icon bar and its icons are determined.

Of course, assigning functions to your icons is the real heart of the matter. With *Icon-It!*, you can assign either a menu item (a command, a font, a desk accessory) or a keyboard command to an icon. Menu items are earmarked either by name (if the name is unvarying and the position, as in fonts and desk accessories, possibly unpredictable) or by position (if the name changes, as in **Undo** and **Redo**, but the position doesn't). Keyboard commands can be shortcuts already built into the application, or *Tempo II*, *QuicKeys*, *Excel*, or other types of macros.



In the Icon/Function Editor window, you design an icon's appearance and establish its function.

Once the icon bar is designed and installed, it will open each time you launch your application (as long as the *Icon-It!* file is in the System folder). If you need the space, you can click on the icon bar close box to make it go away and later select *Icon-It!* to bring it back. To move the icon bar, you drag it with the mouse while pressing the Command key. If the bar is covering something you want to get to with the mouse, you press the Option key to make the icon bar "transparent." Most importantly, to activate an icon's function, you simply click on the icon.

Naturally, as with any utility of this sort, such as *Tempo II* or *HFS Navigator*, *Icon-It!* is only as useful as you make it. But if you spend a little time analyzing the sorts of tasks you do most often, especially ones you usually perform with the mouse instead of the keyboard, you should quickly get a sense of the kinds of things *Icon-It!* can make more accessible and more efficient.

In some cases, *Icon-It!* merely provides quicker mouse access to commands which are already accessible to mouse-lovers. In other cases, *Icon-It!* makes it possible to invoke a command (or set of commands, using *Tempo II*) with the mouse which would otherwise only be available from the keyboard. Just how and how well it will work for you depends on all sorts of things: whether you use a macro processor such as *Tempo II* or *QuicKeys*, whether you use *FKey Manager*'s *Pop-Keys* menu, the applications and the menu commands you use most frequently, and so on. Since *Icon-It!* tends to make you more, not less, likely to grab for the mouse, I do not recommend it for people who prefer to use the mouse as little as possible or for people who are trying to move in that direction. But if you are a die-hard mouse fan or aspire to that position, then *Icon-It!* might be just the thing for you.

RAM Caches and RAM Disks

All of the techniques previously described will speed up the way you work, but the most significant increases in the speed with which the Macintosh itself works (from the software perspective, at any rate) come from RAM disks and RAM caches. This is because these two types of utilities store your applications (and System files) entirely in RAM, and there's no faster way to run a program than to run it entirely in RAM.

Of course, when you start an application, part of what happens is that, as they say, "the program is loaded into RAM." Although some programs are, in fact, copied completely into RAM right at the very beginning, in most cases only the core part of the program is initially copied from the disk. Less frequently used segments of the program (which can mean anything from a single menu command to an entire 90,000-word dictionary) remain on the disk, awaiting the moment when you decide to use them. Only then are they copied from disk and moved into RAM, a process which can cause a noticeable delay in program operation, especially if you are running the program from a floppy disk.

To make matters even slower, a newly requested program segment may erase the RAM version of a previously requested segment. Thus when you use that previously requested but now erased segment again, it has to be read from disk—again. A smart program would evaluate memory usage, make note of which program segments get called for frequently, avoid flushing anything from RAM except when absolutely necessary, and when necessary, erase only older, rarely used segments. Few programs, however, are that smart.

Desk accessories, of course, also reside on disk until you call for them from the Apple menu, and they (usually) remove themselves from RAM every time you close them. Fonts remain quietly tucked away in the System file until you specifically ask for them; thus each new font you request must be copied from disk before it can be displayed on the screen. In other words, fonts and desk accessories (and FKeys and other parts of the System file) are copied to and erased from RAM, just like the program segments of applications.

The purpose of all this code shuffling is to use as little RAM as possible, in order to leave as much RAM as possible for work space. For those programs which require that their documents be entirely in RAM, for example, this means a larger maximum document size. Programs which do

not require that the entire document be in RAM can have even larger documents, but once they run out of work space, they must copy document segments, as well as program segments, from disk. In the process of editing this chapter, for example, the first part of the chapter was flushed from RAM to make room for this part as I scrolled through the document. If I now scroll back to the beginning of the chapter, my word processor will pause briefly—to save this section to disk and recopy the beginning of the file into RAM—before it displays the opening paragraph on the screen. This strategy of storing pieces of documents, applications, and the System file on disk until you really need them makes possible both large application size and large document size despite a small amount of RAM. But moving all that stuff in and out of RAM, off and on disks, takes its toll, and overall speed can suffer considerably.

For Macintoshes with a small amount of RAM and for those who want to have as many applications as possible open at the same time using MultiFinder, these trade-offs make sense. But for those of us with RAM to spare, our Macs are slowed down unnecessarily. The solution, simply enough, is to put the application, System folder, and any auxiliary files (such as dictionaries, help files, and so on) in RAM and keep them there, despite the programmer's good intentions.

There are two common ways of placing and keeping files and applications in RAM: with a RAM disk or a RAM cache. There are important differences between the two methods, differences in the way they are set up and run, in memory requirements, in the safety of the documents you are working on, and in the options available. Which one of the two methods makes sense for you depends on the amount of RAM you have, your work habits, and just how much of a speed increase you want (or need). In the next two sections we'll look at all of these factors, so you can decide for yourself which type of RAM utility is for you.

By the way, before rumors of the complex nature of RAM caches and RAM disks tempt you to skip this section, I should point out that those rumors apply primarily to the IBM PC side of the fence. Working with RAM caches and RAM disks on a Macintosh is not at all difficult, and it's a far sight easier than it is on a PC. So even if the thought of what goes on in RAM makes you a bit queasy, I suggest that you read this section, at the very least, to get an idea of how RAM caches and disks are used. Even better, give one or the other strategy a try. I suspect you'll be surprised at how easy it is and pleased with the results.

RAM Disks

RAM disk utilities are the brute force method of keeping programs in RAM. The technique they employ is thus simple and rather brainless. A RAM disk program marks off a portion of RAM and fools the computer into thinking that this RAM partition is actually a real, physical disk. This piece of RAM, this illusory, electronic disk, is what is called a RAM disk. From the desktop it looks like a real disk (since it has an icon, just like a physical disk). The Macintosh thinks it is a real disk (since the Macintosh copies to and from a RAM disk, just like a physical disk). Indeed, except for the fact that there is no physical object which you can hold in your hand, it acts like a disk. But it is, in fact, nothing more than a slice of RAM pretending to be physical disk.

So what's the big deal about a RAM disk? If you copy an application to a RAM disk and run it from there, any copying of program segments from the RAM disk into active RAM will occur at RAM speed, which is far faster than the speed with which a real disk drive mechanically reads the information from a floppy or hard disk and then copies it into RAM. The less often the computer needs to mechanically read or write to a physical disk, the faster the computer will run. Ideally, if you were to copy all the software you use onto a RAM disk (applications, System files, downloadable fonts, and so on), then there would be no reason ever to (mechanically) read a hard or floppy disk, and everything would take place at RAM (electronic) speed. Although, as we'll see, this ideal is rarely achieved, RAM disks are, in most instances, the utility of choice for those who want raw speed.

To see what all this theorizing means in a practical sense, consider for a moment, the setup and use of a typical RAM disk. Although you can automate the setup of a RAM disk fairly easily (see the discussion of automated startups in the earlier part of this chapter and the section on using RAM disks that follows), we will for the moment perform each step of the process by hand, in order to see just what a RAM disk is all about. For this example, let's assume that your standard work arrangement is *Word* and *SuperPaint* running under MultiFinder, and that your Macintosh possesses a full complement of four megabytes of RAM (we'll consider the significance of the latter assumption shortly).

After turning on your hard disk and Macintosh (with MultiFinder turned on), rather than launching *Word* and *SuperPaint*, you launch your RAM disk utility instead. You instruct it to create a RAM disk large enough

to hold your System folder, the two applications, plus a little extra RAM for the application's temporary files (and for good measure, as well). Note that you need to know how large the RAM disk should be before you can create it. In this case, you instruct the program to make a two-megabyte RAM disk. When your RAM disk program is finished and you return to the desktop, there is now a RAM disk icon (usually the standard floppy disk icon named `RAM Disk`) below the icon for your master System disk—your hard disk, for example. If you double-click on the RAM disk icon, the RAM disk window opens, revealing two-megabytes of unused space.

Now, you copy the files and folders—System folder, *Word*, and *SuperPaint*—from your hard disk to the RAM disk, using the standard copying method. Then you make the RAM disk the startup disk (and thus the System file on the RAM disk the active System file) by double-clicking on the RAM disk's Finder while holding down both Command and Option keys. Finally, you launch *Word* and *SuperPaint* (making sure to run the applications on the RAM disk, not those on the hard disk) and go to work. All your applications are in RAM, all your desk accessories and fonts are in RAM, everything is in RAM, so everything is going as fast as is "computerly" possible. (Well almost everything is going as fast as it can, as you'll see.)

Now if you've never seen a RAM disk in action, you're in for a pleasant surprise. With the applications and the System file in a RAM disk, needed program segments are copied to active RAM in a flash. Menu selections which used to just chug along (because they had to be read from disk first) are now instantaneous. Applications open and close almost in the blink of an eye. Scrolling in long word processing files is speeded up considerably and calculations on large spreadsheets are greatly improved as well (although the amount of improvement depends upon the application, the location of your documents, and the application's temporary files). If it's raw speed you want, you can't do much better than this.

RAM Disk Pros and Cons Although the setup process may sound complex, it really isn't difficult, since most RAM disk programs automate the entire procedure so that it runs by itself every time you turn on your Macintosh. But RAM disks are not a universal panacea. They have their drawbacks as well as their virtues. The following list highlights both the good and the bad. When considering ways to speed up your Mac, you should consider carefully the points in this list in relation to your needs and

your Macintosh's free RAM, and compare it to the list of pros and cons of RAM cache programs later in this chapter.

First the good points:

- *Set it and forget it.* Assuming you use a more or less predictable collection of applications, once you get your RAM disk utility configured the way you want it, you never need to think about it again. From that point on, if you make it the startup application or part of an automated startup routine, a good RAM disk utility will automatically create the RAM disk, copy the desired files and folders to the RAM disk, and make the RAM disk the startup disk (if the System folder is copied to the RAM disk).

- *No disk reads, ever.* Well, almost never. If you're running an application from a RAM disk, the entire program is in RAM. Thus the program will never need to read a program segment from a real (and hence slow) disk. If you can fit your System folder onto the RAM disk, then it will never need to read desk accessory or font or FKey code from a real (slow) disk. But if you store your documents on a hard or floppy disk (and you should), it may need to read document segments from a real (slow) disk. For more about that, refer to the drawbacks that follow.

- *It's fast.* About as fast as they get. The only way to make it faster is to change the microprocessor, add a co-processor, or buy a faster program, or get a VAX.

- *A RAM disk is a RAM disk.* RAM disk programs vary in the way in which they automate the setup procedure. Some RAM disk programs are easier to use; others may contain a greater variety of options. But concerning the RAM disk itself, there are no real differences. One RAM disk functions more or less the same as any other (with one important exception: the degree to which they are protected from accidental trashing). There are few surprises. In contrast, the same cannot be said for RAM caches.

- *You can see what you're doing.* Because a RAM disk is considered by the computer to be a real (physical) disk, it has an icon on the desktop. You can open the icon's window, see what is stored there, copy files onto it, or throw some or all of its contents into the trash. In short, it has the "look and feel" of a physical disk, and

for this reason, you can monitor what your RAM disk is doing as easily as you can monitor any other disk.

Now for the not so good points:

- *RAM disks waste RAM.* I never use mail merge or footnotes, even though they are built-in features of my word processor. But like it or not, I've got to copy the entire program onto my RAM disk, mail merge, footnotes, and all. In fact, every program on my RAM disk contains elements that I never use. Thus there is a significant fraction of my RAM disk which sits idly by. This space could have been used for other programs or as active RAM to accommodate larger documents. It is a waste of good RAM, and there's no way to avoid it with a RAM disk.
- *The System folder should be on the RAM disk.* Applications often call upon segments of the System file; fonts and desk accessories are the most obvious examples, but there are others as well. If you are opening and closing applications (without using a transfer utility or a Finder substitute) or using MultiFinder, then the Finder must be readily available as well. If you use downloadable LaserWriter fonts and rely on automatic downloading, the PostScript font files must be stored in the System folder. These are some of the more obvious reasons why, to take full advantage of a RAM disk's speed, the System folder must be on the RAM disk (and the RAM disk must be made the startup disk). The problem is that System folders can consume and waste a lot of valuable RAM.
- *RAM disks are inflexible.* As I pointed out earlier, the size of the RAM disk must be set when the RAM disk is first installed. If you make it just large enough to contain the programs you intend to run, then you won't be able to add other programs should you wish to later (unless, of course, you erase something else or create a second RAM disk, which some RAM disk programs cannot do). If you make the RAM disk large enough for potential additions, you waste all that RAM until you find an application you want to add.
- *RAM disks are ephemeral.* Unlike floppy and hard disks, a RAM disk is not permanent (or as permanent as a floppy or hard disk can be). Turn off your Mac and it's gone, for good. Poof. If your

application bombs and you have to reset or restart your Macintosh, the RAM disk probably won't be there when you get back. As a result, RAM disks are unsuitable places for storing the documents you're working on. When you perform an intermediate save, you should always save to a real disk, a physical disk that spins and works with read/write heads and doesn't depend on a power supply to maintain its memory from one minute to the next. If you ignore this recommendation, you run the serious risk of irretrievably losing data when you least expect it. If you follow this common sense advice, the result is the problem discussed in the next paragraph.

- *Despite everything, applications still have to read data from real disks.* If you store your documents on a floppy or hard disk (where else?), your application will, of course, have to read from a physical disk when you first open the document (and possibly at other times as well) and write to the disk whenever you save (which should be often). There are also other occasions when an application may need to read the data on a real disk. For example, your application may create temporary files, and those temporary files may end up on a physical disk instead of the RAM disk. These mechanical reads and writes are, unfortunately, unavoidable.
- *Some copy-protected programs won't run on a RAM disk.* One of the flies in the RAM disk ointment is copy protection. If one or more of your favorite applications is copy protected, and there is no way to bypass or subvert the protection scheme (see Chapter 8), then a RAM disk is not likely to help you. Simply put, this is because you won't be able to copy your applications onto the RAM disk.

RAM Disk Strategies Although RAM disks are not perfect, there are many situations in which a RAM disk makes good sense. To figure out if they make sense for you, you need to consider, in addition to the points in the previous section, your work habits in relation to the amount of RAM you have available for a RAM disk, and the sizes of your System folder, applications, and their associated files (such as dictionaries, help files, etc.). What follows are discussions of strategies appropriate to four different internal RAM configurations. In each section, different working strategies are compared in terms of their ability to take advantage of a RAM disk.

Even if you currently have only one or two megabytes of RAM, I still recommend that you read the other strategies, if only to find out what life could be like.

One Megabyte of RAM No matter how you look at it, it is a tight squeeze with only one megabyte of RAM. Even a RAM disk containing only a fairly modest System folder can eat up a third or a half of all your RAM. Add just a single, large application to your RAM disk and you could be, as the saying goes, all dressed up with nowhere to go.

Unless you use one program all day long, there really aren't any good RAM disk strategies that fit within a single megabyte. (It is, in fact, in such tight memory situations that RAM caches make the most sense.) But for those who want to enjoy at least some of the benefits of a RAM disk, the following are a few simple strategies you can use until you have more memory to work with.

- If you really do use just one application all day long, then it is possible, if the application is not monster-size, to create a RAM disk containing a stripped down System folder and your application, and still have enough RAM left over to run the application and its documents. For example, you could have a RAM disk containing *MacWrite*, the ImageWriter file, the Finder (or a Finder substitute, such as *PowerStation*), and a System file with only the most essential desk accessories and fonts. That should leave you with enough room to run *MacWrite* and a large document. You could run other applications as well, but only one at a time.
- If you use a number of applications during a single work session (and most of us do, I think), there is little likelihood that you can fit all the applications and the System folder onto a RAM disk and have enough room left over for the Mac to work in. Instead, you might consider putting just the System folder on a RAM disk. This will speed up your access to fonts, FKeys, and desk accessories (and other, more esoteric System resources) and speed up the Finder if you are using it. But since you won't see any improvement in application speed, unless you are a heavy user of fonts and desk accessories, this strategy is not a very useful one.
- Alternatively, you could place your most frequently used applications (if they are slim enough) on a RAM disk and leave your System folder on a floppy or hard disk. Your applications will

launch and quit much faster this way, and they will need to access real disks much less often than they might otherwise. On the other hand, fonts and desk accessories will still have to be read from a real disk, and the Finder will open and close as lethargically as usual. This, however, is probably the best alternative for one-megabyte Macs. It is, at the very least, worth trying to see if it improves the performance of your applications.

- If you want to use MultiFinder (or *Switcher* or *Servant*, for that matter) and run one or two or more applications at once, a RAM disk in addition is nearly impossible in a single megabyte of RAM. Indeed, if MultiFinder's advantages are important to you, you might as well appreciate the benefits of MultiFinder and forget about the RAM disk. Even with the entire megabyte of RAM available, running just two applications under MultiFinder can require the electronic equivalent of a trash compactor.

Two to 2.5 Megabytes of RAM With 2 or 2.5 megabytes, you've got some room for maneuvering. While you can't afford to be profligate, neither do you need to strip your System file and folder to bare skeletons to fit them on a RAM disk. You should have room for MultiFinder and one or two applications as well. The following are some of the possibilities.

- If you are just a one- or two-application person, you should be able to fit a System folder and your one or two applications on a one-megabyte (more or less) RAM disk, although if one or both of those applications is a big one such as *HyperCard*, or requires sizable help files or dictionaries, you might have some difficulty doing so. That would leave you with 1 or 1.5 megabytes of RAM, enough to run both applications under MultiFinder in some comfort.
- If you like to have four or five or six applications at your fingertips, then 2.5 megabytes will require some compromises. You might, for example, choose to put your System folder and all your favorite applications on the RAM disk. That, of course, would shrink working RAM to 1 megabyte, or even less—although you shouldn't let it drop below this level if you use memory-hungry programs like *PageMaker*, *SuperPaint*, or *HyperCard*. In this situation, you would probably keep only one application open at a

time and use a Finder substitute or a transfer utility (as described in Chapter 6) to switch from one application to another. But because your applications are all stored on the RAM disk, quitting one and opening another can be accomplished in just a second or two.

- Another strategy for those who like to use many applications is to place your System file and only the most important one or two applications on the RAM disk. As with the first strategy, this should result in a RAM disk of about 1 megabyte and free RAM of about 1.5 megabytes. Using MultiFinder, for example, you could then keep the one or two applications on the RAM disk open at all times and keep enough free RAM for one additional application. (If there's not enough RAM left over for another application, cut back to having one open all the time. If there is plenty of room, you could try keeping a third application open all the time as well.) The additional application slot can be used as needed to run first this application, then that one, from your hard disk or floppy disks. They won't run as speedily as the applications on your RAM disk, and quitting one and launching another will also be slower. But the additional memory left free for another application provides the flexibility to launch any application, even while your major one or two programs continue to run at all times—and from the RAM disk.

Four or Five Megabytes of RAM You would think that the sky's the limit with four or five megabytes of RAM. Unfortunately, there is a limit even with this much RAM, though it can take some effort to reach it.

The obvious strategy is to create a RAM disk large enough to hold your System folder (which can be a rather full one) and your major applications, up to about two megabytes worth. You would then run, using MultiFinder, whatever combination of the programs on your RAM disk as you felt were useful at the time. You would also keep enough RAM free so you could add (and later subtract) other applications as needed.

The only problem with this simple-sounding strategy is that applications (and system files) are getting larger all the time. Programs such as *HyperCard* and others can require as much as one megabyte all by themselves. Thus any strategy that reduces your working RAM to two or three megabytes will seriously limit your ability to keep a number of programs

running under MultiFinder if those programs include *HyperCard* and other memory hogs.

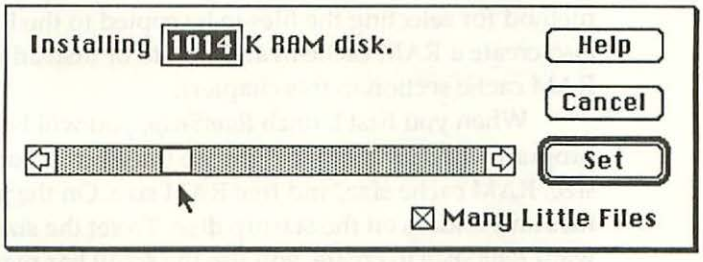
If you find that the applications you want to use together all require big allotments of RAM, then even four or five megabytes may not provide enough room for a fully loaded RAM disk (one that contains a System folder and your most frequently used applications) and adequate working RAM in which to run your applications. At the time this book was written, few people were likely to find themselves in this position. But the computer world moves quickly, and by the time you read this, quite a few more may be bumping into the four- or five-megabyte ceiling.

If you are one of these people, refer back to the strategies in the two-megabyte section and adapt those strategies to your situation. A healthy System folder and a few memory-hungry applications, for example, may be all you want to or can fit on your RAM disk. You could then use *HyperCard* or a transfer utility such as *SkipFinder* to switch quickly from one application to another (and by using this method, save space by not having to copy the Finder onto the RAM disk). On the other hand, if you want to run several memory-intensive applications under MultiFinder, you may find that you can only put a small System folder or a small number of applications on the RAM disk, but not both.

More Than Five Megabytes of RAM Some day, eight megabytes will be the equivalent of what 128K is today. But at least for now, eight megabytes is enough to satisfy almost everyone. A three-megabyte RAM disk can accommodate a very large System folder and quite a few applications (unless you have some very large and very specialized programs). That will still leave you with five megabytes with which to run MultiFinder and a more-than-adequate number of applications.

You don't really need a strategy for dealing with this much memory. Again, unless you have one or more very large and very specialized applications, you can afford to waste some and maybe even a lot of this RAM. With one, two and one-half, four, or even five megabytes, a RAM cache may be a better bet than a RAM disk because it is less wasteful, and because it requires fewer compromises (as we'll see). But with eight megabytes, wasted RAM isn't much of an issue (today—ask me again next year), and a three-megabyte RAM disk is big enough to hold most anything. Thus if you want to dream about the ultimate setup, with a RAM disk, MultiFinder, and a half-dozen applications all open at once, eight megabytes is the stuff that dreams are made of.

Using a RAM Disk Program One of the most popular RAM disk programs is a public-exchange program called *RamStart* (public-exchange). It works in two different ways, which I call folder mode and script mode. In either mode, the size of the RAM disk to be created is stored in the program and set, or reset, by quickly clicking on the RAM disk scroll bar (in the program's only window) as *RamStart* begins its work.



In RamStart, *you can set the size of your RAM disk with the scroll bar or by typing the appropriate number.*

In folder mode, after *RamStart* creates the RAM disk, it copies all the files in the same folder as *RamStart* (but not *RamStart* itself, nor any folders within the folder) to the RAM disk. Then if the System and Finder were copied to the RAM disk, it ejects the startup disk (if it was a floppy) and switches control to the System file on the RAM disk.

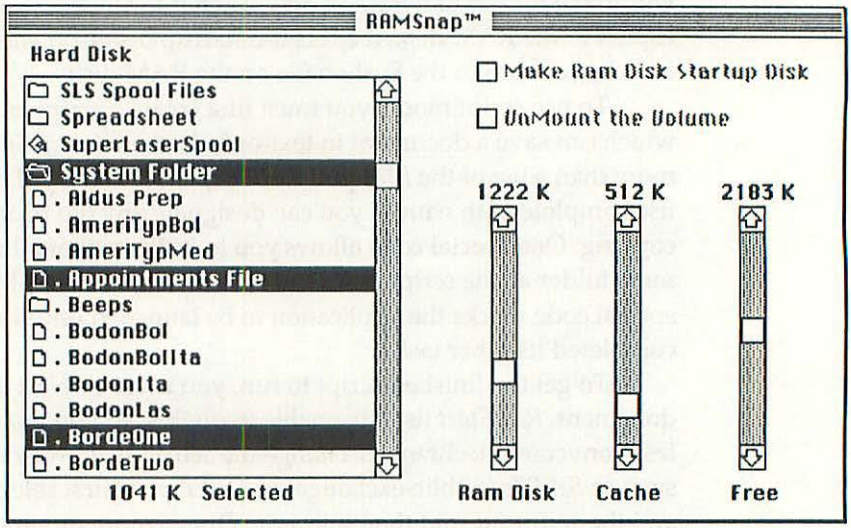
To use script mode, you must first create a script in a word processor which can save a document in text-only format. A *RamStart* script is little more than a list of the files you want copied to the RAM disk. Since you can use complete path names, you can designate any file in any folder for copying. One special code allows you to indicate that all the files in the same folder as the script file should be copied to the RAM disk. Another special code marks the application to be launched once *RamStart* has completed its other tasks.

To get the finished script to run, you must convert it to a *RamStart* document. *RamStart* itself is unable to do this, so you must use one of two less-convenient techniques: change the script file's creator with a program such as *Set File* (public-exchange) or *DiskTop*, or first select both *RamStart* and the script file and then select the **Open** command in the Finder's file menu to get *RamStart* to run the file and convert it to a *RamStart* file in the process. Once the conversion is complete, the script will display a *RamStart*

icon, and you can run the script simply by double-clicking on it. If you want *RamStart* to run a script automatically, you must put a copy of the script in *RamStart* itself with a program such as *MDS Edit*.

Despite its popularity, *RamStart* is, as the previous paragraph demonstrates, not as simple to use as it could be. One of the easiest RAM disk programs to use is *RamSnap* (Dove Computer). It contains all the features a good RAM disk program should have, and it provides a relatively easy method for selecting the files to be copied to the RAM disk. *RamSnap* can also create a RAM cache in addition to or instead of a RAM disk (see the RAM cache section in this chapter).

When you first launch *RamSnap*, you will be greeted with the main program window. On the right are three scroll bars, one each for RAM disk size, RAM cache size, and free RAM size. On the left is a scrolling list of the files and folders on the startup disk. To set the size of the RAM disk you want *RamSnap* to create, you use the scroll bar marked Ram Disk. The size you have selected is displayed at the top of the bar, and the amount of free RAM remaining (which appears at the top of the far right scroll bar) adjusts accordingly to give you a clear picture of how your Macintosh's RAM will be divided.



RamSnap's *main window allows you to set the size of your RAM disk and/or RAM cache, and to select the files to be copied to the RAM disk.*

The file selection box on the left does not work quite like the usual file list. To select the files to be copied to the RAM disk, scroll through the files and folders in the list, and click on the name of each file you want copied (you do not need to Shift-click to make multiple selections). If a file is in a folder, double-click on the folder to open it; the names of the files and folders it contains will be added to the list, each preceded by a period. You can select any combination of folders and files simply by clicking on the appropriate names. As you make your selections, a running tally of the total size of all the files is displayed below the list box. Use this total to make sure that you have enough room on your RAM disk for all the files you have selected.

Once the desired files are selected and the RAM disk size is correct, you have three other options to decide on: if you want *RamSnap* to transfer to an application when it quits, locate the application in the file selection box and click on the icon to the left of the name. The icon will change to an arrow to indicate that this is the "launch application." The two other options are activated with check boxes on the upper right and are only available if the System and Finder are copied to the RAM disk. When checked, the top one will make the RAM disk the startup disk. If you started with a floppy disk, the second option will eject the floppy disk ("unmount the volume" is the term used by *RamSnap*).

To save your selections, you use the **Save Set...** command from the File menu, which will create a file containing the names and locations of the files to be copied, the size of the RAM disk, and the other options selected. To have *RamSnap* actually create the RAM disk, copy files, and so on, you use one of the two **Install** commands: one uses the current settings on the screen, the other uses the settings in a previously saved file. But you don't actually have to open *RamSnap* and select **Install** to create a RAM disk and copy a set of files. All you need to do is name the set containing the appropriate directions `SnapStart` and make sure that *RamSnap* and `SnapStart` are both in the same folder or on the desktop (not in any folder), or that `SnapStart` is in the System folder. Then, if you double-click on *RamSnap*, set it as the startup application, or use it as part of a startup routine, *RamSnap* will automatically install a RAM disk and copy files to it, all according to the instructions in `SnapStart`.

My one gripe about *RamSnap* is that is does not allow you to edit a previously saved set of instructions. If you discover that you need to alter an existing *RamSnap* set, even just slightly, you have to recreate the whole

thing from scratch. Other than that, however, I find *RamSnap* to be about the best and certainly the easiest to use RAM disk program I've seen.

What to Look for in a RAM Disk Program Should you decide to implement a RAM disk strategy, there are many RAM disk utilities to pick from, some of them commercial, some shareware, some free for the asking. Choosing among them can be difficult, however. To help you select the right one, the following is my list of the features every good RAM disk program should have.

- *Variable RAM disk size.* Being able to choose the size RAM disk you want to create is an essential feature for any RAM disk program. Fortunately, I can't think of a RAM disk program that doesn't have this ability, so there's not much that needs to be said about that.
- *RAM disk lock.* Unless special measures are taken by the programmer, it is possible to throw a RAM disk in the trash and, as a result, have it disappear completely and forever. Thus the RAM disk that the program creates should be protected from accidental erasure. Good RAM disk programs lock their RAM disks so that they cannot be thrown in the trash; they simply bounce back to their original position if you accidentally drag them there. This small bit of thoughtfulness on the programmer's part can save you lots of heartache.
- *Works with a hard disk.* This one is so obvious it hardly bears repeating. You should always be sure a utility such as this runs from a hard disk. Even if you don't own one today, you may tomorrow, and you'll surely want your software to run on it. By running on a hard disk, I don't just mean that you should be able to copy the program to your hard disk and still expect it to run. It should also be able to perform its startup routine (see the following paragraph) from the hard disk, and be able to locate files anywhere on the hard disk and copy them to the RAM disk. You should not have to resort to putting the files you want copied to your RAM disk onto a floppy so that the program can find them. Nor should you have to put all the files you want copied to the RAM disk in the same folder on your hard disk. Wherever your files are, your RAM disk program should be able to find them.

- *Automated procedures.* A good RAM disk program should be able to create a RAM disk of a previously designated size and copy a set of previously selected files to the RAM disk automatically. Ideally, you should be able to store such automated procedures as separate files, so you can create a number of them and, depending upon the needs of the moment, choose the one you want to run. If you set the RAM disk program as the startup application or run it as part of a more complex startup procedure, there should also be a mechanism by which a specially designated set can be run automatically.

- *Makes RAM disk the startup disk.* If the RAM disk program has copied the System folder, or at least the System file and Finder, to the RAM disk, it should be able to switch control from the System on your hard or floppy disk to the System on the RAM disk (the RAM disk, in other words, should now appear as the startup disk in the top, left corner of the screen). If it cannot do this, then you'll have to do so manually to enjoy the benefits of having the System and Finder in RAM. If you are working from a floppy setup, the RAM disk program should also be able to eject the original startup disk from the drive, once control has switched to the other System, in expectation of the insertion of a document disk.

- *Launches an application when done.* Once the RAM disk program has completed its job (creating the RAM disk, copying files to it, and, if desired, switching control to the System file on the RAM disk), it should be able to launch one of the applications on the RAM disk as its very last task. In other words, it should have a kind of built-in transfer utility which you can preset, along with all the other parameters, to launch a specific application when all the program's other tasks have been completed.

RAM Caches

If using a RAM disk is the brute-force way of keeping your application in RAM, then a RAM cache program is the effete, intellectual way of doing the same thing. Like a RAM disk, a RAM cache is a RAM partition, set up, in this case, by a RAM cache program. Unlike a RAM disk, however, a RAM cache is invisible. There is no icon for it on the desktop, no sign that it exists.

Since it is invisible, you can't copy programs to it, as you would with a RAM disk. Instead, the RAM cache program runs in the background and carefully controls the contents of the cache, copying and deleting bits and pieces of program code as it sees fit.

If the cache is big enough, the cache program simply adds pieces of code to the cache at the same time they are copied from disk and added to working RAM. These pieces of code can be the core portions of an application (the part which is placed in RAM when you launch the program), or small program segments which are requested later by the application itself.

But that's not all a cache program does. Equally important is the fact that it intercepts subsequent requests for program segments and answers the requests with code from the RAM cache, rather than from the physical disk. In this way, program segments are read from a real (slow) disk only once and from the (fast) RAM cache every time thereafter. Thus once an application is launched, once each of its commands is selected just one time, that application will run as fast as if it were on a RAM disk.

If only a small amount of RAM is allocated for the cache, then the cache program must perform yet a third task: make room for newly requested program segments by erasing others. As you use an application and the application calls up program segments which are stored on disk, the cache program makes note of which segments are needed most often. Using this information, the cache program is able to discard rarely used program segments to make room for new requests. Commonly used segments are erased only as a last resort. Because it keeps track of your work habits, even with a small amount of RAM allocated, RAM cache programs can dramatically reduce the number of disk reads, if not eliminate them entirely.

Keeping in mind, for comparison's sake, the RAM disk startup procedure, consider an example of manually starting and using a RAM cache. (As with a RAM disk, the creation of a RAM cache is easily automated, but for now we'll look at the individual steps.) In this example, we'll assume the same work habits as those in the RAM disk illustration: that your standard work arrangement is *Word* and *SuperPaint* running under MultiFinder, and that your Macintosh possesses four megabytes of RAM. We'll also assume that *PowerDisk* (Software Power Company) is your RAM cache utility.

After turning on your hard disk and Macintosh, you go to the Apple menu and select *PowerDisk* from among your desk accessories. When the *PowerDisk* window appears, you use the scroll bar to set the size of the RAM cache and click on the On radio button. Then you click on the *PowerDisk*

close box. That's all there is to setting up the RAM cache—on your end, at least. Now the RAM cache program starts its job.

Back at the Finder (with MultiFinder turned on), you launch *Word* and *SuperPaint* and go to work. Immediately, the RAM cache program starts storing program code in the cache, watching your usage habits and monitoring the total capacity of the cache. At first (during the period when the cache first stores called-for program segments), your applications will seem no faster than before. But soon you will notice that things are happening much more quickly. Menu commands that used to pause before opening a dialog box now display their dialog boxes instantly. Desk accessories open much more quickly. Often, even menus themselves are more responsive. (If you were not using MultiFinder, you'd notice that transferring from one application to another is much faster, if your cache is a large one.)

Although they usually take a while to get up to speed, RAM caches can, under ideal circumstances, be just as fast as RAM disks once they get going. In less than ideal circumstances, they can still allow the Mac to run a good deal faster than it would without a RAM cache, although slightly slower than it would with a RAM disk. The major consideration in the choice between RAM disk and RAM cache is whether or not you have enough RAM to accommodate an effective RAM disk. But there are other issues that factor into the decision as well; these are the topic of the following section.

RAM Cache Pros and Cons As with RAM disks, RAM caches can be set up to start automatically when you turn on your Mac. Thus the configuration procedure just described need only be done once, assuming your work habits do not change dramatically from one day to the next. In most other ways, however, RAM disks and RAM caches are quite different. To see just how different they are, compare the following list of good and bad points with the RAM disk list that appears earlier in this chapter.

The good points:

- *Set it and forget it. Really forget it.* RAM caches work in the background. Once they are started and configured, everything a RAM cache does occurs behind the scenes. There is no disk icon on the desktop which can be thrown in the trash by accident. You don't have to think about which copy of an application you are using (the one on the real disk versus the one on the RAM disk, for

example). You don't have to alter your work habits at all, or think about the RAM cache ever again. Once you set it up, you can forget it's even there.

- *Set it and then change it.* A good RAM cache can be set to a new cache size in the middle of your work (unless memory constraints prevent it). If you're not getting the performance increase you're looking for, for example, and have the extra RAM, you can enlarge the cache to make it more effective. RAM caches are thus much more flexible than RAM disks, which can only be reset by erasing the RAM disk and starting over.

- *RAM caches do not waste space.* As long as the appropriate RAM cache size is designated during the initial setup, a RAM cache is the most efficient way to use RAM. A RAM cache will not keep in RAM those parts of an application you never use. It will store only those parts of the System file that you or your application actually need. All the rest is left on disk, where it belongs. Especially considering the amount of room a System folder can occupy, this means that compared to a RAM disk, a tremendous amount of RAM remains available for you to use in other ways and is not tied up in the storage of never-used program segments.

- *RAM caches are often safer.* As previously noted, there is an element of risk involved with using a RAM disk: you can lose files stored (unwisely) on a RAM disk if an application bombs or the power goes out. In some RAM disks, you can throw the entire RAM disk in the trash by mistake. But since RAM caches are essentially invisible, you can't throw them in the trash accidently, and as long as your RAM cache is of the write-through variety (more on this in a moment), all your saves will always be safely written to a real disk. For these reasons, RAM caches can be safer to use than RAM disks, especially for people who are a bit careless to begin with.

- *Copy protection is usually not a problem.* A big plus in favor of RAM caches is that they usually work just as well with copy-protected software as they do with programs lacking copy protection. Since a RAM cache's job is to keep in RAM only what the application has put in RAM already, there is no need to subvert copy protection. Thus a RAM cache will work with many copy-protected

programs which can't be used with a RAM disk. So if some of your favorite applications are copy protected, and you have no way to get around the copy protection, a RAM cache is really the only game in town. (There are exceptions to this generalization, however. Many of the exceptions are games; so if you are in a playful mood, remember to turn the cache off.)

The bad points:

- *A RAM cache is not as fast as a RAM disk.* Because of the way a RAM cache works, every piece of program code must first be read from a real disk during normal program operation (not in advance, as in a RAM disk). Also, if the RAM cache is small and rarely used program segments must be erased to make room for new segments, some program segments may be read from a real disk a number of times. Because these mechanical disk reads are unavoidable, RAM caches are not as fast as RAM disks, especially during the initial fifteen or twenty minutes that you use an application.
- *A RAM cache is not as bombproof as a RAM disk.* Macintoshes seldom crash because there is a RAM disk tucked away somewhere. They do sometimes crash because there is a RAM cache tucked away somewhere. This is because RAM caches are not static entities but instead work away in the background, watching everything you do and frequently intercepting the messages that pass between disk drive and RAM. RAM cache programs are manipulative. They alter the way things are supposed to be, and on rare occasions, they make a fatal mistake, or the program they are manipulating protests too much. In either event, the System crashes and you've lost some data. Some RAM cache programs are more prone to this problem than others. Some RAM cache programs have sterling records and may never have problems of this sort. But you need to be aware that such things are possible.
- *Not all RAM caches are alike.* As with most other types of applications, but unlike RAM disks, all RAM caches are not created equal. Different program designers have different ideas about how RAM caches should work, and the finished products show just how

divergent these views can be. As a result, you have to be much more careful in selecting a RAM cache. If you aren't careful or just plain make a mistake, you can end up with a RAM cache program that is inappropriate for you, your hardware, and your way of working.

- *You can't see what it's doing.* Because the RAM cache itself is invisible, and because the cache program works behind the scenes, there is no way to really see what is going on. You should be able to tell that it's working simply from the fact that your applications are running faster. (Some RAM caches also display a flashing icon in the menu bar to show that the program is paying attention.) You may be able to tell just how well it is working by checking the statistics section of the cache program. But, because they lack the tangible, visible presence that RAM disks have, RAM caches don't provide the same level of direct feedback.

  More important, however, is the fact that you may not be able to tell whether or not your cache is a write-through cache, and thus whether or not your saved files go directly to disk. To understand the concept of a write-through cache, we need a little background first. The fastest RAM caches minimize disk writes as well as disk reads. In other words, they save all disk writes (as well as disk reads) in the RAM cache until you close the document you are working on, or until the RAM cache runs out of space, or until some preselected milestone (time, amount of information to be saved, etc.) is passed. Only then does the cache program save the document to disk.

  Although this makes for greater speed (since both disk reads and writes are reduced), it is a dangerous habit, as dangerous as saving documents to a RAM disk. To avoid this risky situation, the RAM cache program must be a write-through cache or must have an option which allows a write-through mode of operation. Write-through simply means that all disk writes pass directly through the cache and onto a physical disk. Some caches work only in write-through mode (and thus don't cache disk writes at all), some work only in write-to-cache mode (and thus always cache disk writes), and some can work either way, depending on how you configure them.

I strongly advise that you use a cache that is a write-through cache or always set yours to be in write-through mode. Your data will be much more secure that way. The problem is, you can't always tell if a RAM cache is write-through or not. You certainly can't tell just by looking at the screen (or by listening, if you have a quiet hard disk and work in a noisy environment), and some manuals, such as Apple's, don't tell you either.

- *RAM caches are more expensive.* There are good RAM disk programs available as shareware or even for free. But RAM caches are pretty much a commercial software phenomenon, and thus they tend to cost more. The Macintosh does have a RAM cache built into the Control Panel, but for an important reason I cover shortly, I cannot recommend that you use it. Your only alternative is thus to go out and buy one.

Optimizing RAM Cache Performance How much RAM do you need to use a RAM cache? How do you get the most from a RAM cache? What size cache should you use? The answer to the first question is that you can use a RAM cache with virtually any amount of RAM. If you have only a small amount of RAM, you will probably end up with a very small cache. But you can still use a cache, and it will still be of some benefit.

As for the answers to the second and third questions, the important criteria—how much RAM you have, your work habits, your favorite applications, and so on—vary from one person to the next. The best way to answer these questions is to use a RAM cache with the programs you regularly work with and monitor its performance. Of course, such a process requires a RAM cache program that can provide you with some statistics about its operation, which is why I recommend (in the following section) that you use a RAM cache utility that can keep track of its performance and report back on how it's doing.

PowerDisk (Software Power Company) offers the Macintosh owner a wide variety of up-to-the-minute statistics and is one of the best RAM cache programs available. Thus in the paragraphs that follow, I use it to illustrate how to analyze and optimize RAM cache performance.

(Note that the descriptions that follow refer to *PowerDisk* 1.5, the version available when this was written, and that *PowerDisk* 1.5 is not Multifinder compatible. Version 2.0 of *PowerDisk*, which should be available by the time you read this, will make the program MultiFinder-compatible.

According to the manufacturer, other changes will include increased speed, elimination of the optional dynamic cache, described in the following paragraphs, and a choice of either Control Panel or desk accessory format. The desk accessory will be a standard Font/DA Mover file and thus, unlike the version I'm currently using, will not require installation via a special installer program.)

For those of you who are using RAM cache programs other than *PowerDisk*, your program probably provides some, if not all, of the statistics described in the following example. Use your manual to help you translate if the terminology used here is not quite the same as that used by your program. In terms of features, the most likely difference between your cache program and *PowerDisk* is the lack of a second dynamic cache (see following description). You should have few problems, however, adapting the following optimization process to any good-quality RAM cache utility.

Now let's go step-by-step through the process of optimizing RAM cache performance for your combination of RAM size, work habits, and applications.

1. The first step is to configure your RAM cache for the initial test flight. To do this in *PowerDisk*, you use the Control Panel which appears when you select *PowerDisk* from the Apple menu.

   a. First set the size of the RAM cache. In the upper left corner of the Control Panel is the horizontal scroll bar which you use to set the size of *PowerDisk*'s permanent cache. The size of this cache is fixed (unchanging, unless you reset it from the Control Panel) and "immortal," which is to say, it is not erased when you quit an application but continues to store code and analyze your needs throughout your work session.

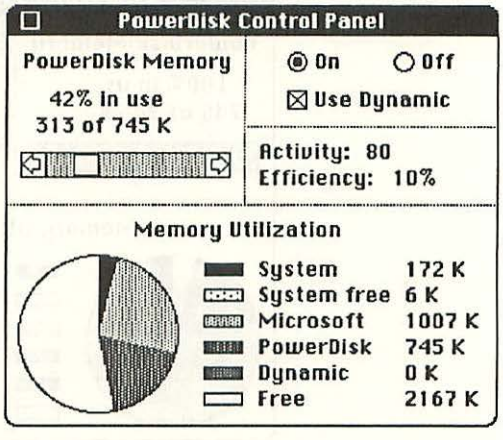
   Of course, the point of this whole process is to set the size of the permanent cache to something that makes sense for you and the way you work. Thus at this point, we can only guess. For starters you might try about 300K for each megabyte of RAM in your machine.

   b. *PowerDisk* is unusual in that it also has a second cache, the dynamic cache. This cache only stores code for the application that is currently running (it is not permanent). It adjusts in size according to the needs of the moment and the free RAM available (it is not fixed in size). You may turn the dynamic

cache on and off with the Use Dynamic check box; for now, click the check box to turn it on (if it is not already on).

c. Turn the disk cache on. In the upper right corner of *PowerDisk*'s Control Panel, click on the **On** radio button (if it is not already on).

d. Finally, make sure that the cache is configured as a write-through cache. Open the *PowerDisk* menu and check to make sure that Delay Writes... is not checked. If it is checked, select Delay Writes..., click on the Write Immediately button in the Delay Writes dialog box, and click on the OK button. The Write Immediately option is just another name for write-through. As I noted earlier, I strongly advise that you always operate your cache in write-through mode.



When you first start using PowerDisk, *cache utilization and efficiency will be very low.*

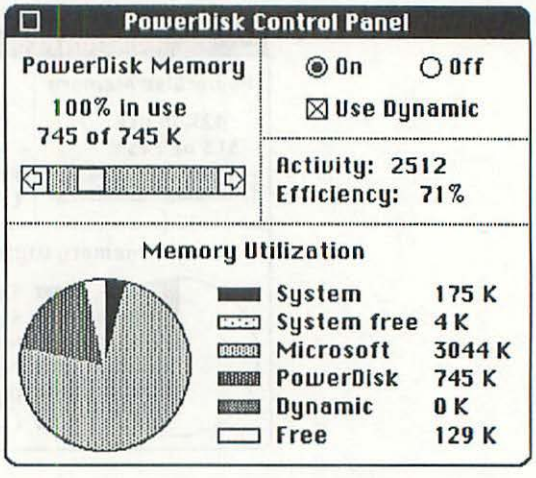
2. Now go to work. Use your Macintosh just as you normally would, except pay attention to how fast things are happening: commands, dialog boxes, menus themselves, quitting and launching applications, and so on.

3. After a half-hour or more of work, open the *PowerDisk* Control Panel again to jot down the following statistics:

a. Check to see how much of the permanent cache has been utilized. In the upper left corner of the Control Panel, you should see a phrase something like "53% in use," and below that the exact figures, "478 of 900K."

b. Now check the cache's efficiency. In the middle right section of *PowerDisk*'s Control Panel, you will see an Activity value, a number like 357, and an Efficiency value, a percent such as 41%.

c. Now determine how much free RAM currently remains. To do that, turn to the numbers beside the pie chart at the bottom of the panel. The free RAM figure should be the last one on the list. Also make note of the amount of RAM used by *PowerDisk*'s dynamic cache.



As you continue to use the PowerDisk *RAM cache, cache utilization and efficiency should increase.*

4. After working for at least another half-hour, check the Control Panel again, and continue to do so until the figures seem relatively stable. As before, jot down the figures each time you check the control panel so you can keep track of them. Once the figures have become reasonably predictable, it's time to analyze them.

5. So let's look at the figures from your last visit to the *PowerDisk* Control Panel, and consider what they mean and how they can be used to fine-tune your RAM cache.

   a. *Cache utilization.* The figures in the upper left corner of the *PowerDisk* Control Panel reveal the percentage and the actual number of kilobytes in the permanent cache that were used. For example, your figures might read "88% in use" and "244 of 300K."

      i. If the percentage of the permanent cache is below 100%, even after a few hours work , then the permanent cache is probably too large. Unless you have lots of RAM to spare (which you can check on with the free-memory-remaining figure, as we'll see), and don't mind wasting some RAM for those rare moments when you might need an especially large RAM cache, you should reduce the size of the cache. How much should you reduce it? Use the figure indicating the actual number of kilobytes used as a guide. For example, if the second line reads "244 of 300K," then reset the permanent cache to 250K.

      ii. If the percentage of the permanent cache is 100%, then the permanent cache may not be large enough. I say *may* because there may not be any extra RAM to give it. Indeed, before you go and enlarge the permanent cache, you need to look at the RAM cache's efficiency (as described in point b) and how much free RAM remains available (which is discussed in point c). Don't forget that in *PowerDisk*, if the permanent cache is full, the dynamic cache may be able to handle the overflow without any sacrifice in efficiency.

   b. *Activity and efficiency.* These two statistics are displayed in the right, center part of *PowerDisk*'s Control Panel. The first figure, labeled Activity, is the total number of read and write requests which have been made since *PowerDisk* was turned on. The second number, Efficiency, is the percentage of times that the RAM cache was able to fulfill the request and prevent the Macintosh from performing a disk read or write, or a "real I/O operation," as they say.

i. Although the Activity figure is not of much help in fine-tuning your RAM cache, it is a revealing number nonetheless. It is not uncommon to see a number in the thousands after a few hours of work. In fact, if you time yourself and divide the Activity figure by the number of minutes you have been using the computer, you can get a good idea of how frequently the Macintosh actually reads from or writes to disk.

Another way to use this number is as an indicator of whether or not you have been working long enough to get a reliable Efficiency figure. If the Activity number is low (in the low hundreds for most applications), or has changed little since the last time you looked at the Control Panel, then you should go back and work some more before you place much stock in the other statistics on the Control Panel.

ii. Efficiency is the hub around which all the other statistics revolve. The goal of the whole RAM cache tuning process is to raise efficiency as high as possible. Do not expect, however, to achieve 100% efficiency very often (or ever, if the cache is in write-through mode). If you save your documents frequently while you work, if limited RAM necessitates a small RAM cache, efficiency may be very low. But even reducing the number of times the Macintosh must go to disk by a quarter, a third, or a half is better than no reduction at all.

If permanent cache usage is below 100%, then what you see in terms of efficiency is what you get; there's little you can do to improve it. If permanent cache usage is 100%, however, you may be able to increase the cache's efficiency, especially if it is low (and low, unfortunately, is a relative term). If efficiency is low and you have some free RAM available, try increasing the size of the cache. If efficiency is already high (80% or more), on the other hand, you probably won't gain all that much by increasing the size of the permanent cache, although you have little to lose by trying.

Remember, if RAM cache efficiency is low, it may be because you haven't used the cache long enough. As I pointed out earlier, RAM caches take a while to get up to speed, and initially, until they start to fill up, they do little to enhance a computer's performance. Thus if you are impatient and check the Control Panel too soon, you will no doubt discover an abysmally low efficiency. So for efficiency and all the important statistics described, be sure to check the Control Panel several times and wait for the figures to level off before drawing any conclusions.

c. *Memory utilization.* The figures that accompany *PowerDisk*'s Memory Utilization map can also be helpful. One of the most interesting of these numbers is the one which shows how much memory your current application occupies. Obviously, this number depends on the size of your documents and which aspects of the program are currently in use (if you have used the spell-checking part of your word processor, for example). But if you take that kind of variation into account, you can use the memory utilization figure as a guide for its optimal memory allocation when run under MultiFinder.

More to the point of the current exercise, however, are the last two numbers on the list: the memory used by the dynamic cache and the amount of RAM which remains unused (and is therefore "free").

i. The last of the two figures on the list is the amount of free RAM still available. If your RAM cache is working at 100% capacity and if its Efficiency is low, you will probably want to increase the size of the permanent cache—if you still have some free RAM available. Use this last figure on the list to tell you if any free memory is available to add to the permanent cache.

For many of us, however, RAM is in short supply (that's frequently why we use a RAM cache), and the free RAM figure will be down to nothing. With most RAM cache utilities and with no RAM to add to the permanent cache, the cache's Efficiency is as good as it's going to get. But with *PowerDisk*, this answer is really not quite correct—because of the dynamic cache.

ii. The next-to-last figure in the Memory Utilization list is the memory occupied by *PowerDisk*'s dynamic cache. If your permanent cache is large enough or too large, this figure will be zero or close to it. It is only when permanent cache usage is 100% that the dynamic cache kicks into gear. When it does kick into gear, the Memory Utilization list will show you just how much memory the dynamic cache is using.

An important aspect of the dynamic cache is that it adjusts automatically to the needs of the moment, increasing and decreasing as your applications require. This fact is important if you use a number of different applications which vary in size: one application may utilize the entire permanent cache and employ a large dynamic cache as well, for example, but another may not even use all of the permanent cache.

If you do use a variety of applications, you should keep track of the dynamic cache as you work in different ones. You may find that it makes sense to optimize the permanent cache for one of your smaller applications and let the dynamic cache help out for the larger applications. Also, if you need lots of RAM for large documents from time to time, the dynamic cache can release the needed RAM, whereas the permanent cache cannot.

On the other hand, the dynamic cache is flushed each time you quit your application (if you are working one application at a time). If you want to cache the most-used parts of several applications, so you can optimize program operation in several applications at once, then you should make the permanent cache as large as possible and not rely on the dynamic cache at all.

In the end, as you work with the statistics and try to optimize *PowerDisk*, it is important to keep in mind that cache efficiency applies to the sum of the permanent and dynamic caches. If you want to know the total cache size at any given moment, you must add the current size of the dynamic cache to that of the permanent cache. But

you must also keep in mind the different ways in which the two caches work.

6. If you changed the size of the permanent cache as a result of the analysis in step 5, monitor the cache's performance once again to see if your change causes an improvement in the statistics. Continue adjusting the cache size and monitoring the statistics until you are satisfied that the cache is performing as well as conditions allow.

Once you have configured your RAM cache the way you want it, don't sit on your laurels from that point on. Check on cache statistics from time to time. You may find that changing habits or new projects have altered the performance of the RAM cache, for better or worse. If worse, use the new statistics to analyze the situation and make an adjustment to the cache.

What to Look for in a RAM Cache If you have decided that a RAM cache makes good sense for your combination of work habits, RAM size, and personal preferences, you're going to need to go out and buy one since, as you will soon see, the RAM cache built into the Macintosh Control Panel is not recommended. Because not all RAM cache programs are created equal, it is important to be aware of the features that distinguish good RAM cache utilities from mediocre or bad ones. Here's a list of the features which I consider important:

- *Works from a hard disk.* As I mentioned in the RAM disk section, this one is so obvious it hardly bears repeating. You should always be sure utilities such as this run from a hard disk. Even if you don't own one today, you may tomorrow.
- *Does not assume installed RAM of a particular size.* A few out-of-date RAM caches are programmed in such a way that they assume you have one megabyte of RAM. In these days of RAM upgrades and Macintosh II memory limits, this is a very poor assumption. If you are purchasing a RAM cache program, make sure it is designed to work with any amount of RAM, even if you have just one megabyte. As with a hard disk, even if you own only one megabyte today, you may own two or four (or more) tomorrow.

- *Does not empty cache when you quit the current application.* If you are working one application at a time (with MultiFinder turned off, in other words), a cache can speed up the switch from one application to another, as well as the internal workings of the applications themselves. But only if the cache is permanent and not emptied each time you quit an application. Some caches do just that; they erase everything in the cache when you quit your application, assuming, for some reason, that you are interested only in speeding up one application at a time. Others caches do not, making it possible to speed up two or three or more applications all at once as you switch effortlessly from one to the next. This is the most useful kind of cache program.

- *Offers automated startup.* Since RAM caches work behind the scenes, they should also start without human intervention. A good RAM cache program should include an automatic startup option.

- *Allows you to change RAM cache size or parameters while you work.* A useful RAM cache feature is the ability to change the way you use the RAM cache without resetting or restarting your Mac. You should be able to reconfigure the whole RAM cache operation at any time.

- *Has a write-through option.* We've already discussed how caches which cache disk writes as well as reads are fast but dangerous, as dangerous as when you save documents on a RAM disk. A good RAM cache program must thus be a write-through cache or must have an option which allows a write-through mode of operation.

  Apple's RAM cache, which can be turned on and off via the Control Panel, is not write-through (or was not at the time this was written). For this reason alone, I cannot recommend that you use the Control Panel's RAM cache.

- *Provides statistics.* As you'll see in the following section, feedback from a RAM cache program is essential if you want to use it most effectively. Thus at the very minimum, a RAM cache utility should be able to tell you how much of the cache's reserved space is in use and what percentage of disk requests have been answered by the cache.

RAM Disk or RAM Cache?

So should you use a RAM disk or RAM cache? The answer, as usual, depends.

It depends, to a great extent, on how much RAM your Mac has and to what extent you would have to compromise to use a RAM disk. If you have a small amount of RAM (one megabyte, for example), if you simply cannot use a RAM disk because of memory limitations, or if you must compromise a great deal to use a RAM disk (by stripping down your System file, for example), then you may be better off using a RAM cache. On the other hand, if your Mac has more than adequate RAM and you don't have to compromise to use a RAM disk, then a RAM disk may be the better way to go.

The answer also depends on the pros and cons for each type of utility. You should be sure to check the advantages and disadvantages sections for both utilities to see if there are any fatal flaws for one or the other. You may, for example, use several copy-protected programs, which would rule out a RAM disk no matter how much memory your Macintosh has.

Simple preferences count for something, too. You may prefer the instant acceleration that RAM disks offer, rather than having to wait for a RAM cache to get up to speed. Or you may prefer the hands-off nature of a RAM cache: having all the fancy stuff occur in the background, out of sight.

In all likelihood, however, there will be a combination of factors which tilts the scale. Some of the pros and cons are objective, others more subjective, but collectively, once you've had the chance to weigh the factors presented in the previous sections, you will probably find that one or the other dominates your thinking. So once again, you should consider all the issues—available RAM, fatal flaws, personal preferences—and decide for yourself.

Print Spoolers

Like a RAM cache, a print spooler hides in the background and waits for you to print a document. Then when the printing process is initiated, the spooler quickly redirects (or spools) the printer instructions for the document you have chosen to print to a special storage area. Because the two usual storage areas (disk or RAM) can accept printing instructions much

faster than most printers can, a print spooler speeds up document preparation and saves you time even before printing itself begins.

Then once the document is spooled, the spooler goes back into hiding again and feeds your document to the printer (despools it) piece by piece. As the spooler deals with one or more spooled documents and manages the whole printing process, you, meanwhile, are free to go back to work. Through a little programmer's sleight of hand, the Macintosh does two things at once. As a result, a print spooler can dramatically shorten the time your Macintosh is directly involved in the printing process (although behind the scenes, it still takes some time to print a document). A spooler can thus give you more usable time on the Macintosh to perform other, more worthwhile tasks.

Print spoolers come in two basic forms, as software or hardware (with software to run it). Hardware buffers (as they are called) have their advantages, but they are expensive and generally speaking, are probably cost-effective only on networks of ten Macintoshes or more. Software spoolers, on the other hand, are more commonly used on small networks or on single Macintoshes and are the spoolers to which I refer in this book. Most software spoolers can be used in two different ways; I call these ways the "hands-off" and "hands-on" modes. A brief look at these two different modes should give you a good idea of what it's like to use a spooler.

The hands-off mode is certainly the easiest. All you need to do is make sure that your spooler is running, and that's all there is to it. Once it is turned on, you need never think about it again. You simply work as usual and print as usual—well, almost print as usual. When you click on OK in the Print dialog box, the printing process should be over and your application ready to go to work again in a matter of seconds. You can then continue working, select another document and print that, quit and start another application, do whatever you like. The point of hands-off mode is that you never bother about print queues or where the spooled files are stored, or any of that. You just work and print.

The hands-on method starts out much the same way. The difference is that once the spooler has something to print, you use the spooler's manager from time to time to check up on what has been printed and what is still waiting, to reorder the print queue to have a particular document printed next, or to delete a document in which you have found a few spelling mistakes and which you have subsequently corrected and reprinted. Or perhaps a colleague is printing a lengthy book manuscript on your shared

LaserWriter and doesn't want your correspondence appearing between the book's chapters. So you go to the spooler manager and tell the spooler not to send any more files to the printer until it hears from you.

Whether you use your spooler in hands-off or hands-on mode, you should find that your Macintosh spends much less time tied up with printing tasks. As a result, you spend less time waiting for the print routine to put control back into your hands, and more time doing productive work. After all, getting more work done is what the fully powered Macintosh is all about.

With the arrival of MultiFinder and its ability to print in the background with the accompanying PrintMonitor file, the need for independent print spoolers is much less than what it once was. Indeed, print spooling to the LaserWriter with MultiFinder and PrintMonitor is fast, easy, and cheap. (How to use PrintMonitor is explained in the MultiFinder manual; I won't repeat that material here.) But PrintMonitor is not perfect, and there are reasons why it may not work for you. These include the following:

- PrintMonitor's biggest problem is that two megabytes seems to be the minimum RAM size for effective use. PrintMonitor, MultiFinder, and just a single, good-sized application (such as *HyperCard*) will not fit within a single megabyte of RAM. So if you have a one-megabyte machine, you may not be able to use it.
- PrintMonitor only spools files for the LaserWriter. If you use an ImageWriter, you're out of luck.
- PrintMonitor cannot (yet) cope with programs which make an end run around the Laser Prep file. If you want to spool *PageMaker* files, for example, you are, once again, out of luck.

So although PrintMonitor will work fine for some, it may not work at all for others. For those who find PrintMonitor adequate, my only recommendations are to turn it on (in the Chooser window), use it, and spend the time you save wisely. For those who find PrintMonitor inadequate, to reap the benefits of a print spooler you'll have to go out and buy one.

Unfortunately, like RAM caches, print spoolers are not all created equal. It is, therefore, worth a moment or two to consider what to look for in a spooler if you are in the market for one. Of course, it is of utmost importance that the print spooler work with your printer. Beyond that, a good print spooler should have the following features:

- *Fast, faster, fastest.* There are two types of spoolers, and they can differ significantly in the speed with which they spool a file. QuickDraw spoolers capture printer output at an earlier stage and thus free up your Macintosh quickly, often in a matter of seconds. PostScript spoolers, on the other hand, capture printer output at a somewhat later stage and thus give you your Macintosh back a little more slowly (with simple documents) or quite a bit more slowly (with complex documents). The trade-off is that QuickDraw spoolers spend more time working on the file in the background, and thus print more slowly.

  But if you want the best of what a spooler can offer, then the time required to spool a document to disk should be as short as possible. Your spooler, in other words, should be a QuickDraw spooler. At the time this was written, the only two spoolers were of the QuickDraw type, *SuperLaserSpool* (SuperMac Software) and MultiFinder's PrintMonitor. *LaserSpeed* (THINK Technologies) is among the fastest of the PostScript spoolers, and not too far behind the QuickDraw spoolers.

- *Compatibility with your applications.* Some applications do not use standard Apple printing routines. *PageMaker*, for example, does not use Apple's LaserWriter driver and prep file (the LaserWriter and Laser Prep files) to print its documents. Instead, it uses its own special prep file, Aldus Prep. *PageMaker* is not the only application that bypasses the standard printer files; *Cricket Draw* and *Illustrator* both use modified versions of the Laser Prep file. Other programs have their own unique methods as well, and few spoolers can handle them all. Be sure that the spooler you choose works with your applications.

- *Compatibility with MultiFinder.* At the time this was written, MultiFinder had only recently been released and at least one top-ranking spooler had been temporarily knocked out of the ring with MultiFinder incompatibility. If you use MultiFinder or ever think you will, be sure the spooler you buy works with MultiFinder.

- *Automatic startup.* Since spoolers work in the background, it makes sense that they should be launched in the background as well. Some spoolers, such as *LaserSpeed*, start up automatically when

you turn on your Macintosh. Others, such as *SuperLaserSpool*, launch automatically only if you set them as the startup application or include them in a self-running startup routine. In any case, your spooler should be able to fit into your existing startup procedures.

- *Ability to select spooled file storage area.* Spooled files have to be stored somewhere until they are sent to the printer. Spooled files can be large, especially with PostScript spoolers, where they can run into the many hundreds of kilobytes. Thus it is helpful to be able to select the place where the files will be stored: a hard disk, a special blank floppy, or somewhere else. A spooler which does not offer this option is just that much less flexible than one which does. A few spoolers store spooled files in RAM instead of on disk. Unless you have more RAM than you know what to do with, avoid them.
- *Minimal RAM required.* Spoolers lie in wait in the background and thus are what is called memory-resident. Memory-resident programs occupy a small amount of RAM which would otherwise be available for some other purpose. The less RAM a spooler requires, the more RAM you have for other things, such as large documents, multiple applications, and so on. Although the differences are not immense (from 30K to 65K for most spoolers), if your Mac is short on RAM, you should use a spooler which needs as little RAM as possible. *LaserSpeed* and *SuperLaserSpool* both require about 65K. (The one spooler that does not work in this way is PrintMonitor. It uses memory only when actually called upon to spool a file.)
- *Automatic printer initialization.* For LaserWriter printing, normal (non-spooled) print routines include initializing the printer—if it has not been initialized already. Some spoolers (such as *SuperLaserSpool*) continue in this tradition and initialize the printer as needed, as a matter of course. Others (such as *LaserSpeed*) require that you manually initialize the printer before you print your first document. This may make the programmer's task a little easier, but it makes yours a little more complex by adding a step to the printing process, a step which is performed automatically under normal (non-spooled) circumstances. Since spoolers should make

your life easier, not more complex, I recommend that you choose a spooler which performs printer initialization automatically.

- *Downloadable fonts.* Many LaserWriter owners use disk-based, downloadable fonts in addition to those which reside in the printer. A smart LaserWriter spooler should have no problem dealing with documents which require downloadable fonts and should not insist that you download them manually ahead of time. An even smarter spooler should allow you to manually download fonts whenever you want to, and it should still be able to download fonts automatically whenever it is called upon to do so.

- *Printer problems reported.* If the paper supply runs out, or paper gets stuck in the printer mechanism, the spooler should inform you. Otherwise, nothing will get printed until you discover the problem for yourself. Most spoolers notify you of printer problems, but some don't. Get one that does.

- *Spooling can be suspended.* There will be times when you want to suspend spooling and print a document directly. Your spooler, for example, may interfere with a nonconformist program's printing procedure, and you may want to turn off the spooler while you are working with that application. Or you may need a little extra RAM—the RAM which the spooler usually occupies as it waits in the background. Thus you should be able to terminate the spooler at any point and restart the spooler at some later time, without restarting the computer.

- *Print now, print later.* Much of the time, you will want to have your documents spooled and printed as you work. There will be times, however, when your printer is unavailable, for example, that you simply want to spool files for later printing. Thus a good spooler should allow you to spool files for printing at some other time later in the day, the following day, or whenever. Of course, it should be able to print no matter what application you are using at the time.

- *Reorder or cancel spooled files.* As spooled files stack up, you may want a particular file to print next, or you may change your mind and decide to delete a spooled file altogether. A good spooler

should allow you access to the print queue, and permit you to change the order of the files to be printed, or delete files still in the queue. For the greatest flexibility, the spooler control center should be in a desk accessory, so that you can use it at any time.

- *Slow down your application as little as possible.* Inevitably, you will experience some degradation of performance in your applications while the spooler is spooling out files to the printer. But while your application may sputter and hesitate some, it should not come to a grinding halt. If there is a bottleneck in processing time, printing should halt, not your application.

  Spoolers vary in the degree to which they effect application performance; QuickDraw spoolers have a slightly more noticeable effect because they capture a document at an earlier stage and thus need to spend more time working on it in the background. Although it is hard to determine what its effect will be until you actually use it, try to choose a spooler which degrades application performance as little as possible. If you need to get that spreadsheet calculation as fast as you can, remember that you can suspend printing until a later time, perhaps when you're not even around to notice.

- *Good error handling.* If the LaserWriter overloads and resets itself, if your Macintosh crashes for one reason or another, if your network goes down, the spooler should be able to pick up where it left off. Some spoolers are better at this than others. If possible, use one of the better ones, such as *SuperLaserSpool* or *LaserSpeed.*

Summary

- Automated startup routines are a simple and easy way to free yourself from the drudgery of preparing your Macintosh for work every morning.
  - ◊ *StartMac* 1.1 (and its relatives *StartMac* 2.0 and *MacPS*) uses a straightforward command language to create complex startup routines which can include displaying images, speaking instructions, and launching applications.

- ◊ *Tempo II*'s Set Startup file makes it possible to turn anything you can do in *Tempo II* into a startup sequence.

- You can speed up your interactions with the Macintosh by using the many keyboard shortcuts which are always available in dialog boxes and file selection boxes. It also helps to take the time to learn the keyboard shortcuts in your applications which are applicable to the way you work.
- Mouse-oriented Macintosh owners can enhance their speed and efficiency through several utility programs:
  - ◊ *HFS Navigator* collects diverse folders all under one menu which is available in any Save or Open file selection box. It can also be used to find files and rename or delete them, all with the click of the mouse button.
  - ◊ You can use *Icon-It!* to create icon bars specific to each of your applications. Icon bars can be displayed anywhere on the screen and can contain buttons which call up desk accessories, invoke commands, or play *Tempo II* or *QuicKeys* macros.
- By placing your applications and/or System folder in RAM, RAM disks and RAM caches are the most effective way to increase the speed with which the Macintosh works.
  - ◊ If you have RAM to burn, if you don't use copy-protected programs, and if you like to see what it is that makes your Macintosh such a speed demon, then a RAM disk is probably the better choice.
  - ◊ If you have limited RAM, if you use copy-protected programs (other than games), and if you prefer to keep the mechanics of speed enhancement out of sight, then a RAM cache is more likely your style. Do not be tempted to use the one in the Macintosh Control Panel, however. It is not write-through and thus should be avoided.
- Print spoolers, such as Multi]Finder's PrintMonitor, can dramatically shorten the time your Macintosh is directly tied up by the printing process and thus give you more usable time on the computer. You may see some performance degradation in your applications as the spooler works in the background to complete the printing process; if it bothers you, just tell the spooler to wait to print until you go on a break.

8

Fragmentation, Copy Protection, Backups, Disk Problems, and More

Speed Traps that Slow the Macintosh Down *476*
- Copy Protection *476*
- Fragmented Files *482*

Backups: The Computer Owner's Safety Net *493*
- A Fully Powered Backup Strategy *494*
- Hard Disk Backup Programs *500*

What to Do if Your Safety Net Has a Hole in It *502*

Summary *508*

This chapter is about roadblocks on the highway to speed and efficiency. One of those roadblocks has been commented on in a number of places throughout the book: infrequent but nonetheless frustrating software incompatibility—often between several utilities, such as two Init files, or a utility and one or two applications. As I have mentioned before, the incompatibility problem is really only solved by avoiding it. But there are other problems which surface from time to time. Some of them can be confronted directly and successfully. Others are less amenable to solution. Both kinds of problems are what this final chapter is about.

Speed Traps that Slow the Macintosh Down

There are (at least) two significant obstacles to getting your Macintosh to run at top speed. One which you may or may not encounter depending on what applications you use, is copy protection. Of course, copy protection has many drawbacks, but an important one is that it can prevent you from copying applications onto a hard disk or a RAM disk and thus keep you from experiencing the speed enhancements hard disks and RAM disks can offer. The other obstacle, which you most certainly will encounter if you have a hard disk, is file fragmentation.

Copy-protected software is not always easy to deal with, although fortunately, it is not as common as it used to be and is still on the decline. Fragmented files, on the other hand, become more common and more of a problem the more you use your hard disk; they are also rather easily unfragmented. The next two sections look at these two problems and what can be done about them.

Copy Protection

Commercial software companies have every right to feel protective of their products. They have, after all, spent months or years and tens or hundreds of thousands of dollars designing the program, writing the software, and putting together the manual and the rest of the package. Only by selling the finished product at a profit can they expect to earn back their investment

and go on to create other products. But the sad fact is that antique technologies such as books are more expensive and more work to duplicate, and hence better protected against illegal duplication than software. Since software costs so much more than the average book, the enticement to illegally copy software is even greater. Thus was born software copy protection.

Unfortunately, copy protection usually does more than just protect the legitimate interests of the software vendor. Individual copy-protection schemes (and there are many different kinds) may do one or more of the following:

- Prevent you from copying the program onto a hard disk.
- Prevent you from copying the program onto a RAM disk.
- Prevent you from copying the program onto a floppy disk with a larger capacity (from a 400K to an 800K disk, or from an 800K to a 1600K disk) or onto a specially configured work disk.
- Place a hidden file on your hard disk which may cause serious problems sometime in the future. If the copy-protection scheme goes haywire (and they sometimes do), the file may, for example, grow to enormous size and thus consume valuable disk space. You, meanwhile, will have no idea where the space has gone, since the file is an invisible one.
- Force you to keep the master disk at hand and produce it on demand, even if you have copied the program to a hard disk.
- Cause copy errors or other problems when you back up your hard disk (if a special copy-protection scheme is used when the program is transferred to the hard disk).
- If the hard disk to which the copy-protected application has been copied is optimized by a file fragmentation program, the result may be a nonfunctional copy of the program. (Again, this would happen if a special copy-protection scheme is used when the program is copied to the hard disk).
- If the copy-protection scheme is keyed into the ROM-based serial number of your LaserWriter (or some other piece of hardware), major repairs to the LaserWriter may change its serial number and render your software useless.

Needless to say, none of these copy-protection "features" is likely to make life with Macintosh any easier. On the contrary, the more a company is concerned about pirated software and the more complex the protection it uses, the more unfriendly that company's programs become.

Fortunately, in the past year or two, copy-protection hysteria seems to have died down, and more and more software producers have come to realize that copy-protection is a strong negative selling point. Consumers, it appears, have finally made it clear to manufacturers that copy protection causes more hassles for users than benefits for vendors. As a result, fewer and fewer companies are copy protecting their programs. Indeed, at the time this was written, most major business software for the Macintosh was not copy protected.

But until the time comes when we live in the best of all possible worlds, there will probably always be some programs that are copy protected. I offer two solutions to this less than perfect situation:

Solution One: Don't buy copy-protected programs. By far the best solution to the copy-protection dilemma is not to buy copy-protected programs. If there are non-copy-protected alternatives to a copy-protected program, and the alternatives perform the tasks that need to get done, buy one of them instead. It is definitely worth the time and effort it takes to investigate all the possibilities before you invest in a copy-protected program, even if that program is only "mildly" protected (in the words of one manufacturer).

Of course, there are times when only a copy-protected piece of software will do the job. In that situation, there is nothing to do but bite the bullet, buy the software, and go to solution two.

Solution Two: Break the copy-protection if you can. Lest this provocative statement be misinterpreted, let me say at the outset that I do not condone illegal copying or copying for illegal purposes (such as giving a copy to a friend or selling copies through the mail). If you end up buying a piece of copy-protected software, I strongly suggest that you write to the offending software vendor and complain loudly of your displeasure, explaining exactly what kind of inconvenience the copy protection has caused. Then after you have vented your spleen, I suggest that you use whatever means are necessary to make a legal, archival backup and a copy which allows unhindered use on your own hard disk, RAM disk, or application work disk. Let me repeat

that: make a *legal*, archival backup and a copy that allows unhindered use on *your own* hard disk, RAM disk, or application work disk *only*.

Ideally, what you want is to convert the recalcitrant application into one that can be employed in any of the strategies described in this book, an application which behaves like a nonprotected program in the ways that matter. You should be able to copy it freely, using the standard methods, to your hard disk or RAM disk. You should be able to back it up, along with all your other files, with a standard backup program. You should be able to optimize your hard disk, with a file fragmentation program, without fear of ruining the application. You should also be able to store the original disk "in a safe place" (as all the manuals say) and not have to keep it next to the computer in case the program asks for it.

Of course, converting an inconvenient, copy-protected program into the kind of cooperative application described in the previous paragraph is rarely easy. That, after all, is the point of copy protection. But sometimes it actually is easy. Other times it is not so easy, but it still can be done. Either way, it usually takes a special utility program to overcome an inconvenient copy-protection scheme.

My first line of defense against copy-protection hassles is a program called *Hard Disk Util* (FWB Software). This program's main purpose in life is to disengage any copy-protection schemes which prevent you from using software on a hard disk, RAM disk, or special applications work disk, or which require you to insert a master disk each time you want to use a valid copy. Because different programs have different protection schemes, *Hard Disk Util* uses special "patches" to disable the copy protection; each patch is specific to a particular application. *Hard Disk Util* comes with a long list of patches for many of the more popular programs that are still copy protected. You can also buy one or more Patch Disks, which contain collections of patches for other programs not included in the basic *Hard Disk Util* package. If you live in the San Francisco Bay area (or anywhere else, and don't mind the toll call), you can subscribe to The Bay electronic bulletin board and obtain patches directly over the phone that are not included in the *Hard Disk Util* package.

Using *Hard Disk Util* is easy. After launching the program, you make a copy of the offending software's master disk with the **Copy A Disk...** command in the File menu. Then, with the **Patch A Disk...** command, you convert the (often nonfunctional) copy into a fully functional one by install-

ing the appropriate patch. Of course, you must have the patch that applies to the program in question in order to use the **Patch A Disk...** command. That's all there is to it. If all has gone as planned, you should be able to copy the new copy to a hard disk, a RAM disk, or a work disk in the usual way. The new copy will not ask you to insert the master disk—ever.

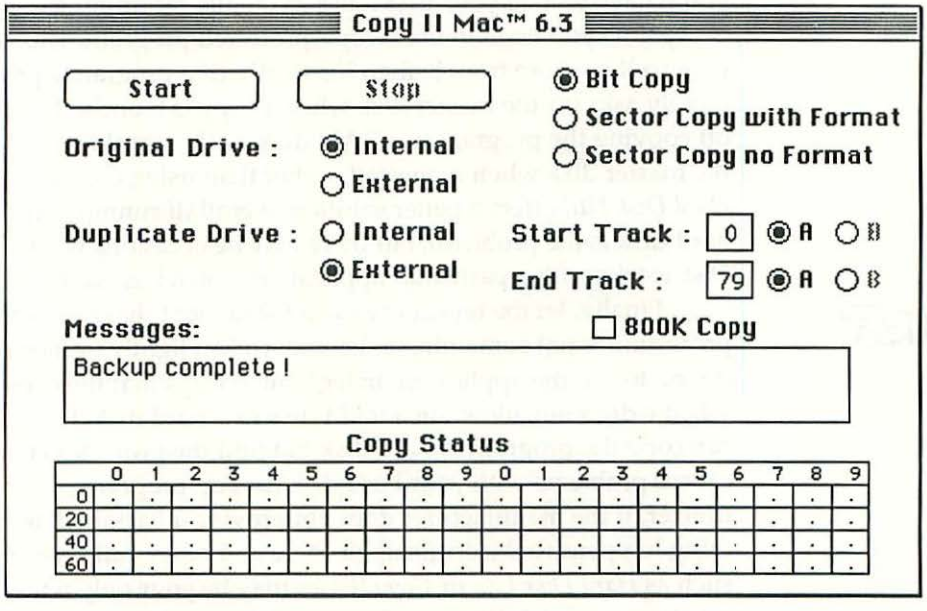
Because it uses special, program-specific patches, *Hard Disk Util* succeeds where many similar programs fail. It is not fail-safe, however, and like any such utility, must be used with care. So if you use a program such as *Hard Disk Util*, be sure that you do so with your eyes wide open and you are fully aware of the possible consequences. Of course, as long as you always apply a patch to a copy of the master disk and lock your master disk before making the copy, you shouldn't experience any unpleasant surprises.

My second line of defense is a popular utility called *Copy II Mac* (Central Point Software). The *Copy II Mac* package actually contains three programs, *Copy II Mac*, *Copy II Hard Disk*, and *MacTools*. *MacTools* is a disk and file utility whose most distinguishing features are its ability to restore deleted files (those which have been thrown in the Trash, and the Trash then emptied) and repair damaged disks. Of course, *MacTools*' ability to restore such files and disks depends on a number of factors, and success is not assured. (See the last section in this chapter for more on file and disk problems.)

But it is *Copy II Mac*'s other two programs, *Copy II Mac* itself and *Copy II Hard Disk*, which concern us here. *Copy II Mac* is a disk backup program which is specifically designed to help you create backups of copy-protected applications. It does so by creating exact duplicates of the original disk. Thus if the original disk is copy protected, so is the *Copy II Mac* copy. This, of course, is no problem if all you need is a backup copy. If you want to copy the application to a hard disk or a RAM disk, however, it is indeed a problem. In that situation, you must use *Copy II Hard Disk*.

Copy II Mac is easy to use if the copy-protection scheme is relatively straightforward: you simply launch *Copy II Mac*, click on the Start button, and follow the instructions which appear in the Messages box. On the other hand, if read or write errors occur, or other problems arise, a certain amount of ingenuity is required to snatch victory from the jaws of defeat (an understanding of Macintosh file structure and copy-protection methods can also help). *Hard Disk Util* makes all the decisions for you, but *Copy II Mac* requires that you figure out what to do next. The manual offers some assistance in such situations, but it is rather short on specific solutions to particu-

lar problems. Nonetheless, copy protection and its conquest being what they are (unpredictable), *Copy II Mac* can sometimes create a backup of a program that stumps *Hard Disk Util*. Although a backup on a floppy disk won't help you put your application on a RAM disk, it is better than having no backup at all.



Copy II Mac *gives you the power to back up copy-protected programs, but its many different options can be confusing.*

If you want to use *Copy II Mac* to transfer copy-protected applications to a hard disk or a RAM disk, you must use *Copy II Hard Disk*. Like *Copy II Mac*, *Copy II Hard Disk* does not convert a copy-protected program into a nonprotected one. What it does is copy a protected program to a hard disk or a RAM disk, altering it only enough so that it will run unimpeded on the hard or RAM disk. Unfortunately, *Copy II Hard Disk* works with only a limited number of copy-protected applications (although they are usually the most popular ones); the current list of applications which *Copy II Hard Disk* can deal with is in the program's Information menu.

Of course, you need to copy a program to a hard disk only once. But since RAM disks disappear every time you turn your Macintosh off, you must use *Copy II Hard Disk* to copy a protected program to a RAM disk every time you turn your Macintosh on. As a result, unless you want to make a rather convoluted *Tempo II* startup macro which runs *Copy II Hard Disk* and prompts for the master disks of your copy-protected programs, you'll have to load your RAM disk manually by running *Copy II Hard Disk* every time you want to use a copy-protected program. This sort of thing may well be more trouble than it's worth. (If a program's protection scheme merely asks for the master disk when a copy is launched, you may be better off copying the program to a RAM disk by the usual method and providing the master disk when requested, rather than using *Copy II Hard Disk*.) Again, *Hard Disk Util* offers a better solution overall (if running on a RAM disk or a hard disk is the problem), but there may be occasions when *Copy II Hard Disk* works with a particular application which has *Hard Disk Util* stymied.

Finally, let me repeat my earlier statement that overcoming copy protection is not something to be undertaken lightly or illegally. If you intend to use the application in legitimate ways that the copy-protection scheme does not allow (on a RAM disk or a hard disk, for example), if you can copy the program to hard disk but find the protection that goes with it causes problems with your hard disk backup program or hard disk optimizer, if the manufacturer does not provide a backup disk in case something happens to the original, for these and other valid reasons, a program such as *Hard Disk Util* or *Copy II Mac* may be your only relief. If you have a valid reason for doing so and are careful never to compromise the rights of the manufacturer (by selling or giving away illegal copies), then you should be able to use a program such as *Hard Disk Util* with a clean conscience and thereby enhance your computer's speed and boost your own productivity.

Fragmented Files

To understand fragmented files, you first need to know a little about disks and how the Macintosh stores files on a disk (a floppy disk or a hard disk, it doesn't really matter which). First, you need some information about disks in general. Inside its hard plastic case, a floppy disk contains a circular piece of flexible plastic coated with a magnetic film, much like the coating on audio or video tape. You can see this flexible disk if you slide the disk's

metal cover to one side (but please, don't touch the magnetic surface). A hard disk consists of one or a stack of several magnetic-coated disks which differ from floppy disks in that they are inflexible (hence the name, hard disk). Information is stored on the magnetic coating by turning microscopic magnetic bits "on" and "off." A special read/write head, analogous to the recording head of your tape player, moves back and forth across the face of the spinning disk, recording and retrieving information stored on the magnetic surface.

Direct from the factory, floppy and hard disks are blank: there is no information of any kind stored on the magnetic layer. Before the Macintosh can put any information there, the disk must be formatted, or initialized. Hard disks are usually formatted by the company assembling the drive from its component parts, but floppy disks must be initialized by you and your Macintosh the first time they are used. Formatting divides the disk into sectors, which are the basic unit of disk storage. Although the number and arrangement of sectors varies among 400K, 800K, 1600K, and hard disks, the basic concept is the same for all disks. First, the disk is divided into a sequence of narrow, concentric rings. Then each ring is divided into many equal-sized, arc-shaped segments. Each arc-shaped ring segment is a disk sector; sectors are numbered sequentially so that each one has a unique identification number. A disk may be comprised of hundreds or thousands of sectors.

Each sector can contain no more than 512 bytes (0.5 kilobytes, or 0.5K) of information: data from a document or code from an application. Since most documents and applications are larger than half a kilobyte, a document or an application is divided up and stored in a number of sectors which are, whenever possible, contiguous (the sector numbers form an ordered sequence, in other words). For example, the document `Income-July`, which is 6K in size, would fill 12 sectors. When copied to your daily backup disk, it might occupy sectors 511 to 522. If the next file saved to the backup disk, `Expenses-July`, were 12K in size, it would fill 24 sectors and if they were available, occupy sectors 523 to 546.

As files are revised, reworked, and resaved, the file's size may increase, and new sectors may be needed to store the additional information. If a neighboring sector is not available, then the new information must be saved to a sector elsewhere on the disk. For example, if we revised `Income-July` by adding income which we failed to report in June and thus increased its size by an additional 2K, the Macintosh would have to jump to

sectors 547 to 560 to store the new information, since all the sectors in between are occupied by `Expenses-July`. If we then revised `Expenses-July` as well, and its size is increased, the Macintosh would have to jump to sectors 561 and beyond to store that new information.

A file whose sectors are not all contiguous, as in the revised versions of `Income-July` and `Expenses-July`, is called a fragmented file. The problem with fragmented files is multifold. The most immediate and obvious objection to fragmentation is that it increases the time it takes to locate and read a file from disk. This is because it takes the read/write head more time to search out and read many sectors strewn about the disk than it does to read a block of contiguous sectors. On a floppy disk, the resulting slowdown usually isn't all that severe. But on a hard disk which has been used for a long period of time, the performance penalty can be profound. As time passes, the fragmentation problem on a hard disk only gets worse.

Another problem is that fragmentation exposes files to a somewhat greater risk of accidental damage. This is because the file directory (the area on the disk where the location of all the files is cataloged) must create special entries, in addition to the usual entries it keeps for all files, for especially large and highly fragmented files. Since information about a fragmented file is kept in two places instead of one, there are now two opportunities for a problem to arise (unlikely as that may be) in the file directory. Finally, fragmentation, and the constant copying, revising, and erasing of files that usually accompanies it, can result in unused sectors scattered all over the disk. Since these sectors are often isolated or in small clusters, they are passed over when storing even moderate-sized chunks of data, and thus are simply wasted space. The more a hard disk is used, the more of a problem unused, scattered sectors become, and the more space on the disk is lost.

Fortunately, there are ways to unfragment files and recover lost disk space. Some are cheap but labor-intensive, and others are more expensive but automatic.

The cheapest, if not the easiest, method is to copy fragmented files to a new, blank disk. Since the disk is blank, there are no impediments to storing the files in contiguous sectors. Thus a fragmented file can be unfragmented merely by copying it onto a blank disk. If you work with a floppy-disk based system, this technique of copying to a new disk is really all you need. If you have a hard disk, it may be a bit too time-consuming and cumbersome.

The following are several simple variations on the technique:

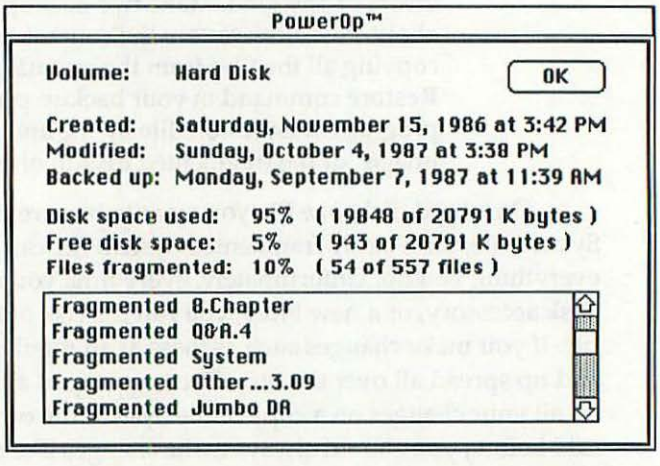
- For floppy disk users, select all the files on the disk (using the **Select All** command, if there are many files) and drag them to a newly initialized disk. Confirm that the files on the new disk all function correctly, then re-initialize the old disk before you recycle it.
- Hard disk owners, of course, face a far more difficult task:
  - ◊ *Reconstituting a single file.* If you have revised a file many times and suspect it is badly fragmented, you can most likely defragment it by copying it to a floppy, erasing the original, then copying it back to the hard disk. If your hard disk is nearly full and already badly fragmented, this method may not work, however. Even if it does work, it accomplishes little in the way of recovering lost disk space.
  - ◊ *Reconstituting an entire hard disk.* To recover lost disk space and reconstitute all the files on a hard disk (and there may be dozens of fragmented files if you've been using your disk for six months or more), back up the entire disk onto floppies, manually or with a reliable backup program (see the next section for more on backup programs). Reinitialize the entire hard disk. Since this will destroy all the data on the disk, you *must* be certain that your backup is complete and without errors of any kind. Make two backups if you have any doubt about this. Now restore the contents of the hard disk by re-copying all the files from the manual backup or using the **Restore** command in your backup program. (Your backup program must restore file by file and not make a "mirror image" of the fragmented disk in order for this to work.)

On a hard disk, one file you want to be sure is not fragmented is the System file, for a badly fragmented System file can slow down almost everything you do. Unfortunately, every time you install a new font, a new desk accessory, or a new FKey, you run the risk of fragmenting the System file. If you make changes such as these at all regularly, the System file may end up spread all over the disk. The best way to alleviate this problem is to do all your changes on a copy of the System file on a floppy disk (as I've said before, you should always make changes like these to a copy, never to the original), delete the old System file and then copy the new one onto the hard disk.

If you do choose to work directly on the System file on your hard disk, then you should employ the single-file defragmentation technique previously described (copy the file to a floppy disk, then copy it back to the hard disk) from time to time to prevent the file from becoming badly broken up.

If you have a hard disk and are unwilling to go through the lengthy backup-and-restore ordeal described above, there are several commercial programs which can take the pain out of defragmenting files. These programs are called disk optimizers, and they vary in their effectiveness, speed, and other, file-related features.

One such disk optimizer is *PowerOp* (Software Power Company). Before optimizing a disk, *PowerOp* can tell you how much disk space is in use, how much is free, and how many of the files on the disk are fragmented. It will also list all of the fragmented files by name. As it optimizes your disk, *PowerOp* moves the pieces of a fragmented file from one part of the disk to another in order to defragment, or reconstitute, the file. Files that are in use (such as the System file, if you are running *PowerOp* from the hard disk), and files that are too large relative to the amount of free disk space, are not moved and thus are not reconstituted. When the optimization process is over, and it is usually over in a matter of minutes, *PowerOp* will tell you which files were moved (and thus reconstituted) and which (if any) were not.



PowerOp's *Info window tells you a lot about your hard disk and lists all the fragmented files.*

PowerOp can also perform two other tasks, neither of which are anywhere near as useful as the disk optimization. One, executed with the **Verify Volume** command, will test your hard disk and report any bad sectors (sectors that contain imperfections which can corrupt the data stored in that sector). This command is useful if you suspect a hardware problem and are looking for confirmation that the problem is caused by the disk. **Verify Volume** cannot, however, correct the problem.

PowerOp can also erase the free space on the disk. This requires some explanation. When a file is trashed (erased), the file itself is not, in fact, removed from the disk. What happens is that the listing for the file in the File Directory is deleted. This is the functional equivalent of erasing the file, since the Macintosh is no longer aware of the file's existence once its listing in the directory is gone. But the file's data is still on the disk (until it is written over by some other file). Prying eyes can read it if they look hard enough. So *PowerOp*'s **Erase Free Space** command will write over all your free space and prevent that from happening. But unless you work for the CIA or the Mafia, you're not likely to have much use for this command.

There are two important features that *PowerOp* lacks. One is the ability to recover lost disk space. *PowerOp* prides itself on being fast, but it does so at the expense of completeness: it often fails to reconstitute large files, and because of the way it works, it recovers little in the way of unusable space.

The other feature the program lacks is the ability to compress the Desktop file. The Desktop file is an important System file that is normally invisible. It is the place where all the icons used on the desktop are stored, and it is also the repository for all the comments you may have made in file and folder Get Info boxes. (You may recall that we used *ResEdit* to look at and change the icons in a Desktop file in Chapter 5.) The problem with the Desktop file is that once it copies an icon from an application or other file, it stores that icon forever, even if the application or file that uses it is long gone. Thus, on a hard disk especially, the Desktop file can balloon to immense size over time, even though many of the icons it contains are no longer relevant.

To reduce the size of overstuffed Desktop files, the *PowerOp* manual suggests that you reconstruct the Desktop file by holding down the Command and Option keys when you quit *PowerOp* (or any other application); a dialog box will then ask if you want to rebuild the Desktop file. But as I pointed out in Chapter 5, using this method will sacrifice all the information in your Get Info boxes. Thus I don't suggest that you reconstruct the Desktop file in this way, unless you have no other options available to you.

Although *PowerOp* is good as far as it goes, it doesn't go quite far enough for a fully powered Macintosh. The best and most effective disk optimizer that I know of is *DiskExpress* (ALSoft). *DiskExpress* is slower than *PowerOp*, but I suppose that's the price you pay for thoroughness, and if *DiskExpress* is anything, it is thorough. It can defragment all the files on a hard disk, no matter how full it is. It can even "prioritize" your files, which means that files least likely to be revised or deleted (such as System file and applications) are copied to the "inside" of the disk, and files most likely to be revised or deleted (such as documents) are copied to the "outside," closest to whatever free space remains. *DiskExpress* is also good at recovering lost disk space, which can amount to quite a bit on a heavily-used hard disk. *DiskExpress* can also compact the Desktop file without losing your Get Info comments. (Like *PowerOp*, it can also erase free disk space, but as with *PowerOp*, few people are likely to need this feature.)

To show how these two programs work and to illustrate their differences, the following is what happened when I recently used first *PowerOp*, then *DiskExpress*, to optimize my twenty-megabyte hard disk. First with *PowerOp*:

1. I inserted the *PowerOp* disk and restarted the Macintosh so the *PowerOp* System file was in control (and thus none of the files on the hard disk were in use).

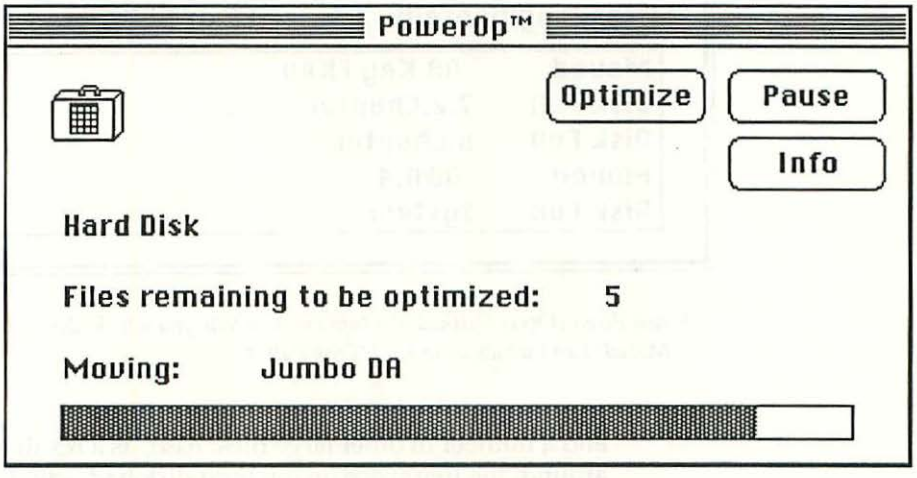
2. Next I launched *PowerOp* and selected my hard disk from the Volumes menu (since I did not want to optimize the *PowerOp* disk).

3. The *PowerOp* Control Panel then appeared. I clicked on the Info button to find out how many fragmented files there were before optimizing the disk.

4. After a minute or so of contemplation, *PowerOp* displayed its Info Panel. It told me that about 95 percent of the disk space (19,848K of 20,791K) was in use, that about 5 percent of the disk space (943K) was free, and that 10 percent (or 54) of the files on the disk were fragmented. It also listed all of the fragmented files by name in a scrolling list at the bottom of the panel. Each listing began with the term "Fragmented."

5. I closed the Info Panel by clicking on the OK button, and, back in the Control Panel, I clicked on the Optimize button.

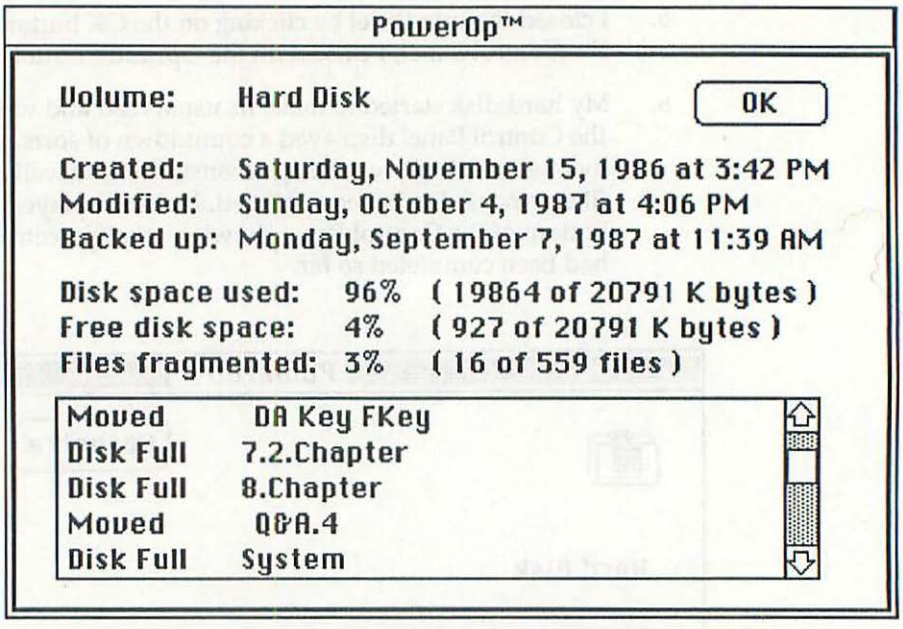
6. My hard disk started to make its usual read and write noises, and the Control Panel displayed a countdown of sorts. It told me, one by one, which file was being reconstituted, as well as how many files remained to be reconstituted. It also displayed a meter at the bottom of the Control Panel showing what percentage of the work had been completed so far.



As PowerOp *works, it keeps you informed of its progress.*

7. In just a few minutes, *PowerOp* had finished optimizing my disk. The Control Panel stated that sixteen of my files had not been reconstituted, presumably because my hard disk was pretty full and because these still-fragmented files were large (although none were over 750K).

8. I returned to the Info Panel to check and see which files had yet to be reconstituted. The ones which were not reconstituted were labeled with the term "Disk Full." They included my System file



When PowerOp *is finished, the Info window tells you which files were de-fragmented ("Moved") and which were not ("Disk Full").*

and a number of other large files. Also, as a result of moving files around, the free space on my hard disk had actually decreased by a small amount. The Desktop file was presumably the same size, since *PowerOp* does not have a Desktop file compression command.

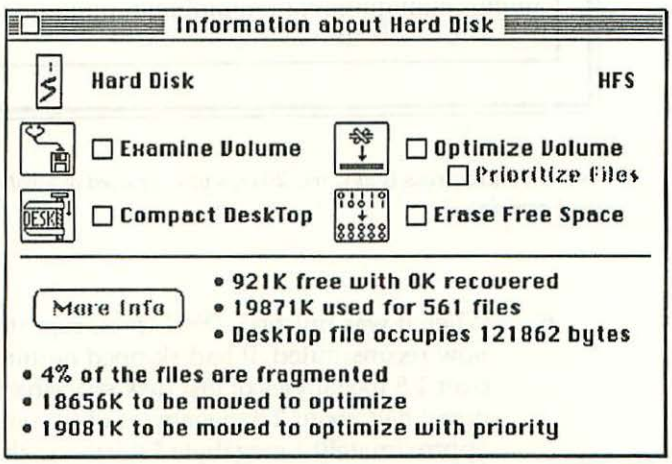
When *PowerOp* had finished its work, I quit *PowerOp*, restarted the Macintosh with the *DiskExpress* disk, and immediately launched *DiskExpress*. The *DiskExpress* test was thus conducted on a hard disk already optimized by *PowerOp*. Nonetheless, as you will see, *DiskExpress* found plenty to work on.

1. The *DiskExpress* screen looks disconcertingly like the desktop. The major difference is that there are fewer menu titles. To work on a disk, you double-click on the disk icon on the right to open the *DiskExpress* window for that disk. (*DiskExpress* will not open the

current startup disk.) In this instance, I double-clicked on the hard disk icon.

2. At the top of the disk window are four options, each with a check box next to it. Below that are a few statistics for the selected disk (free disk space, occupied space, total number of files, and the size of the Desktop file). My Desktop file, *DiskExpress*, was about 122K. To find out more about my hard disk, I clicked on the More Info button to the left of the statistics.

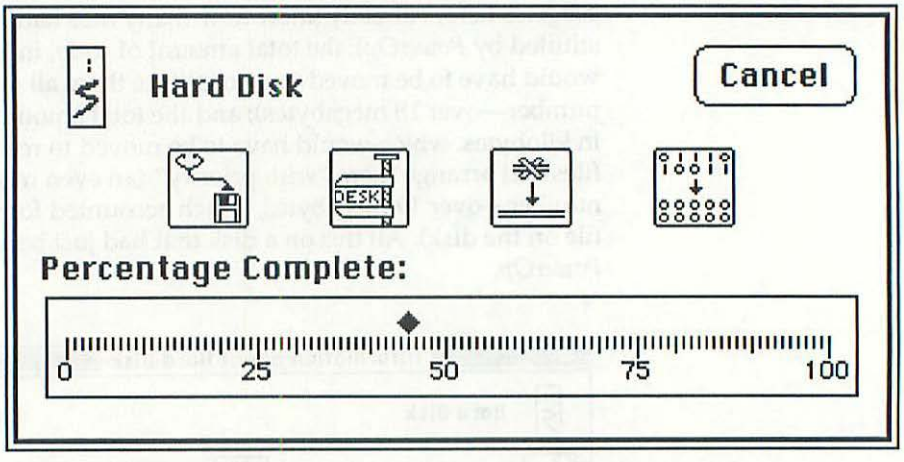
3. My hard disk made its usual noises, *DiskExpress* calculated, and after a few minutes (longer than it took to optimize my disk with *PowerOp*), the following information was added to the list: the percentage of the total number of files that were fragmented (no surprise here; I already knew how many files had not been reconstituted by *PowerOp*); the total amount of code, in kilobytes, that would have to be moved to reconstitute them all (a staggering number—over 18 megabytes); and the total amount of code, again in kilobytes, which would have to be moved to reconstitute all my files and arrange them "with priority" (an even more staggering number—over 19 megabytes, which accounted for virtually every file on the disk). All this on a disk that had just been optimized by *PowerOp*.



After analyzing your hard disk, DiskExpress *displays the bad news.*

4. Back up at the top of the disk window, I clicked on the Optimize Volume check box to turn that option on. I then selected the Express command from the Special menu.

5. The disk window was immediately replaced by a row of four icons, representing each of the program's four functions, and a thermometer-like gauge, which shows how much of the disk has been optimized so far. *DiskExpress* went to work, reading from and writing to my hard disk almost without stopping, its red light flashing almost continually. The red light continued to flash, and *DiskExpress* continued to work for well over an hour. For my part, I had a nice, long coffee break.



When DiskExpress *is at work, it keeps you apprised of what it is doing and how much more work remains.*

6. When it was finished, *DiskExpress* reported that all my files were now reconstituted. It had skipped nothing. It also revealed that over 1.5 megabytes of lost disk space had been recovered, so that now I had about 2.5 megabytes of free space, instead of the approximately 1 megabyte I started with.

7. Finally, just to see what would happen, I turned on the Compact Desktop option, turned off the Optimize Volume option and

selected the **Express** command again. In less than a minute, *DiskExpress* finished its work and returned me to the disk window. My Desktop file, which had been about 122K, was now 96K.

Clearly, *DiskExpress* works much harder and much longer to reconstitute fragmented files. The extra hour *DiskExpress* took to do the job may be a sign that it is highly inefficient (as compared to *PowerOp*). But the resulting 1.5 megabytes of reclaimed disk space was well worth the additional wait. As long as you know in advance that *DiskExpress* is going to put your Macintosh out of commission for a while, the lost time is not a terrible inconvenience. Of course, my hour-long wait was for a 20-megabyte hard disk; I have no idea how long it might take or how much disk space it might reclaim for a hard disk of a much greater capacity.

Backups: The Computer Owner's Safety Net

Some technological advances seem to take one step backward in addition to (or in spite of) their five steps forward. Take data storage, for example. If you are typing a document on a typewriter and a few letters fail to print properly on the paper, the contents of your report is hardly affected. Or say you are typing along and the electricity goes out. Your electric typewriter stops, you grab a bite of lunch, and when the electricity comes back on again, you take up where you left off and finish your report as if nothing had happened. Or say the typewriter experiences a momentary hiccup and types a few incorrect letters in the middle of a word. The typewriter's hiccups and your misspellings will be indistinguishable when you are done.

But if you are creating your report on the computer and the computer version of any one of these events occurs—the disk where you save your file fails to record the data properly, the electricity goes out while you are working, or the computer (or your word processor) hiccups briefly and flips a few bits of information—you could lose your entire file. Information stored on paper may be harder to manipulate, but it is much less vulnerable to damage and loss than information stored on a disk. Take a magnet sometime and pass it over the printed version of a document and then over the electronic version of the same document on a floppy disk, and you'll see what I mean. (Actually, don't put a magnet on your floppy disk—just take my word for it.)

Experienced computer users have long recognized the fragile nature of electronic data storage and, in defense, invented the bane of all personal computer owners: the backup. The only way to protect yourself against accidental data loss is to make a backup copy with which the lost file (or, heaven forbid, disk) can be replaced. Even if the backup copy lacks the latest revisions, at least some of your work can be recovered, but only if you make the effort to back up your files.

Because electronically stored data is more vulnerable to accidental loss, backing up your files is imperative; imperative, yes—fun, no. In fact, making backups is about as dull and boring as it gets. And if it is a large hard disk you have to back up, it's even worse. This is especially true because in the Macintosh world, the era of the cheap tape backup has yet to arrive. When it does arrive, everyone with a hard disk will probably rush out and buy a tape backup and thus convert a dreaded chore into a simple point and click. Until that time, however, backing up to floppy disks is the norm for most of us and the method assumed in this book.

A Fully Powered Backup Strategy

For backup purposes, the documents on your hard disk or floppy disks can be conveniently divided into three types: applications, system files, and documents. The first two are relatively easy to keep track of and to back up properly. Documents, however, are another matter entirely. Of course, for complete insurance against disaster, you need to keep backups of all three kinds of files.

Whether you work with floppy disks or a hard disk, one thing which may make creating and keeping backups easier is dividing each backup set into two parts: an application (plus System folder) subset and a document subset. The application subset will not change much over time (only when you purchase new applications or alter your System file), so you can put it somewhere out of the way. You will be working with the document subset frequently, however, so it should be easily accessible.

Now, since the most effective way of backing up each type of file is somewhat different, let's look at each in turn.

System Be sure to keep a copy of your most recent master System folder and store it as part of your application backup subset. Although it is true for

most people that the System file, even entire System disks, can be recreated from their component parts, the time and effort involved are definitely worth the extra backup disk and the time it takes to copy files to it. Every time you make a change to your System file or System folder—add a desk accessory, add a font, add an FKey, change to a new version of the System software, add an Init file—back up the System disk to a *new* disk (manually or with a backup program) and add that to your application subset, alongside the backup disk for the earlier version of the System disk. Thus if you later encounter software incompatibilities with the newly added parts of the System disk, you can easily revert back to the previous version of the System folder by using the earlier backup disk.

Applications Always, always make backups of your application disks. If you back up your files manually, every time you acquire a new application, make a backup of it *immediately* and put the original in your application backup subset. Don't wait until it is too late. If you have several backup sets, put the original disk with the backup set that is stored in the safest place (your safety deposit box, for example). If the application is copy protected and there is no manufacturer-supplied backup, write to the manufacturer to complain in the strongest possible terms. Then get a copying utility such as one of the ones previously described and make a backup for each of your backup sets.

If you have a hard disk and purchase a backup utility, backup all your applications the first time you use it and then once every six months or so thereafter (or even less frequently, if you rarely buy new applications and rarely reorganize the folders on your hard disk). You don't have to backup applications every week or every month, as long as you dutifully back up applications as soon as you purchase them and back up the applications on your hard disk approximately every six months.

Documents Because of their proliferation, documents are the hardest type of file to keep track of. They are also the most important type to back up. Lost System files and damaged applications can be replaced (preferably with backups, if need be with new copies), but lost documents must be recreated from scratch if a backup does not exist.

There are a number of ways you can make and keep backups of all your documents. The following is only one possible strategy, but one that works well and keeps your backups both complete and current. It works

with both floppy disk and hard disk systems (a backup utility program is recommended for hard disk owners). It is actually a composite of several different backup methods, each on a different schedule:

1. *Back up as you work.* This is not the same as making intermediate saves. Intermediate saves simply save the document you are currently working on, some or all of which is in RAM, to your current work disk. After an intermediate save, there is still only a single copy of your document on disk. To back up as you work, you must follow one of two procedures:

   Stop working at an appropriate stopping place, save your document, and close it and switch to the Finder if MultiFinder is turned on (quit your application if it is not). Using the usual file copy techniques, copy the document you have been working on to a backup disk (your daily backup disk, for example). Now return to your application, reopen the document, and continue working.

   Or again at an appropriate stopping place, save your document. Then select the **Save As...** command and save it again, this time to your daily backup disk or some other backup disk. Now close the document on the screen (since it is now the "saved as" backup document, and that is not the one you should be working on), reopen the original, and go back to work.

As you can see, this sort of thing can disrupt your work a bit, and therefore is only recommended when the document you are working on is especially complex and/or important.

2. *End-of-day backups.* At the end of each work day, back up the files you have created or revised during that day. How you do this depends on whether you use a floppy-disk-based system or a hard disk.

   Hard disk. If you own a backup utility program, simply set it to copy only documents modified on the current date, then either create a separate backup set or add to an existing set. Put the disk or disks that it creates at the head of your document subset and mark them clearly as daily backups. Continue to do this each day, a new disk (or group of disks) each

day, until the next weekly or monthly backup. If you do not own a backup utility, adapt the floppy disk techniques described next as required.

Floppy disk. The hardest thing about floppy disks is keeping track of the files you have altered or created during the day. One way to do this is to set aside a spot on your desk for disks containing new or revised files. Then as you work, if a disk you no longer need contains a file that should be backed up, instead of putting it back in its drawer or box (or wherever you keep your disks), put it on that special place on your desk. At the end of the day, all the disks which contain files to be backed up are all in one place. All you need to do is back up the appropriate files (located using the By Date view) and then put the disks away in their proper place.

There are several ways you can structure the actual process of daily backups:

One way is to create a special disk for daily backups. At the end of the day, you take each disk in the backup pile, locate the new or revised files, and copy them to the backup disk. Continue with all the disks in the daily pile, and repeat the process each day. When the backup disk is full, add it to your backup set, create a new daily backup disk, and continue as before until it is time to perform a weekly or monthly backup.

The other technique involves copying directly to the backup disk which corresponds to each of the disks in the daily stack. Assume that you have revised two files on your *Excel* document disk and written three letters on your correspondence disk. With this method, you could do one of two things:

◊ You could pull out your document backup disk and copy the revised files to it, and then you could pull out your correspondence disk backup and copy the three new letters to that disk.

◊ Alternatively, you could simply drag the disk icon of the *Excel* document disk onto the disk icon of the *Excel* document backup disk. You'd be copying more files than is necessary

(since you're copying the whole disk), and it would take the Macintosh longer to do it, but you wouldn't have to search for the files changed or created that day.

In either case, there are no daily backup disks to deal with and there is no need to perform a monthly backup to consolidate your backup files.

3. *Weekly or monthly backups.* At the end of each week, two weeks, a month, or whatever, you should perform a complete backup of all your document disks. This consolidates all the files on your daily backup disks, catches any files you may have missed earlier, and makes it easier for you to find files in the event that you need them. Complete document backups are a relatively simple, though perhaps time consuming, affair. Indeed, hard disk owners need only fire up their backup program, set it to documents only, and feed it disks. Floppy disk users must work a little harder. If you have a well-organized set of working disks, you pull out all your data disks (or disks that contain data plus other things) and backup all the documents thereon. If your working disks are disorganized, well, the idea is the same, although the execution may be more laborious. Once the periodic complete backup is finished, all those daily backup disks from the period now covered by the complete backup can be recycled.

4. *Printed backups.* This is not as peculiar or exotic as it sounds: (Remember, paper is less fragile than magnetic media.) By printed backups, I mean paper copies of the files you have on disk. For word-processing documents and database files, printed copies can make file reconstruction a simple matter of retyping. (Of course, this only makes sense if there are no backup files on disk.) For graphics and spreadsheets (where the formulas are hidden from view), paper versions of the document are of more limited use.

   An easy way to maintain a printed backup collection is to make two copies instead of one when you print a new file. Toss one copy in a box set aside for backup purposes, or you can just throw your test printouts in there. Either way, when the box gets full, discard material from the bottom of the pile and continue to add new copies to the top. Since this is a real last-resort type of backup, there's no need to be systematic or obsessive about paper backups; think of it as a special kind of wastebasket.

The strategy just described assumes for the most part that you have only one complete backup set. You may want to have two or more backup sets, however, as insurance in case one set (or more likely, the disk or file you need in that one set) is faulty. For extra safety, you could store one of those sets off-site in your home (if your computer is at work) or in your safety deposit box, for example. While safer from calamities such as flood, fire, or theft, off-site backup sets are more of a hassle to keep up-to-date. If you have a backup set in your safe-deposit box, for example, you'll have to establish, and follow, a procedure whereby you regularly bring in the off-site set and perform a complete backup.

If you do have more than one complete backup set, there are (at least) two different ways you can use them:

1. *Identical backups.* This means that each backup set is the same as every other backup set. In order for that to be true, all your backup sets must be created and updated at the same time. Thus if you have two backup sets, set A and set B, you will backup all your documents twice in the same sitting, first for set A and then for set B.

2. *Rotated backups.* In the rotation technique, you only make one backup at a sitting. But you back up to a different set in rotation each time. In this way, one set will have the most recent versions of all your files, another set will have somewhat older versions. For example, assume again that we have two backup sets, set A and set B. At the end of the first backup period, you make a complete document backup to set A. At the end of the second backup period, you make a complete document backup to set B. At the end of the third backup period, you make a complete document backup to set A again, overwriting all the documents which had been stored there previously. Then you overwrite set B, and so on.

   The reason for using this second strategy is that the current (working) version of a file might be flawed, but the flaw might not yet be discovered. If you rotate your backup sets, newly backed up files (which are on set B, for example) do not overwrite the immediately previous version (which was backed up on set A). If you then discover that the file you just backed up is nonfunctional (on both your working disk and on backup set B), you can go to backup set A and retrieve the earlier and, one hopes, functional

version. If you had only one backup set or two identical backup sets, the older version (which is still on set A in the rotated sets) would have been written over, and you would be out of luck.

Hard Disk Backup Programs

Barring catastrophes like fire or flood, the worst-case scenario for accidental loss in a floppy-disk-based system is one or two irreparably damaged disks due to a spilled drink or an errant magnet (like the ones in your stereo headphones). If you work with a hard disk, however, the worst case scenario is much, much worse. Your entire hard disk could crash, and all the files on it could be lost, completely and irrevocably. Believe me, I know. I've had it happen several times (but not in the last year or so, knock on wood). It need hardly be said that the failure of a hard disk can be a disaster of major proportions—if you haven't backed it up.

Backing up a hard disk is not unlike backing up a large collection of floppies, only more so. But let's face it, manually backing up a hard disk is a major pain in the neck. A very important pain in the neck, but a pain in the neck nonetheless. That's why there are commercial backup and restore utilities.

Some hard disks come with a backup program, others do not. If yours does not, or the one it comes with is not to your liking, you can buy a backup program; there are several excellent ones on the market.

One such program is *HFS Backup* (Personal Computer Peripherals Corporation). *HFS Backup*'s strongest point is its flexibility. You can back up an entire hard disk or you can back up only those files and folders which you select from an on-screen directory of the disk. You can also select only certain kinds of files (either by inclusion or exclusion): all the files modified or created since a certain date and time, certain kinds of files (applications, documents, System files), or files whose names start with, end with, or contain certain characters (all your correspondence files, for example, if they all start with `ltr/` or end with `.csd`). These options make it easy to perform application backups on an infrequent basis, complete document backups every week or month, and perform daily backups of changed files every day or two. Once you select the kind of backup you want to perform, you click on the OK button and feed in floppy disks until *HFS Backup* has done with it.

You can also create custom "backup sets" that back up the same files every time. Thus if you have accounting files or product and customer databases, which are constantly being revised, you can create a custom set that always backs up only those files. Then at the end of each day, you need only select the custom set to back up your accounting or database documents.

Files too large to fit on a single disk are split across several disks. File restoration has most of the same features that file backup does: you can restore an entire volume, files changed since a certain date, or selected files only. Of course, files which are divided during backup are put back together during restore.

HFS Backup is a powerful and (from all reports) highly reliable backup program. A similar program in most respects is *Fastback* (Fifth Generation Systems). But from my perspective, a major drawback of these two programs is that they store backed up files in a special format. This speeds up the transfer of files to and from your backup disks. Unfortunately, it also adds two potential (and I have to emphasize the word *potential*) pitfalls to a process that should be as safe, reliable, and problem-free as is humanly possible. One is that there is now an additional step in the backup process (the translation of files into or out of the special format), an additional gear in the machine which might, some day, somehow, break down.

The other potential pitfall is that your backup files are accessible only through the backup program. You cannot copy a file directly from the backup disk to your hard disk and expect it to work. Thus you are totally dependent on the backup program for retrieving your backed up files. If something goes awry with the backup program or if you lose your only copy of it, you might as well have no backup files at all. I must emphasize that these are potential problems which, like disk crashes, are highly unlikely. But they could conceivably happen and therefore are worth considering when you buy a backup program.

Another popular, high-quality backup program is *DiskFit* (SuperMac Software). *DiskFit* differs from *HFS Backup* and *Fastback* in several important ways. One is that *DiskFit*'s backup files are standard Macintosh files. As a result, once a backup set is made, you do not need *DiskFit* to retrieve files. In fact, *DiskFit* is unable to perform a partial restore; if you want to recover only selected files, you must use the *DiskFit* Report file to locate the disks those files are on and then copy them, manually, back to your hard disk.

The other major difference between *DiskFit* and *HFS Backup* is that *DiskFit* offers few backup or restore options. You can back up an entire disk. You can back up only those files which have changed since the last *DiskFit* backup (but not since an arbitrary date). You can back up documents only. You can restore an entire disk, but not selected files or folders. The advantage to such a small number of possibilities is that backing up becomes a hands-off, automatic process that requires little, if any, thought. (Automatic as it is, it runs well in the background when you are using MultiFinder.) The disadvantage, of course, is that your hands are tied if your preferred backup strategy differs from what *DiskFit* offers.

Which backup program should you use? As long as the program is safe and reliable, the specific program is, as in so many things, largely a matter of personal choice. I use *DiskFit* because, for reasons which may or may not be totally rational, I want my backup files in standard Macintosh format. But others may find *HFS Backup*'s (or *Fastback*'s) speed and flexibility more to their taste. Indeed, you may prefer yet another backup program (there are many to choose from). But whatever backup program you use, it won't help you very much if you don't use it on a regular basis.

What to Do if Your Safety Net Has a Hole in It

So you've finally done it. You threw the only copy of a valuable file in the Trash, instead of dragging it to your backup disk as you had intended. Then, against all odds, you ejected the disk and thus automatically emptied the Trash before you realized what had happened.

Perhaps you were working on that year-end results spreadsheet for the boss when the electricity went out. You were just about to make a backup copy of the day's efforts when everything went black, and now you can't open the document, because *Excel* won't even list the document in the file selection box.

It may have come right out of the blue. The disk that was working just fine yesterday is now greeted with the "This disk is unreadable" dialog box. All the files on the disk are inaccessible. What happened? What to do? Where is Peter Norton when you really need him?

Unfortunately, all of these problems occur from time to time to just about everyone. Even more unfortunate, no matter how often you back up

your files, if one of these small (or large) misfortunes hits just before the next backup, you may well lose one day's work, or one week's work, or worse. When your safety net fails, it's time to look for a medic who can help repair the damage.

There are many helpful disk and file recovery utilities around, including the *Disk First Aid* utility sold with every Macintosh, but the best thing I've seen for dealing with small and large disasters is a package called *1st Aid Kit* (1st Aid Software). Actually, *1st Aid Kit* is useful both before and after disaster strikes. The book-length manual is loaded with good advice, important information, and essential disk and file recovery techniques (some of which require the accompanying *1st Aid* program). Every fully powered Macintosh owner should buy a copy and read the background chapters in the first part of the manual; they contain a great deal of information which can help protect you against the almost inevitable glitches which can destroy your data. When a calamity does occur, *1st Aid Kit* is unsurpassed in its step-by-step troubleshooting and recovery procedures.

1st Aid Kit's well-written manual and recovery software deserve equal billing and equally high marks. The manual starts at ground zero and begins with three chapters of useful and sometimes essential basic information and advice. If you ever wondered what resource and data forks were, or whether bundle bits or parameter RAM had any relevance in the modern world, this part of the manual will tell you. It explains how data is organized on a disk, what causes disk and file errors, and explains the how and why behind a number of important recovery techniques. But perhaps most important, these introductory chapters are full of advice on ways to minimize the likelihood of data loss. As everyone knows, a bit of prevention is worth a kilobyte of cure.

Next, the manual lists almost all the error messages and abnormal behaviors you are ever likely to experience. The messages, which range from the dreaded "This disk is unreadable" to the usually innocuous "Couldn't empty the Trash," are listed alphabetically. The behavior problems are listed, in great detail, by functional groups. Both listings refer the reader to one or more specific problems (the probable sources of the error message or strange behavior), which are described in one of the two largest sections in the manual, the Troubleshooting chapter.

The Troubleshooting chapter contains careful, step-by-step solutions to specific file and disk problems: Desktop file problems, disk drive problems, disks with physical damage such as a warped case or a coffee-stained

magnetic surface (in which you learn how to remove the flexible disk from the hard plastic case for a final, last-ditch recovery effort), memory problems, insufficient disk space, missing files, unreadable disks, printer and print-quality problems, sad Macs and System crashes, and quite a lot more. In many instances, the techniques and advice in this chapter will be enough to pull you through. But in extreme circumstances, such as those described at the opening of this section, you will be referred to the second of the two largest chapters in the manual, the one on Recovery Procedures.

The sections in the Recovery Procedure chapter cover a wide variety of techniques you can use to bail out of the most serious problems. Most of these techniques use *1st Aid Kit*'s file and disk recovery program, *1st Aid*. Because you are not expected to read the manual like a book, and because you are not expected to "learn" the software and all its features before you need to use it, each technique performs a very specific task (such as undeleting files, restoring boot blocks, extracting text from damaged files, and so on), and the description includes one-step-at-a-time directions, cautions, possible side effects, examples, and references to related parts of the manual.

The *1st Aid* application can also perform several preventative measures. For example, it can create a catalog of your hard disk folders and files which includes useful file information (such as creator code, type code, and file ID number) should one or more files become damaged or accidentally erased. It can also create an archive file containing your hard disk's volume information, which is important for the proper operation of the Macintosh, as well as for many disk recovery techniques. Should the volume information become damaged, *1st Aid* can replace it with this backup file.

Finally, appendices to the *1st Aid Kit* manual contains additional information and a set of practice tutorials. For the more technically inclined, there is a list of Macintosh error codes. You can, for example, find out what ID=02, which often appears in the bomb dialog box, means. (Whether you can understand what it means is another matter entirely.) There is also an extensive list of application creator and type codes which can be useful in resurrecting files. The tutorials offer a means of practicing different recovery techniques on nonessential files by taking you through seven classic problems using sample files provided on the *1st Aid* disk.

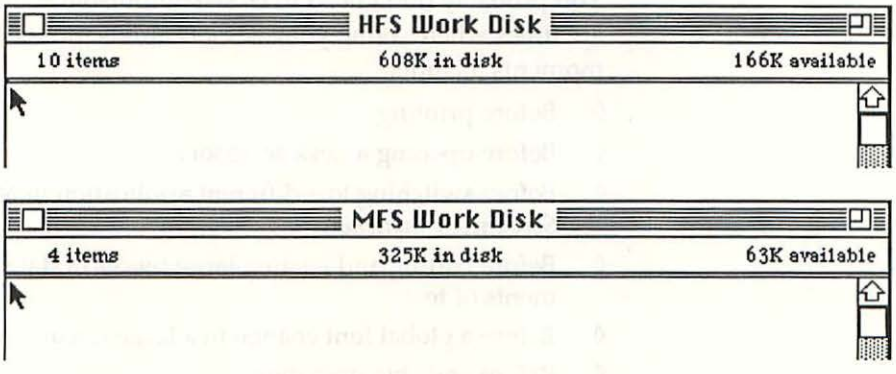
Nobody likes to plan ahead for unlikely or unwanted disaster. But *1st Aid Kit* can help you prevent data loss as well as help you to understand and overcome some of the more common, annoying, and less-than-disas-

trous error messages. If disaster does strike, I can't think of a better way to deal with it than with *1st Aid Kit*.

Of course, there are undoubtedly many who will wait to buy *1st Aid Kit* until a critical file is lost or an important disk refuses to work. For these people, the following are some useful pieces of advice (some of them suggested by *1st Aid Kit*'s invaluable background chapters) which may save you some heartache, and which can make successful disk or file recovery with *1st Aid Kit* more likely when the inevitable happens.

- If you are still using 400K (single-sided) disks, format them as HFS disks. HFS is the formatting standard used by all 800K (double-sided) and larger disks initialized in the usual way. With 400K disks, however, the standard format is MFS. (If you've never used a 400K disk, you may not even realize that there is another way of formatting disks.) HFS stands for Hierarchical Filing System and MFS stands for (the original) Macintosh Filing System.

  You can tell how a disk is formatted by looking at the double line below the title bar and the information section at the top of the disk window. At the extreme left end of the double line, there is, in HFS disks, an extra pixel joining the two horizontal lines. With MFS disks, the extra pixel is lacking.



HFS disk windows display an extra pixel at the left end of the title bar; MFS disk windows do not.

To format 800K and larger disks in HFS, you simply initialize the disks the way you always do. To format 400K disks in HFS, you

must hold down the Option key when you click on the One-Sided button in the disk initialization dialog box, and continue to hold it down while the disk is initialized.

Why go to the trouble to format 400K disks in HFS? In part, to make the folder structure on your 400K disks the same as that on all your other disks (MFS folders are not the same as those in HFS, and as a result they do not appear in file selection boxes). Just as important, or perhaps more important, is the fact that file recovery (if you need it) with *1st Aid Kit* is more likely to be successful if the disk is an HFS disk.

- Do not initialize 400K (single-sided) disks as 800K (double-sided) disks. Even though they look the same, 400K and 800K disks are not manufactured in the same way and thus are not interchangeable. *1st Aid Kit* says reformatting 400K disks as 800K disks causes nothing but problems, and people in the know at Apple say the same thing. If you value the files on your disks, initialize disks only in the format for which they were manufactured.

- Most manuals tell you to save frequently. As a nervous habit more than anything else, I save my documents constantly as I work. You probably don't need to be so compulsive about it, but there are times when saving your work is almost imperative. Such moments include:

  - ◊ Before printing.
  - ◊ Before opening a desk accessory.
  - ◊ Before switching to a different application in MultiFinder, *Servant*, or *Switcher*.
  - ◊ Before cutting and pasting large pieces of data or long segments of text.
  - ◊ Before a global font change in a large document.
  - ◊ Before any "big operation."
  - ◊ Before leaving your Macintosh unattended (when you go on a coffee break or to walk the dog).

- Pay attention to *where* you are saving your documents. Many lost files are not lost at all but were simply saved to the wrong folder.

- Never place more than one System folder on a single hard disk. You're asking for trouble if you do. Thus, when copying new applications to your hard disk, make certain that you are not copying the System folder on the application disk.
- Don't put copy-protected software on a hard disk, unless you first use a utility that removes the copy protection.
- If you encounter a disk with problems or a disk containing a file which has problems:

  1. Stop using the disk *immediately*.
  2. Lock the disk with the sliding plastic write-protect lock *immediately* (if you can see through the hole, the disk is locked).
  3. *Do not* save any file to the problem disk. Eject the disk (use Command-Shift-1 or Command-Shift-2 if all else fails to eject the disk) and save to another disk if you need to save a file.
  4. *Do not* initialize the problem disk. If the "This disk is unreadable" dialog box appears, click on Cancel. If you initialize the disk, all the files on that disk will be lost for good.
  5. Go out and buy a copy of *1st Aid Kit*, if you don't own one already, and follow the directions therein.

- If you encounter problems while working on a document (that is, you are in an application and you have reason to believe that something is wrong with the current document), use the **Save As...** command to save your work as a new document. If you use the **Save** command, you may replace an uncorrupted, though slightly older, version of the document with a new but corrupted version and lose the entire document in the process.
- Most applications do not assign a keyboard shortcut to the **Save As...** command. Since using the **Save As...** command is an important part of backing out of a problem, and since less-than-fatal problems sometimes freeze the pointer without crashing the System, it is a good idea to assign a universal keyboard shortcut to the **Save As...** commands in all your applications with a program such as *MenuMaster* (as described in Chapter 6). Then if the mouse

freezes up on you, you may still be able to save your document from the keyboard, as a new file with a new name.

Naturally, this list is not exhaustive by any means. Nor does it imply that you should neglect all the other admonitions mentioned elsewhere in this book. It is equally important that you minimize file fragmentation, back up your files regularly, use *ResEdit* only on copies of files, store your master disks in a safe place (if you know of one), and so on.

If you use everyday good sense and follow the recommendations in this and previous chapters, your risk of serious data loss should be at a minimum. But if ever something untoward does happen, there is always *1st Aid Kit*'s recovery procedures to get you out of your predicament.

Summary

- Copy-protected software can be a real obstacle to honest attempts toward greater speed and efficiency. I offer two solutions to this fast-disappearing bug-a-boo:
  - ◊ Don't buy any software that is copy protected.
  - ◊ If you must buy copy-protected software, complain to the manufacturer and then use a program such as *Hard Disk Util* or *Copy II Mac* to convert it into a well-behaved, nonprotected program.
- Over time, the files on your hard disk can become badly fragmented (split into pieces scattered across the disk) and greatly slow down the disk's performance. To defragment your files and bring your hard disk back up to speed:
  - ◊ Back up the disk (twice, if there is any doubt about the backup's reliability), reformat it, and then copy the files back onto the disk.
  - ◊ Use a disk optimizer such as *DiskExpress*.
- If you use a floppy disk setup and regularly add desk accessories or fonts to your System file, your System file may be fragmented. You can defragment all the files on your System disk by selecting them all with the **Select All** command and dragging them to a newly initialized disk.

- Because of the fragile nature of magnetic media such as floppy and hard disks, backing up your files is essential. A well-organized, regularly-scheduled backup procedure can save you time and money later on, and it might look something like this:
  - ◊ If you are working on especially important files, back up those files as you work, several times a day.
  - ◊ At the end of each day, back up the files which were created or modified that day.
  - ◊ At the end of each one-, two-, or four-week period, consolidate your daily backups by backing up all your documents.
  - ◊ Each time you modify your System file, make a new backup of it. Keep the old backup in case the modification turns out to have an unforeseen incompatibility.
  - ◊ Back up your applications once every six months or longer, and always make a backup of every new application you acquire in the interim.
- If a file won't open, a disk becomes unreadable, or your hard disk crashes, I hope you made a backup. If not, *1st Aid Kit* may be able to recover the lost file or files.

Appendix

A

The Fully Powered Mac Utilities Disk

Unpacking the Software on the Disk *512*
Software Notes *514*
- *Apfont 515*
- *AutoBlack 515*
- *FKey Maker 516*
- *FKey Manager 516*
- *Focal Point Document Launcher 516*
- *Fontsie 517*
- *Layout 517*
- *MenuMaster 517*
- *Oasis 518*
- *Other... 518*
- *PowerStation/E 518*
- *QuicKeys Demo 518*
- *RamStart 519*
- *Randomizer 519*
- *ResEdit 519*
- *SafeLaunch 520*
- *Screen Maker 520*
- *Set Paths 520*
- *StartMac 1.1 521*
- *SuperFinder 521*

Tucked inside the back cover of this book is *The Fully Powered Mac* Utilities Disk. On it are many of the utilities described in the main part of the book. Indeed, the disk symbol which appears from time to time in the text and in Appendix C indicates that the program described in the neighboring paragraph can be found on *The Fully Powered Mac* Utilities Disk.

All of the programs on the disk are public-exchange programs, and all are included with the generous permission of the author. All programs and the accompanying documentation retain the author's copyright, and their use and further distribution is regulated by the author. In most cases, the rights granted by the author are described in the documentation file or in the program itself. For example, some authors state that their programs may be used and distributed freely; others permit further copying and distribution only under certain circumstances; and others require payment of a fee if you decide to use the program. (Remember, however, that experimentation and a decision not to use the program cost you nothing.) If you are in any doubt as to what rights the author is granting, please refer to the program's documentation or read the Help or About... screens in the program itself.

> If you use a program on The Fully Powered Mac utilities disk and its author requests a fee for such use, PLEASE send a check to the author. Only by supporting public-exchange authors can we expect to see other, equally useful programs in the future.

Unpacking the Software on the Disk

In order to fit as many programs on the disk as possible, all but one of the files it contains have been compressed. Thus in most cases, in order to use a file on the disk, you must decompress it first, using the one thing on the disk that is in its natural state, the program *StuffIt*. *StuffIt* is a fast and convenient utility which allows you to compress and decompress files, combine a number of smaller files into one large *StuffIt* archive file (such files usually end with the suffix `.sit`), and retrieve individual files that are stored in a *StuffIt* archive (an act usually referred to as unpacking or decompressing files; the two terms are more or less interchangeable). It can also unpack files which have been archived with a similar utility by the name of *PackIt* (these files usually end with the suffix `.pit`).

No fee is required if you use *StuffIt* to unpack the files on *The Fully Powered Mac* Utilities Disk. You can also use it freely to unpack files downloaded over the phone (from CompuServe's Macintosh software libraries, for example) or received from other sources. If you decide to use it in other ways, however, to archive old documents for example, then you must send the author a fee as described in *StuffIt*'s documentation.

Using *StuffIt* to unpack files is easy. To show you how easy it is, I'll walk you through the process of unpacking the *StuffIt* documentation. If any questions remain about using *StuffIt*, you should be able to find the answers in the newly reconstituted documentation.

To unpack the *StuffIt* documentation file:

1. Find a spare floppy disk; if there are any old files on it, throw them in the trash. Copy *StuffIt* itself and the documentation archive file, `StuffitDoc.sit`, from *The Fully Powered Mac* Utilities Disk to the floppy disk, which I will call the Work disk. If you are using a hard disk drive, insert your Work disk in any floppy disk drive. If you have a floppy-disk-based setup, start your Macintosh with your master System disk and then insert the Work disk in the other drive.
2. Launch *StuffIt* by double-clicking on it.
3. Select the **Open StuffIt Archive...** command from the File menu, and in the ensuing file selection box, locate and double-click on `StuffitDoc.sit`. A window labeled StuffitDoc.sit will open.
4. In the StuffitDoc.sit window, select the only file displayed in the file list, `StuffIt Documentation`, so that it is highlighted, then click on the Extract... button.
5. In the Extract dialog box that follows, locate in the usual manner the disk and folder where you want the file saved. (If you choose to save to a disk other than the Work disk, make sure it has enough free disk space to hold the unpacked file.) Unless you enter a different name or edit the current name in the file name text box, the file will be saved with its current name.
6. Click on the Save button. The file will be unpacked and saved, and you will be returned to the StuffitDoc.sit window.
7. If you want to continue to unpack other files, close the StuffitDoc.sit window, select the **Open StuffIt Archive...** command (or press Command-O) and select another archive file. If you are finished unpacking for now, close the StuffitDoc.sit window and quit *StuffIt*.

These directions work just as well for unpacking the other files on *The Fully Powered Mac* disk. For the other files on the disk, I would only add two additional tips which will make using *StuffIt* a little easier:

- To save an extra step or two, you can launch *StuffIt* by double-clicking on the first file you want to unpack. *StuffIt* will then open to that file's window (after you click on the opening screen to make it go away), ready for you to select a file and click on the Extract... button.
- Many of the files you will be unpacking contain a number of related files: program files, documentation, and so on. You can ask *StuffIt* to extract all these files at once by shift-clicking on each file name or shift-dragging down the list of files. Then, as before, click on the Extract... button, select the disk and folder where you want the files saved, and click on the Save All button or press Command-Return. *StuffIt* will take over from there.

For more detailed information on *StuffIt*, how it works, what it does, and how to use it, I refer you to the documentation file we just unpacked.

Software Notes

It is important to keep in mind that there is no System folder on the disk. It was omitted in order to squeeze as many different programs on the disk as possible. Since anyone who's gotten this far in the book should be working with his or her own master System file and folder, and should be using that System folder no matter what application he or she is running, the lack of a System folder on the Utilities Disk should not prove to be a hardship. Just remember, to use any of the stand-alone applications on the disk, you must first unpack it and then run the program using the System folder on your hard disk or master System disk. Init files, FKeys, and desk accessories must, of course, be installed in the usual way (after they are unpacked) in order to use them.

Most of the programs on *The Fully Powered Mac* disk come with documentation, either as separate text-only or MacWrite files, or as built-in help files. These files, plus the comments and examples in the main part of the book, should be more than enough to get you going. The following notes include a few additional comments concerning what you will find on the disk, cautions about known incompatibilities, references to the pages where the program is discussed, and the like.

For information on last-minute changes to the contents of *The Fully Powered Mac* disk, look for a text-only Read Me file on the disk (it may need to be unpacked before you can read it with your word processor). If there is no Read Me file, then the comments which appear here and elsewhere in the book currently require no modification.

ApFont

To use *ApFont*, simply place a copy of the file in your System folder and restart your Macintosh. Then call up the Control Panel, select the *ApFont* icon and select the desired font. While *ApFont* requires little more explanation than that, I do suggest that you look over the accompanying documentation file, since *ApFont* is only discussed on pages 334–335 in the book. That file covers a number of useful details, including *ApFont*'s ability to choose from the fonts added by a font extender such as *Suitcase*.

> **Note:** The author requests a fee if you decide to use this program. Please send him a check if you do add it to your software library.

AutoBlack

Two versions of *AutoBlack*, the standard Init version and the special *MacsBug* version, plus a documentation file, are included in the *StuffIt* file. The program is also discussed on pages 233 to 235. Neither version of *AutoBlack* works on a Macintosh II.

> **Note:** The author requests a fee if you decide to use this program. Please send him a check if you do add it to your software library.

FKey Maker

FKey Maker has no documentation. The section on *FKey Maker* on pages 98 to 99, however, covers the territory. *FKey Maker* does not work on a Macintosh II.

FKey Manager

The *FKey Manager StuffIt* file includes *FKey Manager*, the Init files *Pop-Keys* and *KeyPad*, and illustrated documentation in *MacWrite* format. The documentation is quite complete, and the section on *FKey Manager* fills in the details.

Please note that at the time this was written *FKey Manager* was being revised to correct some incompatibilities with MultiFinder and the Macintosh II. It was not known, however, if the new version would be finished in time to be included on *The Fully Powered Mac* disk. To determine if the new version is on the disk, check the Read Me file on the disk (if there is one). If there is no Read Me file, or if it contains no information on *FKey Manager*, then the version on the disk is the unrevised one (Version 3.0). Anyone using MultiFinder or a Mac II should thoroughly test Version 3.0 before using it, or check their sources of public-exchange software to see if the revised version has been released since *The Fully Powered Mac* went into production.

Focal Point Document Launcher

Focal Point Document Launcher is a *HyperCard* stack. It is the document launcher part of *Focal Point*, a commercial stackware program from Activision, which is described on pages 374 to 377. Although all the other parts of *Focal Point* have been disabled, this segment retains much of its functionality and can thus be used as a *HyperCard* document launcher (a concept which is covered on pages 376 to 377).

For help in setting up and using *Focal Point Document Launcher*, follow the directions on the opening card. If you run out of room for buttons in the first card of the *Focal Point Document Launcher* stack, you can create as many additional cards as you need with the New Card command. To add a *Focal Point Document Launcher* button to your Home Card, follow the directions enumerated in step 7 of the illustration on pages 370 to 373, substituting the *Focal Point Document Launcher* stack for the Transfer stack which is the focus of those instructions.

Needless to say, to use *Focal Point Document Launcher* you must have a copy of *HyperCard*. For information on obtaining the complete *Focal Point* package, click on any of the icons on the left side of the main *Focal Point Document Launcher* card.

Fontsie

Two versions of *Fontsie* are provided on the disk, and each one is in its own *StuffIt* file. *Fontsie* 1.53 is in FKey format, and installs fonts in the Font menu of the application that is currently running. *Fontsie* 3.0 is an Init that installs certain designated fonts in the Font menu of an application (either automatically or on request) during the launching of an application.

For a general overview of *Fontsie*, see pages 176 to 180. For precise descriptions of and directions for each version, see the corresponding documentation file.

> **Note:** The author requests a fee if you decide to use this program. Please send him a check if you do add it to your software library.

Layout

There is no documentation for *Layout*, but the on-screen instructions plus the extensive description on pages 222 to 231 should be more than sufficient to see you through.

MenuMaster

Not only is *MenuMaster* covered in two parts of the book (pages 76 to 77 and 246 to 248), but it also comes with brief but quite adequate instructions in a separate file. *MenuMaster* itself is provided in both FKey and desk accessory versions; use the format you are most comfortable with.

> **Note:** The author requests a fee if you decide to use this program. Please send him a check if you do add it to your software library.

Oasis

The Read Me First! file explains how to get *Oasis* up and running. General instructions and tips can be found on pages 347 to 351. For more detailed instructions, invoke the *Oasis* Help command (it's in the Apple menu).

> **Note:** The author requests a fee if you decide to use this program. Please send him a check if you do add it to your software library.

Other...

Other... is described briefly on pages 130 to 132. More details are provided in the accompanying documentation file, along with a few important warnings you should be sure to take note of.

> **Note:** The author requests a fee if you decide to use this program. Please send him a check if you do add it to your software library.

PowerStation/E

This is a demo version of *PowerStation* (which is described on pages 354 to 359). Although several significant restrictions have been imposed (most notably the fact that you can configure no more than one full page of buttons), none of the program's important functions have been removed or disabled. Thus *PowerStation/E* can be used, albeit in a limited way, just like the real thing. A text file which cursorily covers most of the program's less-than-obvious features is included. The real version, of course, has a real manual.

QuicKeys Demo

This is a demo version of *QuicKeys* (which is described in a number of places throughout the book). It contains all of *QuicKeys'* important features and is limited only in the number of macros you can store at one time and in its inability to remember your macro definitions from one work session to

the next. You can thus use it exactly as you would the full-fledged program—except that you'll run out of macro storage space long before you run out of key combinations and ideas, and you'll have to redefine your macros each time you turn on your Macintosh.

The accompanying *QuicKeys* Demo text file is rather brief. You may find it useful to consult the various sections in the book where *QuicKeys* is discussed for additional help.

RamStart

RamStart, which is described briefly on pages 447 and 448, contains a complete set of help files within the program. To get to them, put a copy of *RamStart* in your startup disk's main window, double-click on *RamStart* to launch it, and then, when the *RamStart* control box appears, quickly press the Help button (or click on the scroll bar to get the program's attention, and then press the Help button). I mean quickly; if your Mac has an accelerator board or you own a Mac II, you have barely a second or two before *RamStart* completes its work and the control box disappears. Once in the Help section, click on the appropriate radio button and use the scroll bars to read the entire file.

RamStart works with System 4.2/Finder 6.0, but only when MultiFinder is turned off. (If you have over 16 megabytes of RAM, it supposedly will work with MultiFinder, but few of us have the equipment needed to test that claim.)

Randomizer

Randomizer and its associated files, *BeepInit* and *SoundInit*, comes with a thorough documentation file. The use of these three Init files is also covered in some detail on pages 306 to 310. They do not work, unfortunately, on a Macintosh II. (The Sound section of the Mac II's Control Panel serves much the same function as *BeepInit*, however.)

ResEdit

Where *ResEdit* is used in the course of this book, it is discussed in detail, since there is no instruction manual per se for the program. If you wish to use *ResEdit* to accomplish tasks other than those outlined in the book, however, you are completely on your own.

But no matter what you use *ResEdit* for, no matter whose instructions you are following, be sure to review the cautions and caveats outlined on pages 242 to 245. In a nutshell, they boil down to what I once overheard someone say at a conference: only use *ResEdit* on a totally expendable disk, one you wouldn't think twice about using to scrape mud off your boots.

SafeLaunch

SafeLaunch is described in general terms on pages 341 to 345 and in more detail in the illustrated documentation file. It comes in desk accessory and FKey versions; use the format you are most comfortable with.

> **Note:** The author requests a fee if you decide to use this program. Please send him a check if you do add it to your software library.

Screen Maker

Screen Maker does not have any documentation, but the instructions on pages 300 to 302 should tell you all you need to know. Keep in mind, however, that *Screen Maker* creates startup images which fit the standard Mac Plus/SE screen. A Mac II will display *Screen Maker* images, but they will not fill the entire screen.

Set Paths

The *Set Paths* package includes the associated file, Boot Paths, and an illustrated documentation file. If you have questions concerning *Set Paths* that are not answered in the discussion in Chapter 2 you should be able to find the answers in the documentation file.

> **Note:** The author requests a fee if you decide to use this program. Please send him a check if you do add it to your software library.

StartMac 1.1

StartMac 1.1, a simple one-line *StartMac* playscript, and the *StartMac* documentation are stored in the *StartMac* archive file. Although there is a fairly extensive example of how to use *StartMac* on pages 403 to 410 for complete descriptions of all *StartMac* commands and other important information, as well as another example of a complete playscript, you should refer to the documentation file.

StartMac 1.1 will not work with MultiFinder turned on. (Its close relative, Mac PS, does work with MultiFinder.) To use *StartMac*'s Talk command on a Macintosh II, you must use a Mac II compatible version of MacinTalk.

> **Note:** The author requests a fee if you decide to use this program. Please send him a check if you do add it to your software library. Remember, you get *StartMac* 2.0 (and its extra features) at no charge when you register as a *StartMac* 1.1 user.

SuperFinder

The *SuperFinder* package includes *SuperFinder* itself, a special Run *SuperFinder* file, and an illustrated documentation file. *SuperFinder* basics and tips are covered on pages 351 to 354, but for a complete explanation of all the menu commands, among other things, consult the documentation file.

> **Note:** The author requests a fee if you decide to use this program. Please send him a check if you do add it to your software library.

Appendix

B

Troubleshooting

Problems that Occur During Startup *526*
- No Smiling Mac Icon at Startup *528*
- Smiling Mac Icon at Startup, No Startup Screen or Welcome Message *530*
- Startup Screen or Welcome Message Appears, but Finder or Startup Application Does Not Launch *532*
- Startup Process Completed Successfully, but the Startup Screen Fails to Appear, or the Startup Sound Fails To Play *534*
- Mysterious Problems, Last-ditch Solutions *535*

Problems that Occur While Working *536*
- Desk Accessory Fails to Work Properly *539*
- Init Fails to Work *540*
- BeepSound Fails to Play *541*
- BeepSound Sounds Weird *541*
- Screen Saver Fails to Work *542*
- Windows Do Not Change after Using *Layout* *542*
- New Command Shortcuts Fail to Work Properly *543*
- Macros Fail to Work Properly *544*
- Font Extender Fails to Make Available Certain Screen Fonts, or Certain Font Sizes, or Styles *545*
- Styled Screen Fonts Fail to Appear When Using Style Commands *546*
- Downloadable Fonts Fail to Download *547*
- Strange Tool Palette or Mouse Pointer Symbols in Some Applications *547*
- Application Bombs When Quitting with a Transfer Utility *548*
- System Crashes of Unknown Origin *548*

The best laid schemes o' mice and men..." I think that's how it goes. Well, whether you're male or female, whether you like to use the mouse or favor the keyboard, things can come undone and software can fail to work as expected, no matter how carefully you and the programmer have planned.

Sometimes these failures, which can range from a simple lack of activity (where some was expected) to dramatic crashes, are caused by a bug in an application. Unfortunately, in most instances, the only way to fix such bugs is to point them out to the manufacturer and await a new, improved, and updated version of the program, and work around the bug until the update arrives.

But software failures can also occur if you add Init files to your System folder, load up on FKeys, desk accessories, and fonts, or alter standard System settings with special utilities—in other words, if you go beyond the plain vanilla Macintosh, as we do in this book. Such failures are usually the result of unexpected incompatibilities between two or more programs, or one program and a nonstandard System configuration. It is a sad fact that Macintosh programming is complex and Macintosh applications and utilities have become so numerous and diverse that it is almost impossible to test the compatibility of every combination of programs under every conceivable circumstance. As a result, it is possible to use two or more programs and discover that they are, indeed, incompatible.

For us ordinary people, incompatibility problems usually crop up between an application and a lesser program, such as a memory-resident Init file or a desk accessory, or between two utilities, such as two Inits or two desk accessories. Often the errant program is public-exchange software, since most public-exchange programmers do not have the resources for exhaustive testing. But this is certainly not always the case: I have run into more than a few incompatibilities with commercial programs. With an early version of *Suitcase*, for example, it was impossible to switch from *PageMaker* to another application when using MultiFinder. Then there is the ongoing problem of commercial desk accessories which interfere with each other when installed in the same System file.

The point of all this is that incompatibilities and the resulting performance aberrations (at best) or crashes (at worst), can and do occur. For better or worse, they are a predictable part of life at the cutting edge of computing. They are, of course, the reason I encourage you to thoroughly test the changes you make and the software you use as you work through this book—*before* going back to work as usual. If you carefully test the addition

of an Init, alterations in the keyboard shortcuts for menu commands, new combinations of desk accessories and FKeys, and so on, you will detect most of the incompatibilities which may result from such changes; and you'll catch them before they can cause serious problems.

Of course, there are incompatibilities that are not so easily discovered, ones that arise when you least expect them, despite the most careful testing. Thus it is always a good idea, after adding a new utility or making some other change, to work cautiously for a while, to save often, and to keep a wary eye out for the unexpected. Most of the time, your caution will prove unnecessary. But true calamities are far less likely if you maintain a healthy skepticism until time and experience under actual working conditions confirm the results of your testing.

> **Remember, the points in the last two paragraphs apply to *any* change you make and *any* newly added utility, no matter what their source (commercial or public-exchange).**

But the question remains, when you do discover a problem during testing or later on, what do you do next? How do you figure out what is wrong, how do you determine which pieces of software are in conflict, and how do you resolve the problem? That's where this appendix comes in, for it describes a number of common problems that are often due to software incompatibilities, followed by suggestions about how to solve them. It also covers problems which are the result of human error or misunderstanding.

This appendix is divided into two parts. The first part covers problems which occur during the startup sequence, problems of commission (the System crashes just after it displays the startup screen) and of omission (the startup sound fails to play). The second part covers problems which may occur while you are working; once again, problems both of commission (your paint program acquires a new and bizarre set of tools) and omission (*Camera* appeared to take a screen shot, but you can't find it). If you encounter a problem, look for the listing in this appendix which best describes your particular calamity, then follow the suggestions on how to resolve the problem.

Keep in mind, however, that the emphasis of this appendix is on those problems which are most likely to arise while using the software on the enclosed disk or following the suggestions in this book. It in no way pretends to cover every conceivable problem that might occur with your

Macintosh. If you encounter a problem that isn't covered here and can't figure it out, then the best thing I can suggest is to go back to a software setup which you know works, and then slowly, piece by piece, add the elements of the problem setup, testing each addition to see if it is the cause of the problem. With care, you should be able to isolate the cause of your problem and, if not cure it, figure out how to work around it.

Problems that Occur During Startup

Sometimes a change to your System folder can wreak havoc with the startup sequence and prevent the Macintosh from ever getting to the point where it launches the Finder and displays the desktop (if the Finder is your startup application). Although such mishaps are inconvenient (in order to repair the errant startup disk, you'll have to run your Mac temporarily from some other startup disk), they do have one point in their favor: you can get a fairly good clue as to the source of the problem by taking note of what happens just before the bomb hits.

Thus the following list of startup problems is arranged according to the last thing that happens in the startup sequence, the last thing you see on the screen before the crash. If you failed to notice exactly what happened before the crash, simply start your Macintosh using the problem disk once again, and this time take careful note of what occurs. Then refer to the appropriate entry in the list that follows.

As I mentioned, if this sort of problem does happen to you, you'll have to run your Macintosh with a different startup disk in order to work on the startup disk that's giving you problems. If you are working with a floppy-disk-based setup, simply use a backup master System disk or some other System disk that you know will work properly. Then after starting up with the other disk, insert the problem disk in the other drive and follow the suggestions given in this appendix.

If the startup disk that won't start up correctly is a hard disk, you may need to start your Macintosh from a floppy disk. I usually use a specially configured, emergency floppy startup disk in such circumstances, one with a System folder small enough to fit on a single disk. Since putting together an emergency startup disk is a lot easier when your hard disk is running, I always create one right after setting up a hard disk and keep it handy, like the antacid in the medicine cabinet, for those rare occasions when I need it. Since the hard disk's utility program (often called the hard disk Manager) is

sometimes useful in emergencies, I always include a copy of it on my emergency startup disk as well.

Keep in mind that a hard disk that has crashed takes more time to perform standard disk maintenance tasks (which are preliminary to putting its icon on the desktop) than a hard disk which was properly shut down. Thus the first time you get going with your emergency startup disk, your hard disk's icon may not appear on the desktop, because the startup sequence and the Finder's launching process are both completed before the hard disk is ready to display its icon.

If the hard disk icon fails to appear on the desktop, don't panic (at least not right away). First make sure that the Finder is not, in fact, still waiting for the hard disk to get itself together. The easiest way to test this is to try to pull down a menu. If a menu pops open when you click on it, the Finder's launch sequence is over and what you see is all you're going to get for now. If you can't get a menu to drop down, then the Finder is still waiting, and there's still hope. So wait until you are certain that the Finder's launch sequence is over (wait until you can pull down a menu), and if the hard disk icon is still missing, then select the Restart command to cycle through the System startup and Finder launch sequences once again. Repeat one or two more times if necessary. Ninety-nine times out of a hundred, this trick will put your hard disk's icon on the desktop—eventually. Once the icon is on the desktop, you can do something about your startup problem.

If this process doesn't work, however, then it's time to get a little worried. Your hard disk's Manager utility is the place to turn to next. It probably has a command with which you can mount your hard disk; in other words, locate the hard disk and put its icon on the desktop. If so, try that. Even if it doesn't have such a command, start the Manager utility and see if it recognizes the existence of your hard disk. For example, try to reinstall, or update, the hard disk driver; if you are able to do so successfully, then the Manager has located the hard disk. If the Manager has located the hard disk, that may be enough to jog the disk out of somnolence and convince it to put its icon on the desktop when you quit the Manager.

If neither the Finder nor the Manager utility can locate the hard disk, then it's time to get concerned. Something important is probably damaged, and that something is too complex for me to get into here. Your choices at this point are either a disk recovery program, such as *1st Aid Kit*, or a call to the manufacturer and appeal for help. Fortunately, such downright disasters are rare.

One final comment before we begin: All of the diagnoses and suggestions in this section assume that, in the work session immediately prior to the failed startup, you altered your System folder in some way or added an

automatic-loading, RAM-resident program; that you added an Init file, installed something in the System file, turned on an automatic-loading RAM disk or RAM cache, that sort of thing. If, in fact, you have made no recent changes to the System folder or added no self-loading programs, if the failure to startup properly comes right out of the blue, then there is a fair likelihood that one of the programs involved in the startup process has been damaged somehow.

If you experience one of these out-of-the-blue startup failures, take note of where in the startup sequence the problem occurs. Then go to the appropriate section in the following list and follow the suggestions there. If that fails to solve the problem, turn to the last entry on the list, "Mysterious Problems, Last-Ditch Solutions," and follow the remedy described in that section. If that doesn't work, it's time to turn to a disk recovery program such as *1st Aid Kit*.

Finally, to recap briefly, if you alter the contents of the System folder in some way or turn on a self-loading program such as a RAM cache, and your Macintosh crashes during the startup procedure the next time you turn it on:

1. Take note of the point in the startup sequence where the crash occurs, and refer to the entry in the following troubleshooting list which corresponds to that point.
2. Restart your Macintosh with a backup or emergency startup disk.
3. Insert the problem startup disk, if you are running a floppy-disk-based setup. If the problem startup disk is a hard disk, do what needs to be done (if anything) to get your hard disk to put its icon on the desktop.
4. Follow the suggestions in the appropriate section that follows for diagnosing and fixing the problem.

No Smiling Mac Icon at Startup

The first thing you see when you start your Macintosh is an all-gray screen, followed shortly by a small disk icon flashing a question mark in the center of the screen. If all goes well, a smiling miniature Macintosh should quickly replace the disk icon. If it does not, if the startup process fails to get even this far, then there is probably something wrong with the disk's boot blocks.

The boot blocks are a specific location on the disk containing information the Macintosh needs to continue the startup sequence. After the boot

blocks are read, the smiling Macintosh icon appears. Thus, if the smiling Mac fails to appear, it is most likely that the boot blocks could not be read. (If you want a second opinion, you can use *1st Aid Kit* to determine the condition of a disk's boot blocks.) Fortunately, bad boot blocks are unlikely to be the result of anything we do in this book. Unfortunately, the problem may, in fact, lie in the disk itself.

If your startup disk is a floppy disk, the best and easiest thing to do is put the problem disk aside, make a copy of your backup startup disk, and use this new disk as your startup disk. Unless the problem is software related, which is unlikely, the problem should go away. (You may, however, want to work cautiously for a little while, just in case I'm proven wrong.) If you want to reuse the problem startup disk for some other purpose, test it first by reinitializing it two or three times. If it fails to reinitialize even once, discard the disk; it's not to be trusted.

Although, as we shall see, there are a number of relatively easy ways in which damaged boot blocks can be restored on a hard disk, the fact that the boot blocks were damaged at all is something to be at least mildly concerned about. Because if you have a hard disk, you can't just discard it to make the problem go away. Indeed, once you get the disk icon to appear on the desktop, you should use the Test function of your hard disk Manager utility, or the one in *1st Aid Kit*, to confirm that the disk is indeed in good condition. If the test reports any bad sectors, or if the same problem occurs again (and you are fairly certain that it is not caused by software), back up everything on the disk and reinitialize it. (If the problem occurs again after reinitializing, it's time to call your dealer or the manufacturer for help.)

If the Manager's test function reports that the disk is in fine shape, there are a number of fairly simple ways you can restore the boot blocks on a hard disk short of reinitializing the disk and restoring all your files from your backup. Try the following suggestions in the listed order until you reach one that works:

- If you have a Macintosh SE or II and a hard disk, use the Startup Device section of the Control Panel to reset the hard disk as the startup drive. Then restart the computer.

- Use your hard disk Manager utility to reinstall the hard disk's driver. Then reset the PRAM (parameter RAM) in the computer. If you have a Mac Plus, pull out the battery in the compartment in the back of the computer, keep it out for a few minutes, then put it back in (be sure you replace it the right way around). If you have a Macintosh SE or II, reset the PRAM by holding down the Option,

Command, and Shift keys while opening the Control Panel. You'll have to reset all your preferences (except for the time, which remains unaffected), but the procedure may resolve the problem. Finally, restart the computer.

- Throw your System folder in the trash. To be on the safe side, use a file locator utility such as *DiskTop* or *DiskTools II* to make sure that there are no stray Systems or Finders buried away somewhere. If you find any, be sure to delete them as well. Now use the Finder (not a file utility) to copy a new System folder onto the disk. Restart the computer.
- Use *1st Aid Kit*'s **Fix Boot Blocks** command to repair the boot blocks.
- If the disk still won't perform a proper startup, it may be time to back up any not-yet-backed-up files on the disk and reinitialize it.

Finally, let me repeat that boot blocks are unlikely to be damaged by anything we do in this book. If you do experience this problem and you use a hard disk, it would be wise to test your disk to make sure that it is not the source of the problem.

Smiling Mac Icon at Startup, No Startup Screen or Welcome Message

There are a number of things that happen between the appearance of the smiling Mac and the display of the startup screen (or the standard welcome message, if no startup screen is present in the System folder). Only one of them has much relevance to the kind of problems you are likely to run into using this book, and it occurs at this point only on the Mac SE and II. This is the opening of the System file.

On the Mac Plus, the System file is opened after the startup screen or welcome message is displayed. Thus, if you are a Mac Plus owner and have a startup problem which falls into this section, its cause is probably not a result of something we do in this book, and its diagnosis and cure is thus beyond the scope of this appendix. For help, turn to a package such as *1st Aid Kit*.

On the Mac SE and II, the display of the startup screen or the welcome message indicates that the System file was opened successfully. If the smiling Mac icon appears but the startup screen or welcome message does

not, then it is likely that the System file could not be opened; and the System file probably couldn't be opened because there was something wrong with it.

If you just installed a desk accessory or used an installer program to alter the System file, the installation may have gone awry; or a system crash may have not only carried away your unsaved data, but also damaged the System file. If you can't think of any reason why the System file should have become corrupted since you last used it, it may be some fluke for which there is no easy explanation.

Whatever the cause, however, the first thing to try is replacing the damaged System file with a new one, either a System file from a backup disk, or a basic one from Apple's Macintosh System Disk. Then restart the computer and make sure that the new System file does indeed complete the startup sequence properly. It should.

If the old System file was damaged during a crash or for some unknown reason, replacing it is about all you need to do. If something went wrong in the course of an alteration to or installation in the System file, again, all you need to do is replace the System file and perform the alteration/installation again—this time correctly. The third possibility, that the alteration/installation is incompatible with your current System file configuration, is more difficult to rectify. But first, you need to confirm that this is indeed the problem.

To do so, perform the installation or alteration on a basic System file, one copied directly from a Macintosh System Disk, and see if the startup problem repeats itself when you use this test System file in a no-frills System folder at startup. If the startup problem does occur again, then the alteration/installation is somehow damaging the System file. If the test System file works as expected, but you continue to have problems with your fully configured Master System file (after the alteration/installation is made), then something in your Master System file is incompatible with the alteration/installation.

If the problem is incompatibility, then you'll have to search out the source of the incompatibility. Start with the test System file from the previous paragraph, the one whose only non-Apple feature is the problem alteration/installation. Starting with this System file, reconstruct your Master System file step-by-step, piece-by-piece. After adding each new piece (a small group of desk accessories, a collection of fonts, etc.) or making a new change (such as installing *HFS Navigator*), test it by using it as the startup System file, and see what happens. Continue in this vein until the second part of the incompatibility pair makes itself known. Unless a different order of installation/alteration, or some other esoteric item, resolves the

problem, you'll have to choose between the two incompatible pieces (one hopes it is only two—it may be more than that).

In any case, remember that in a Macintosh SE or II, if the startup problem occurs after the smiling Mac but before the startup screen or welcome message has a chance to appear, the incompatibility (if that is what it is) should lie entirely within the System file. Init files and such, an otherwise likely place to look for problems, are not executed until the next stage of the startup process on the SE and II, and thus should not be part of the problem.

Startup Screen or Welcome Message Appears, but Finder or Startup Application Does Not Launch

After the startup screen or welcome message appears, the System file is opened on the Mac Plus. (As described above, it is already open at this point on a Mac SE or II.) Immediately thereafter, if you use its *MacsBug* version, *AutoBlack* is opened. After that, any Init files in your System folder are opened, in alphabetical order according to the Init file's name. Finally, the startup application, either the Finder or the program you have designated as the startup application, is launched.

If you have a Mac Plus and your startup failure occurs after the startup screen or welcome message but before the startup application is able to launch, the first thing you need to do is make sure that the problem does not lie in the System file. To do so, follow the suggestions in the previous section, the one titled "Smiling Mac Icon at Startup, No Startup Screen or Welcome Message."

If you have a Mac Plus and have determined that the problem does not lie in the System file, or have a Mac SE or II, then the most likely cause of this sort of startup failure is either *AutoBlack* (if you use it) or one of your Init files.

Let's consider *AutoBlack* first. (Skip this and the following three paragraphs if you don't use *AutoBlack*.) *AutoBlack* comes in two versions. One, which is a standard Init file, is a file named *AutoBlack*. The other version is a mini-application named *MacsBug*. The *MacsBug* file name is actually a trick used to ensure that this version of *AutoBlack* is launched immediately after the System file is opened. It's a trick because *MacsBug* is the name of a debugger (a programmer's tool) used by many Mac programmers. The System file is set up so that if *MacsBug* is present in the System folder, *MacsBug* is loaded into memory right after the System file is opened. Thus, by giving this version of *AutoBlack* the name *MacsBug*, the System is

fooled into thinking the file is a debugger, and it loads it into memory before anything else (before any Init files, for example).

The problem is that some programs can't handle mutual coexistence with a debugger. You and I know that the *MacsBug* version of *AutoBlack* isn't really a debugger, but your software doesn't know that; the mere fact that something called *MacsBug* is present can be enough to make some programs very upset. *MacsBug*-intolerant programs range from invisible items such as hard disk drivers to stand-alone applications or utilities (about which I'll have more to say later). For example, the driver for my new hard disk cannot deal with a *MacsBug* file if a startup screen is also present in the System folder; if both are present, the startup process comes to a grinding halt when it encounters the *MacsBug* file.

The moral of this somewhat complex tale is that if you use the *MacsBug* version of *AutoBlack*, or anything else that uses the *MacsBug* trick to get loaded first during startup, be aware that it may cause problems during startup (or later on when you are working). Thus, if you encounter a post-startup screen problem, the first thing you should do is disable the *MacsBug* version of *AutoBlack* (if you use it) to determine whether or not it is at fault.

An easy way to do that is to create a folder within your System folder, call it `Maybe/Maybe Not` or some other name, and drop *AutoBlack* in it; if *AutoBlack* is in a folder other than the System folder, it will not load when you start your Macintosh. So with *AutoBlack* tucked away in the `Maybe/Maybe Not` folder, restart your Macintosh and see what happens. If you haven't eliminated the problem, then *AutoBlack* is not at fault and can be returned to the System folder.

The other likely cause of the problem is an Init file in your System folder which is incompatible with another Init file, or is perhaps messing up in some other way. The problem Init file could be the Init version of *AutoBlack*, *Suitcase*, *Tempo II*, or any of a number of others. If you added an Init file to your System folder just before the problem began, you can be fairly certain that Init is one part of the problem. Even if you haven't recently added an Init file, because Inits can affect the workings of your Macintosh in basic ways, you should always suspect an Init problem until you're proven wrong.

Testing for an Init problem is relatively easy. All you need to do is "disengage" them, just as we did with *AutoBlack* a few paragraphs back. Simply put them away in the `Maybe/Maybe Not` folder, or any folder for that matter, as long as it is not the System folder. Now restart your Macintosh and see what happens. If you still have the same problem, then its solution must lie elsewhere. If the problem disappears, then you're onto something.

If you are onto something, the question remains which Init or Inits are the source of the problem. It could be that you have two incompatible Inits or an Init which is incompatible with something else. Thus you need to experiment, to try different Inits alone and in combination with others (leaving the rest in the `Maybe/Maybe Not` folder), to see if you can further isolate the problem. If you recently added an Init to your System folder, that Init is certainly a good place to start your experimentation. If you recently altered your System file (by modifying it with something like *HFS Navigator*, for example), then you may have an Init which conflicts with the modification. If you suspect this might be the problem, try testing your Inits first without the System file modification, then with the modification.

Ultimately, you should be able to isolate one or two Inits which are the source of your problem. Once they are isolated, you will have to decide what to do about them. Most likely, you will have to forego using one or both of them. If you are willing to continue experimenting, however, there is one other thing you can try to see if you can restore order without losing the use of the problem Init file. As I mentioned earlier, Init files are loaded in alphabetical order. You can thus alter the order in which they are loaded simply by renaming them. Loading order sometimes makes all the difference with Inits. Thus, if you are so inclined, you can experiment with the loading order of your Inits and see if that helps. If not, there's not much you can do but throw the offending Init in the trash.

Startup Process Completed Successfully, but the Startup Screen Fails to Appear or the Startup Sound Fails to Play

First, concerning a missing startup screen, check to make certain that the startup screen is in the System folder, and not within a folder in the System folder, or anywhere else. Then check to be sure that the file is in startup screen format (saved by *Screen Maker* or by your paint program as a startup screen), and that it is named `StartupScreen` (no more, no less). If everything appears to be in order, take a startup screen that you know works and try it to make sure that the startup screen process works properly. If the startup screen you know is good doesn't work either, then something very peculiar is happening—and unfortunately, I have little to offer in terms of a solution.

If the good startup screen works but the problem one doesn't (still), then there's probably something wrong with that startup screen file. To remedy the problem, locate the original graphic used to create the problem startup screen (or if you don't have the original image, convert the errant

startup screen back to a Paint file with *Screen Maker*) and convert it to a startup screen again, using *Screen Maker*, *SuperPaint*, or some other appropriate program. Then put this new file in the System folder, restart your Macintosh, and see if this clears up the problem.

If your startup sound fails to play, first check to be sure that both *SoundInit* and your sound file are in the System folder, and not in any other folder. Then make sure that the sound file is named `StartupSound` (no more, no less). If these two conditions are met and your startup sound still fails to work, consult the first beep sound problem listing on page 305, for some other possibilities. If the startup sound does play but sounds weird, the cause of the problem is probably described in the second beep sound listing on page 307.

Mysterious Problems, Last-ditch Solutions

If you've tried all my suggestions so far, and still can't solve your problem (no matter which kind of startup problem it is), then the following is a real last-ditch solution:

1. Start with a clean slate by throwing the existing startup System folder in the trash:
   a. If you have a hard disk, use your emergency startup disk to start your Macintosh and, if necessary, coax your hard disk to display its disk icon on the desktop. (If you cannot convince your hard disk to display its icon when you start up with the emergency startup disk, then once again, there is little I can recommend other than a disk recovery program such as *1st Aid Kit* or a call to your hard disk's manufacturer.) Back up the contents of the current System folder first (if you haven't done so already), then throw the entire System folder in the trash.
   b. If you are working with a floppy-disk-based setup, start your Macintosh with an alternate System disk, insert a brand new disk in the other drive, and initialize it.
2. Copy a new System folder from the most recent Macintosh System disk to your hard disk or to the newly initialized floppy disk. Drag the folder from the Macintosh System disk to the startup disk using the Finder; do not use a file utility such as *DiskTop*. Make no changes to the System folder for now.

3. Restart your Macintosh using the new floppy System disk or the hard disk with the new System folder. (If you still have a problem, then you really do have a problem and should consult your dealer for help.)

4. Finally, re-create your previous System file and folder, bit by bit, using software from the manufacturer's original disks whenever possible, or if that's not possible, from backup disks created before the problem struck.

   To save time, you may want to experiment and copy the entire System file from the backup disk back onto the new floppy disk or your hard disk and see if it works. If it does, you need not reconstruct the System file all over again. If it is the source of the problem, however, then you will have to reconstruct it from scratch. Test each piece you add—by restarting your Macintosh—before adding something else. If the startup failure crops up again, you now know that the problem is probably related to your most recent additions to the System file or folder. If the startup failure does not reappear, then you probably had a damaged file in the System folder and have now replaced it with an undamaged copy.

Problems that Occur While You Work

Startup failures are one kind of problem you may encounter. Problems which occur during the course of your work are another. Unfortunately, such problems are often harder to pin down than those which occur during startup. The problem may be due to a software incompatibility, for example, or it may be caused by something more mundane, such as improper setup on your part. Although some problems are directly related to the actions which precipitated them (a system bomb immediately after opening a desk accessory is probably the desk accessory's fault, for example), others have causesthat are more difficult to define.

What follows, then, is a list of some of the more commonly encountered software failures. Each is listed by its symptom (the problem you experience) then diagnosed, and treatment suggested in the paragraphs that follow. But before you resort to searching through the list for a description of your Mac's current illness, you should take a few minutes to make sure that you aren't experiencing the Macintosh equivalent of the common cold. In other words, you should be sure to rule out two of the most common

causes of a sick Mac before going any further. Those two common causes are MultiFinder and Init files:

- If MultiFinder was on when the problem occurred, then try to reproduce the problem with MultiFinder turned off. This is an especially useful test if the problem involves either a desk accessory or an application which has not been updated since the advent of MultiFinder. Of course, there are some problems which cannot be reproduced when MultiFinder is turned off (if the Mac crashes when you click on the MultiFinder switching icon, for example). But others can be tested and not a few resolved, simply by turning MultiFinder off.

  If the problem disappears when MultiFinder is turned off, then you are probably experiencing one of two problems. The first possibility applies to applications (and not to desk accessories) and is what might be called "memory stress": not enough memory allocated to the problem program's MultiFinder partition. To see if this is the problem, turn MultiFinder back on, open the application's Info window, and increase the number in the Application Memory Size box. If you have enough RAM, set it to 1024K or more. Now close the Info window, start the application, and see if that solves the problem. If it does, then the original memory size was too small. Experiment until you find a memory size which is neither too small nor too large.

  The second possibility is MultiFinder incompatibility: some programs have not been taught how to tolerate the MultiFinder environment. Although the number of such programs is becoming smaller every day, they do exist, especially among desk accessories and applications which have not been upgraded since August 1987. If the problem goes away when MultiFinder is off, and returns when it is on (and if the problem program is an application, resetting the size of the memory partition doesn't change anything), then the problem is very likely MultiFinder incompatibility.

  If you run into such a MultiFinder incompatibility, call the manufacturer of the errant software and see if the problem has been, or will be fixed by an upgrade. If not, there's probably not much you can do except avoid using the problem program whenever MultiFinder is turned on.

- If MultiFinder is not the source of the trouble, and if you have Init files in your System folder, disengage all your Init files and try to duplicate the problem once again. Disengage your Inits by placing all of them in a folder, any folder—as long as it is not the System folder itself—and restarting your Macintosh (as described on page 526).

  If all your Inits are disengaged (make sure that there are no stray ones still hiding in the System folder) and the problem still occurs, then your Inits are not at fault. But if the problem fails to occur when your Inits are disengaged, then at least for now, look no further. It's time to check out your Init files, one by one, to see which is causing the problem.

If after trying these two quick fixes, you still have a problem, then it's time to check the following. But first, make sure that you know precisely what the problem is. Try to duplicate the problem. Do whatever it was you think caused the problem and see if it happens again. If it does happen again, then you should have a good fix on the immediate cause (if not the ultimate cause). Now with a clear and certain notion of what's going wrong, check the list for help.

If you cannot reproduce the problem, then one of at least three things is happening. One possibility is that the problem was a bizarre fluke and may not strike again; an unlikely possibility, but a possibility nonetheless. Try to duplicate the problem a number of times; if it doesn't recur, you may find you have nothing to fix.

Another possibility is that the sequence of events that invokes the problem is not quite what you thought it was. Since knowing the sequence of events that led up to a problem is important, it is worth your while to play around a little bit to see if you can re-create the problem in some other way. Continue to experiment until you can reliably reproduce the problem (and describe it accurately as well), or until you've convinced yourself that you can't re-create it.

The third possibility is that the problem is erratic or unpredictable. This is the worst possible case, since it is difficult to know what causes an erratic problem, and well nigh impossible to be sure that it is fixed. Let's just hope your problem is not of this sort, since I have no suggestions for help on this one, other than the standard "come back when you can show me what's wrong with it."

Finally, as in the previous section, keep in mind that this list is not exhaustive, since it would be impossible to cover every conceivable problem

and its cure. If you run into a problem which is not listed here and haven't a clue how to solve it, it may be worth consulting your local Macintosh guru. He or she should be easily reached through your local Macintosh users group.

Desk Accessory Fails to Work Properly

Problems in this category include desk accessories that never appear when selected; desk accessories which place a menu on the menu bar, but the menu is nonfunctional, or the one that appears belongs to some other desk accessory; desk accessories that crash sooner or later; and similar failures to work as advertised.

To solve a problem such as one of these, you first need to determine if it is caused either by MultiFinder or an Init. Refer to the quick fixes previously outlined if you're not sure how to test for these two relatively common incompatibilities. For Init incompatibilities, there is not much you can do but stop using either the desk accessory or the Init. For MultiFinder incompatibilities, there is one thing you can try, which is to force the desk accessory to open in the currently active application, rather than in MultiFinder's special desk accessory holding tank. To open a desk accessory in an application while MultiFinder is running:

1. Click on the Apple menu to reveal the list of your desk accessories.
2. Depress the Option key and continue to hold it down.
3. Drag the pointer down to the desk accessory you want to open and release the mouse button.
4. You low release the Option key.

Note that the order in which you perform these steps is important. Don't try to press the Option key first and then open the Apple menu, for example. Note also that this trick only works with some desk accessories, not all of them. Thus it may or may not resolve your problem, but it is worth a try to see what happens.

Another common problem is incompatibility with desk accessory extenders such as *Suitcase*. If you are using a desk accessory extender and the improperly functioning desk accessory is stored in a file which is added by the extender, the problem desk accessory may simply be trying to tell you that it doesn't like being manipulated in that way. Try installing it in the System file and see if that clears up the problem.

If you are able to rule out MultiFinder, Init, and desk accessory extender problems, however, then in all probability the misbehaving desk accessory is incompatible with some other desk accessory under your Apple menu. The trick is to figure out which desk accessory is the culprit (or which desk accessories; there may be more than two in conflict).

If you are using a desk accessory extender such as *Suitcase*, the first thing to do is to remove the extender-added desk accessories one by one (or group by group) and see if you can isolate the conflicting desk accessory that way. If the conflicting desk accessory does turn out to be among those added by your extender, move both it and the misbehaving desk accessory to the System file and see if the problem goes away.

If you are not using a desk accessory extender, or you have ruled out all the desk accessories added by the extender, then simply delete desk accessories one at a time from the System file until the problem no longer occurs. The desk accessory which was deleted just before the problem disappeared should be the problem one. It's a good idea, however, to test to make sure.

Once you have isolated the desk accessories that are in conflict, and if you use an extender such as *Suitcase*, you have determined that moving the problem desk accessories into the System file does not solve the problem, it's time to call the manufacturers. Ask if there is a simple solution to the problem, or a newer version that fixes the problem, or perhaps a "workaround" until an update is available. If one of the conflicting desk accessories is public-exchange, then consult your local user group or electronic bulletin board to see if there is a more recent version or some other solution to the problem. If your inquiries lead you nowhere, then there's not much you can do but choose between the conflicting desk accessories and install one but not the other.

Init Fails to Work

If you have an Init file which is not performing as expected, first check to make certain that the malfunctioning Init file is in the System folder, not within a folder in the System folder or in any other folder. Next, make sure that you have restarted the computer after copying the Init to the System folder. Since Inits must be read by the System file at startup, simply copying them to the System file is not enough to get them going; you must restart the Macintosh before it will work as advertised.

If everything checks out so far, make sure you are using a recent version of the System file. You must be using System 3.2 or higher; earlier versions cannot read Init files. Of course, if you've come this far in the book,

you should be well aware of how important it is to use the most recent System files. System 3.2 is pretty old at this point; if you're using it, you haven't been paying attention.

If the Init has been properly installed and you are using an up-to-date version of the System file, but the Init still does not work properly, you may have a faulty piece of software. Try replacing the one you are using with a new copy from the original disk and see what happens.

Finally, if the Init still fails to work and you have several Inits in your System folder, you might want to try changing the order in which they are read by the System file at startup. Since some Inits prefer (or insist on) being loaded first, changing the loading order may bring a moribund Init file back to life. Inits are loaded in alphabetical order according to their names, so you need only rename the files or add a number or letter to the beginning of each name to change the loading order. Try naming the problem Init so that it is loaded first (if it isn't already), then restart the Macintosh and see what happens. If that doesn't work, try some other possible arrangements.

Beep Sound Fails to Play

If you have a Mac Plus or an SE and are using the technique described in Chapter 5 to change your Beep Sound, first make sure that *BeepInit* is in the System folder and not in any other folder. Then, make sure that the sound file is named `BeepSound` (no more, no less).

If these important points check out, then look to see if the sound file is a large one. Very large sound files sometimes fail to play. (Although it is difficult to pinpoint how large is too large; if the file is over 50K, you should consider this a possibility.) If you think the file may be too large, try a smaller one (one on the order of 10K or 15K) and see if it works. If it does, continue to experiment with larger files until you reach some sort of size limit. Then be sure not to use files larger than the limit.

One other possibility is that your sound file is in the wrong format. Unfortunately, not all digitized sound files are created equal. There are several formats for sound files, and not all of them work with *BeepInit* (or *SoundInit*). Thus, it is possible that your sound file doesn't play because it is in the wrong format. The solution: use a sound file in the correct format (MacNifty's *SoundCap* format or Impulse's *SoundWave* format).

Beep Sound Sounds Weird

Just as records can come in 78 rpm, 45 rpm, or 33 1/3 rpm formats and must be played back accordingly, digitized sound files are recorded at a specific

sampling rate and must be played back at the same rate if you want to hear the sound as it was recorded. If a digitized sound is replayed at something other than the correct sampling rate, it sounds too fast, too slow, or just plain strange. (Of course, if you're looking for a strange sound, like Rocky J. Squirrel with a drawl, playing a sound at the wrong sampling rate may suit you just fine.) Both *SoundInit* and *BeepInit* replay digitized sounds at only one of the more common sampling rates. If your sound file was created at some other sampling rate, you're out of luck. You'll have to use some other sound file.

Screen Saver Fails to Work

I haven't the foggiest idea how screen savers do what they do, but it clearly involves some hexadecimal sleight of hand. Unfortunately, there are times when such programming tricks just don't go down well with an application. For example, you can't activate *AutoBlack* with the pointer when you are using *SuperPaint*. *AutoBlack* still turns the screen off after the preset inactivity period, but for reasons of its own, *SuperPaint* disables *AutoBlack*'s "sleep now" corner (or used to disable it; the problem may be fixed by now) so that you can't turn *AutoBlack* on or off with the mouse.

If your screen saver behaves oddly with a few applications or doesn't work at all with some applications, it's probably because those applications are preventing the screen saver from performing its job. There's not much you can do about it, except perhaps bring it to the attention of the application's manufacturer.

If your screen saver doesn't work at all, then you have a different problem. If your screen saver is an Init, see the earlier entry on Inits which fail to work for suggestions about what to do. If your screen saver is a desk accessory, consult the section on poorly behaved desk accessories for help.

Windows Do Not Change after Using *Layout*

A common misconception about *Layout* is that your windows will take on all the characteristics of your new *Layout* settings as soon as you quit the program. The fact is, only some of the parameters which can be changed in *Layout* appear immediately after using the program. These include the size and position of the default window (the window which opens from a newly created folder or newly initialized disk), the font and font size of folder and file names, and the arrangement of list views.

Existing windows with icon views (either large or small), on the other hand, will look exactly the same as they did before you used *Layout*. If you did not turn on *Layout*'s **Always Grid Drags** option, you can continue to place new icons and drag existing icons to new locations anywhere in the window, as if your new *Layout* icon pattern didn't exist.

To order all your icon view windows in the *Layout* arrangement, you need to perform one more step after quitting *Layout*. That step involves activating each window and selecting the **Clean Up** command from the Special menu. Simply select the **Clean Up** command if you want each icon in the window moved to the nearest grid location in your *Layout* arrangement. Hold down the Option key when you select the **Clean Up** command if you want all your files neatly ordered, row by row according to your *Layout* arrangement, with no empty spots between icons.

The easiest way to keep your icon view windows in *Layout* order, no matter how many new files you add later on or how much you move them around, is to turn on *Layout*'s **Always Grid Drags** option. With **Always Grid Drags** on, new or moved icons will automatically snap into the correct position in the window. If you prefer not to relinquish the ability to place icons anywhere you like, turn **Always Grid Drags** off. Then when you want an icon to snap to the *Layout* grid, hold down the Command key when dragging the icon to a new location.

New Command Shortcuts Fail to Work Properly

After using *ResEdit*, *MenuMaster*, or *MenuKey* to assign a new keyboard shortcut to a menu command, you may discover that your shortcut doesn't work. The key combination might do nothing at all, it might select a command quite different from the one to which you assigned it, or in the worst possible case, it might make your application crash.

There are several reasons why a new keyboard shortcut might fail to work properly. One is that you have duplicated a shortcut already used on some other menu (probably a menu to the left of the one in question). If two commands have the same shortcut, only one will actually work, and the one that doesn't work may well be the one you added. The solution is to assign a different keyboard combination, one that you know is not already in use.

Also, keep in mind that some programs have Command-key shortcuts which are not displayed on the menus. *Word*, for example, uses virtually every letter of the alphabet for its Command-letter keyboard combinations, which is many more than are shown on its menus. Thus, even if you haven't

duplicated a shortcut displayed on any menu, you may have accidentally duplicated one of these "hidden" keyboard combinations. If your new keyboard shortcut doesn't work, check your application's manual or quick reference card to see if you have run afoul of a keyboard command which is not displayed on a menu.

Finally, remember that some programs are not able to respond to keyboard shortcuts for some commands. With such programs, added shortcuts may never work or may function on some occasions but not on others. If you suspect that this might be the source of your problem, then use a macro processor such as *Tempo II* or *QuicKeys* to assign a keyboard shortcut for the command(s) in question.

Macros Fail to Work Properly

The most common cause of this problem is the failure to turn on or install the macro processor program properly. If you use *QuicKeys*, for example, you must be sure that the *QuicKeys* file was in the System folder (and not in another folder inside the System folder) when you started your computer. If it was not, put it there now and restart your Macintosh to launch it properly. Once installed, *QuicKeys* is on and awaiting a command; you don't have to turn it on.

Tempo II must also be in the System folder (and not in another folder inside the System folder) when you start your computer. In addition, to use a *Tempo II* macro, you must first turn *Tempo II* on—either manually by selecting it from the Apple menu, or automatically at startup via *Tempo II*'s Autoboot file (as described on pages 411 to 413). Only after *Tempo II* is turned on and its menu symbol is visible in the menu bar will *Tempo II* run your macros. Also with *Tempo II*, be sure that you have stored the macros in question in the correct macro file and, using the **Configure...** command, that *Tempo II* is reading the correct macro file and not some other file at startup.

Also, remember that macro processors can run afoul of (be incompatible with) Init files and other programs which are memory-resident, programs such as spoolers, RAM caches, and the like. If your macro processor is acting strangely, disengage all your Init files and turn off all other RAM-resident programs and see if that clears up the problem. Then, one by one, turn your Inits and other programs back on to determine which one is causing the problem.

Font Extender Fails to Make Available Certain Screen Fonts, or Certain Font Sizes, or Styles

For people who own a font extender such as *Suitcase*, it makes a lot of sense to install all the roman screen fonts in the System file, create several font files containing all the corresponding italic, bold or bold-italic screen fonts, and use *Suitcase* to install the italic font collection, for example, only when italic characters are needed. It sounds logical, but unfortunately, it doesn't work with the font extenders current when this was written.

The main reason it doesn't work is that all the screen fonts for a given font family must be in the same file when using a font extender like *Suitcase*. For reasons too complex to get into here, only that collection of screen fonts of a given font family which is first encountered is actually available for use. Any other screen fonts in this font family which are encountered later are simply ignored.

For example, assume that you have New Baskerville roman and italic screen fonts installed in the System file, and New Baskerville bold and bold-italic screen fonts in a separate font file that *Suitcase* loads automatically at startup. Since the screen fonts in the System file are always encountered first, only the New Baskerville roman and italic screen fonts will be available for use. Those in the font file will not. The solution to this problem is simple: always put all the screen fonts of a font family in the same file, either the System file or a font file.

There are two reasons why extender-added fonts may not appear on the Font menu at all. One is that if you attempt to add a font with an extender while your application is running, the fonts in the newly added font file will not appear in the Font menu. The reason is that nearly all Font menus are created when an application launches and not re-created until the next time you launch the program. Thus, any changes you make to the list of available fonts while an application is running, such as adding more fonts with *Suitcase*, are ignored. The solution is to always add fonts with your font extender from the Finder, with all relevant applications closed (if MultiFinder is turned on). Once the font file has been added, relaunch your application and go back to work.

The other reason extender-added fonts may not appear in your font menu is font ID conflicts: two fonts having the same font ID number. Font ID conflicts and one way to deal with them are discussed on pages 186 to 187. Another method of dealing with them, if you own *Font/DA Juggler Plus*, is to use one of that program's accompanying utility programs, either *Font/DA Utility* or *Font Resolver*, to eliminate the conflict for you.

Styled Screen Fonts Fail to Appear When Using Style Commands

Here's the problem: you have both roman and italic screen versions of a certain font installed, but when you use the Italic style command on characters of that font, you get Macintosh-generated italic characters, not true italic characters from the italic screen font.

If you use a font extender such as *Suitcase*, the first thing to check is that all font sizes and styles for the entire font family are in the same file. As I mentioned previously, if you have one style screen font, such as Palatino roman, in the System file and another style screen font from the same font family, such as Palatino italic, in a separate font file which is installed by *Suitcase*, then only those screen fonts in the System file, only Palatino roman, will be available. The solution, if you are using a font extender such as *Suitcase*, is to always put all of the screen fonts, all sizes and styles, in a single file (either a font file or the System file).

Next, check to make sure that the font you are using is a smart font, that it knows to use the italic screen font when you use the Italic style command. Nowadays, the fonts from most font manufacturers are smart fonts. You should run into problems only with fonts from lesser known, smaller companies. As far as I am aware, for example, all the fonts from Adobe, Casady, and Century (to name only the obvious ones) which have styled screen fonts are smart fonts. If the font you are using is not a smart font (and if the manual isn't clear on this point, the only way to be sure is to ask the manufacturer), then you'll have to select the italic screen font and not the Italic style command when you want true italic text.

Finally, if you have followed my suggestion and renamed your styled screen fonts so that they disappear (as described on pages 187 to 188), go back and check your work to be sure you didn't make a mistake. If you inadvertently changed the font name (even by adding just an extra space), altered the FOND number, or slipped up somewhere else, your font may not work properly. If you suspect this may be the source of your trouble, the best thing to do is to start over again from the beginning, using a fresh copy of the problem screen fonts. Don't try to repair a botched job; it's too easy to make a bad situation worse.

Downloadable Fonts Fail to Download

A common font problem is that a downloadable font file fails to download, and your document prints in some other font or as an exact copy of the screen at screen resolution. There are many possible causes, including:

- The correct PostScript font file must be in the System folder. Check to be sure that it is, and that it's not in some other folder or missing altogether.
- Your application must be able to download fonts automatically. Check the manual to make sure that it can. If it cannot, you should download them manually before you print, with a font downloader application or with a desk accessory such as *LaserStatus*.
- The LaserWriter must have enough free memory to store the fonts that need to be downloaded. Some programs manage downloadable fonts in such a way that you can use an almost unlimited number per document. In most applications, however, the maximum number of fonts per document is limited to the number of fonts that can fit in the LaserWriter's memory all at one time, which is usually only three or four. If you try to use too many fonts in a single document, later fonts may fail to download, or the document may fail to print altogether. Check the manual to see how your application manages downloadable fonts. Also reread pages 148 to 154 concerning LaserWriter memory management.

There are numerous other PostScript font problems as well. Many are discussed on pages 158 to 161.

Strange Tool Palette or Mouse Pointer Symbols in Some Applications

In some programs, *MacPaint* 1.5, *SuperPaint* 1.0, and *HyperCard* 1.0 among them, the tools in the tool palette and the special mouse pointer symbols are actually characters from a font which is stored within the application. If you have a lot of fonts installed in your System file or added by a font extender such as *Suitcase*, the application's private font may run into a font ID conflict with one of your fonts. The result of such a conflict is letters or an unrecognizable mess for tools and the mouse pointer, instead of the expected symbols.

The best way to resolve this problem is to fix the application's special font using *ResEdit*, so that the problem goes away permanently. The procedure for doing so is fairly complex, and unfortunately, too long to repeat here; if you use *Suitcase*, you'll find the necessary instructions in the `Please Open Me!` file on the *Suitcase* disk.

A less elegant way to resolve the problem is to identify the conflicting font and avoid using it. Experiment by removing fonts, one or more at a time, and then testing the problem application. When you have isolated the problem font, put it in its own font file and then install it with a font extender only on those occasions when you actually need it, and when you are not using the affected application.

Application Bombs When Quitting with a Transfer Utility

Transfer utilities such as *SkipFinder* and *SafeLaunch* keep you from having to return to the Finder when you quit an application; instead, they take you directly to another application. There are a few programs, however, that resent such an intrusion, and they let you know by crashing when you attempt to quit. Some programs, if they lack the Command-Q shortcut for Quit, can be convinced to behave in a more civilized manner simply by assigning Command-Q to the Quit command. If the offending application already has such a shortcut, or fails to respond to the addition of one, then you have two choices: try using a different transfer utility and see if that solves the problem; otherwise, avoid using a transfer utility with the problem program, and quit to the Finder in the usual manner instead.

System Crashes of Unknown Origin

Finally, there are those fatal crashes which seem to come out of nowhere, or the times when your application unexpectedly quits when you are running under MultiFinder. If you want to understand why an oddball crash has occurred, the first thing you need to do is try and reconstruct what you were doing when it happened. Then you should test your notion of what happened by re-enacting the crime, by attempting to crash once again (not with critical data on the screen, of course).

If you can reconstruct what happened, and re-create the crash, then you need to think about what it was that triggered the crash. Were you selecting a specific desk accessory, for example, or an FKey, a new font? If you were doing anything that involved going outside the application itself—for a desk accessory, an FKey, a font—either at the time of the crash or a short time before it, then you should start your search for solutions there. It may be that you have a faulty FKey, for example, one that needs to be replaced; or the desk accessory may be incompatible with your application and should not be used when the application is running.

Around the time the crash occurred, if you were not doing anything that involved going outside of the application, then the task of resolving the problem is more difficult. First you need to make sure that it's not a bug in the application. You can do that by making copies of the application's program and System disks, restarting the Macintosh with these two disks, and then attempting to re-create the crash. If it happens again, there's a good chance that the problem is in the application. If it does not happen again, then the problem is probably in your System folder. Try removing Init files and otherwise paring away at your System folder until the problem goes away. Whatever it was that you last eliminated should contain a clue to the problem.

Finally, if you can't reconstruct what happened accurately and can't re-create the crash, then you are in that never-never land of not quite knowing what happened and not quite knowing how to keep it from happening again. I suppose the take-home message is this: if you want to solve a problem, you've got to know just what the problem is. If you don't have a good grasp of the problem, the solution is just about anyone's guess.

Appendix

C

Products Guide

To help you locate the hardware, commercial software, and public-nexchange software mentioned in the text, this appendix lists the name, address, and phone number of the manufacturer of each product mentioned, and the product's price at the time the book went to press. If your dealer or local software store does not have the item you are looking for, you can call the manufacturer either to obtain it directly or to find out where you can purchase it.

For items that receive more than passing comment, the pages on which relevant descriptions and illustrations begin are listed after the product name. Where the version number of a program may make a difference (in the context of the examples described in the book), the version of the program used during the course of writing the book is also listed. Since popular programs are updated frequently, some of the versions listed here may have been replaced by more recent ones by the time you read this.

As for public-exchange software, I make no attempt to list the many places where you can obtain it. (Public-exchange software, you may recall, is "freeware" or "shareware" software that you may legally copy and give to or receive from others.) For more information about obtaining public-exchange programs, consult *Free (and almost free) Software for the Macintosh* (listed in Appendix D) or ask during the open question session of your next user group meeting.

1st Aid Kit, page 503
1st Aid Software
42 Radnor Road
Boston, MA 02135
(617) 847-4190
(800) 843-3497
$99.95

3D Tic-Tac-Toe (desk accessory)
Steve Christensen
(public-exchange)
free

Acta (desk accessory)
Symmetry Corporation
761 East University Drive
Mesa, AZ 85203
(602) 844-2199 or
(800) 624-2485
$59.95

Adobe Type Library
Adobe Systems, Inc.
P.O. Box 7900
Mountain View, CA 94039-7900
(415) 961-4400
(800) 292-3623
$95 and up per font package

AffiniFile (desk accessory)
Affinity Microsystems, Ltd.
1050 Walnut Street, Suite 425
Boulder, CO 80302
(303) 442-4840
(800) 367-6771
$79.95

Ammortization (desk accessory)
Roger V. Mitchell
(public-exchange)
contribution requested

Ap Font 3.0, page 334
Jim Hamilton
(public-exchange)
$5 shareware fee

Art Grabber+ (part of M.U.D.)
Spinnaker Software Corporation
1 Kendall Square
Cambridge, MA 02139
(800) 826-0706
$49.95

Artisto 1.0 (desk accessory)
Tom Taylor
(public-exchange)
$5 or $10 shareware fee

AutoBlack 1.5, page 233
Itty Bitty Computers
(public-exchange)
$5 shareware fee

BackDown (desk accessory)
Ron Risley
(public-exchange)
$30 shareware fee

Bagels (desk accessory)
Steve Christensen
(public-exchange)
free

BCS•Sleep (desk accessory)
Ken Winograd
(public-exchange)
free

BeepInit 1.3, page 306
Bill Steinberg et al.
(public-exchange)
free

BigCursor FKey
Andy Hertzfeld
(public-exchange)
free

Blackjack (desk accessory)
Brad Slutsky
(public-exchange)
shareware fee

Boston II ImageWriter font
Charles E. Maurer
(public-exchange)
$10.50 shareware fee

Calculator Construction Set
Dubl-Clik Software, Inc.
18201 Gresham Street
Northridge, CA 91325
(818) 349-2758
$59

Calendar (desk accessory)
Mike Schuster and David Oster
(public-exchange)
$5 shareware fee

Camera (desk accessory) page 98
Keith A. Esau
(public-exchange)
free

Canvas (desk accessory)
Deneba Software
7855 N.W. 12 Street, Suite 202
Miami, FL 33126
(305) 594-6965
(800) 622-6827
$99.95

CheapPaint (part of M.U.D.)
Spinnaker Software Corporation
1 Kendall Square
Cambridge, MA 02139
(800) 826-0706
$49.95

Coach Merriam-Webster Thesaurus
Deneba Software
7855 N.W. 12 Street, Suite 202
Miami, FL 33126
(305) 594-6965
(800) 622-6827
$59.95

Coach Professional
Deneba Software
7855 N.W. 12 Street, Suite 202
Miami, FL 33126
(305) 594-6965
(800) 622-6827
$195

Coach Spelling Checker 3.0
Deneba Software
7855 N.W. 12 Street, Suite 202
Miami, FL 33126
(305) 594-6965
(800) 622-6827
$99.95

Comment (desk accessory)
Deneba Software
7855 N.W. 12 Street, Suite 202
Miami, FL 33126
(305) 594-6965
(800) 622-6827
$99.95

Complementary Type
Software Complement
8 Pennsylvania Avenue
Matamoras, PA 18336
(717) 491-2492 or 491-2495
$69 per disk

Copy II Mac, Version 7, page 480
Central Point Software, Inc.
9700 SW Capitol Highway, #100
Portland, OR 97219
(503) 244-5782
$39.95

Copy Screen FKey
Joe Miller
(public-exchange)
free

DA Key FKey, page 74
Loftus E. Becker, Jr.
(public-exchange)
$5 shareware fee

DA-Switcher 1.0
Olduvai Software
6900 Mentone
Coral Gables, FL 33146
(305) 665-4665
(800) 628-2828
$49.95

Date Key FKey
Loftus E. Becker, Jr.
(public-exchange)
$5 shareware fee

dCAD Calculator (desk accessory)
Desktop CAD
(public-exchange)
$19.95 shareware fee

Desk Necessities
MicroSparc
52 Domino Drive
Concord, MA 01742
(617) 371-1660
$29.95

DeskPaint (desk accessory)
Zedcor, Inc.
4500 East Speedway, Suite 22
Tucson, AZ 85712-5305
(602) 881-8101
(800) 482-4567
$59.95

Desktop Express
Dow Jones & Company
P.O. Box 300
Princeton, NJ 08543-0300
(609) 452-1511
$149

DialogKeys, page 276
see *QuicKeys*

DiskExpress, page 490
ALSoft, Inc.
P.O. Box 927
Spring, TX 77383-0927
(713) 353-4090
$39.95

DiskTools II (part of DiskTools Plus), page 111
Electronic Arts
1820 Gateway Drive
San Mateo, CA 94404
(415) 571-7171
(800) 448-8822
$49.95

DiskTop 3.0.3, (desk accessory) page 111
CE Software
P.O. Box 65580
West Des Moines, IA 50265
(515) 224-1995
(800) 523-7638
$49.95

Dvorak FKey
(public-exchange)
free

EZ-Menus, page 232
see *MockPackage Plus*

Expressionist
Alan Bonadio Associates
1579 Dolores Street
San Francisco, CA 94110-4928
(415) 282-5864
$54.95

FadeKey FKey
Carlos Weber
(public-exchange)
free

Fastback
Fifth Generation Systems, Inc.
11200 Industriplex Boulevard
Baton Rouge, LA 70809
(504) 291-7221
(800) 225-2775
$99

FEdit+
MacMaster Systems
939 East El Camino Real, Suite 122
Sunnyvale, CA 94087
$40

Findswell
Working Software, Inc.
321 Alvarado, Suite H
Monterey, CA 93940
(408) 375-2828
(800) 851-1986
$49.95

FKey Installer (part of Quick & Dirty Utilities, Volume On
Dreams of the Phoenix
P.O. Box 10273
Jacksonville, FL 32247
(904) 396-6952
$39.95

FKey Maker, page 98
Loftus E. Becker, Jr.
(public-exchange)
free

FKey Manager 3.0, page 81
Carlos Weber
(public-exchange)
free

FKey Sampler
David Kallin
(public-exchange)
free

FKey-DA Sampler
John Holder
(public-exchange)
$10 shareware fee

Fluent Fonts
CasadyWare
P.O. Box 223779
Carmel, CA 93922
(408) 646-4660
$49.95

Fluent Laser Fonts
CasadyWare
P.O. Box 223779
Carmel, CA 93922
(408) 646-4660
$69.95 per font package

Demo

Focal Point, page 374
Activision, Inc.
P.O. Box 7286
Mountain View, CA 94039
(415) 960-0410
$99.95

Font/DA Juggler Plus 1.0, pages 180 and 181
ALSoft, Inc.
P.O. Box 927
Spring, TX 77383-0927
(713) 353-4090
$49.95

FONTastic Plus, page 188
Altsys Corporation
720 Avenue F, Suite 108
Plano, TX 75054
(214) 424-4888
$79.95

Fontographer, page 198
Altsys Corporation
720 Avenue F, Suite 108
Plano, TX 75054
(214) 424-4888
$395

Fontographer Fonts
Altsys Corporation
720 Avenue F, Suite 108
Plano, TX 75054
(214) 424-4888
$59.95 per font

Fontsie 1.53 and *3.08*, page 75
Loftus E. Becker, Jr.
(public-exchange)
$10 shareware fee

FullPaint
Ann Arbor Softworks, Inc.
2393 Teller Road, Suite 106
Newbury Park, CA 91320
(805) 375-1467
$99.95

Fullwrite Professional
Ann Arbor Softworks, Inc.
2393 Teller Road, Suite 106
Newbury Park, CA 91320
(805) 375-1467
$295

GetInfo FKey
Carlos Weber
(public-exchange)
free

Graphidex
BrainPower
24009 Ventura Boulevard
Calabasas, CA 91302
(818) 884-6911
(800) 345-0519
$124.95

Hard Disk Partition, page 50
FWB Software, Inc.
2040 Polk Street, Suite 215
San Francisco, CA 94109
(415) 474-8055
$54.95

Hard Disk Util, page 479
FWB Software, Inc.
2040 Polk Street, Suite 215
San Francisco, CA 94109
(415) 474-8055
$89.95

HFS Backup, page 500
Personal Computer Peripherals Corporation
6204 Benjamin Road
Tampa, FL 33634
(813) 884-3092
(800) 622-2888
$49.95

HFS Locator Plus
PBI Software, Inc.
1163 Triton Drive
Foster City, CA 94404
(415) 349-8765
$39.95

HFS Navigator, page 427
Think Technologies Division
Symantec Corporation
135 South Road
Bedford, MA 01730
(617) 275-4800
(800) 648-4465
$59.95

HyperCard, page 364
Apple Computer, Inc.
20525 Mariani Avenue
Cupertino, CA 95014
(408) 996-1010
$49.95

HyperCharger 020
General Computer Corporation
215 First Street
Cambridge, MA 02142
(617) 492-5500
(800) 634-9737
$1499

Icon-It!, page 433
Olduvai Software
6900 Mentone
Coral Gables, FL 33146
(305) 665-4665
(800) 628-2828 ext. 751
$79.95

Idle (desk accessory)
Larry Rosenstein
(public-exchange)
free

Illustrator
Adobe Systems, Inc.
P.O. Box 7900
Mountain View, CA 94039-7900
(415) 961-4400
(800) 292-3623
$495

Init FKey
Scott Winders and Alan Pavlish
(public-exchange)
free

JClock
James T. Sulzen
(public-exchange)
free

LaserFonts
Century Software, Inc.
c/o MacTography
702 Twinbrook Parkway
Rockville, MD 20851
(301) 424-1357
$34.95 or $44.95 per font

LaserSpeed, page 470
Think Technologies Division
Symantec Corporation
135 South Road
Bedford, MA 01730
(617) 275-4800
(800) 648-4465
$99

LaserStatus, page 115
see *DiskTop*

LaserWriter Plus Screen Fonts, page 152
Adobe Systems, Inc.
P.O. Box 7900
Mountain View, CA 94039-7900
(415) 961-4400
(800) 292-3623
$50

Layout 1.3, page 222
Michael C. O'Connor
(public-exchange)
free

Long Beach ImageWriter font
(public-exchange)
free

Lookup
Working Software, Inc.
321 Alvarado, Suite H
Monterey, CA 93940
(408) 375-2828
(800) 851-1986
$74.95

MacDraw
Claris Corporation
440 Clyde Avenue
Mountain View, CA 94043
(415) 960-1500
$195

MacinTalk
Apple Computer, Inc.
20525 Mariani Avenue
Cupertino, CA 95014
(408) 996-1010
(public-exchange)

MacKeymeleon, page 264
Avenue Software, Inc.
1173 West Charest Boulevard,
Suite 390
Quebec City, Quebec
Canada G1N 2C9
(418) 682-3088
$49.95

MacPaint 1.5
Claris Corporation
440 Clyde Avenue
Mountain View, CA 94043
(415) 960-1500
$125

MacPS, page 410
Neff Systems Group
3286 Knightswood Way
San Jose, CA 95148
(408) 274-1110
$19

MacServe, page 49
Infosphere, Inc.
4730 SW Macadam Avenue
Portland, OR 97201
(503) 226-3620
$250

MacWrite 4.5
Claris Corporation
440 Clyde Avenue
Mountain View, CA 94043
(415) 960-1500
$125

Maxwell's Demon desk accessory
(public-exchange)
free

MDS Edit
Bill Duvall
(public-exchange)
free

MenuFonts 1.0, page 290
Beyond
3865 North Oracle Road
Tucson, AZ 85705
(602) 323-4547
$15

MenuMaster, pages 76 and 245
Fred Reed
(public-exchange)
$5 shareware fee

MicroPhone 1.06
Software Ventures Corporation
2907 Claremont Avenue, Suite 220
Berkeley, CA 94705
(800) 336-6477
(800) 336-6478 in CA
(800) 336-6479 in Canada
$149

Microsoft Chart
Microsoft Corporation
16011 NE 36th Way
Box 97017
Redmond, WA 98073
(206) 882-8080
$125

Microsoft Excel
Microsoft Corporation
16011 NE 36th Way
Box 97017
Redmond, WA 98073
(206) 882-8080
$395

Microsoft File
Microsoft Corporation
16011 NE 36th Way
Box 97017
Redmond, WA 98073
(206) 882-8080
$195

Microsoft Word 3.01
Microsoft Corporation
16011 NE 36th Way
Box 97017
Redmond, WA 98073
(206) 882-8080
$395

MiniWRITER
David Dunham
(public-exchange)
$12 shareware fee

MockPackage Plus
CE Software
P.O. Box 65580
West Des Moines, IA 50265
(515) 224-1995
(800) 523-7638
$35

Modern Jazz
Lotus Development Corporation
55 Cambridge Parkway
Cambridge, MA 02142
(617) 623-6572

Moose Phrazer
Jan Eugenides
(public-exchange)
free

More
Living Videotext
117 Easy Street
Mountain View, CA 94043
(800) 822-3700
(800) 443-4310 in CA
$295

Moving Lines FKey
John Raymonds
(public-exchange)
free

Multi-Scrap
Bob Luce
(public-exchange)
free

Navigator
CompuServe, Inc.
5000 Arlington Centre Boulevard
Columbus, OH 43220
(614) 457-0820
(800) 848-8199
$59.95

New Scrapbook
Tim Wasko
(public-exchange)
contribution requested

Oasis 2.01, page 347
Jan Eugenides
(public-exchange)
$15 shareware fee

Omnis 3 Plus
Blyth Software, Inc.
2929 Campus Drive, Suite 425
San Mateo, CA 94403
(415) 571-0222
$495

Other..., page 130
Loftus E. Becker, Jr.
(public-exchange)
$5 shareware fee

OverVue 2.0
ProVue Development Corporation
222 22nd Street
Huntington Beach, CA 92648
(714) 969-2431
$295

PageMaker 2.0a
Aldus Corporation
411 First Avenue South, Suite 200
Seattle, WA 98104
(206) 622-5500
$495

Paint Cutter (part of Accessory Pak I)
Silicon Beach Software
P.O. Box 261430
San Diego, CA 92126
(619) 695-6956
$39.95

Password FKey
William Steinberg
(public-exchange)
free

Pathname FKey
(public-exchange)
free

PictureBase
Symmetry Corporation
761 East University Drive
Mesa, AZ 85203
(602) 844-2199 or
(800) 624-2485
$99

PowerDisk 1.5, page 457
Software Power Company
P.O. Box 14133
Fremont, CA 94539
(415) 490-6086
$59.95

PowerOp 1.0, page 486
Software Power Company
P.O. Box 14133
Fremont, CA 94539
(415) 490-6086
$59.95

Demo

PowerStation 2.3, page 486
Fifth Generation Systems, Inc.
1120 Industriplex Boulevard
Baton Rouge, LA 70809
(504) 291-7221
(800) 225-2775
$59.95

Prague ImageWriter font
(public-exchange)
free

Prodigy 4
Levco
6160 Lusk Boulevard, Suite C-203
San Diego, CA 92121
(619) 457-2011
$3499

Prodigy SE
Levco
6160 Lusk Boulevard, Suite C-100
San Diego, CA 92121
(619) 457-2011
$1499

Professional Type Fonts
Kensington Microware Ltd.
251 Park Avenue South
New York, NY 10010
(212) 475-5200
(800) 535-4242
$49.95 or $69.95 per font

Pyro! 2.1, page 233
see *PowerStation* or *Suitcase*

QDial desk accessory
Leo Laporte
(public-exchange)
free

Quick & Dirty Utilities, Volumes One and Two
Dreams of the Phoenix
P.O. Box 10273
Jacksonville, FL 32247
(904) 396-6952
$39.95 per volume

QuicKeys 1.1, pages 99, 100, 101, 102, 252, 256 and 270
CE Software
P.O. Box 65580
West Des Moines, IA 50265
(515) 224-1995
(800) 523-7638
$99.95

QuickPaint (desk accessory)
EnterSet, Inc.
2380 Ellsworth
Berkeley, CA 94720
(415) 549-0539
(800) 621-0851, ext. 305
$49

RamSnap, page 448
Dove Computer Corporation
1200 North 23rd Street
Wilmington, NC 28405
(919) 763-7918
(800) 622-7627
$29.95

RamStart, page 447
George A. Nelson
(public-exchange)
free

Randomizer, page 306
Jon Pugh
(public-exchange)
free

ResEdit, pages 187 and 242
Apple Computer, Inc.
20525 Mariani Avenue
Cupertino, CA 95014
(408) 996-1010
(public-exchange)

RPN Calculator (desk accessory)
Apple Computer, Inc.
20525 Mariani Avenue
Cupertino, CA 95014
(408) 996-1010
(public-exchange)

Rubik's Cube (desk accessory)
(public-exchange)
free

SafeLaunch 2.2, page 341
Fred Reed
(public-exchange)
$5 shareware fee

Science Calculator (desk accessory)
Rainmaker Research
(public-exchange)
free

Screen Maker, page 340
Bill Atkinson
(public-exchange)
free

Screen To Paint FKey
Steve Maller
(public-exchange)
free

Screen Saver FKey
Mike Cohen
(public-exchange)
free

Servant 0.952, page 385
Andy Hertzfeld
(public-exchange)
free

Servant documentation
Scott Kronick
Servant Documentation
1442A Walnut Street, Suite 278
Berkeley, CA 94709
$10

Set Paths, page 46
Paul F. Snively
(public-exchange)
$20 shareware fee

SideKick 2.0
Borland International
4584 Scotts Valley Drive
Scotts Valley, CA 95066
(408) 438-8400
$99.95

SkipFinder, page 119
Darin Adler
(public-exchange)
contribution requested

Sleep desk accessory
(public-exchange)
free

Smart Alarms and Appointment Diary
(desk accessory)
Imagine Software
19 Bolinas Road
Fairfax, CA 94930
(415) 453-3944
$49.95

Smartcom II
Hayes Microcomputer Products, Inc.
P.O. Box 105203
Atlanta, GA 30348
(404) 449-8791
$149

SmartScrap & The Clipper
Solutions, Inc.
P.O. Box 989
Montpelier, VT 05602
(802) 229-9146
$59.95

SoundInit 1.3, page 306
Bill Steinberg et al.
(public-exchange)
free

Space Warp FKey
(public-exchange)
free

Spellswell
Working Software, Inc.
321 Alvarado, Suite H
Monterey, CA 93940
(408) 375-2828
(800) 851-1986
$74.95

SS Key FKey
(public-exchange)
free

StartMac 1.1, page 403
Neff Systems Group
(public-exchange)
$13 shareware fee (includes upgrade to StartMac 2.0)

Suitcase 1.2, pages 180 and 181
Fifth Generation Systems, Inc.
1120 Industriplex Boulevard
Baton Rouge, LA 70809
(504) 291-7221
(800) 225-2775
$59.95

SuperFinder 4.2, page 351
Matthew Russotto
(public-exchange)
$20 shareware fee

SuperLaserSpool, page 470
SuperMac Technology
295 North Bernardo Avenue
Mountain View, CA 94043
(415) 964-8884
$150

SuperPaint, page 302
Silicon Beach Software
P.O. Box 261430
San Diego, CA 92126
(619) 695-6956
$99

Switcher (Switcher Construction Set), page 389
Apple Computer, Inc.
20525 Mariani Avenue
Cupertino, CA 95014
(408) 996-1010
$19.95

Talking Moose desk accessory
Steve Halls
(public-exchange)
$15 shareware fee

Tempo II, pages 100, 258 and 410
Affinity Microsystems, Ltd.
1050 Walnut Street, Suite 425
Boulder, CO 80302
(303) 442-4840
(800) 367-6771
$149.95

ThinkTank 2.0
Living Videotext
117 Easy Street
Mountain View, CA 94043
(800) 822-3700
(800) 443-4310 in CA
$195

Thunder! 1.1
Electronic Arts
1820 Gateway Drive
San Mateo, CA 94404
(415) 571-7171
(800) 448-8822
$49.95

Time Logger
Loftus E. Becker, Jr.
(public-exchange)
$5 shareware fee

TopDesk 2.1, page 116
Cortland Computer, Inc.
P.O. Box 9916
Berkeley, CA 94709
(415) 845-1142
$59.95

Version Reader FKey
John Holder
(public-exchange)
free

VideoWorks II
MacroMind, Inc.
1028 West Wolfram Street
Chicago, IL 60657
(312) 871-0987
$195

Vancouver ImageWriter font
Jason Kanter
(public-exchange)
free

Wall Street ImageWriter font
(public-exchange)
free

Willowdale ImageWriter font
(public-exchange)
free

Window (desk accessory)
Marsh Gosnell
(public-exchange)
$5 shareware fee

Window FKey
Charlie Bounds
(public-exchange)
free

Window Manager FKeys
Michael Long
(public-exchange)
free

Word Finder
Microlytics, Inc.
300 Main Street
East Rochester, NY 14445
(716) 377-0130
(800) 828-6293
$59.95

WordPerfect
WordPerfect Corporation
288 West Center Street
Orem, UT 84057
(801) 225-5000
$395

World Class Fonts, Volumes One and Two
Dubl-Clik Software, Inc.
18201 Gresham Street
Northridge, CA 91325
(818) 349-2758
$39 per volume

Appendix

D

Further Reading

The following is a list of the books and magazines mentioned in the text. Some are easy reading, while others will take a great deal of concentration and effort. All have something to offer the fully powered Mac owner.

The Complete HyperCard Handbook
by Danny Goodman
Bantam Computer Books
666 Fifth Avenue
New York, NY 10103
(212) 765-6500

Free (and almost free) Software for the Macintosh
by Robert C. Eckhardt
Crown Publishers
225 Park Avenue South
New York, NY 10003
(212) 254-1600

Inside Macintosh
by Apple Computer
Addison-Wesley
Route 128

Reading, MA 01867
(617) 994-3700

LaserSampler II
MacTography
702 Twinbrook Parkway
Rockville, MD 20851
(301) 424-1357

Macintosh Revealed, Volumes One and Two, Second Edition
by Stephen Chernicoff
Howard W. Sams and Company
4300 West 62nd Street
Indianapolis, IN 46268
(317) 298-5699
(800) 428-7267

MacUser
Subscription Department
P.O. Box 52461
Boulder, CO 80321-2461
(800) 852-5200

Macworld
Subscription Department
P.O. Box 51666
Boulder, CO 80321-1666
(800) 524-3200

Index

About the Finder...window, 28, 381–382
accelerator board, 15–16
accessories, *see* Desk accessories
Acta, 44, 103, 117, 118, 252–254, 552
add-on boards, 5, 8
AffiniFile, 105, 106, 552
alarms, 108, 118
Aldus Prep, 24, 25, 31, 38, 185, 470
alert box, 79, 411
alphabetical-plays sequence, 60–63
Always Grid Drags command, 229, 543
Amortization, 241, 552
animation, 229
Apfront, 334, 398, 515, 552
Apple Menu, 123, 286–289, 311, 452, 458
Apple Talk, 23, 49, 109
AppleLink, 53
applications, *xx*, *xxiv*
  copy-protected, 34
  desk accessories in, 126–130
  file sorting by, 27, 40–42
  FKeys in, 97
  fonts in, 173–175
  megaprogram, *see* Megaprogram, applications
  memory size, 380–382, 386
  in MultiFinder, 385
  organization on hard disk, 40–45
  organization in System folder, 35–40
  in *Servant*, 386
  shortcuts, *see* Keyboard shortcuts
  support files for, 41, 46, 48
  window, 41, 46, 48
Appointment Diary, 107, 108, 125
Area Code Lookup, 105, 106
arrow keys, 423–424
art, ready-made, 113–114, 300
Art Grabber+, 113, 552
Artisto, 103, 113, 552
ASCII, 60, 200, 201, 207
Auto Mounter file, 50
AutoBlack, 26, 119, 233–235, 311, 332, 515, 542, 552
  MacsBug version, 532–533
Available Key, 73–74

BackDown, 121, 552
background programs, 27, *see also* Print spooler, RAM cache, Screen-saving function
Backgrounder file, 27
backup, *xx*, *xxv*, 243, 321, 508, 509
  applications, 495
  automated, 410
  of copy-protected applications, 479–482
  documents, 495–500
  for fragmented files, 485
  identical, 499
  periodical, 496–498, 509
  printed, 498
  rotated, 499–500
  system, 494–495
Bagels, 112, 552
*BCS*Sleep*, 119, 552
BeepInit, 306, 312, 519, 541–542, 552
  and error beep, 307, 310
BigCursor, 74, 553
bitmap fonts, 144–146, 149
Blackjack DA, 112, 113, 553
Blank, 119
boot blocks, 528–530
Boot Paths file, 46
Boston II, 147, 553
brackets, 79
Browse tool, 370, 373
Build Paths, 47
built-in fonts, *see* Fonts, resident
bulletin boards, *xv*, 29, 121, 479
burned-in image, 119, 233, 311

By Icon command, 57, 240, 291–295
By Name command, 57, 240
By Name view, 58, 59, 68, 226

Calculator Construction Set, 107, 553
Calculator+, 107
calculators, 106–107
Calendar, 107, 553
Calendar Book, 107
Calendar Key, 74
calendars, 74, 107, 375
Camera, 98, 108, 115, 237–239, 275, 311, 312, 553
Canvas DA, 112, 553
Character command, 335
Character editing window, 200, 202, 203, 209
Chart, 103, 117, 559
charts, 116–117
CheapBeep, 110, 306, 307–309
CheapPaint, 112, 553
Chooser, 28, 29, 109, 138
  changing name, 287–289
  installing, 125
circuit board, 5
Clean Up command, 57, 222, 240–241, 543
  in *PowerStation*, 357, 358
Clipboard, 6, 74, 77, 119, 394
Clipper, The, 119, 563
clocks, 108, 232, 233, 234, *see also* Alarms
Close Window, 80
Coach, 120
Coach Merriam-Webster Thesaurus, 122, 553
codes, *xxii*, 60–62, 64–66, 68, 78, 447
commands, 236, 317–321, 410, 425–426
  combined, 240–241, 322–324, 397
  new, 255–263
  shortcuts, 245–252, 255–263, 311
  standardization of, 317–323
comma, changing, 79, 264, 265, 268, 270–272
Command-D, 320–321
Command-Q, 548
Command-X, 321
Comment, 109, 553
Compaq, 10, 140
Compatibility problems, *xvi*, 16, 221, 389
Complete HyperCard Handbook, The, 365, 373, 567
CompuServe, *xv*, 28, 29, 147, 411
  Adobe forum, 151
  MAUG, 300, 306, 307
  Navigator, 53
configuration files, 25
Configure KeyPad...command, 95
Configure Pop-Keys...command, 90, 91
Control Panel, 25, 100, 110, 138, 235, 295
  error beep in, 307–309
  files, 25, 110, 334
  RAM cache, 404, 457, 458, 465, 466, 474
  installing, 125
  versions, 28
Copy command, 203, 316–318, *see also* Commands
Copy protection, *xx*, *xxv*, 32, 34, 124, 476–482, 508
  overcoming, 478–482, 508
  and RAM cache, 454–455
  and RAM disks, 442
Copy Screen, 74, 554
Copy II Hard Disk, 480–482
Copy II Mac, *xv*, 480–482, 508, 554
Copyright symbol, 265–267
Cricket Draw, 470
Custom help screen, 110–111
Cut command, 316–318, 321, *see also* Commands

DA Helper file, 27
DA Key, 74, 75, 130–132, 554
DA-Switcher, 111, 554
Data →Mask command, 281–282
Date Key, 74, 75, 554
dBASE Mac, 38
dCAD Calculator, 107, 554
default application font, 334, 398
default window, 230–231
Define Styles...command, 335
Delete Column command, 321
Desk accessories, *xix*, *xxii*, *xxiii*, 70–71, 98, 103–136, 138, see also *FKey Maker*, FKeys, *Top Desk*
  adding, 132–133
  extenders, *xxiii*, 71, 72, 111, 125, 130–136, 138
  failure, 539–540
  vs. FKeys, 72–73
  ID numbers, 128–130, 131, 136
  installing in applications, 126–130
  installing in System file, *xiv*, 124–126
  slots for, 70, 72, 126
  and support files, 125
DeskPaint, 103, 113, 114, 554
Desktop, *xiv*, 23, 181, 220, 221, 231
  command shortcuts, 236–312
  menus, 232–235

pattern, 295–296, 309, 312
Desktop Express, 338–339, 341, 346, 392, 554
Desktop file, 282–285
compacting, 487–488, 490, 493
desktop publishing, 25, 172
dialog box
custom, 276–278, 312
File Selection, xiv
shortcuts, 423, 474
in *Word*, 425–426
DialogKeys, 277–278, 312, 422, 554
dictionaries, 6, 41; *see also* Spelling checkers
direct-transfer utilities, *xxiv*, 337–345, 398, 445, 548
Disk First Aid, 503
Disk optimization, 486–491, 508; *see also DiskExpress, PowerOp*
DiskExpress, *xv*, 488–493, 508, 554
DiskFit, 501–502
DiskInfo, 111–112
DiskTools II, 111, 530, 554
direct transfer, 338, 340, 346, 361–363
Finder substitute, 355, 359, 376, 396
Launching On option, 362–363
Quick Launch list, 363, 399
DiskTop, 111, 238, 292, 393, 530, 555
Add to Menu command, 360
direct transfer, 338, 340, 346, 355, 359–361
Finder substitute, 355, 359, 376, 396, 398–399
and script files, 447
Double QuickQuotes, 272–273, 312
DRVR resource, 287, 289
Dvorak, 75, 555

electronic mail, 120, 411
em-dash, changing, 265, 268
error beep, 305, 306, 307–309, 310, 312
Excel, 6, 38, 103, 116, 354, 559
Expressionist, 111, 555
EZ-Menu, 232–233, 311, 555

FadeKey, 75, 555
Family Builder, 169
Fastback, 501, 555
FEdit+, 77,555
file organization, *xix*, *xxi*, 20–21, 33–36
hierarchical system, 33
rules for, 33–34, 67
sorting by application, 40–42
sorting by file type, 43–44
file selection box, 59, 60, 64, 337, 398
shortcuts, 423–424, 427, 474
file transfer utility; *see* Direct-transfer utility
files
fragmentation; *see* Fragmented files
naming, xxii, 59–66
prioritizing, 58–59
restoration, 501, 502
sorting, 40–44
splitting, 32
subdividing, 42
Find File, 112
Finder, *xii*, *xxiii–xxiv*, 23, 51, 227, 310, 376
vs. DOS, 314
and FKeys, 70, 78
and fonts, 228
launch sequence, 526, 527
LAYO, 222
and MultiFinder, 279, 380
and shortcuts, 236, 242, 245, 251, 256
substitutes, *xxiv*, 337, 345–364, 445
versions, *xvi*, 28, 29
View menu, 291–295
viewing files, 56
Findswell, 432, 555
1st Aid Kit, *xv*, 503–508, 509, 527–530, 535, 552
FKey Installer, 80, 555
FKey Maker, 98–99, 100, 137, 515, 555
FKey Manager, *xxii*, 23, 80–99, 137, 516, 555
FKey Sampler, 75, 89, 555
FKey-DA Sampler, 75, 555
FKeys, *xix*, *xxii*, 23, 70–103
in applications, 97
vs. desk accessories, 72–73, 137
homemade, 98–99
installing, *xxii*, 80, 86–89
naming, 82–84
removing, 84–86, 137
as screen savers, 119
slots for, 70
testing, 80
FKeys command, 101
FKToggle, 79
Floppy disk drive, *xxi*, 3
and *ResEdit*, 243, 244
Floppy disks, 9, 482
and applications, 25

backup to, 10, 243; *see also* Backup
daily backup, 497–498
and downloadable fonts, 183–184
failure, 9
with *FKey Manager*, 81–82
formatting, 483
and fragmentation, 484, 485, 508
organizing files on, 35–40, 67
single-sided/double-sided, 4, 32, 505–506
and startup, 526
windows, 231
Focal Point, 364, 374–376, 517, 556
Document Launcher, 364, 375, 516
Folders, *xiv*, *xxii*, 33–34, 40; *see also* System folder
Desk Accessory, 131, 132, 133
naming, 59–66
sorting, 43–44
storing, 42
for support files, 46
windows for, 53–55
FOND resource, 195–196, 546
Font Downloader, 413–414, 416
FONTastic Plus, *xxiii*, 187, 188–192, 213, 217, 556
Font/DA Juggler Plus, 136, 138, 180–182, 216, 255, 556
and desk accessory ID numbers, 130, 175
and FKeys, 89
Font Resolver, 545
Font/DA Utility, 136, 175, 181, 188, 217, 286, 311, 545
and sound files, 111
Font/DA Mover, xiv, 23
and font ID numbers, 175
installation of desk accessories, 70, 123, 124–126, 130, 131
installation of fonts, 144, 159, 164–165, 174, 180
installation of *Set Paths*, 46
versions, 28, 29
Font/desk accessory extender, 173, 176–182, 216; *see also Font/DA Juggler Plus*, *Suitcase*
Fontographer, *xxiii*, 169, 198–215, 217, 556
Fonts, *xix*, *xvi*, *xxiii*, 70–71, 133, 141–217, 227–229
application-specific, 32
bitmap, 144–146, 149
border, 157
categories, 142–144
customizing, 185–186
decorative, 144, 148, 156, 215
downloadable, xxiii, 26, 59, 115, 151, 152–154, 156, 161, 172, 182–185, 472
extenders, 33, 72, 75, 176–182, 545
free, 147
ID numbers, 159, 175, 177, 181, 188, 190, 199–200
and master System file, 31, 140, 141
menu, 177, 217, 241, 290–291
private, 547
problems, 158–161
resident, 151–152, 154, 157, 161
and screen legibility, 168
sizes, 145, 162–172
window, 201, 210, 211
Fontsie, 75–76, 173, 176–180, 216, 517, 556
as configuration program, 178–179
FKey, 177, 178
as Init resource, 177, 178–180
versions, 178
Fontsie Special, 76
fractions, 198–215, 217
fragmented files, *xx*, *xxv*, 482–493, 508
copying, 484
and disk optimizers, 486–491
Free (and Almost Free) Software for the Macintosh, 147, 551, 567
freeware, *see* Software, public exchange
FullPaint 1.0, 278, 560
FullWrite, 120, 556
function keys, 102; *see also* FKeys
dedicated, 95–97

Games, 112
Get Info, 76
Get Info box, 28, 56, 282, 380, 385, 386, 432, 487
Get Info command, 285, 432
Get Info Key, 76, 556
graphics, 112–114
Graphidex, 114, 556

hard disk drive, 9–10, 527
ID number, 12
internal/external, 12–13
multiple, 12
removable media, 14
subvolume utility, 49
Hard Disk Partition, 50, 68, 557
Hard Disk Util, *xv*, 479–482, 508, 557
hard disks, *xxi*, 2, 9, 17, 450, 483
backup, xv, 34, 67, 90, 494–498, 500–502, 508;

see also Backup
boot blocks on, 529–530
and copy protection, 477–482, 507
cost, 11, 14
failure, *xv*, 9–10, 13
Finder, 23, 527
fragmented, 484–486, 508
inserting/removing, 14
and optimization process, 487–493
organizing files on, 40–50
reliability, 12
size, 10–12
and startup failure, 526–528
subdividing, *xxiii*, 48–50
and System folder, 30, 66–67, 72
and System files, 32, 66–67
volume information, 504
windows, 231
hardware, *xix*
buffers, 468
and copy protection, 477
failure, xx
ideal setup, 5
minimum setup, 2, 4, 17
Helvetica, 155
HFS Backup, 500–502, 557
HFS Locator Plus, 112, 557
HFS Navigator, *xv*, 43, 474, 557
installing, 427, 531
menu, 428–432
Hierarchical filing system (HFS), 33, 48, 505–506
Hot spot, 90–93
HyperCard, *xxiv*, 5, 6, 7, 112, 398, 516, 557
as Finder substitute, 364–377
organization schemes, 376–377
and RAM, 444, 445–446
search paths, 373–374
HyperCard, 1.0, 457

IBM PC, 10–11, 314, 315, 319
icon, *xiv*, *xviii*, *xx*, 34, 221, 278–279, 427
custom, 278–285, 312
displaying, 52–53, 56
in Desktop file, 487
document, 27
in Finder, 40
ID numbers, 283, 285
mask, 280
in *Servant*, 387, 388
size, 58, 223–227
spacing, 223–225
special, 27–28
icon bars, 433–435, 474
icon views, *xxii*, 58–59, 68, 221, 222, 223–226
small, 225–226
standard, 223–225, 231
Icon-It!, *xv*, 427, 433–435, 474, 557
Icon Bar Editor, 433, 434
installing, 433
Idle, 119, 557
Illustrator, 40, 110, 126, 470, 557
ImageWriter, *xix–xx*, 23, 26, 121, 402
with FKeys, 82, 84
fonts, 142–148, 150, 158, 162–168
print spooler for, 49
quotation marks on, 268
Init, 76, 557
Init resources, 25–26, 80, 232, *see also Fontsie*
problems, 533–534, 539, 540–541
Inside Macintosh, 245, 567
Install EZ-Menu command, 232
Instruction cache, 16
Integrated programs, 314, 335–397; *see also* Megaprogram, applications
ample-memory, 337–397, 399
limited-memory, 336–337, 398–399

JClock, 233, 311, 558
JustText, 24

Key Caps, 114–115, 125, 157
window, 265, 266, 274–276, 312
Keyboard, 72
configurations, 75, 76, 79, 80
custom, 264–276
Dvorak layout, 264, 265
maps, 108, 115, 274, 312
QWERTY row, 200
shortcuts; *see* Keyboard shortcuts
keyboard shortcuts, *xx*, 101–102, 116, 245–263, 276, 297, 311, 312
applications, 424–426, 474
failure in, 543–544
menu commands, 317–327, 397–398
strategy for, 322–324
KeyMap, 76
KeyPad, 80, 81, 95–97, 137, 516
keypad, 322

Laser Prep file, 23, 24, 28, 31, 185, 216, 470
replacing
LaserSampler II, 157, 568
LaserSpeed, 470, 471, 473, 558
LaserStatus, 115, 185, 416, 558
LaserWriter, *xix–xx*, *xxiii*, 23, 217
files, 23–25, 28, 31, 38
fonts, 26, 142–143, 148–162, 167–172, 182–185, 215, 472
and fractions, 209–211
and LaserStatus, 115
memory, 214–216, 402, 547
quotation marks on, 264, 268
LaserWriter Plus, 26, 140, 184, 215, 216
fonts, 143, 151, 156, 157, 161, 167, 215, 558
Launch Oasis file, 347
Layout, *xxiii*, 222, 223, 227–230, 311, 517, 542–543, 558
Adjust menu, 230
Options menu, 229–230
line breaks, 170
Linotronic, 151, 160, 161, 217
List views, *xxii*, 59, 68, 227–227
Loadkbd Key, 76
Lockup, 120, 558

MacClock, 108
MacDialer, 120
MacDraw, 75, 169, 558
Macintalk, 120, 388, 521, 558
Macintosh 512K, 3, 10
Macintosh Plus, *xviii*, 2–5
accelerator board, 15–16
disk drive, 3, 13
FKeys, 82
memory limitations, 50
RAM size, *xxi*, 5, 7
screen, 55, 300
SIMM sockets, 8
startup, 529, 530–532
Macintosh Revealed, 245, 568
Macintosh SE, 2–5
accelerator board, 15–16
disk drive, 13
FKeys, 82
RAM size, *xxi*, 7
SE-Bus slot, 8
SIMM sockets, 8
startup, 300, 529, 530, 532
Macintosh System Software Update, 379
Macintosh II, *xviii*, 2–4, 15–16
disk drive, 13
FKeys, 82
keyboard, 72
NuBus slot, 8
RAM size, *xxi*, 5, 6–7
SIMM sockets, 8
startup, 529, 530, 532
Macintosh Utilities User's Guide, 141
MacKeymeleon, *xxiv*, 97, 115, 264–270, 312, 558
keyboard editor, 265
MacPaint, 108, 174, 237–239, 300, 327, 388, 558
changing icons in, 279–285
MacPaint 1.5, 547
MacPlan, 103, 115–116
MacPS, 410, 473, 521, 559
Macro processors, *xvi*, 252, 322, 327, 333, 544
Macros, *xxiv*, 252; *see also* Megamacros
adding, 419
applications, 323–399
creating, with *Tempo II*, 411, 415–421
Desktop, 259
examples, 329–334
naming, 263
specialty, 364
MacsBug file, 234, 515, 532–533
MacServe, 49–50, 559
Manager, 49
MacTerm, 120
MacTools, 480
MacUser, 3, 29, 568
MacWorld, 3, 29, 568
MacWrite, 169, 514, 516, 559
MacWrite 4.5, 317–319, 324–327, 329–331
New command, 329
Open command, 329–330
Quit command, 330
master System file, *xxi*, *xxii*, 4, 6, 21, 31–32, 35
with *FKey Maker*, 98
fonts in, 140, 141, 152
problems, 531
MasterKey, 75
Maxwell's Demon, 112, 559
MCI Mail, *see* Electronic mail
MDS Edit, 448, 559
megamacro, 417–421
megaprogram, applications, *xx*
choosing modules for, 315–316

keyboard shortcuts in, 317
organization, 349–350
and RAM capacity, 378–382
memory, *see also* RAM
applications in, 335, 380–382, 399
1 megabyte, 5–7, 315, 377, 398
standard size, 5
system files in, 6
memory management software, 8
memory-resident programs, 471
menu
custom, 286–295
Edit, 57, 203, 214, 238–239, 284, 311–312
Font, *xx*, 161–172, 176–177, 187, 217, 227
ID numbers, 249–250
map, 236–239
with *Pop-Keys*, 80, 178
scrolling, 133–136
shortcuts; *see MenuKey, MenuMaster*
View, 57, 291–295
MenuFonts, 290–291, 312, 559
Menukey, 245, 246, 259, 311, 322–327, 398
installer program, 116
MenuMaster, *xxiv*, 76–77, 86, 517, 559
customizing, 291–292
and menu shortcuts, 245–248, 252, 259, 311, 322–327, 398
MFS (Macintosh Filing System), 505
MicroPhone, 53, 54, 103, 259, 261, 263, 559
miniWRITER, 121, 560
MockChart, 103, 116
MockPackage Plus, 232, 560
MockTerminal, 120, 232
MockWrite, 121, 232
modem, 235
Modern Jazz, 314, 425, 560
dialog box, 276
Hotview function, 384
Moose Phrazer, 120, 560
More, 103, 320, 323, 560
Mouse, 422; *see also* Pointer
shortcuts, 427–435, 474
Moving Lines, 77
MultiFinder, *xvi*, *xix*, 5, 27, 103, 109, 111, 314
alternatives to, *xxiv*, 385–391, 399
and Finder, 379, 380
and print spoolers, 469
problems, 537–539
and RAM, 15, 336, 379, 444, 445, 446
shortcuts, 256, 333
techniques, 379–385
Multi-Scrap, 119, 560
multitasking, 378–379

names, files/folders, 59–66, 68
New Scrapbook, 119, 560
NotePad+, 121
notepads, 117, 121, 375
numeric calculation routines, 16

Oasis, *xxiv*, 111, 344–351, 378, 398, 518, 560
launching, 347, 398
screen, 347, 349–351, 398
operating system; *see* System file
Optima, 143, 156, 160
Other..., *xxiii*, 111, 123, 125, 130–132, 138, 518, 560
Other...command, 80, 90
outliners, 117–118
Outlook, 118

PackIt, 512
PageMaker, 24, 25, 31, 38, 112, 172, 470, 560
commands, 318–319, 320
and downloadable fonts, 153, 185, 187
in file sorting, 44
and print spoolers, 469, 470
and RAM, 444
spacing rules, 169–170
Paint Cutter, 300, 561
partitions, 50, 381, 382, 438, 451
Password, 77, 561
passwords, 49, 77
Paste command, 203, 316, 393
Patch A Disk...command, 479–480
Pathname, 77, 561
paths, 46–48, 77
period, changing, 79, 264, 265, 268, 270–272, 312
phone dialer, 120, 375
phone log, 375
Phone Pad, 107
Phone Book, 120–121
PictureBase Retriever, 114, 561
pointer, 74, 89, 91, 93, 101, 547; *see also* Mouse
Pop-Keys, 75, 80, 81, 516
conflicts with, 97, 220–221
installing, 89–90
menu, 80, 82, 89–95, 98, 99, 137
PostScript, *xxiii*, 24, 25, 31, 141, 149, 217

adding fractions to font, 198–215
fonts, 149–154, 160–161, 168–169, 182–184, 188, 216, 217
spoolers, 470, 471
PowerDisk, 118, 125, 452–453, 457–465, 561
activity figure, 461–462
cache utilization, 461
Control Panel, 459–461
efficiency figure, 461–462, 463
memory utilization, 463–464
versions, 457
PowerOp, 486–491, 493, 561
PowerStation, *xxiv*, 235, 346, 354–359, 398, 404, 408, 561
Clean Up options, 355, 358
Cut Page command, 357
Shortcuts command, 358
"type added" feature, 359
PowerStation/E, 518
Print queue, 473
Print spooler, *xv*, *xx*, 49, 378, 385, 402, 467–473, 474
and automatic startup, 470
and downloadable fonts, 472
and *LaserStatus*, 416
manager, 468–469
RAM required, 471, 472
and *Servant*, 389
Print Monitor, 27, 31, 379, 385, 469, 471, 474
Print command, 246
Print Sample command, 209, 210
Printing, 23–25, 141, 160, 184; *see also* ImageWriter, LaserWriter, PostScript
as background task, 378, 379, 399, 468, 474; *see also* Print spooler
drivers, 23–25
initializing, 471
Puzzle, 112
Pyro!, 137, 233, 235, 311, 355, 561

QDial, 121, 561
Quick Launch list, 363
QuickDraw, 24
spoolers, 470, 473
QuicKeys, *xvi*, *xxii*, *xxiv*, 98, 99–102, 544, 562
Alias command, 271, 312
and *AutoBlack*, 332, 333
Buttons...command, 278
to customize keyboard, 137, 246, 270–274
demo, 518–519
document launching, 399
as Finder substitute, 346, 363
and keyboard shortcuts, 252–258, 311–312, 322, 327, 332, 398
Menu/Da...command, 253, 329, 330, 331
Specials command, 272
window management shortcuts, 332
QuickPaint, 113, 562
QuickQuotes, 272–274, 312
QuickEdit, 118
QuikSheets, 109, 118
Quotation marks, changing, 264, 265, 268–269, 272–274, 312

RAM, 5, 103, 104, 315, 377–378
and *Available Key*, 73–74
and configuration files, 25
fragmentation, 382
and memory constraints, 336, 379–382
size decisions, 6–8
speed gains, 4
speed limitations, 436–437
and *StartMac*, 404
RAM cache, *xx*, *xxv*, 336, 377, 402, 443, 451–467, 474
advantages, 453–455
automated, 466
and copy protection, 454–455
disadvantages, 455–457
failure, 455
va. RAM disk, 467
size, 465, 466
and speed, 118, 337, 452, 453, 455
in startup procedures, 452–453
statistics, 456, 457–461, 466; *see also PowerDisk*
write-through, 456–457, 466, 474
RAM disk, 402, 404, 437–451
advantages, 439–441
automated, 438, 439, 474
and copy protection, 442, 474, 479–482
disadvantages, 441–442
icon, 439, 440
lock, 400
and MultiFinder, 444, 445, 446
and power supply, 441
vs. RAM cache, 467
vs. real disks, 441–442
setup process, 438–439

size, 439–440, 441
and speed, 438, 439, 474
strategies, 442–446
RamSnap, 407, 408, 448–449, 562
RamStart, 447–448, 519, 562
Randomizer, 306, 309–310, 312, 519, 562
read/write head, 14, 442, 483
ResEdit, *xxii*, *xxiii*, 123, 187, 217, 242–244, 311–312, 508, 519–520, 562
do's and don'ts, 243–24
in customizing, 220, 222, 248–251, 252, 279, 282, 285–289, 292
hiding fonts with, 192–197
and keyboard shortcuts, 322–327, 398
Revert command, 244
and scroll bars, 297–299
with special fonts, 547
resource number, 81
ROM (read-only memory), 22
Macintosh vs. LaserWriter, 24
root directory, 415
RPN Calc, 107, 562
RPN Calendar, 107
Rubik's Cube, 112, 562

SafeLaunch, *xxiv*, 77, 341–345, 398, 520, 562
memory option, 345
with *Oasis*, 344
SANE, 16
sans serif fonts, 143–144, 148, 215
decorative, 148
LaserWriter, 155–156
Save command, 200, 231, 507
Save As command, 175, 200, 269, 507
and backup, 496
shortcut, 318–319
and startup screen, 303–305
Scale command, 205
Science Calculator, 107, 562
Scrapbook, 27, 117, 119
scrapbooks, 119
screen, 16, 383–384
and *Copy Screen*, 74
file organization on, 51–59
help functions, 110–111
startup, 299–309, 312, 534
screen fonts, 140, 144–146, 148, 149, 158, 169–172, 181–182
failure, 546
renaming, 186–198
resident, 151–152
sizes, 162, 215
Screen Maker, 300–302, 312, 405, 520, 562
Screen Saver, 77, 562
screen-saver function, 26, 77, 78, 119, 136, 232, 311, 355; *see also AutoBlack*, *Pyro*!
automated, 119, 125, 233–235
problem, 542
screen shots, 84, 108, 115, 146, 238–239, 311
of Key Caps window, 274, 312
ScreenToPaint, 77, 562
scroll bars, custom, 297–299, 312
Select All command, 57, 209, 212, 318–319, 485, 508
Send To Rear, 80
serif fonts, 142–14, 147–148, 215
LaserWriter, 154–155
public-exchange, 147
Servant, *xxiv*, 377, 399, 563
compatibility problems, 389
emergency eject command, 386
icons, 387, 388
and MacinTalk, 388
vs. MultiFinder, 385–389, 399
"Show Page," 387
Set Default Font, *xxiv*
Set EZ-Menu Delay command, 232
Set Paths, 27, 46–47, 68, 520, 563
Set Startup command, 385, 409, 421
SetFile, 47
SetFile Key, 78
shareware; *see* Software, public exchange
single in-line memory modules (SIMMS), 8
Skip Trash Warnings command, 229–230
SkipFinder, *xxiv*, 119–120, 338–341, 342, 350, 378, 398, 446, 563
slash, in fractions, 207
Sleep, 119, 563
Smart Alarms, 108–109, 563
SmartScrap, 407
Software, *xv–xvi*
copy-protected, 476–482, 508
failure, *xx*, 536–549
illegally copied, 477, 478
print spoolers, 468
public-exchange, *xv*, 478
print spoolers, 468
public-exchange, *xv*, 512

sound
digitized files, 306
startup, 305–310, 312, 404, 535
problems, 541–542
SoundCap, 306–307, 541
SoundInit, 306, 307, 309, 312, 535, 541, 542, 563
SoundWave, 541
Space Warp, 78, 563
speaker, internal, 78
special characters, 115; *see also* Symbols
Special Files, 47–48
speech synthesis, 120; *see also* MacinTalk, Sound
with *StartMac*, 405–409
speed, *xx*, 16, 402–474
increasing, 4, 6, 15, 402
obstacles to; *see* Copy protection, Fragmented files
spelling checker, 6, 41, 121
Spellswell, 120, 563
spooled files, storage, 471
Spreadsheet desk accessory, 115–116
SS Key, 78, 564
StartMac, *xxiv*, 403–411, 473, 521, 564
and Finder, 409
instructions for, 405–409
RAM disk in, 407
script file, 403, 407–410
vs. *Tempo II*, 410–411
versions, 410, 473, 521
Startup Device, 25, 529
startup routines, automated, 403, 473
problems, 526
and RAM disk, 440, 451
with *StartMac*, 404–410
with *Tempo II*, 410–422
startups, custom, 299–310
storage, 36, 38
on floppy disks, 38, 40
long-term, 9
StuffIt, 512–514, 515–517
subvolumes, *xxii*, 48–50, 68
Suggested Memory Size, 380–381
Suitcase, *xxii*, 26, 33, 71, 545, 547, 564
and adding fonts, 160, 167, 180–182
as desk accessory extender, 130–136, 137, 138, 255
with FKeys, 89, 111
and *Pyro!*, 235
with *Set Paths*, 46
SuperFinder, *xxiv*, 346, 351–354, 399, 521, 564
button sets, 352–354
installing, 451
SuperLaserSpool, 470, 471, 473, 564
SuperPaint, 112, 339, 341, 346, 392, 542, 564
and desktop publishing, 38
with *QuicKeys*, 256–258
and RAM, 444
screen startup, 300, 302–305
super-macro, *see* Megamacro
support files, *xxii*, 41–42, 46–48, 67–68, 138; *see also* Dictionaries
Switcher, *xxiv*, 243, 377, 389–397, 399
Configure Then Install...command, 394, 395
emergency eject command, 391, 396
launching, 396
memory, 390–391, 394–395
vs. MultiFinder, 390–391
open-ended sets, 391–393
preconfigured sets, 393–394
Save Screen option, 394
screen, 390
set naming, 396
tips, 394–396
Symbol, 143, 157
symbols, 62, 68
System files, *xvii*, *xix*, 21, 22–23, 25–26, 31–33; *see also* Master System file
backing up, *xxi*, 23
components of, 23
and Control Panel, 110
and desk accessories, *xxii*, 104, 105, 109, 110, 114, 117, 123–126, 130, 138
duplicating, 29
and FKeys, 70–71, 81–82, 89, 90, 97, 100
fonts in, 140–141, 144, 146, 159–162, 167, 188, 228
fragmentation in, 485–486
HFS Navigator in, 427
keyboard in, 269–270
in memory, 6
problems, 530–532
recreating, 536
size, 35–36
updating, 28–29, 71
System folder, 21, 35, 37–40, 41
adding extras to, 10
backup for, 494–495
dedicated, 37

files in, *xvii*, 21–28, 35, 46, 125, 133, 179
and FKeys, 80
and icons, 59
master, 30–31, 67
PostScript fonts in, 158, 217, 441
problem, 549
on RAM disk, 440, 441, 443–444, 446, 474
re-creating, 536
size, 36, 38
sounds, 310
and startup routine, 526, 534
System/application disk, 38

Tab key, 423
Talking Moose, 120, 564
Telecommunications programs, 53–54, 55, 77, 120–121, 378–379, 411
Tempo II, 26, 100, 106, 220, 221, 246, 334, 564
Autobot file, 544
and complex macros, 411, 415–416
Configure...command, 544
and keyboard shortcuts, 252, 258–263, 312, 322, 327–334, 398
launching, 258–259, 327–328
Play By Name option, 259, 263
vs. *StartMac*, 410–411
startup routine, 403, 410–422, 474, 544
versions, 258
window management shortcuts, 332
temporary files, 27
text editors, 121
text fonts, *see* Serif fonts
Text Views command, 226
The Clipper, 119, 563
thesaurus, 122
ThinkTank, 103, 564
3D Tic-Tac-Toe, 112, 552
Thunder!, 120, 564
ThunderScan, 40, 388
Time Logger, 78–79, 565
Toggle Key, 79
Toggle Window Size, 80
Touch•n•Go, 122
transfer utilities, *see* Direct-transfer utilities
Trash can, *xiv*, 52
troubleshooting, 158–161, 524–549
typefaces, *see* Fonts

underline, 268
Use Zoom Rects command, 229
user groups, *xv*, 28, 29, 123, 307
Utilities Disk, 512–521

VAX, 440
version numbers, 28–29, 79, 279
Version Reader, 79
VideoWorks II, 113, 565
volume manager, 48

Week-at-a-Peek, 107
Widgets, 300
width tool, 208
Window, 79
Window DA, 122–123, 565
Window Manager FKeys, 80
Windows, 122–123
windows, *xiv*, *xxii*, 42, 51–59, 68, 79–80, 383–384
custom, 221–231
default, 230–231
and FKeys, 73
main, 40, 43, 45, 52
offset rows, 52, 54, 55, 58
placement of, 51–56
viewing files and folders in, 56–59, 68, 190
Word, 6, 27, 36, 41, 46, 75, 331, 559
commands, 319, 320
dialog boxes, 276, 425–426
Help file, 41, 110
key combinations, 323, 543
keyboard shortcuts, 424–426
and screen fonts, 169
settings file, 41
spelling checker, 6, 41, 121
as startup disk, 38
and Switcher, 392–393
Word 3.0, *xvi*, 6
Word 3.01, 278
Word Finder, 122, 565
word processing, 111, 120, 403, 405, 411, 447
WordPerfect, 120, 565
write-through, 456–457, 466

zeros, 62–63
zoom box, 332, 383
Zoom Windows command, 332

About the Author

Robert C. Eckhardt is a Macintosh author and user. He is a contributing editor for *Macworld* and co-author of the Q & A column for *Publish!*. Eckhardt wrote *Free (And Almost Free) Software for the Macintosh* (Crown Publishers) and is at work on a new Macintosh book.

Eckhardt, an avid marine wildlife fan and expert, was Assistant Professor of Zoology at the University of Maine at Orono and Naumann Fellow at the American Museum of Natural History in New York. He has a B.A. from Harvard in Biology and a Ph.D. in Ecology and Evolutionary Biology from Cornell.

SIMON & SCHUSTER, INC. MAKES NO WARRANTIES, EXPRESS OR IMPLIED IN CONNECTION WITH THE SOFTWARE, AND EXPRESSLY INCLUDES ALL WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. SIMON & SCHUSTER SHALL HAVE NO LIABILITY FOR CONSEQUENTIAL, INCIDENTIAL OR EXEMPLARY DAMAGES.

The Fully Powered Mac®

REPLACEMENT ORDER FORM

Please use this form when ordering a replacement for a defective diskette(s).

A. If Ordering within Thirty Days of Purchase

If a diskette is reported defective within thirty days of purchase, a replacement diskette will be provided free of charge. *This card must be totally filled out and accompanied by the defective diskette and a copy of the dated sales receipt.* In addition, please complete and return the Limited Warranty Registration Card.

B. If Ordering after Thirty Days of Purchase but within One Year

If a diskette is reported defective after thirty days but within one year of purchase and the Warranty Registration Card has been properly filed, a replacement diskette will be provided to you for a nominal fee of $5.00 (send check or money order only.) *This card must be totally filled out and accompanied by the defective diskette, a copy of the dated sales receipt, and a $5.00 check or money order made payable to Simon & Schuster, Inc.*

NAME _____ PHONE NUMBER ( )_____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

PURCHASE DATE _____ PURCHASE PRICE _____

COMPUTER BRAND & MODEL _____

Please send all requests to: MicroService/Customer Service, 200 Old Tappan Road, Old Tappan NJ 07675

NOTE: Simon & Schuster reserves the right, at its option, to refund your purchase price in lieu of providing a replacement diskette.

0-13-332230-0

The Fully Powered Mac®

LIMITED WARRANTY REGISTRATION CARD

In order to preserve your rights as provided in the limited warranty, this card must be on file with Simon & Schuster within thirty days of purchase.

Please fill in the information requested:

NAME _____ PHONE NUMBER ( )_____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

COMPUTER BRAND & MODEL _____ DOS VERSION _____ MEMORY _____K

Where did you purchase this product?

DEALER NAME _____ PHONE NUMBER ( )_____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

PURCHASE DATE _____ PURCHASE PRICE _____

How did you learn about this product? (Check as many as applicable.)

STORE DISPLAY _____ SALESPERSON _____ MAGAZINE ARTICLE _____ ADVERTISEMENT _____

OTHER (Please explain) _____

How long have you owned or used this computer?

LESS THAN 30 DAYS _____ LESS THAN 6 MONTHS _____ 6 MONTHS TO A YEAR _____ OVER 1 YEAR _____

What is your primary use for the computer?

BUSINESS _____ PERSONAL _____ EDUCATION _____ OTHER (Please explain) _____

Where is your computer located?

HOME _____ OFFICE _____ SCHOOL _____ OTHER (Please explain) _____

0-13-332230-0

PUT
FIRST-
CLASS
STAMP
HERE

Simon & Schuster, Inc.
Brady Books
One Gulf+Western Plaza
New York, New York 10023

ATTN: **PRODUCT REGISTRATION**

UNDERNEATH THIS FLAP is an 800K disk containing 20 of the most useful utilities described in *The Fully Powered Mac*. Among the programs included on the disk are:

ApFont ■ AutoBlack ■ FKey Manager ■ Fontsie ■ Layout ■ MenuMaster ■ Oasis ■ Ramstart ■ Randomizer ■ ResEdit ■ SafeLaunch ■ Screen Maker ■ Set Paths ■ SuperFinder

All of the programs on the disk, and their associated documentation files, have been packed (compressed) in order to squeeze as much on the disk as possible. You must unpack each program before you can use it. Complete unpacking instructions can be found in Appendix A. For those of you who never read manuals, here is a quick summary of the unpacking process:

1. *The Fully Powered Mac* utilities disk is not a System disk (a disk containing a System folder), so start your Macintosh with one of your own System disks.
2. Copy StuffIt, the unpacking program, and the file you want to unpack from the utilities disk to a blank work disk.
3. Launch StuffIt either by double-clicking on the StuffIt icon, or by double-clicking on the file you want to unpack. If you launch StuffIt by the former method, use the Open command to open the window of the file you want to unpack.
4. If there is one name listed in the file window, click on it to select it. If there is more than one, shift-click to select them all.
5. Click on the Extract... button.
6. In the resulting Save dialog box, make sure that StuffIt intends to save the unpacked files on your work disk, then click on the Save button (if you selected one file) or the Save All button (if you selected more than one).
7. Quit StuffIt.

Your decompressed program is now ready to run and its documentation file, if it has one, is now in a format readable by most word processors. For information and advice on getting the most from each of the programs on the disk, consult the appropriate pages in *The Fully Powered Mac*.

Cover Illustration by Eugene Mihaesco

THE FULLY POWERED MAC®

Fasten your seat belt and get ready to shift into overdrive with this valuable book/disk package.

The book is loaded with sage advice, special techniques, and sound strategies, and the disk is stuffed with valuable software. Together, they're everything you need to make your Mac faster, more efficient, and better tailored to your needs.

Inside, you'll cover basics (including file and window management), desk accessories and FKeys, and tips on using fonts, an important subject for everyone with a printer. Step-by-step instructions for customizing your Mac include adding or changing keyboard shortcuts, individualizing your desktop, and installing your own startup screen. After you customize your Mac to fit the way you work, learn how to pour on the speed with:

- Keyboard shortcuts
- Automated startup procedures
- Macros
- Special mouse techniques
- RAM caches
- RAM disks
- Pitfall warnings
- File backup techniques
- Troubleshooting techniques

The Fully Powered Mac utilities disk contains 20 of the most useful freeware/shareware programs described in the book, programs that will unleash your Mac's hidden potential. (See the other side of this flap for a list.) Once you know how to make your Mac work to its full potential, there's no going back. No matter if you're a novice or expert, it's time to power up your Mac!

Brady Utility Software ■ Distributed by Prentice Hall Trade ■ New York

ISBN 0-13-332230-0

0 21898 32230 2